Doc-Start

Magma codes for Quotient modular curves $X_0(N)/W_N$

Francesc Bars*

section.1 1 Magma functions for quotient modular curves.

Here, we present Magma functions presented in github in order to obtain the modular forms f_i that appears its associated abelian variety A_{f_i} (by Shimura) in the $\mathbb{Q}$-decomposition of $J_0(N)/W_N$ (up to $\mathbb{Q}$-isogeny, the Jacobian of the quotient modular curve $X_0(N)/W_N$):

$$J_0(N)/W_N \sim_{\mathbb{Q}} \prod_{i=1}^{n} A_{f_i}^{m_{f_i}},$$

and an approach to determine the naturals m_{f_i} for each modular forms, (recall W_N is a subgroup generated by some Atkin-Lehner involutions of the modular curve $X_0(N)$).

The n_i naturals associated to A_{f_i} computed in the Magma functions bellow give an upper bound for the m_{f_i}'s, and after comparing with the genus of $X_0(N)/W_N$ we can decide if $n_i = m_{f_i}$ for all i or not. If some of the bounds $n_i > m_{f_i}$, then, we need to deal a detailed study by following Lema 2.2 and Proposition 2.2 (i)(ii) in [**?**] to obtain m_{f_i} for such factors A_{f_i}. See next parts of the file for detailed discussion of such point.

All computations was done by Magma Online Computation, V2.26-1.

subsection.1.1 1.1 Magma function to compute genus of a quotient modular curve

Function that computes $\nu(m, n)$ the number of fixed points in $X_0(m)$ of the Atkin-Lehner involution w_n where $n \leq 4$ with $(n, m/n) = 1$. We use classical formula given at P.G.Kluit "On the normalizer of $\Gamma_0(N)$, LNM 601, (1977), pp.239–246.

Input: level m and $n|m$ with $n \leq 4$ and $(n, m/n) = 1$.
Output: Number of fixed points of w_n at $X_0(m)$.

```
fixedpointsALinvsmall:=function(m, n)

if n eq 3 then
         q:=Factorization(Numerator(m/3));
         nu_3:=2;
         for d in [1 .. #q] do
           if q[d][1] eq 2 then
```

*F.Bars is supported by PID2020-116542GB-I00.

```
              if q[d][2] gt 2 then
               nu_3:=0;
              else
              end if;
            else
              nu_3:=nu_3*(1+LegendreSymbol(-n,q[d][1]));
            end if;
          end for;
          return nu_3;
else
          if n eq 1 then
          else
              if n eq 2 then
                q:=PrimeDivisors(Numerator(m/n));
                nu_2factorminus1:=1;
                nu_2factorminus2:=1;
                for d in [1 .. #q] do
                   nu_2factorminus1:=nu_2factorminus1*(1+LegendreSymbol(-1,q[d]));
                   nu_2factorminus2:=nu_2factorminus2*(1+LegendreSymbol(-2,q[d]));
                end for;
                return nu_2factorminus1+nu_2factorminus2;
              else
                 if n eq 4 then
                   q:=PrimeDivisors(Numerator(m/n));
                   l:=Divisors(Numerator(m/n));
                   nu_4factorminus1:=1; nu_4sumeuler:=0;
                   for d in [1..#q] do
                            nu_4factorminus1:=nu_4factorminus1*(1+LegendreSymbol(-1,
                   end for;
                   for dd in [1..#l] do
                      ss:=Numerator(m/(n*l[dd]));
                      ssh:=GCD(l[dd],ss);
                      nu_4sumeuler:=nu_4sumeuler+EulerPhi(ssh);
                   end for;
                   return nu_4factorminus1+nu_4sumeuler;
                 end if;
              end if;
           end if;
end if;

end function;
```

Function that computes $\nu(m,n)$ the number of fixed points in $X_0(m)$ of the Atkin-Lehner involution w_n where $n > 4$ with $(n, m/n) = 1$. We use classical formula given at book of William Kuyk, Modular functions of one variable I, LNM 320, Antwerp 1972, or more precisely P.G.Kluit "On the normalizer of $\Gamma_0(N)$, LNM 601, (1977), pp.239–246.

Input: level m and $n|m$ with $n > 4$ and $(n, m/n) = 1$.
Output: Number of fixed points of w_n at $X_0(m)$.

```
fixedpointsALinvbig:=function(m, n)

PD:=PrimeDivisors(Numerator(m/n)); D:=Divisors(Numerator(m/n));

n_mod2:=n mod 2; n_mod4:=n mod 4;


if n_mod2 eq 1 then
      if n_mod4 eq 3 then
           if 2 in PD then
               if 8 in D then
nu_nodd3mod4_8:=2*(ClassNumber(-n)+ClassNumber(-4*n))*(1+KroneckerSymbol(-n,2));
                    if #PD eq 1 then
                    else
                        for p1 in PD do
                         p1_mod2:=p1 mod 2;
                         if p1_mod2 eq 1 then
nu_nodd3mod4_8:=nu_nodd3mod4_8*(1+LegendreSymbol(-n,p1));
                         end if;
                        end for;
                    end if;
                    return nu_nodd3mod4_8;
               else
                   if 4 in D then
                      nu_nodd3mod4_4:=
(2*ClassNumber(-4*n)+2*(1+KroneckerSymbol(-n,2))*ClassNumber(-n));
                     if #PD eq 1 then
                       return nu_nodd3mod4_4;
                     else
                       for p2 in PD do
                         p2_mod2:=p2 mod 2;
                           if p2_mod2 eq 1 then
nu_nodd3mod4_4:=nu_nodd3mod4_4*(1+LegendreSymbol(-n,p2));
                           end if;
                       end for;
                       return nu_nodd3mod4_4;
                     end if;
                  else
nu_nodd3mod4_2:=ClassNumber(-4*n)+3*ClassNumber(-n);
                       if #PD eq 1 then
                          return nu_nodd3mod4_2;
                       else
                          for p3 in PD do
                             p3_mod2:=p3 mod 2;
                             if p3_mod2 eq 1 then
```

```
nu_nodd3mod4_2:=nu_nodd3mod4_2*(1+LegendreSymbol(-n,p3));
                                end if;
                              end for;
                              return nu_nodd3mod4_2;
                           end if;
                     end if;
                end if;
           else
                if #PD eq 0 then
nu_nodd3mod4_no2:=ClassNumber(-n)+ClassNumber(-4*n);
                  return nu_nodd3mod4_no2;
                else
nu_nodd3mod4_no2:=ClassNumber(-n)+ClassNumber(-4*n);
                   for j in [1..#PD] do
nu_nodd3mod4_no2:=nu_nodd3mod4_no2*(1+LegendreSymbol(-n,PD[j]));
                   end for;
                   return nu_nodd3mod4_no2;
                end if;
           end if;
      else
           if #PD eq 0 then
              nu_nodd1mod4:=ClassNumber(-4*n);
              return nu_nodd1mod4;
           else
              if 2 in PD then
                   if 4 in D then
                     return 0;
                   else
                      nu_nodd1mod4_2:=ClassNumber(-4*n);
                      PD2:=PrimeDivisors(Numerator(m/(2*n)));
                       for k in [1..#PD2] do
nu_nodd1mod4_2:=nu_nodd1mod4_2*(1+LegendreSymbol(-n,PD2[k]));
                        end for;
                    return nu_nodd1mod4_2;
                   end if;
              else
                  nu_nodd1mod4_no2:=ClassNumber(-4*n);
                     for i in [1 .. #PD] do
nu_nodd1mod4_no2:=nu_nodd1mod4_no2*(1+LegendreSymbol(-n,PD[i]));
                     end for;
                   return nu_nodd1mod4_no2;
              end if;
           end if;
      end if;
  else
     nu_neven:=ClassNumber(-4*n);
     if #PD eq 0 then
       return nu_neven;
     else
```

```
      for u in [1 .. #PD] do
        nu_neven:=nu_neven*(1+LegendreSymbol(-n,PD[u]));
      end for;
      return nu_neven;
    end if;
  end if;
end function;
```

A magma code function to compute the genus of $X_0(b)$, we use the usual formula for computing its genus, that one can obtain for example Prop. 1.43 in Shimura book "Introduction to the Arithmetic Theory of Automorphic Functions", Princeton University Press, Princeton, 1971.

Input: b the level.
Output: the genus of $X_0(b)$.

```
generexoN:=function(b)

   m:=PrimeDivisors(b);
   l:=Divisors(b);
   factor_b:=Factorization(b);
  psiEulerindex:=b;
  order4elliptic:=1;
  order3elliptic:=1;
  cusps:=0;

  for x in [1 .. #m] do
  psiEulerindex:=psiEulerindex*(1+1/m[x]);
  end for;

    for y in [1..#m] do
      if factor_b[y][1] eq 2 then
          order3elliptic:=0;
        if factor_b[y][2] gt 1 then
          order4elliptic:=0;
        end if;
      else
        if factor_b[y][1] eq 3 then
            if factor_b[y][2] gt 1 then
              order3elliptic:=0;
order4elliptic:=order4elliptic*(1+LegendreSymbol(-1,factor_b[y][1]));
            else
order3elliptic:=order3elliptic*(1+LegendreSymbol(-3,factor_b[y][1]));
order4elliptic:=order4elliptic*(1+LegendreSymbol(-1,factor_b[y][1]));
            end if;
        else
```

```
order4elliptic:=order4elliptic*(1+LegendreSymbol(-1,factor_b[y][1]));
order3elliptic:=order3elliptic*(1+LegendreSymbol(-3,factor_b[y][1]));
         end if;
       end if;
     end for;

  for a in [1 .. #l] do
    n1:=Numerator(b/l[a]);
    t1:=GCD(l[a],n1);
    cusps:=cusps+EulerPhi(t1);
   end for;

genus:=1+(psiEulerindex/12)-(order4elliptic/4)-(order3elliptic/3)-(cusps/2);
return genus;
  end function;
```

It is well-knwon for a W_N a subgroup of $B(N)$ of all Atkin-Lehner involutions of $X_0(N)$, the genus g_{W_N} of the quotient modular curve $X_0(N)/W_N$ is given by the formula

$$2g_{X_0(N)} - 2 = 2^r(2g_{W_N} - 2) + \sum_{w_d \in W} \nu(N, d)$$

where $g_{X_0(N)}$ is the genus of $X_0(N)$, and 2^r means the order of W_N.

We implement such formula in a function on Magma.

Input: level N, WN a list with all d where $w_d \in W_N$ (we can omit $d = 1$ if one wish, but not the others, NOT the a generators of W_N, we need list all non-trivial elements), and t the order of the subgroup W_N.

Output: the genus of $X_0(N)/W_N$.

```
genereX0NQuotientWN:=function(N,WN,t);

FixedpointsALinvolutions:=[* *]; vv:=0; L:=Divisors(N);


for i in [1..#WN] do
    u:=GCD(WN[i],Numerator(N/WN[i]));
    if WN[i] in L then
    else
    vv:=1;
    end if;

    if u eq 1 then
       if WN[i] eq 1 then
       else
```

```
          if WN[i] gt 4 then
             nu_Ddi:=fixedpointsALinvbig(N,WN[i]);
FixedpointsALinvolutions:=Append(FixedpointsALinvolutions,nu_Ddi);
          else
             nu_Ddi:=fixedpointsALinvsmall(N,WN[i]);
FixedpointsALinvolutions:=Append(FixedpointsALinvolutions,nu_Ddi);
          end if;
       end if;
     else
      vv:=1;
     end if;

end for;

CountAllFixedPointsALinvolutions:=0;

for u in FixedpointsALinvolutions do
CountAllFixedPointsALinvolutions:=CountAllFixedPointsALinvolutions+u;
end for; if vv eq 0 then
  genusxoN:=generexoN(N);
  genusxoNQuotient:=1+t^(-1)*(genusxoN-1-(CountAllFixedPointsALinvolutions/2));
else
  genusxoNQuotient:=-1;
  end if;
return genusxoNQuotient;

end function;
```

Once we introduced such functions we can do computations in different examples:

```
N := 366; // Level

d:=61; // d an integer with (d,N/d)=1

d2:=2; // d2 an integer with (d2,N/d2)=1

Fixedsmall:=fixedpointsALinvsmall(N,d2);

print Fixedsmall; /* Return the number of fixed points of w_{d2}
when d2 is 1,2,3 or 4*/

Fixedbig:=fixedpointsALinvbig(N,d);

print Fixedbig; /* Return the number of fixed points of w_d when d
is strictly bigger than 4*/

 genus:=generexoN(N);
```

```
 print genus; //Return the genus of X_0(N)

TN := [* 61, 122, 2, 1 *]; /* List the elements of the subgroup W_N
of involutions*/

t:=#TN; //Order of the subgroup of involutions

genusquotientcurve:=genereX0NQuotientWN(N,TN,t);

print genusquotientcurve; /* Return the genus of the quotient
modular curve X0(N)/TN */

TN := [* 61, 122, 2 *]; /* List the elements of the subgroup W_N of
involutions*/

t:=4; //Order of the subgroup of involutions

genusquotientcurve:=genereX0NQuotientWN(N,TN,t);

print genusquotientcurve; /* Return the genus of the quotient
modular curve X0(N)/TN */
```

subsection.1.2**1.2 Magma function to a first approach to obtain $\mathbb{Q}$-decomposition of the Jacobian of a quotient modular curve**

We introduce a magma function to obtain $\mathbb{Q}$-decomposition of a quotient modular curves in a lot of situations, in particular for square-free levels. Consider a quotient modular curve $X_0(N)/W_N$ where $W_N = \{w_1, w_{d_1}, \ldots, w_{d_k}\}$ with $t = |W_N|$.

Input: the level N, the list of naturals $[*1, d_1, \ldots, d_k*]$ or $[*d_1, \ldots, d_k*]$ corresponding to the AL involutions of W_N, a length (which will give the q-expansion length for the modular forms related with the Jacobian of the quotient modular curve, and t the order of W_N. Thus 4 inputs with this order.

Output: The result is a LIST. In the fist position of such list appear 11111111 or 0. If appear 11111111111 means that the function with output the LIST computes the $\mathbb{Q}$-Jacobian decomposition of the quotient modular curve $X_0(N)/\langle w_{d_1}, \ldots, w_{d_k}\rangle$. If appears 0 means that the function computes all f_i's that appears in the Jacobian decomposition, BUT some f_i is repeated too much times in the list.

The second position of the list are the f_i's that appears in the $\mathbb{Q}$-decomposition of $Jac(X_0(N)/W_N)$ repeated each of them n_i times (they appear not consecutive the repeated modular forms). If appeared 1111111 in the first position, then $n_i = m_{f_i}$ and we obtain the exact Jacobian decomposition.

The third position of the LIST is the number field where the modular form is defined, always following the ordering that are list the modular forms in the second position.

The fourth position if the levels that appears each modular form (when is a repeated modular

form in the list, the repeated modular form appeared the level as a multiple of the exact level of the modular form). The ordering follows the order of the list of modular forms in the second position.

The fourth position is the number field where the modular form is defined, always following the ordering that are list the modular forms in the second position.

The fifth position is a control output where gives the modular forms joint with the action of $w_{p_p^{v_p(N)}}$ for primes $p|N$ if belongs to W_N, this may be improved in the future. If has action, if the size of the matrix is not the dimension over the rationals of the field of definition of the modular form, we know that is already an old form that is repeated at the level that is quotied (in third position).

```
JacobianDecompositionQuotientX0NWN:=function(n,WN,length,t); N:=n;
AtkinLehnerfix:=WN; Involutions:=#AtkinLehnerfix; L:=[*
*];F:=[*
*];Level:=[* *]; ALaction:=[**];

Pd:=PrimeDivisors(N); mt:=[**];

for prime in Pd do
      op:=Valuation(N,prime);
      primepower:=Gcd(prime^(op), N);

  mt:=Append(mt, primepower);
end for;


Nd:=Divisors(N); countergenus:=0;

for j in Nd do

MS:=NewformDecomposition(CuspidalSubspace(ModularSymbols(j,2,1)));

       m:=#MS;

       M:=PrimeDivisors(j);

       Nr:=Numerator(N/j);

       divi:=GCD(j,Nr);

       jj:=Numerator(j/divi);

       Mm:=PrimeDivisors(jj);

       Nn:=Divisors(jj);

       mm:=#Mm;

       mn:=#Nn;
```

```
        D:=Factorization(jj);

          for i in [1..m] do

             f:=Eigenform(MS[i],length);

             f2:=MS[i];

             K:=Parent(Coefficient(f,3)); d:=Dimension(MS[i]);

             X:=IdentityMatrix(Rationals(), d);

             u:=0;
                 for jo in [1..Involutions] do
                   dd:=GCD(j,AtkinLehnerfix[jo]);

                  if dd eq AtkinLehnerfix[jo] then

                    Y:=AtkinLehner(MS[i],dd);

                     if Y eq X then

                     else

                     u:=1;

                     end if;

                  else

                     if dd eq 1 then

                     else

                     end if;

                  end if;

                end for;

           if u eq 0 then
              ALactionf:=[**];
              for i in [1..#Pd] do
                if GCD(mt[i],j) eq mt[i] then

                  ALactionf:=Append(ALactionf, [*AtkinLehner(f2,mt[i]),mt[i]*]);
```

```
              end if;
              ALaction:=Append(ALaction,[*f,j,ALactionf*]);
            end for;


          L:=Append(L,f);


          F:=Append(F,K);

uu:=#Basis(K);

countergenus:=countergenus+uu;

          Level:=Append(Level,j);
                else
                 end if;
       end for;

   end for;
genuscorrect:=genereXONQuotientWN(N,WN,t);

if genuscorrect eq countergenus then

  CorrectJacobian:=11111111111111;

  else

  CorrectJacobian:=0000000000000;

  end if;
M:=[*CorrectJacobian,L,F,Level, ALaction*];

return M;

end function;
```

An application to a example for $X_0(366)/W$ with $W = \langle w_2, w_{61} \rangle$ to obtain the $\mathbb{Q}$-decomposition is the following code:

```
N := 366; // Level

TN := [* 61, 122, 2, 1 *]; /* List the elements of the subgroup of
involutions*/

t:=#TN; //Order of the subgroup of involutions

genusquotientcurve:=genereXONQuotientWN(N,TN,t);
```

```
print genusquotientcurve; /* Return the genus of the quotient
modular curve X0(N)/TN */

prec := 20; /*Number of coefficients of the q-expansion*/

 HH :=JacobianDecompositionQuotientX0NWN(N, TN, prec, t);

print HH[1]; /* Should return 11111111111111 if compute the exact
Jacobian decomposition (if one factor appears n times, appears n
times in Jacobian decomposition)  If return 0 then each factor
appears in Jacobian decomposition BUT if one factors appears n times
in Jacob.decom, here could appear m times with m ge n. For N square
free as our example should appear 11111111111 and compute the
programme directly the Jacobian decomposition.*/

print HH[2]; /* List the modular forms with q-expansion that are
factor of the Jacobian, (could appear repeated if some factor of the
Jacobian can appeared repeated)*/

print HH[3]; /* List the number fields of the list of modular forms
given in HH[2] */

print HH[4]; /* List of the levels (except when appears repeated) of
modular forms given in HH[2]
*/
```

The output of this code by Magma V2.27-2 is:

```
13 /*the genus of the quotient modular curve*/


11111111111111 /* The programme compute the Q-decomposition */


[*
    q - q^2 - 2*q^3 - q^4 - 3*q^5 + 2*q^6 + q^7 + 3*q^8 + q^9 + 3*q^10 - 5*q^11
        + 2*q^12 + q^13 - q^14 + 6*q^15 - q^16 + 4*q^17 - q^18 - 4*q^19 +
        O(q^20),
    q - q^2 - 2*q^3 + q^4 + q^5 + 2*q^6 - 5*q^7 - q^8 + q^9 - q^10 - 3*q^11 -
        2*q^12 - 3*q^13 + 5*q^14 - 2*q^15 + q^16 - q^18 + O(q^20),
    q + a*q^2 - q^3 + (-2*a - 1)*q^4 - q^5 - a*q^6 + (-a - 2)*q^7 + (a - 2)*q^8
        + q^9 - a*q^10 + (-a - 2)*q^11 + (2*a + 1)*q^12 - 3*q^13 - q^14 + q^15 +
        3*q^16 - 6*q^17 + a*q^18 + (4*a + 6)*q^19 + O(q^20),
    q + a*q^2 + q^3 + (a^2 - 2)*q^4 + 1/2*(a^5 + 2*a^4 - 10*a^3 - 16*a^2 + 21*a
        + 20)*q^5 + a*q^6 + 1/2*(-2*a^5 - 3*a^4 + 18*a^3 + 22*a^2 - 34*a -
        23)*q^7 + (a^3 - 4*a)*q^8 + q^9 + 1/2*(2*a^5 + a^4 - 18*a^3 - 10*a^2 +
        30*a + 17)*q^10 + 1/2*(-a^4 + 6*a^2 - 2*a - 5)*q^11 + (a^2 - 2)*q^12 +
```

```
        1/2*(-a^5 + 10*a^3 - 21*a + 2)*q^13 + 1/2*(-3*a^5 - 4*a^4 + 26*a^3 +
        28*a^2 - 43*a - 34)*q^14 + 1/2*(a^5 + 2*a^4 - 10*a^3 - 16*a^2 + 21*a +
        20)*q^15 + (a^4 - 6*a^2 + 4)*q^16 + (a^5 + a^4 - 9*a^3 - 6*a^2 + 16*a +
        5)*q^17 + a*q^18 + (a^5 + a^4 - 8*a^3 - 8*a^2 + 11*a + 13)*q^19 +
        O(q^20),
    q - q^2 - 2*q^3 - q^4 - 3*q^5 + 2*q^6 + q^7 + 3*q^8 + q^9 + 3*q^10 - 5*q^11
        + 2*q^12 + q^13 - q^14 + 6*q^15 - q^16 + 4*q^17 - q^18 - 4*q^19 +
        O(q^20),
    q - q^2 + q^3 + q^4 + q^5 - q^6 - 2*q^7 - q^8 + q^9 - q^10 + 6*q^11 + q^12 +
        2*q^14 + q^15 + q^16 + 3*q^17 - q^18 + O(q^20),
    q - q^2 - 2*q^3 + q^4 + q^5 + 2*q^6 - 5*q^7 - q^8 + q^9 - q^10 - 3*q^11 -
        2*q^12 - 3*q^13 + 5*q^14 - 2*q^15 + q^16 - q^18 + O(q^20)
*]

/* The list of modular forms $f_i$ where Q-decomposition prod
A_{f_i} but could be repetitions, like first modular form and the
fifth one, and second with the seventh */


[*
    Rational Field,
    Rational Field,
    Number Field with defining polynomial x^2 + 2*x - 1 over the Rational Field,
    Number Field with defining polynomial x^6 - 11*x^4 + 2*x^3 + 31*x^2 - 10*x -  1
    Rational Field,
    Rational Field,
    Rational Field
*]

/* The list of number fields corresponding to the field of
definition of $f_i$ with same order */


[* 61, 122, 183, 183, 183, 366, 366 *]

/*The list of the levels corresponding to the $f_i$ in L[2] with the
same order, but f_1 is the same that f_5 thus the level of f_1 is
61, the level of f_5 is not the fifth position of the conductors 183
is 61. Also f_2 and f_7 are the same thus the level of f_7 is 122
and not 366 */
```

After the result we obtain that the $\mathbb{Q}$-decomposition of $Jac(X_0(366)/\langle w_2, w_{61}\rangle$ is $\mathbb{Q}$-isogeny of

$$\prod_{i=1}^{7} A_{f_i} \sim (E61a)^2 \times (E122a)^2 \times A_{f_3} \times A_{f_4} \times E366c$$

because by Table 4 of Cremona tables we obtain that $A_{f_1} \sim A_{f_5} \sim E61a$, $A_{f_2} \sim A_{f_7} \sim E122a$ and $A_{f_6} \sim E366c$. Moreover we obtain that A_{f_3} has dimension 2 and new at level 183, and A_{f_4} has dimension 6 of level 183 (we recall $\sim$ means a $\mathbb{Q}$-isogeny).

subsection.1.3 1.3 Magma function to compute the $\mathbb{F}_{p^n}$-points of a Quotient modular curve of level N with $p \nmid N$

Here we ASSUME that one obtained a $\mathbb{Q}$-decomposition of the Jacobian of the quotient modular curve.

We introduce the following Magma function:

Input: N, the level, a prime p which $p \nmid N$, the list of modular forms with repetition that appears in the $\mathbb{Q}$-decomposition of the Jacobian of $X_0(N)/W_N$, and the list of corresponding number field (with the same order that we listed the modular forms), and a bound.

Output: The number of $\mathbb{F}_{p^n}$-points of the reduction of $X_0(N)/W_N$ at p, with $n = 1$ until $n = bound$.

```
FpnpointsforQuotientcurveX0NWN:=function(N,prime,JacDecomp,FieldDefinition,bound);
L:=JacDecomp; p:=prime; F:=FieldDefinition; felm:=# F; bod:=bound;


C:=ComplexField(100); R<x>:=PolynomialRing(C); pj:=0*x+1; Roo:=[**];
for j in [1 .. felm] do

   if Degree(F[j]) eq 1 then

       cc:=Roots(x^2-Coefficient(L[j],p)*x+p,C);

       Roo:=Append(Roo,cc);

      pj:=pj*(x^2-Coefficient(L[j],p)*x+p);

   else

    dd:=Degree(F[j]);

    u:=Roots(DefiningPolynomial(F[j]),C); uu:= # u;

       for m in [1 .. uu] do

       f := hom< F[j] -> C | u[m][1]>;

       cc2:=Roots(x^2-f(Coefficient(L[j],p))*x+p,C);

       Roo:=Append(Roo,cc2);

       pj:=pj*(x^2-f(Coefficient(L[j],p))*x+p);

       end for;

   end if;
```

```
end for; pjdegree:=Degree(pj);

PR:=[* *];

  d2:=Degree(pj);

long:= # Roo;




for nn in [1 .. bod] do s:=0;

   for i in [1 .. long] do

     for j in [1..2] do

      if Roo[i][j][2] gt 0 then

s:=s+(Roo[i][j][2])*(Roo[i][j][1])^(nn) ;

   else

   s:=s;

   end if;

end for;

end for;
 a:=Round(1+p^(nn)-s);
 PR:=Append(PR,a);

 end for;

return PR;

end function;
```

And we can run it easily, for example in the previous example we have:

```
p:=3;// A prime number not dividing the level

N bound:=20;/* For bound n where we will compute F_{p^n} points of
the modular curve*/

FpnpointsQuotientCurve:=FpnpointsforQuotientcurveX0NWN(N,p,HH[2],HH[3],bound);
```

```
print FpnpointsQuotientCurve; /*List the number of F_{p^n}-points
for the modular curve X0(N)/TN for n=1 until bound IF HH[1]=
11111111111111*/
```

And returns the following:

```
[* 7, 63, 28, 75, 97, 456, 2947, 7443, 18244, 56703, 175237, 541032,
1612111, 4748247, 14307148, 43097763, 129171481, 387621384,
1162245739, 3485359755 *]
```

Which are the list of the number of $\mathbb{F}_{3^n}$-points of $X_0(366)/\langle w_2, w_{61}\rangle$ from $n = 1$ until $n = 20$. In particular the number of $\mathbb{F}_{3^5}$-points of $X_0(366)/\langle w_2, w_{61}\rangle$ is 97.

subsection.1.4 1.4 Magma function if we can discard a degree d map between a quotient modular curve to a fix elliptic curve

Here we ASSUME that one obtained a $\mathbb{Q}$-decomposition of the Jacobian of the quotient modular curve. And p is a prime with $p \nmid N$ the level of $X_0(N)/W_N$.

We present a Magma code function such that...

Input:

Output:

```
MapdegreedtoEC:=function(prime,degree,bound,apCoefficientEC,FpNpointsModularCurveLi

p:=prime; a3:=apCoefficientEC; bod:=bound; deg:=degree;

PR2:=[* *];

C:=ComplexField(100); R<x>:=PolynomialRing(C);
cearrels:=Roots(x^2-a3*x+p,C);



for i in [1..bod] do



b:=deg*(p^i+1-Round(cearrels[1][1]^i+ p^i/cearrels[1][1]^i));

PR2:=Append(PR2,b); end for;

el:=#FpNpointsModularCurveList; tt:=Min(el,bod);

NoDegreeMaptosuchEC:=[**];
```

```
for k in [1..tt] do

difference:=(FpNpointsModularCurveList[k])-(PR2[k]);

Rr:=RealField(10); difference:=Rr!difference;

case Sign(difference):
 when 1:
 NoDegreeMaptosuchEC:=Append(NoDegreeMaptosuchEC,[*difference,p^k*]);

 end case;




end for;

return NoDegreeMaptosuchEC;

end function;
```

We can use for the case $X_0(366)/\langle 2, 61\rangle$ the Magma code:

```
degree:=2; // degree of the map of Quotient curve to elliptic curve

apcoefficient:=-2;/* Corresponds to E61a the a_p-coefficient of the
q-expansion of the associated modular form which appears in Jacobian
decomposition of quotient modular curve*/

NondegreedmaptoEC:=MapdegreedtoEC(p,degree,bound,apcoefficient,FpnpointsQuotientCur
print NondegreedmaptoEC ;/* List if any of the set (p^n, Integer)
n<bound such that
|X_0(N)/W_N(F_{p^n})|-degree*|EllipticCurve(F_{p^n})|=Integer, thus
no degree map between quotient modular curve to such elliptic curve
defined over the rationals if the list is not empty, recall p does
not divide N. */
```

And the result obtained running Magma is:

```
[* [* 39.00000000, 9 *] *]
```

For $X_0(366)/ < w_6 1, w_2 >$ and the elliptic curve $E61a$, we obtain that there is no $\mathbb{Q}$-rational map of degree 2 from the quotient modular curve to $E61a$ (by counting the $\mathbb{F}_9$-points), and the difference:

$$|X_0(366)/\langle w_2, w_{61}\rangle(\mathbb{F}_9)| - 2 * |E61a(\mathbb{F}_9)| = 39.$$

section.22 An ad-hoc modification to compute $\mathbb{Q}$-Jacobian decomposition of a quotient modular curve

Consider $X_0(308)/\langle w_{44}, w_{77}\rangle$.

By use the following Magma code with the Magma functions introduced (that we not reproduce here):

```
N := 308; // Level

TN := [* 44, 28, 77 *]; /* Need to list all non-trivial involutions
of the subgroup WN*/

t:=4; //Order of the subgroup of involutions that we work

genusquotientcurve:=genereX0NQuotientWN(N,TN,t);

print genusquotientcurve; /* Return the genus of the quotient
modular curve X0(N)/WN*/

prec := 30; // Number of coefficients of the q-expansion

HH := JacobianDecompositionQuotientX0NWN(N, TN, prec, t);

print HH[1]; /* Returns 0 or 000000000 implying that the function
NOT obtain que Q-Jacobian decomposition (there are too much reaped
modular forms!!!)*/

 print HH[2]; /* List the modular forms with
q-expansion  until O(q^prec) that appear in the Jacobian, (and
repetitions also some of the repetitions unnecessary if HH[1] is
0)*/

print HH[3]; /* List the number fields of the list of modular forms
given in HH[2].*/

print HH[4];/* List of the levels (recall a repeated form that
appears at level y can appear repeated at level y*k)*/
```

One obtains the following result:

```
11

0


[*
    q - 2*q^2 - q^3 + 2*q^4 + q^5 + 2*q^6 - 2*q^7 - 2*q^9 - 2*q^10 + q^11 -
        2*q^12 + 4*q^13 + 4*q^14 - q^15 - 4*q^16 - 2*q^17 + 4*q^18 + 2*q^20 +
```

```
        2*q^21 - 2*q^22 - q^23 - 4*q^25 - 8*q^26 + 5*q^27 - 4*q^28 + O(q^30),
    q - q^2 - 2*q^3 + q^4 + 2*q^6 + q^7 - q^8 + q^9 - 2*q^12 - 4*q^13 - q^14 +
        q^16 + 6*q^17 - q^18 + 2*q^19 - 2*q^21 + 2*q^24 - 5*q^25 + 4*q^26 +
        4*q^27 + q^28 - 6*q^29 + O(q^30),
    q - 2*q^2 - q^3 + 2*q^4 + q^5 + 2*q^6 - 2*q^7 - 2*q^9 - 2*q^10 + q^11 -
        2*q^12 + 4*q^13 + 4*q^14 - q^15 - 4*q^16 - 2*q^17 + 4*q^18 + 2*q^20 +
        2*q^21 - 2*q^22 - q^23 - 4*q^25 - 8*q^26 + 5*q^27 - 4*q^28 + O(q^30),
    q - 3*q^3 - 2*q^4 - q^5 - q^7 + 6*q^9 - q^11 + 6*q^12 - 4*q^13 + 3*q^15 +
        4*q^16 + 2*q^17 - 6*q^19 + 2*q^20 + 3*q^21 - 5*q^23 - 4*q^25 - 9*q^27 +
        2*q^28 + 10*q^29 + O(q^30),
    q - q^2 + q^4 - 4*q^5 - q^7 - q^8 - 3*q^9 + 4*q^10 - q^11 + 2*q^13 + q^14 +
        q^16 - 4*q^17 + 3*q^18 - 6*q^19 - 4*q^20 + q^22 + 4*q^23 + 11*q^25 -
        2*q^26 - q^28 - 2*q^29 + O(q^30),
    q + q^2 + q^4 + 2*q^5 - q^7 + q^8 - 3*q^9 + 2*q^10 - q^11 + 2*q^13 - q^14 +
        q^16 + 2*q^17 - 3*q^18 + 2*q^20 - q^22 - 8*q^23 - q^25 + 2*q^26 - q^28 -
        2*q^29 + O(q^30),
    q + q^2 + a*q^3 + q^4 - a*q^5 + a*q^6 + q^7 + q^8 + (-2*a + 1)*q^9 - a*q^10
        + q^11 + a*q^12 + (-a - 2)*q^13 + q^14 + (2*a - 4)*q^15 + q^16 +
        2*a*q^17 + (-2*a + 1)*q^18 + (-a - 6)*q^19 - a*q^20 + a*q^21 + q^22 +
        4*q^23 + a*q^24 + (-2*a - 1)*q^25 + (-a - 2)*q^26 + (2*a - 8)*q^27 +
        q^28 + (2*a + 2)*q^29 + O(q^30),
    q - 3*q^3 - 2*q^4 - q^5 - q^7 + 6*q^9 - q^11 + 6*q^12 - 4*q^13 + 3*q^15 +
        4*q^16 + 2*q^17 - 6*q^19 + 2*q^20 + 3*q^21 - 5*q^23 - 4*q^25 - 9*q^27 +
        2*q^28 + 10*q^29 + O(q^30),
    q + a*q^3 + (-a^2 + 4)*q^5 + q^7 + (a^2 - 3)*q^9 + q^11 + (a^2 + a)*q^13 +
        (a^2 - 2*a - 2)*q^15 + (-a^2 - 3*a + 4)*q^17 + 2*a*q^19 + a*q^21 + (a^2
        + 2*a - 6)*q^23 + (-a^2 - 4*a + 9)*q^25 + (-a^2 + 2)*q^27 + (-2*a^2 -
        2*a + 10)*q^29 + O(q^30)
*]


[*
    Rational Field,
    Rational Field,
    Rational Field,
    Rational Field,
    Rational Field,
    Rational Field,
    Number Field with defining polynomial x^2 + 2*x - 4 over the Rational Field,
    Rational Field,
    Number Field with defining polynomial x^3 + x^2 - 6*x - 2 over the Rational
    Field
*]


[* 11, 14, 22, 77, 154, 154, 154, 154, 308 *]
```

Because HH[1] is 0, (we have genus 11 and the dimension of the modular forms computed is

12) we need to choose conveniently the repeated modular forms that appear in HH[2] and fit the fields that corresponds to such modular forms to obtain the $\mathbb{Q}$-decomposition of the Jacobian of $X_0(308)/\langle w_{44}, w_{77}\rangle$. For doing this we use the results of Journal of Algebra paper [**?**] (which follows ideas from the Annals paper of Atkin-Lehner entitle "Hecke operators of $\Gamma_0(N)$"), for a detailed choose of such list of modular forms, consult the next subsection of this pdf file.

Ad-hoc magma code modification in order to obtain the $\mathbb{Q}$-decomposition

```
Newforms:=[*HH[2][1], HH[2][2],
HH[2][4],HH[2][5],HH[2][6],HH[2][7],HH[2][9]*]; /* Take only
newforms appears in Jacob.decomp.this can be check with HH[5] where
last factors says the AL-involution action and is old if the size of
the matrix corresponding to the AL-involution is bigger than the
dimension over the rationals of the field of definition of the
corresponding modular form
*/

NewHHForms:=[*HH[2][1], HH[2][2],
HH[2][4],HH[2][4],HH[2][5],HH[2][6],HH[2][7],HH[2][9]*];  /*Choise
the newforms and fix the times that appears repeated in the
Jacobian, see JA paper Bars-González, see also section 2 in
MagmacodeQuotientModularCurves.pdf in this github folder*/

NewHHFields:=[*HH[3][1], HH[3][2],
HH[3][4],HH[3][4],HH[3][5],HH[3][6],HH[3][7],HH[3][9]*];  /* The
corresponding fields of the modular forms*/
```

Now NewHHForms with NewHHFields, give us the $\mathbb{Q}$-decomposition of the Jacobian of $X_0(308)/\langle w_{44}, w_{77}\rangle$

Now in order to compute correctly $\mathbb{F}_{p^n}$-points of quotient modular curve or use the function if could be a degree d map of the quotient modular to a concrete elliptic curve (by counting points over a finite field) we use the ad-hoc list: NewHHForms and NewHHFields.

```
p:=3;// A prime number not dividing the level N

bound:=20;/* For bound n where we will compute F_{p^n} points of the
modular curve*/

FpnpointsQuotientCurve:=FpnpointsforQuotientcurveX0NWN(N,p,NewHHForms,NewHHFields,b

print FpnpointsQuotientCurve; /*List the number of F_{p^n}-points
for the modular curve X0(N)/TN for n=1 until bound (here with
HH[1]=0 with the ad-hoc modification for compute the Jacobian
decomposition)*/.
```

With output result counting the $\mathbb{F}_{3^n}$-points of $X_0(308)/\langle w_{44}, w_{77}\rangle$ for $n = 1$ until *bound*:

```
[* 16, 28, 28, 80, 176, 952, 1808, 6552, 19828, 59788, 177952,
526196, 1603280, 4780972, 14334668, 43035576, 129109984, 387641752,
1162321216, 3486190640 *]
```

section.3 3 An Example of computing the $\mathbb{Q}$-decomposition of the Jacobian of $X_0(N)/W_N$

Consider $(N = 308 = 2^2 \cdot 7 \cdot 11, W_N := \langle w_{28}, w_{44} \rangle)$ of genus 11. From the $\mathbb{Q}$-decomposition of $J_0(308)$ and the assumption $f \in \mathrm{New}_M$ with $M|N$ and for all $d||M$ with $w_d \in W_N$ $f \in \mathrm{New}_N^{w_d}$ we obtain to consider the following modular forms (that we can control Atkin-Lehner involutions for the power of primes that divides M):

```
f1=q - 2*q^2 - q^3 + 2*q^4 + q^5 + 2*q^6 - 2*q^7 - 2*q^9 - 2*q^10 + q^11 -
       2*q^12 + 4*q^13 + 4*q^14 - q^15 - 4*q^16 - 2*q^17 + 4*q^18 + O(q^20)
```

corresponds to $E11a$, now $w_{11} = -1$. We can lift to level 28, we need to leaf with $w_4 = -1$ because have to be fix by w_{44}, thus by the Proposition above, and there is only one possible lift, because we can lift by 2 with ± 1 but to obtain -1 is only a possibility. Now to lift from level 28 to 308 we need to lift by 7, and we need that lift with $w_7 = -1$ in order to obtain that w_{28} is fixed (because $w_4 = -1$ in level 28). Therefore $m_{f_1} = 1$. The number of different liftings of the modular form from level 11 to level 308 we use Lemma 2.1 and Proposition 2.2 in the Journal of Algebra paper F.Bars-J.González "Bielliptic modular curves $X_0^*(N)$" [**?**]. **The references in this section to previous Lemma and Proposition refers to Lemma 2.1 and Proposition 2.2 of such Journal of Algebra paper**.

```
f2:=q - q^2 - 2*q^3 + q^4 + 2*q^6 + q^7 - q^8 + q^9 - 2*q^12 -
4*q^13 - q^14 +  q^16 + 6*q^17 - q^18 + 2*q^19 + O(q^20),
```

corresponds to $E14a$ with $w_2 = +1$ and $w_7 = -1$ at level 14. We need to lift to 308. Lift first to $14 * 11 = 154$, thus by previous lemma and Proposition we only need to lift by one modular form that we choice the one with $w_{11} = -1$ because we want be fixed by w_{77} at level 154. To reach level 308 need to leaf by 2 thus only one possibility to lift with negative at two (following Lemma) because we want $w_4 = -1$. Thus such lift satisfies $w_4 = -1$, $w_7 = -1$ and $w_{11} = -1$ to become fix by W_N, thus $m_{f_2} = 1$.

```
f3:=q - 3*q^3 - 2*q^4 - q^5 - q^7 + 6*q^9 - q^11 + 6*q^12 - 4*q^13 +
3*q^15 + 4*q^16 + 2*q^17 - 6*q^19 + O(q^20),
```

corresponds to $E77a$ with $w_7 = 1$ and $w_{11} = 1$ thus to lift by 4 and reach level $77 \cdot 4 = 308$, we have two possibilities lift by + in step by step (at prime two) or lift by $-$ and $-$ in each step by step (at prime two). This gives two lifts of f_3 in level 308, thus $m_{f_3} = 2$.

```
f4:=q - q^2 + q^4 - 4*q^5 - q^7 - q^8 - 3*q^9 + 4*q^10 - q^11 +
2*q^13 + q^14 + q^16 - 4*q^17 + 3*q^18 - 6*q^19 + O(q^20);
```

is $E154a$ with $w_2 = w_7 = w_{11} = 1$ to lift by two to level 308, where we need $w_4 = 1$ we have only one possibility by act by $+1$ by previous Lemma and Proposition, thus $m_{f_4} = 1$.

```
f5:=q + q^2 + q^4 + 2*q^5 - q^7 + q^8 - 3*q^9 + 2*q^10 - q^11 +
2*q^13 - q^14 + q^16 + 2*q^17 - 3*q^18 +  O(q^20),
```

is $E154b$ with $w_2 = -1$ and $w_7 = w_{11} = 1$ to lift by two to level 308, where we need $w_4 = 1$ we have only one possibility by act by -1 at prime two to lift by previous Lemma and Proposition, thus $m_{f_5} = 1$.

```
f6:=q + q^2 + a*q^3 + q^4 - a*q^5 + a*q^6 + q^7 + q^8 + (-2*a +
1)*q^9 - a*q^10 + q^11 + a*q^12 + (-a - 2)*q^13 + q^14 + (2*a -
4)*q^15 + q^16 + 2*a*q^17 + (-2*a + 1)*q^18 + (-a - 6)*q^19
+O(q^20),
```

of level 154, with $w_7 = w_{11} = I_2$, thus we have to up to level 308 only one way that can be computed if we are interested by asking the action of AL-involution w_2 is this modular form. Here is unnecessary and $m_{f_6} = 1$. We observe that a is a root of $x^2 + 2x - 4$ thus A_{f_6} has dimension two. Finally,

```
f7:=q + a*q^3 + (-a^2 + 4)*q^5 + q^7 + (a^2 - 3)*q^9 + q^11 + (a^2 +
a)*q^13 + (a^2 - 2*a - 2)*q^15 + (-a^2 - 3*a + 4)*q^17 + 2*a*q^19 +
O(q^20)
```

of level 308 that is fix by W_N, where a a root of $x^3 + x^2 - 6x - 2$, thus $dim(A_{f_7}) = 3$. Thus we obtain the Jacobian decomposition of dimension 11 corresponding to the genus.

And easily we make an ad-hoc modification in the Magma programme to compute well the number of $\mathbb{F}_{p^n}$-points, see this example in the previous section.

References

cite.BaGon

[BG19] F. Bars and J. González. Bielliptic modular curves $X_0^*(N)$ with square-free levels. *Math. Comp.*, 88(320):2939–2957, 2019.

cite.BaGon2

[BG20] Francesc Bars and Josep González. Bielliptic modular curves $X_0^*(N)$. *J. Algebra*, 559:726–759, 2020.

Francesc Bars Cortina
Departament Matemàtiques, Edif. C, Universitat Autònoma de Barcelona
08193 Bellaterra, Catalonia
francesc@mat.uab.cat