Coleman Extensions Examples

Sachi Hashimoto and Travis Morrison

September 9, 2020

Abstract

This shows some example calculations using the Coleman Extensions Magma library for computing Coleman integrals to number field points and run effective Chabauty for curves with known infinite order divisors.

1 A curve without an infinite order rational point

To load the code you must load both the Coleman library of Balakrishnan and Tuitman as well as the Coleman Extensions library:

```
> load "coleman.m";
> load "colemanextensions.m";
```

In this example we consider the Chabauty-Coleman calculation for the curve with the plane model $X : y^3 = x^4 + 7x^3 + 3x^2 - x$. First we compute the rank using `RankBounds`.

```
> f:=x^4 + 7*x^3 + 3*x^2 - x;
> RankBounds(f,3:ReturnGenerators:=true)
1 1 [*
    x^3 + 3*x^2 + 3*x - 1,
    x^3 + 3/4*x^2 + 3/16*x - 1/64,
    x,
    x^3 + 7*x^2 + 3*x + 1,
    x^3 + 3*x^2 - 2*x + 1,
    x^3 - x^2 + 3*x - 1,
    x^3 + 6*x^2 + 3*x - 1,
    x^3 + 6*x^2 + 3*x - 1,
    x^3 + 6*x^2 + 3*x - 1,
    x^3 + 6*x^2 + 3*x - 1,
    x^3 + 6*x^2 + 3*x - 1,
    x^3 + 6*x^2 + 3*x - 1
*]
```

Recall that the polynomials g_i in the list give divisors $\mathrm{Div}(g_i)$ on $\mathbb{P}^1$ which lift to rational divisors on X under the map $X \to \mathbb{P}^1$.

Since x represents a divisor on X whose class in the Jacobian J is torsion, we have to choose a different polynomial giving a divisor on X which is a sum of points defined over a degree three extension of $\mathbb{Q}$ in order to get an infinite order point of in $J(\mathbb{Q})$. Note that many degree three options will not be infinite order in $J(\mathbb{Q})$, for example $x^3 + 6x^2 + 3x - 1 = f - x^2$ and is torsion in J. We can check whether a polynomial will give a torsion point in $J(\mathbb{Q})$ by taking Coleman integrals on basis differentials over the associated divisor on X: if the integrals are all zero to p-adic precision, the divisor class is likely torsion.

For example we check if $x^3 + 7 * x^2 + 3 * x + 1$ is torsion. First we define g and f and find the prime over which we can define the divisor associated to g on X, by finding the first split prime not dividing the discriminant of the curve. The `Q_divs` function searches for rational points to a given bound and computes their associated data also computes divisor corresponding to g and its associated data in the form recognized by the code.

```
> f:=x^4 + 7*x^3 + 3*x^2 - x;
> g:=x^3 + 7*x^2 + 3*x + 1;
> first_split_prime(g,33264270000);
47
> p:=47;
> data:=coleman_data(y^3-f,p,10);
> Qdivs:=Q_divs(data,1000,g);
```

For example, here we get a point at $[0, 0, 1]$, the point at infinity, and also the divisor associated to g.

```
> Qdivs;
[
    rec<recformat<x, b, inf, xt, bt, index, divisor, L> |
        x := [ 0 ],
        b := [
            [ 1 + O(47^10), 0, 0 ]
        ],
        inf := false,
        divisor := false,
        L := Rational Field>,
    rec<recformat<x, b, inf, xt, bt, index, divisor, L> |
        x := [ 0 ],
        b := [
            [ 1 + O(47^10), 0, 0 ]
        ],
        inf := true,
        divisor := false,
        L := Rational Field>,
    rec<recformat<x, b, inf, xt, bt, index, divisor, L> |
        x := [ -14141859620773563 + O(47^10), 14932373399921992 + O(47^10),
            -790513779148436 + O(47^10), -14141859620773563 + O(47^10),
            14932373399921992 + O(47^10), -790513779148436 + O(47^10) ],
        b := [
            [ 1 + O(47^10), 5427448791236738 + O(47^10), 14141859620773562 +
                O(47^10) ],
            [ 1 + O(47^10), -1029226802316704 + O(47^10), -14932373399921993 +
                O(47^10) ],
            [ 1 + O(47^10), -4398221988920032 + O(47^10), 790513779148435 +
                O(47^10) ],
            [ 1 + O(47^10), 5427448791236738 + O(47^10), 14141859620773562 +
                O(47^10) ],
            [ 1 + O(47^10), -1029226802316704 + O(47^10), -14932373399921993 +
                O(47^10) ],
            [ 1 + O(47^10), -4398221988920032 + O(47^10), 790513779148435 +
                O(47^10) ]
        ],
        inf := false,
        divisor := true,
        L := Number Field with defining polynomial x^6 + 42*x^5 + 655*x^4 +
            4620*x^3 + 14095*x^2 + 13482*x + 4657 over the Rational Field>
]
```

We can integrate over the divisor associated to g on basis differentials to check if it corresponds to a torsion point in $J(\mathbb{Q})$ using the following commands:

```
> inf:=Qdivs[2];
> gdiv:=Qdivs[3];
```

```
> coleman_integrals_on_basis_to_div(inf,gdiv,data:e:=450);
(-10613812882497*47 + O(47^9) 123401116565*47 + O(47^9) -5409192954888*47 +
    O(47^9) 482515211169635 + O(47^9) -198795405305749 + O(47^9)
    -352379965042056 + O(47^9))
9
```

Showing the divisor corresponding to g is non-torsion can be done automatically using the `is_torsion` function:

```
> is_torsion(data,400,g);
false
> h:=x^3 + 6*x^2 + 3*x - 1;
> p:=first_split_prime(h,33264270000);
> is_torsion(data,200,h);
true
```

We will run Chabauty-Coleman using the divisor associated to g.

We can construct the splitting field of g and set the divisor corresponding to g manually (instead of using `Q_div` shown above) by computing the b_0^i vector $[1, y, y^2]$ using the following code. We start by defining the Coleman data from the Balakrishnan and Tuitman code, then we compute the points to set in the divisor.

```
> p:=47;
> N:=10;
> data:=coleman_data(y^3-f,p,N);
> L, roots:=SplittingField(x^3 + 7*x^2 + 3*x + 1);
> bvals:=[];
> for r in roots do
for> bool,b:=IsPower(Evaluate(f,r),3);
for> bvals:=Append(bvals,[1,b,b^2]);
for> end for;
> bvals;
[
    [
        1,
        1/2096*(3*L.1^5 + 109*L.1^4 + 1382*L.1^3 + 7174*L.1^2 + 13647*L.1 +
            6325),
        1/1048*(-3*L.1^4 - 84*L.1^3 - 682*L.1^2 - 1840*L.1 - 1807)
    ],
    [
        1,
        1/262*(-L.1^4 - 28*L.1^3 - 271*L.1^2 - 1050*L.1 - 646),
        1/524*(3*L.1^4 + 84*L.1^3 + 682*L.1^2 + 1316*L.1 + 235)
    ],
    [
        1,
        1/2096*(-3*L.1^5 - 101*L.1^4 - 1158*L.1^3 - 5006*L.1^2 - 5247*L.1 +
            3035),
        1/1048*(-3*L.1^4 - 84*L.1^3 - 682*L.1^2 - 792*L.1 + 5529)
    ]
]
> OL:=MaximalOrder(L);
> p_splitting:=Factorization(p*OL);
> prime_over_p:=p_splitting[1][1];
> inf:=false;
> divisor:=true;
> D:=set_bad_div(roots,bvals,inf,data,L,prime_over_p,divisor);
> D;
```

```
rec<recformat<x, b, inf, xt, bt, index, divisor, L> |
    x := [ -790513779148436 + O(47^10), -14141859620773563 + O(47^10),
        14932373399921992 + O(47^10) ],
    b := [
        [ 1 + O(47^10), -4398221988920032 + O(47^10), 790513779148435 + O(47^10)
            ],
        [ 1 + O(47^10), 5427448791236738 + O(47^10), 14141859620773562 +
            O(47^10) ],
        [ 1 + O(47^10), -1029226802316704 + O(47^10), -14932373399921993 +
            O(47^10) ]
    ],
    inf := false,
    divisor := true,
    L := Number Field with defining polynomial x^6 + 42*x^5 + 655*x^4 + 4620*x^3+
         14095*x^2 + 13482*x + 4657 over the Rational Field>
```

While manually setting divisor provides more flexibility, in most cases the user will want to define the divisor associated to g on the curve by simply defining the usual Coleman data from Balakrishnan and Tuitman and then run the `Q_divs` function:

```
> g:=x^3 + 7*x^2 + 3*x + 1;
> bound:=1000;
> Qdivs:=Q_divs(data, bound, g);
```

We can run effective Chabauty using this known infinite order divisor class to compute the vanishing differential using the function

```
> L,v:=effective_chabauty_with_Qdiv(data:Qpoints:=Qdivs,e:=200);
```

As in the code of Balakrishnan and Tuitman this returns a list of candidate points and annihilating differentials:

```
> L;
[
    rec<recformat<x, b, inf, xt, bt, index> |
        x := O(47^6),
        b := [ 1 + O(47^10), O(47^4), O(47^2) ],
        inf := true>,
    rec<recformat<x, b, inf, xt, bt, index> |
        x := O(47^6),
        b := [ 1 + O(47^10), O(47^2), O(47^4) ],
        inf := false>,
    rec<recformat<x, b, inf, xt, bt, index> |
        x := 4246669284 + O(47^6),
        b := [ 1 + O(47^10), O(47^2), O(47^4) ],
        inf := false>,
    rec<recformat<x, b, inf, xt, bt, index> |
        x := 1 + O(47^2),
        b := [ 1 + O(47^10), 584 + O(47^2), 870 + O(47^2) ],
        inf := false>
]
> v;
[
    [ 1 + O(47^3), O(47^3), 31133 + O(47^3) ],
    [ O(47^3), 1 + O(47^3), -41418 + O(47^3) ]
]
```

The vanishing differentials are computed by finding regular one-forms which vanishes on the integral over the divisor corresponding to g as well as all known rational points; this integral is computed by summing over the integrals to all finite points in the support of the divisor from the basepoint ∞.

2 An endomorphism example

One can also use the code to compute integrals to number field points, for example to check if a point is a torsion point in the Jacobian under the Abel–Jacobi map. Here we exhibit another example of points on a Picard curve which are not explained by torsion or linearity, but instead come from the presence of extra endomorphisms of the curve. The curve is a Picard curve of Mordell–Weil rank 2 given by the plane model $y^3 - (x^4 - 2)$ and we can run Chabauty as in the example above at $p = 5$ to get 5-adic points which can be recognized as global points $Q_1 = [i, -1, 1]$ and $Q_2 = [-i, -1, 1]$.

There is a function which computes basis integrals to number field points, allowing one to quickly test relations with Coleman integrals:

```
> data:=coleman_data(y^3 - (x^4 - 2), 5, 10);
> K<i>:=NumberField(x^2+1);
> Kpoints:=[[ i,-1,1], [-i,-1,1]];
> ratpts:=[];
> e:=20;
> compute_integrals(data, ratpts, Kpoints, e);
[
    (-57*5^4 + O(5^7) -8*5^2 + O(5^7) -7764*5 + O(5^7) 7398*5 + O(5^7) -4876 +
        O(5^7) -21441 + O(5^7)),
    (57*5^4 + O(5^7) -8*5^2 + O(5^7) 7764*5 + O(5^7) -7398*5 + O(5^7) -4876 +
        O(5^7) 21441 + O(5^7))
]
```

For example, this shows that $[Q_1 + Q_2 - 2\infty]$ is probably torsion while $[Q_i - \infty]$ is not torsion. (We can further verify $[Q_1 + Q_2 - 2\infty]$ is torsion using algebraic tests.) More generally, one can use `compute_integrals` to easily compute all basis integrals between an enumerated sequence of rational points (replacing the empty sequence) and an enumerated sequence of points defined over a number field K.