Magma homework for hypergeometric motives

1. Construct the HGM given by Φ_3^2 and $\Phi_2^2\Phi_4$. What are its degree and weight? Determine the associated genus 2 curve, and for a given t-value (say $t = 8/27$) check that the Euler factors for the curve and the HGM match for all primes $5 \le p \le 100$.

b. Construct the HGM associated to the γ-vector with $\gamma_2 = 2, \gamma_4 = 1, \gamma_8 = -1$ (alternatively the γ-list given by $2, 2, 4, -8$). What are the cyclotomic parameters? What are the degree and weight? Determine the associated elliptic curve. What quadratic algebra is it defined over? For a given nonsquare t-value, check the Euler factors match. How can you check the matching of Euler factors when t is square?

Mathematical question: why is the first example a genus 2 curve and the second one an elliptic curve over a quadratic algebra?

2. Define the L-series of the hypergeometric datum (Φ_3^2, Φ_1^4) at $t = -8$. What does Magma guess the wild prime information is? Verify Magma's guess via `CFENew`.

More difficult: Numerically guess/determine if this L-series is primitive by computing the second moment of the $a_p/p^{w/2}$ values for $p \le 1000$. For this, the best way is to use `EulerFactor` with the `Degree` vararg set to 1, as else the computation of the full degree 4 Euler factor can be too time-consuming.

Construct the L-series for the elliptic curve 54a and Tate-twist it to have motivic weight 3. Take its product with the L-series for the integral newform of level 54 and modular weight 4 that has $c_{11} = -57$, and check (by comparing Euler factors for small primes) that this product L-series is the same as the L-series of the HGM.

b. Define the L-series of the hypergeometric datum (Φ_2^5, Φ_1^5) at $t = -4$. Compute the first and second moments of $a_p/p^{w/2}$ using (good) primes up to 1000. What does the nonzero first moment imply?

Now re-define the HGM L-series with a guess at the wild prime information at $p = 2$, and divide out by a translated ζ-function. Use `CFENew` to check your wild guess (it turns out the Swan conductor at 2 is 0).

3. List all the HGMs of degree 6 and weight 3 that have Hodge vector $[2, 1, 1, 2]$. How many are there? Find those with only one wild prime. Determine the parity of the functional equation for the $t = 1$ special HGM in each of these latter cases.

b. Experimentally determine (using guessing and `CFENew`) the conductor of the special HGM for (Φ_3^4, Φ_1^8) at $t = 1$. (Hint: the Magma guess of 3^6 is too small). Then numerically determine the analytic rank.

4. Consider the unipotent HGM $(\Phi_3\Phi_4, \Phi_1^4)$ at the special-ish point $t = -1$. Experimentally determine the conductor and bad Euler factors. (Hint: look at what Magma is guessing, and make minor modifications).

b. Consider the HGM $(\Phi_2^2\Phi_3, \Phi_1^2\Phi_4)$ at $t = 2$. Experimentally determine the conductor and bad Euler factors.

You can use the online Magma calculator: `http://magma.maths.usyd.edu.au/calc`

Answers to Magma homework for hypergeometric motives

Problem 1:

```
> H:=HypergeometricData([3,3],[2,2,4]);
> Degree(H),Weight(H);
4 1
> C:=HyperellipticCurve(H);
> F1<u>:=FunctionField(Rationals()); // make printing nicer
> C;
Hyperelliptic Curve defined by
  y^2 = x^6 + 2048/729/u*x^3 - 4096/729/u*x^2 + 1048576/531441/u^2
> t:=8/27;
> S:=Specialization(C,t); S; // could use HyperellipticCurve(H,t)
Hyperelliptic Curve y^2 = x^6 + 256/27*x^3 - 512/27*x^2 + 16384/729
> P:=PrimesInInterval(5,100);
> &and[EulerFactor(H,t,p) eq EulerFactor(S,p) : p in P];
true
```

Part b:

```
> H:=HypergeometricData([* 2,2,4,-8 *]); // defined by GammaList
> CyclotomicData(H);
[1,1,2,2] [8]
> Degree(H),Weight(H);
4 1
> E:=EllipticCurve(H);
> F1<u>:=FunctionField(Rationals()); // make printing nicer
> F2<v>:=PolynomialRing(F1); // again to make printing nicer
> E;
Elliptic Curve defined by y^2 + x*y = x^3 + v*x over
 Algebraic function field defined over
  Univariate rational function field over Q by v^2 - 1/4096*u
> t:=8/27;
> S:=EllipticCurve(H,t); // here Specialization does not work!
> K<w>:=BaseRing(S); // make printing nicer again
> S;
Elliptic Curve defined by y^2 + x*y = x^3 + w*x over K
> MinimalPolynomial(w);
y^2 - 1/13824
> P:=PrimesInInterval(5,100);
> &and[EulerFactor(H,t,p) eq EulerFactor(S,p) : p in P];
true
> t:=4;
> A:=EllipticCurve(H,t); A; // here it's actually two ECs
[* Elliptic Curve defined by y^2 + x*y = x^3 + 1/32*x over Q,
   Elliptic Curve defined by y^2 + x*y = x^3 - 1/32*x over Q *]
> &and[&*[EulerFactor(s,p) : s in A] eq EulerFactor(H,t,p) : p in P];
true
```

Mathematically, the second HGM is not primitive (the entries in the `GammaList` have nontrivial gcd).

Problem 2:

```
> H:=HypergeometricData([3,3],[1,1,1,1]);
> L:=LSeries(H,-8);
> BadPrimeData(L);
[ <2, 2, -4*x^2 + 1>, <3, 6, 1> ]
> CFENew(L);
0.000000000000000000000000000000
> P:=PrimesInInterval(5,1000);
> w:=Weight(H);
> &+[1.0*Coefficient(EulerFactor(L,p : Degree:=1),1)^2/p^w : p in P]/#P;
1.93685330418507282644047647201 // likely equal to 2, imprimitive
> E:=EllipticCurve("54a");
> LE:=Translate(LSeries(E),1);
> MotivicWeight(LE);
3
> NF:=Newforms(CuspForms(54,4)); // four forms, need right one
> g:=[f[1] : f in NF | Coefficient(f[1],11) eq -57][1];
> Lg:=LSeries(g);
> PROD:=LE*Lg;
> Q:=PrimesInInterval(5,100);
> &and[EulerFactor(H,-8,p) eq EulerFactor(PROD,p) : p in Q];
true
```

Part b:

```
> H:=HypergeometricData([2,2,2,2,2],[1,1,1,1,1]);
> L:=LSeries(H,-4);
> P:=PrimesInInterval(7,1000); // 5 is actually not good
> w:=Weight(H); // weight is 4
> w2:=w/2; // half the weight
> &+[1.0*Coefficient(EulerFactor(L,p : Degree:=1),1)/p^w2 : p in P]/#P;
-0.993270095522865738183190137971 // likely pole
> &+[1.0*Coefficient(EulerFactor(L,p : Degree:=1),1)^2/p^w : p in P]/#P;
1.98250726950774030607492361839 // again likely imprimitive
> _<x>:=PolynomialRing(Integers()); // define x
> BP:=[<2,4,1-2^w2*x>]; // translated zeta function as part of Euler
> L:=LSeries(H,-4 : BadPrimes:=BP); // 2-conductor is 2^4
> LZ:=Translate(RiemannZeta(),2);
> Q:=L/LZ;
> Q;
L-function for parameter -4 of Hypergeometric data given
   by [ 2, 2, 2, 2, 2 ] and [ 1, 1, 1, 1, 1 ] /
 Translation by 2 of L-series of Riemann zeta function
> Conductor(Q);
80
> CFENew(Q);
0.000000000000000000000000000000
```

Problem 3:

```
> PHD:=PossibleHypergeometricData(6 : Weight:=3);
> HD:=[HypergeometricData(h) : h in PHD];
> S:=[h : h in HD | HodgeVector(HodgeStructure(h)) eq [2,1,1,2]];
> #S;
79
> A:=[h : h in S | IsPrimePower(&*x*&*y) where x,y:=CyclotomicData(h)];
> A;
[ Hypergeometric data given by [ 2, 2, 8 ] and [ 1, 1, 1, 1, 4 ],
  Hypergeometric data given by [ 2, 2, 2, 2, 4 ] and [ 1, 1, 8 ] ]
> ///////////////////////////////////////////////////////////////
> L1:=LSeries(A[1],1);
> CentralValue(L1);
0.000000000000000000000000000000
> Sign(L1); // a bit tricky because Sign is "0" until computed
-1.00000000000000000000000000000 // odd parity
> LTaylor(L1,2,1); // if you want the value of the derivative at s=2
1.95005312742902851445366363287*$.1 + O($.1^2)
> ///////////////////////////////////////////////////////////////
> L2:=LSeries(A[2],1);
> CentralValue(L2);
1.35350562655467964866367304120
> Sign(L2);
1.00000000000000000000000000000 // even parity
```

For the second part, it is likely best to set up a loop and test conductors. Furthermore, using low precision is recommended.

Part b:

```
> H:=HypergeometricData([3,3,3,3],[1,1,1,1,1,1,1,1]);
> for l in [6..12] do
   L:=LSeries(H,1 : BadPrimes:=[<3,l,1>],Precision:=5);
   l,CFENew(L); end for;
6 0.24898
7 0.28503
8 0.10074
9 0.00000    <---- This one looks correct
10 0.11203
11 0.39868
12 0.19934
> L:=LSeries(H,1 : BadPrimes:=[<3,9,1>],Precision:=12);
> CFENew(L);
0.000000000000
> CentralValue(L); // vanishing central value
9.03926805762E-20
> Sign(L); // but even parity
1.00000000000
> _<z>:=LaurentSeriesRing(ComplexField(12),3); // printing
> LTaylor(L,(1+Weight(H))/2,2); // the analytic rank is 2
9.03926805762E-20 - 1.83660791950E-19*z + 3.24742005041*z^2 + O(z^3)
```

Problem 4:

```
> H:=HypergeometricData([3,4],[1,1,1,1]);
> t:=-1;
> BadPrimeData(LSeries(H,t));
WARNING: Guessing wild prime information
[ <2, 6, 8*x^2 + 1>, <3, 5, 1> ]
```

It turns out that Magma is correct at $p = 3$, and has the right Euler factor at $p = 2$, but has the wrong conductor at $p = 2$. Here is a loop to make some tests.

```
> _<x>:=PolynomialRing(Integers());
> for f2 in [1,1+8*x^2],k in [4..10],l in [4..6] do
   BP:=[* <2,k,f2>, <3,l,1> *];
   L:=LSeries(H,t : BadPrimes:=BP,Precision:=5);
   k,l,f2,CFENew(L); end for;
...
7 5 8*x^2 + 1 0.00000   <-- again this looks good
...
> // check this to higher precision
> BP:=[* <2,7,1+8*x^2>, <3,5,1> *];
> L:=LSeries(H,t : BadPrimes:=BP,Precision:=15);
> CFENew(L);
0.000000000000000
```

Part b:

```
> H:=HypergeometricData([2,2,3],[1,1,4]);
> t:=2;
> BadPrimeData(LSeries(H,t));
WARNING: Guessing wild prime information
[ <2, 8, -2*x + 1>, <3, 5, 1> ]
```

Here Magma is completely correct at $p = 2$, but should have 3^4 as the 3-conductor.

```
> _<x>:=PolynomialRing(Integers());
> for f2 in [1,1-2*x,1+2*x],k in [4..12],l in [4..6] do
   BP:=[* <2,k,f2>, <3,l,1> *];
   L:=LSeries(H,t : BadPrimes:=BP,Precision:=5);
   k,l,f2,CFENew(L); end for;
...
8 4 -2*x + 1 0.00000
...
> BP:=[* <2,8,1-2*x>, <3,4,1> *];
> L:=LSeries(H,t : BadPrimes:=BP,Precision:=15);
> CFENew(L);
0.000000000000000
```

Of course, Magma can often be much wronger, and more tests need to be made. There are better ways to modify L-series data so that not so much re-computation needs to be done (e.g., one can modify f_2 in the above without recalculating traces), but those methods are unnecessary here.