Part II. Mucking Around

In this part I collect together some computations covering topics that have been presented during the course of this semester's program.

1. Smoothness and class relations.
2. Modular curves and isogenies.
3. Arithmetically equivalent fields.

1. Smoothness and class relations.

Problem. Given an order in an imaginary quadratic field, and a set of split primes, find a set of defining relations for the subgroup of the class group generated by these ideals.

Solution. Extend the set of split primes to a smoothness base, form random relations and reduce as binary quadratic forms, and test the resulting relations for smoothness.

```
> Q := BinaryQuadraticForms(-10^7+29);
> Q;
Binary quadratic forms of discriminant -9999971
> ClassNumber(Q);
877
> IsPrime($1);
true
```

```
> time ClassRelations(Q,12 : ExtraGenerators := 20);
[ 0  1  0  0  0  0  0  0 -1  1  0 -1]
[ 0  0  1  0  0  1  1 -1  0  0  0  0]
[ 0  0  0  0  1  0  0  1  1  0  1  0]
[ 1 -1  0  0  0 -1  0  0 -1 -1  0  0]
[ 1 -1  0  0  0  0 -1  1  0  0 -1  0]
[ 1  0  0  1  0  0  0  1  1  0  0 -1]
[ 1  0  0  0  0  1  0  0 -1  0  1  1]
[ 0  1 -1  0  0 -1  0  0  0 -1 -1  0]
[ 0  0  0  1  1 -1  0  1  0  0 -1  0]
[ 0  0  0  1  0  0 -1 -1  1  0 -1  0]
[ 1 -1 -1  0  0  0  0 -1  0  0 -1 -1]
[ 1  0  1  0  0  0  0 -1  1  1  1  1]
[ <3,1,833331>, <5,3,499999>, <31,3,80645>,
 <43,25,58143>, <53,49,47181>, <59,23,42375>,
 <61,53,40995>, <67,39,37319>, <71,33,35215>,
 <89,43,28095>, <97,27,25775>, <103,89,24291> ]
Time: 0.109
```

Bonus solution. The Jacobian J of a hyperelliptic curve

$$y^2 + h(x)y = f(x)$$

over a finite field F has an analogous concept of smoothness, using the monoid of ideals in the ring

$$F[x, y]/(y^2 + h(x)y - f(x))$$

which we can apply to solve for the group structure of $J(F)$.

```
> FF<w> := FiniteField(2^3);
> C := HyperellipticCurve(x+1,x^31+x+w);
> C;
Hyperelliptic Curve defined by y^2 + (x + 1)*y =
 x^31 + x + w over GF(2^3)
> J := Jacobian(C);
> Genus(C);
15
```

```
> time JacobianRelations(J,12 : ExtraGenerators := 96);
[  2   0   0   0   0   0   0   0   0   0   0   0]
[  1  -5  -6  -2  -3   1  -3   4   0   6  -8  -1]
[  0   6  -2  -1   3  -2  10   0  -1   2   6   4]
[  1   0  -1   0   8  -7  -6  -1 -10  -3   5  -1]
[  0   6  -4 -14  -4   2  -4  -1   5  -4   3  -3]
[  1  11   4   0   0   1  -5   2  10   7  -1  -6]
[  1   2  -3  11  -2   5  -6   3 -11   0   5  -2]
[  0   7   4 -13  -8   0  -2 -11  -2   4   5   3]
[  0   5   3   5  11   7  10  -7  -7   1  -5  -6]
[  0   9  -5   3  -8   1  -9 -13   9   0   3   1]
[  0  13  -2  -5   1   0  -4  -6   2   2 -13  13]
[  1   3   5   3  -7  -9  -3  11  -5  -3   6 -15]
[ (x + 1, w^4, 1), (x + w, w^4, 1), (x + w^3, w^2, 1),
 (x + w^5, 1, 1), (x, w^2, 1), (x^2 + w^4*x + w^4, w*x + w, 2),
 (x^2 + w^3*x + w^6, x + w^6, 2), (x^2 + w^2*x + w^3, w^6*x + w^6,
 (x^2 + w^3*x + w^4, w^5*x, 2), (x^2 + w^5*x + w^6, w*x, 2),
 (x^2 + x + 1, w^4*x + w, 2), (x^2 + w^6*x + w, w^4*x + w^2, 2) ]
Time: 194.319
> M, gens := $1;
> Determinant(M);
42113713170722
> Factorization($1);
[ <2, 1>, <269, 1>, <150743, 1>, <519283, 1> ]
> S := [ M[i] : i in [1..12] ];
> for v in S do
>    &+[ v[i]*gens[i] : i in [1..#gens] ];
> end for;
(1, 0, 0)
(1, 0, 0)
(1, 0, 0)
(1, 0, 0)
(1, 0, 0)
(1, 0, 0)
(1, 0, 0)
(1, 0, 0)
(1, 0, 0)
(1, 0, 0)
(1, 0, 0)
(1, 0, 0)
```

2. Modular curves and isogenies.

```
> P<x> := PolynomialRing(RationalField());
> K<e> := NumberField(x^2-5*x-1);
> E0 := EllipticCurve([1,0,e^2,0,0]);
> X0 := ModularCurveX0(2);
> ModuliPoints(X0,E0);
[]
> X0 := ModularCurveX0(3);
> ModuliPoints(X0,E0);
[
(-702 + 135*e, -1/27 - 5/27*e)
]
> P0 := $1[1];
> f := ModularIsogeny(E0,P0);
> E1 := Codomain(f);
> E1;
Elliptic Curve defined by y^2 + x*y + (1 + 5*e)*y =
 x^3 + (-5 - 25*e)*x + (-183 - 950*e) over K
```

```
> X0 := ModularCurveX0(5);
> ModuliPoints(X0,E0);
[
(-9 - e, -14 + e)
]
> Q0 := $1[1];
> g := ModularIsogeny(E0,Q0);
> E2 := Codomain(g);
> E2;
Elliptic Curve defined by y^2 + x*y + (1 + 5*e)*y =
 x^3 + (-88 - 457*e)*x + (-1755 - 9113*e) over K
> // Find the last element of the isogeny class.
> Q1 := ModuliPoints(X0,E1)[1];
> h := ModularIsogeny(E1,Q1);
> E3 := Codomain(h);
> E3;
Elliptic Curve defined by y^2 + x*y + (1 + 5*e)*y =
 x^3 + (-7178 - 37272*e)*x + (-1238246 - 6429694*e)
 over K
```

Putting this all together, here is a hypothetical session in which we analyze the isogeny class of `E0`.

```
> C := IsogenyClass(E0);
> C;
Isogeny class of Elliptic Curve defined by y^2 + x*y
 + (1 + 5*e)*y = x^3 over K
> #C;
4
> [ jInvariant(E) : E in Representatives(C) ];
[
-18233 + 3515*e,
-7309228334338 + 1407628760845*e,
-658 - 3515*e,
-271084530113 - 1407628760845*e
]
> { MinimalPolynomial(j) : j in $1 };
{
x^2 + 18891*x - 357911,
x^2 + 7580312864451*x + 144612187806169
}
```

3. Arithmetically equivalent fields.

The construction of Bart de Smit of for arithmetically equivalent fields comes from Galois representation on $E[p]/{\mathbb{F}_p^*}^2$. Since -1 is a square modulo a prime p with $p \equiv 1 \bmod 4$, we can realize the points on modular curves to construct these extensions.

To be continued.