Kamienny Notebook

William Stein

Contents

1 September 24, 2009

From Sheldon:

For each prime N between 19 and 97 (inclusive) we need to choose a p-Eisenstein quotient A of $J_1(N)$ (with p odd), and find 4 weight two newforms attached to A such that the vectors consisting of their first four Fourier coefficients are linearly independent over $\mathbb{Z}[D]$ in characteristic 2 (or possibly in char 3 if the search in char 2 fails), where D is the group of diamond operators at level N. Also, if possible, I'd like to know if the p-Eisenstein prime is unramified in the Hecke algebra. Finally, I don't really need to know the above for $N = 19, 23, 29, 31, 41$, or 59. I have a direct geometric proof of the non-existence of degree 4-torsion points of order N in these cases. However, if it's not very difficult it would be nice to know the answer in these case too.

Anyway, barring more of my stupid mistakes here's a list of levels N, primes p, and n=order of the character through which a generator of the diamond operators acts on the p-Eisenstein quotient. The levels marked with an asterisk aren't really necessary, but would be nice if it's not extra work for you. If the calculation doesn't produce independence in some case, then we can probably just try a different p.

```
19*, 487, 9
23*, 37181, 11
29*, 43, 7
31*, 2302381, 15
37, 19, 9
41*, 431, 5
43, 463, 7
47, 139, 23
53, 96331, 13
59*, 998855361369139381235879427 1, 29
61, 2801, 5
67, 661, 11
71, 211, 5
73, 241, 6
79, 199, 3
83, 17210653, 41
89, 37, 4
97, 367, 3
```

From William:

I wrote some relevant code and ran it. Here is the code.

```
data = [(19, 487, 9), (23, 37181, 11), (29, 43, 7), (31, 2302381, 15),
(37, 19, 9), (41, 431, 5), (43, 463, 7), (47, 139, 23), (53, 96331, 13),
(59, 998855361369139381235879427 1, 29), (61, 2801, 5), (67, 661, 11),
(71, 211, 5), (73, 241, 6), (79, 199, 3), (83, 17210653, 41),
(89, 37, 4), (97, 367, 3)]

def char(N, p, d):
```

```
    G = DirichletGroup(N,CyclotomicField(d))
    return [eps[0] for eps in G.galois_orbits() if
            eps[0].order()==d][0].minimize_base_ring()

def ms(N,p,d):
    return ModularSymbols(char(N,p,d), 2, +1).cuspidal_subspace()

def modp_reductions(f,p):
    # compute mod-p reductions of a newform
    K = f.parent().base_ring()
    fac = K.factor(p)
    v = []
    F0 = None
    for P, e in K.factor(p):
        F = K.residue_field(P)
        if F0 is None:
            F0 = F
            phi = lambda x: x
        else:
            # fix map F --> F0
            phi = F.hom([F.polynomial().roots(F0)[0][0]])
        R = F0[['q']]
        for i in range(F.degree()):
            v.append(R([phi(F(a)^(p^i)) for a in f.list()]))
    return v
```

Then I ran the following:

```
for N,p,d in data:
    print N,p,d
    M = ms(N,p,d)
    for A in M.decomposition():
        f = A.q_eigenform(5,'a')
        G = [g.list()[1:5] for g in modp_reductions(f, 2)]
        K = G[0][0].parent()
        V = K**4
        z = [V(x) for x in G]
        S = V.span(z)
        print S.rank()
```

which output

```
19 487 9
4
23 37181 11
4
29 43 7
4
31 2302381 15
4
37 19 9
4
4
41 431 5
4
43 463 7
4
4
47 139 23
...
```

This computation means the following.

Proposition 1.1. *Let A be any simple abelian variety factor of $J_1(N)$ with character (N, p, d) in the list above (with $N < 47$, since that is all I've done so far) and A having character of order d. Let $f_1, \ldots, f_r$ be the newforms associated to A, so $r = \dim(A)$.*

Let $\overline{f}_1, \ldots, \overline{f}_r$ *be the images of these newforms in* $\overline{\mathbb{F}}_2[[q]]$ *under reduction modulo primes over* 2. *For each, let* v_i *be the vector in* $\overline{\mathbb{F}}_2^4$ *of the coefficients of* q, q^2, q^3, q^4 *of* $\overline{f}_i$. *Then the span of the* v_i*'s has dimension* 4.

Is that equivalent to what we need or not? It may take hours for the data to run up to 100. Let me know if this is headed in the right direction, if I'm completely misunderstanding everything, if something is unclear, if you want to see things more explicitly, etc.

2 January 7, 2010

Is the following what you want to compute?

For each prime N between 19 and 97 (inclusive) choose a p-Eisenstein quotient A of $J_1(N)$ (with p odd). This abelian variety A is defined over $\mathbb{Q}$. Suppose we have 4 weight two newforms (*not just cusp forms?!*) attached to A, say $f_1, \ldots, f_4$. For each of the f_i, consider the corresponding vector $v_i = (a_1(f_i), a_2(f_i), a_3(f_i), a_4(f_i))$ of the first four coefficients. Let $\mathcal{O}$ be the ring generated by all the coordinates of the vectors v_i, for $i = 1, 2, 3, 4$. Let

$$R = \mathbb{Z}[\langle a\rangle : a \in (\mathbb{Z}/N\mathbb{Z})^*] \subset \mathcal{O}$$

be the subring generated by the diamond bracket operators. Let M be the module generated over $\mathcal{O}$ by the v_i (is this right?). Then M is a module over both R and $\mathcal{O}$.

Let ℓ be a prime, typically 2, but maybe 3. Consider the $\overline{R} = R \otimes_{\mathbb{Z}} \mathbb{F}_\ell$-module

$$\overline{M} = M \otimes_{\mathbb{Z}} \mathbb{F}_\ell$$

Let $\overline{v}_i$ be the image of v_i in $\overline{M}$. We say the $\overline{v}_i$ are *"linearly independent in characteristic ℓ over R"* if whenever

$$\sum_{i=1}^{4} \alpha_i \overline{v}_i = 0 \in \overline{M},$$

with all $\alpha_i \in \overline{R}$, then all $\alpha_i = 0$.

Our goal is to decide whether there are f_i such that the corresponding v_i are linearly independent.

2.1 Ideas for a stupid (but correct) algorithm to do the above computation

For starters we'll try to come up with some non-clever way to do this computation. Once that is nailed down we can try to come up with something much faster.

First, explicitly compute a number field K that contains the coefficients of the f_i by taking Galois closures and composite fields, etc. We may thus assume the coefficients of the f_i are all given explicitly as elements of a common number field. Let $\mathcal{O}$ be the ring spanned by these coefficients, where we can compute $\mathcal{O}$ by taking all coefficients up to the Sturm bound, say, and taking their $\mathbb{Z}$-span. Explicitly, we will represent $\mathcal{O}$ by given by a list $b_1, \ldots, b_r$ of $\mathbb{Z}$-independent elements of K.

Write the vectors v_i as explicit elements of the vector space $V = K^4$. Compute the $\mathrm{rank}(\mathcal{O}) \cdot 4$ elements $b_i v_j \in V$ for $1 \le i \le r$ and $1 \le j \le 4$. Then, using Hermite normal form, compute a $\mathbb{Z}$-basis $c_1, \ldots, c_t$ for the $\mathbb{Z}$-span M of all the $b_i v_j$.

Let $\langle d\rangle$ be a generator of the diamond bracket operators group (which is cyclic since N is prime). We will have $\langle d\rangle(f_i) = z_i f_i$ for each i, where $z_i \in \mathcal{O}$ is a root of unity. Somehow compute these z_i explicitly as elements of $\mathcal{O}$. We can use the relationship between diamond operators and Hecke operators: for d prime we have $\langle d\rangle = (T_d)^2 - T_{d^2}$, so $\langle d\rangle$ acts on f_i via $a_d(f_i)^2 - a_{d^2}(f_i)$.

Let $a_1, \ldots, a_s$ be a $\mathbb{Z}$-basis for $R \subset \mathcal{O}$. Of course we can take the a_i to be successive powers of a root of unity, but it is important that we *represent* them as elements of $R \subset \mathcal{O} \subset K$. Note that because R is integrally closed (being the ring of integers of $\mathbb{Z}[\zeta]$, we have that $\mathcal{O}/R$ is torsion free, so the map $R \otimes \mathbb{F}_\ell \to \mathcal{O} \otimes \mathbb{F}_\ell$ is injective. For $x \in R$ let $\overline{x}$ be the image of x in $\overline{\mathcal{O}} = \mathcal{O} \otimes \mathbb{F}_\ell$. The injectivity remark above tells us that the $\overline{a}_i$ are linearly independent as elements of the $\mathbb{F}_\ell$-vector space $\overline{\mathcal{O}}$.

In the expression

$$\sum_{i=1}^{4} \alpha_i \overline{v}_i = 0 \in \overline{M},$$

write $\alpha_i = \sum_{j=1}^{s} w_{ij} \overline{a}_j$ with $w_{ij} \in \mathbb{F}_\ell$. Expanding out we have

$$\sum_{i=1}^{4} \sum_{j=1}^{s} w_{ij} \overline{a}_j \overline{v}_i = 0.$$

We can explicitly compute the multiplication $\overline{a}_j \overline{v}_i \in \overline{M}$ by lifting, using the embeddings $R \subset \mathcal{O} \subset K$ that we fixed above (i.e., the z_i), and reducing. Finally, the α_i are all 0 if and only if the w_{ij} are all 0.

Conclusion: The $\overline{v}_i$ are linearly independent over R in characteristic ℓ if and only if the $4s$ elements $\overline{a}_j \overline{v}_i$ of the $\mathbb{F}_\ell$ vector space $\overline{M}$ are linearly independent over $\mathbb{F}_\ell$. The latter can be determined by linear algebra over $\mathbb{F}_\ell$ (computing a determinant).

3 February 10, 2010

I worked on trying to do the first example. It is worrisome, regarding our entire approach to this problem!

```
Hi,

Take N=19 and eps the character of order 9.  There is one newform f in
S_2(eps), and it's

       f = q + (-zeta18^2 + zeta18 - 1)*q^2 + (-zeta18^4 + zeta18^3 +
zeta18^2 - 1)*q^3 + (zeta18^4 - 2*zeta18^3 + zeta18^2 - 2*zeta18 +
1)*q^4 + O(q^5)

The Galois group of Q(zeta18)/Q is cyclic of order 6, with some
generator phi.   I think I've carried out the computation we discussed
using f, phi(f), phi^2(f), and phi^3(f).   I get that these 4 are
linearly independent in characteristic p over Z[zeta18] for all primes
p *except* p=3 and p=37.

Does this make any sense to you?

The calculation we discussed before is much simpler in case that the
newform f is defined over the field Q(zeta_n), where n is the order of
the character, as we have above.   In this case, we just compute the
4*phi(n) q-expansions

          g_{i,j} = phi^i(zeta^j * f)
                0 <= i < phi(n) and 0 <= j < 4,

then take the coefficients a_m(g_{i,j}) for m=1,2,3,4, and by viewing
each a_m as a vector over Z (in terms of a basis for Z[zeta_n], we get
4*phi(n) vectors over Z each of degree 4*phi(n).  We then form the
4*phi(n) x 4*phi(n) integer matrix A with these vectors as rows, and
compute its integers determinant.  The primes p that divide the
determinant are precisely the primes where the newforms {phi^i(f) : i
=0,1,2,3} are linearly dependent over Z[zeta] in characteristic p.
Anyway, this is the computation I just did.   The matrix A that I got
in case N=19 has rank 14 for p=3, rank 23 for p=37, and rank 24 for
all other p (including p=2).

 -- William
```

Here's the Sage worksheet:

```
{{{id=1|
N=19
J1(N)
///
Abelian variety J1(19) of dimension 7
}}}

{{{id=2|
G = DirichletGroup(N)
///
}}}

{{{id=3|
G.0.order()
///
18
}}}

{{{id=4|
eps = G.0^2; eps.order()
```

```
///
9
}}}

{{{id=5|
S = CuspForms(eps, 2); S
///
Cuspidal subspace of dimension 1 of Modular Forms space of dimension 3, character [zeta18^2] and weigh
}}}

{{{id=11|
len(S.newforms())show(A)
///
1
}}}

{{{id=9|
f = S.newforms()[0].qexp(5); f
///
q + (-zeta18^2 + zeta18 - 1)*q^2 + (-zeta18^4 + zeta18^3 + zeta18^2 - 1)*q^3 + (zeta18^4 - 2*zeta18^3
}}}

{{{id=7|
f.base_ring().degree()
///
6
}}}

{{{id=6|
d = primitive_root(N); d
///
2
}}}

{{{id=12|
eps(d)
///
zeta18^2
}}}

{{{id=13|
K = f.base_ring()
print K
zeta = K.gen()
///
Cyclotomic Field of order 18 and degree 6
}}}

<p>A generator of the Galois group:</p>

{{{id=16|
phi = K.automorphisms()[-1]; phi
///
Ring endomorphism of Cyclotomic Field of order 18 and degree 6
  Defn: zeta18 |--> zeta18^5
}}}

{{{id=14|
z = zeta
for i in range(6):
    print z
    z = phi(z)
///
zeta18
```

```
zeta18^5
zeta18^4 - zeta18
-zeta18^5 + zeta18^2
-zeta18^4
-zeta18^2
}}}

{{{id=21|
def apply(f, phi):
    R = f.parent()
    return R([phi(f[i]) for i in range(5)],prec=5)
///
}}}

{{{id=19|
fv = [f]
for i in range(3):
    fv.append(apply(fv[-1],phi))
///
}}}

{{{id=18|
for g in fv: print g
///
q + (-zeta18^2 + zeta18 - 1)*q^2 + (-zeta18^4 + zeta18^3 + zeta18^2 - 1)*q^3 + (zeta18^4 - 2*zeta18^3
q + (zeta18^5 + zeta18 - 1)*q^2 + (-zeta18^3 - zeta18^2 - zeta18)*q^3 + (-2*zeta18^5 + 2*zeta18^3 + ze
q + (zeta18^5 + zeta18^4 - zeta18 - 1)*q^2 + (-zeta18^5 + zeta18^3 + zeta18 - 1)*q^3 + (-zeta18^5 - 2*
q + (-zeta18^5 + zeta18^4 + zeta18^2 - zeta18 - 1)*q^2 + (zeta18^5 - zeta18^4 - zeta18^3 + zeta18)*q^3
}}}

{{{id=26|
f[1].list()
///
[1, 0, 0, 0, 0, 0]
}}}

{{{id=25|
def qexp_to_integral_vector(g):
    return vector(ZZ,sum([g[i].list() for i in [1..4]], []))
///
}}}

{{{id=24|
for g in fv:
    print qexp_to_integral_vector(g)
///
[1, 0, 0, 0, 0, 0, -1, 1, -1, 0, 0, 0, -1, 0, 1, 1, -1, 0, 1, -2, 1, -2, 1, 0]
[1, 0, 0, 0, 0, 0, -1, 1, 0, 0, 0, 1, 0, -1, -1, -1, 0, 0, -1, -1, 1, 2, 0, -2]
[1, 0, 0, 0, 0, 0, -1, -1, 0, 0, 1, 1, -1, 1, 0, 1, 0, -1, 1, 1, 0, -2, -2, -1]
[1, 0, 0, 0, 0, 0, -1, -1, 1, 0, 1, -1, 0, 1, 0, -1, -1, 1, -1, 1, -2, 2, -1, 1]
}}}

<p>Compute the integer vectors corresponding to the 24 modular forms $\varphi^i(\zeta^j \cdot f)$ for

{{{id=23|
# first compute zeta^j*f for j =0,1,..,5.
w = [zeta^j*f for j in [0..5]]
# next compute images under powers of phi
z = [w]  # copy of w
for i in [0..2]:
    z.append([apply(z[-1][j], phi) for j in [0..5]])
z = sum(z, [])
///
}}}
```

```
{{{id=28|
def qexp_to_vector(g, ell):
    return vector(GF(ell), sum([g[i].list() for i in [1..4]], []))
///
}}}

{{{id=17|
zmod = [qexp_to_vector(g,3) for g in z]
span(zmod).dimension()
///
14
}}}

{{{id=27|
def qexps_to_matrix(z, ell):
    k = GF(ell) if ell else ZZ
    return matrix(k, [sum([g[i].list() for i in [1..4]],[]) for g in z])
///
}}}

{{{id=31|
len(z)
///
24
}}}

{{{id=29|
A = qexps_to_matrix(z,0); A
///
24 x 24 dense matrix over Integer Ring (type 'print A.str()' to see all of the entries)
}}}

{{{id=30|
factor(A.determinant())
///
3^10 * 37
}}}
```

I also tried with all 6 conjugates of f:

```
{{{id=23|
# first compute zeta^j*f for j =0,1,..,5.
w = [zeta^j*f for j in [0..5]]
# next compute images under powers of phi
z = [w]  # copy of w
for i in [0..5]:
    z.append([apply(z[-1][j], phi) for j in [0..5]])
z = sum(z, [])
///
}}}

{{{id=28|
def qexp_to_vector(g, ell):
    return vector(GF(ell), sum([g[i].list() for i in [1..4]], []))
///
}}}

{{{id=17|
zmod = [qexp_to_vector(g,37) for g in z]
span(zmod).dimension()
///
24
}}}
```

```
{{{id=27|
def qexps_to_matrix(z, ell):
    k = GF(ell) if ell else ZZ
    return matrix(k, [sum([g[i].list() for i in [1..4]],[]) for g in z])
///
}}}

{{{id=31|
len(z)
///
42
}}}

{{{id=29|
A = qexps_to_matrix(z,0); A
///
42 x 24 dense matrix over Integer Ring (type 'print A.str()' to see all of the entries)
}}}

{{{id=30|
factor(A.hermite_form(include_zero_rows=False).determinant())
///
3^10
}}}
```