Sage Days 16: ECHIDNA

Elliptic Curves and Higher Dimensional Analogues

Open source Magma extensions in Sage

http://echidna.maths.usyd.edu.au/kohel/alg/

This is a repository of of open source GPL code which represents many years of research with many collaborators. Once you have installed the echidna package, to attach ECHIDNA to your Magma shell in Sage you just do:

```
magma.attach_echidna()
```

In addition there are various databases which may be of interest for your research in arithmetic geometry, complex multiplication, cryptography, modular forms, and other areas of number theory.

First, a few sensible definitions...

```
%magma
ZZ := IntegerRing();
F<x> := FunctionField(ZZ);
Factorization((x^7-x)/(x^8-1));
```

```
[
<x, 1>,
<x^2 - x + 1, 1>,
<x^2 + x + 1, 1>,
<x^2 + 1, -1>,
<x^4 + 1, -1>
]
```

```
%sage
F.<x> = FunctionField(ZZ)
((x^7-x)/(x^8-1)).factor()
```

```
x * (x^2 + 1)^-1 * (x^2 - x + 1) * (x^2 + x + 1) * (x^4 + 1)^-
```

```
%magma
Factorization((2^7-2)/(2^8-1))
```

```
[ <2, 1>, <3, 1>, <7, 1>, <5, -1>, <17,
-1> ]
```

```
%sage
((2^7-2)/(2^8-1)).factor()
```

```
2 * 3 * 5^-1 * 7 * 17^-1
```

```
%magma
F<x,y> := FunctionField(ZZ,2);
f := x*(x^6-y^6)/(x^8-y^8);
f([2,1])
```

```
42/85
```

```
%magma
Factorization(f([2,1]));
```

```
[ <2, 1>, <3, 1>, <7, 1>, <5, -1>, <17,
-1> ]
```

```
%sage
F.<x,y> = FunctionField(ZZ,2)
f = x*(x^6-y^6)/(x^8-y^8)
f([2,1])
```

```
42/85
```

```
%sage
f([2,1]).factor()
```

```
2 * 3 * 5^-1 * 7 * 17^-1
```

Brandt modules

```
%sage
M = BrandtModule(71)
M.decomposition(bound=7)
```

```
[
Subspace of dimension 1 of Brandt module of dimension 7 of lev
of weight 2 over Rational Field,
Subspace of dimension 3 of Brandt module of dimension 7 of lev
of weight 2 over Rational Field,
Subspace of dimension 3 of Brandt module of dimension 7 of lev
of weight 2 over Rational Field
]
```

```
%sage
for N in M.decomposition(bound=7):
   print N
   print N.basis_matrix()
```

```
Subspace of dimension 1 of Brandt module of dimension 7 of lev
of weight 2 over Rational Field
[  1   2   2   2   2   2 2/3]
Subspace of dimension 3 of Brandt module of dimension 7 of lev
of weight 2 over Rational Field
[ 1  0 -1  0  0  1 -1]
[ 0  1 -1  0  1 -1  0]
[ 0  0  0  1  0 -1  0]
Subspace of dimension 3 of Brandt module of dimension 7 of lev
of weight 2 over Rational Field
[   1    0    0 -1/2 -1/2 -1/2  1/2]
[   0    1    0    0   -1    0    0]
[   0    0    1 -1/2  1/2 -1/2 -1/2]
```

```
%magma
M := BrandtModule(71);
Decomposition(M,7);
```

```
[
Brandt module of level (71,1), dimension 1, and degree 7 over
Integer Ring,
Brandt module of level (71,1), dimension 3, and degree 7 over
Integer Ring,
Brandt module of level (71,1), dimension 3, and degree 7 over
Integer Ring
]
```

```
%magma
for N in Decomposition(M,7) do
   print N;
   print BasisMatrix(N);
end for;
```

```
Brandt module of level (71,1), dimension 1, and degree 7 over
Integer Ring
[3 2 6 6 6 6 6]
Brandt module of level (71,1), dimension 3, and degree 7 over
Integer Ring
[ 1 -1  0  0 -1  2 -1]
[ 0  0  1  0  0 -1  0]
[ 0  0  0  1 -1  1 -1]
Brandt module of level (71,1), dimension 3, and degree 7 over
Integer Ring
[ 1  0 -1  1  0 -1  0]
[ 0  1  1 -2  0  1 -1]
[ 0  0  0  0  1  0 -1]
```

```
%sage
M = BrandtModule(71,2)
M.decomposition(bound=7,anemic=False)
```

```
[
Subspace of dimension 1 of Brandt module of dimension 18 of le
71*2 of weight 2 over Rational Field,
Subspace of dimension 1 of Brandt module of dimension 18 of le
71*2 of weight 2 over Rational Field,
Subspace of dimension 1 of Brandt module of dimension 18 of le
71*2 of weight 2 over Rational Field,
Subspace of dimension 1 of Brandt module of dimension 18 of le
71*2 of weight 2 over Rational Field,
Subspace of dimension 1 of Brandt module of dimension 18 of le
71*2 of weight 2 over Rational Field,
Subspace of dimension 1 of Brandt module of dimension 18 of le
71*2 of weight 2 over Rational Field,
Subspace of dimension 3 of Brandt module of dimension 18 of le
71*2 of weight 2 over Rational Field,
Subspace of dimension 3 of Brandt module of dimension 18 of le
71*2 of weight 2 over Rational Field,
Subspace of dimension 3 of Brandt module of dimension 18 of le
71*2 of weight 2 over Rational Field,
Subspace of dimension 3 of Brandt module of dimension 18 of le
71*2 of weight 2 over Rational Field
]
```

```
%magma
M := BrandtModule(71,2);
Decomposition(M,7);
```

```
[
Brandt module of level (71,2), dimension 1, and degree 18 over
Integer Ring,
Brandt module of level (71,2), dimension 1, and degree 18 over
Integer Ring,
Brandt module of level (71,2), dimension 1, and degree 18 over
Integer Ring,
Brandt module of level (71,2), dimension 1, and degree 18 over
Integer Ring,
Brandt module of level (71,2), dimension 1, and degree 18 over
Integer Ring,
Brandt module of level (71,2), dimension 1, and degree 18 over
Integer Ring,
Brandt module of level (71,2), dimension 3, and degree 18 over
Integer Ring,
Brandt module of level (71,2), dimension 3, and degree 18 over
Integer Ring,
Brandt module of level (71,2), dimension 3, and degree 18 over
Integer Ring,
Brandt module of level (71,2), dimension 3, and degree 18 over
Integer Ring
]
```

Quaternion algebras, lattice genera, and Hecke operators

We can enumerate all representative lattices in a genus.

```
%magma
p := 389;
O := QuaternionOrder(p);
L := NormModule(O);
G := Genus(L);
print G;
// time Representatives(G); // 4 minutes to enumerate 319
representatives...
```

```
Genus of Standard Lattice of rank 4 and degree 4
Determinant: 151321
Factored Determinant: 389^2
Inner Product Matrix:
[  2   0   1   1]
[  0   4  -1   2]
[  1  -1  98   0]
[  1   2   0 196]
```

Even without enumeration, we can verify equality of two genera locally:

```
%magma
A = Matrix([ [2, -1, 0, 1], [-1, 2, 1, -1], [0, 1, 260, -130],
[1, -1, -130, 260] ]);
M = RSpace(ZZ,4,A);
```

```
F = Genus(M);
F eq G;
```

```
true
```

Following the idea of Birch, we can create a ternary lattice which captures half of the Hecke eigenspace of weight 2 modular forms.

```
%magma
A = GramMatrix(L);

D = Matrix(3,[ A[1,1]*A[i,j] - A[1,i]*A[1,j] : i, j in [2..4]
]);
N = LatticeWithGram(D);

N = LatticeWithGram(2*p*GramMatrix(Dual(N)));
N;
```

```
Standard Lattice of rank 3 and degree 3
Determinant: 778
Factored Determinant: 2 * 389
Inner Product Matrix:
[ 2  0 -1]
[ 0  4  1]
[-1  1 98]
```

The number of representatives is determined by naive enumeration;

```
%magma

H := Genus(N);
time #H;
```

```
22
Time: 1.550
```

which should be improved using the mass formula (which we can also compute naively):

```
%magma
auts = [ #AutomorphismGroup(X) : X in Representatives(H) ];
print auts;
print "Mass:", &+[ 1/m : m in auts ];
```

```
[ 4, 4, 4, 4, 2, 2, 4, 2, 4, 2, 4, 2, 2, 2, 2, 4, 2, 4, 2, 2,
]
Mass: 97/12
```

Now we can follow Birch's idea for computing Hecke operators as the adjacency matrices of Kneser's neighboring method:

```
%magma
for p in [2,3,5,7] do
   time Tp := AdjacencyMatrix(H,p);
   Factorization(CharacteristicPolynomial(Tp));
end for;
```

```
Time: 1.140
[
<x - 3, 1>,
```

```
<x + 2, 1>,
<x^20 - 3*x^19 - 29*x^18 + 91*x^17 + 338*x^16 - 1130*x^15 -
2023*x^14 + 7432*x^13 + 6558*x^12 - 28021*x^11 - 10909*x^10 +
61267*x^9 + 6954*x^8 - 74752*x^7 + 1407*x^6 + 46330*x^5 - 1087
12558*x^3 - 942*x^2 + 960*x + 148, 1>
]
Time: 1.550
[
<x - 4, 1>,
<x + 2, 1>,
<x^20 - 11*x^19 + 19*x^18 + 204*x^17 - 845*x^16 - 781*x^15 +
8883*x^14 - 6177*x^13 - 40916*x^12 + 63058*x^11 + 85034*x^10 -
215618*x^9 - 46920*x^8 + 342529*x^7 - 84612*x^6 - 241030*x^5 +
112365*x^4 + 51018*x^3 - 28526*x^2 + 3560*x - 100, 1>
]
Time: 2.270
[
<x - 6, 1>,
<x + 3, 1>,
<x^20 - x^19 - 58*x^18 + 69*x^17 + 1338*x^16 - 1962*x^15 -
15578*x^14 + 28633*x^13 + 93460*x^12 - 224324*x^11 - 236982*x^
902782*x^9 - 92649*x^8 - 1549758*x^7 + 1240027*x^6 + 457997*x^
897661*x^4 + 293181*x^3 + 17361*x^2 - 16713*x + 757, 1>
]
Time: 2.540
[
<x - 8, 1>,
<x + 5, 1>,
<x^20 - 12*x^19 - 8*x^18 + 602*x^17 - 1355*x^16 - 11751*x^15 +
44797*x^14 + 105012*x^13 - 632038*x^12 - 274991*x^11 + 4756743
- 2413492*x^9 - 19377380*x^8 + 21737168*x^7 + 37613472*x^6 -
64826048*x^5 - 17117376*x^4 + 68169472*x^3 - 23637760*x^2 -
4162560*x + 1715200, 1>
]
```

We can also graph these neighboring relations.

Now let's check that this gives reasonable Hecke eigenvalues:

```
%magma

E := EllipticCurve(DBEC,389,1,1);

[ TraceOfFrobenius(E,GF(p)) : p in [2,3,5,7] ];
```

```
[ -2, -2, -3, -5 ]
```

Heights on elliptic curves

```
%magma
K<i> := QuadraticField(-1);
EK := E(K);
```

```
P1 := EK![0,0,1];
P2 := EK![1,0,1];
Q1 := EK![ -13/4, 1/8*(33*i - 4), 1 ];
IsLinearlyIndependent([P1,P2,Q1]);
```

```
true
```

```
%sage
show(magma("HeightPairingMatrix([P1,P2,Q1]);").sage())
```

$$\begin{pmatrix} 0.32700077365160495184435215238 & 0.058522674844878951747663984109 \\ 0.058522674844878951747663984109 & 0.47671165934373953738278 \\ -6.7762635780344027125465800054 \times 10^{-21} & 1.0164395367051604068819870008 \times \end{pmatrix}$$

Elliptic curves and canonical lifting algorithms

```
%magma
FF<s> := FiniteField(3,71);
E := EllipticCurveWithjInvariant(Random(FF));
q := #FF;
time t := AGMTrace(E);
print "Trace:", t;
print "Order:", q + 1 - t;;
```

```
Time: 0.230
Trace: -145710267643441352
Order: 75094665149797249496569836016989000
```

For several years this was the fastest algorithm available, but Mike Harrison has now implemented canonical lifting algorithms in Magma...

```
%magma
time "Order:", #E;
```

```
Order: 75094665149797249375037196892106631
Time: 0.140
```

```
%magma
FF := FiniteField(13,117);
E := EllipticCurveWithjInvariant(Random(FF));
time t := TraceOfFrobenius(E);
```

```
Time: 1.680
```

```
%magma
time t eq AGMTrace(E);
```

```
true
Time: 6.510
```

Exercise: Port this code to Sage, then improve and extend it.

... using Magma in Sage ...

Caution: FF = FiniteField(3,71) doesn't do what you might think.

```
%sage
FF.<s> = FiniteField(3^71)
E = EllipticCurve_from_j(FF.random_element())
# Don't try this: E.trace_of_frobenius()
q = FF.cardinality()
t = ZZ(magma(E).AGMTrace())
E._order = q + 1 - t
print "Trace:", E.trace_of_frobenius()
print "Order:", E.order()
```

```
Trace: -158953772870405074
Order: 75094665149797249629004888828662622
```

Quartic CM fields, genus 2 curves, and CM Igusa invariants

```
%magma
DBCM := QuarticCMFieldDatabase();
DBG2 := Genus2CurvesDatabase();
DBIX := IgusaLIXDatabase();
```

```
%magma
DABInvs := QuarticCMFieldInvariantsWithClassNumber(DBCM,8,2);
#DABInvs;
```

```
2108
```

```
%magma
DAB := DABInvs[1];
print "DAB:", DAB;
print "Igusa:", DAB in DBIX;
```

```
DAB: [ 5, 27, 151 ]
Igusa: true
```

```
%magma
K<t> := QuarticCMField(DAB);
K;
```

```
Number Field with defining polynomial x^4 + 27*x^2 + 151 over
Rational Field
```

Exercise: Create a class for quartic CM fields in Sage.

```
%magma
DABInvs[[1..16]];
```

```
[
[ 5, 27, 151 ],
[ 5, 27, 171 ],
[ 5, 28, 191 ],
[ 5, 30, 205 ],
[ 5, 30, 220 ],
```

```
[ 5, 34, 284 ],
[ 5, 35, 305 ],
[ 5, 36, 279 ],
[ 5, 36, 319 ],
[ 5, 37, 311 ],
[ 5, 38, 341 ],
[ 5, 38, 356 ],
[ 5, 39, 319 ],
[ 5, 39, 369 ],
[ 5, 40, 380 ],
[ 5, 40, 395 ]
]
```

```
%magma
#[ DAB : DAB in DABInvs | DAB in DBIX ];
```

```
835
```

The Igusa class polynomials (or ideal) is, for CM genus 2 curves, the analogue of the Hilbert class polynomial for CM elliptic curves.

```
%magma
IgLIX := IgusaLIXInvariants(DBIX,DAB)[1];
IgLIX;
```

```
[
<5112314186600020249170343944188957801051984816 01*x^16 -
3945729417661390367738634862588716060505296443923606990507022
5 +
7146133915704585755312190766883571188405947862812833171140599
06876002304*x^14 -
4807211592019015838843780018245202740492282754331967254928928
4119416054936649728*x^13 -
4032187998111699625577246516793204009897142099522137548111781
1
4518722374330818984509440*x^12 -
9108214928976691566871446582123857378670906503198523251492103
3
5426587482227768002081004191744*x^11 -
8211230878991848717079170712179603475131387105666993580905894
8
4434017654278327495736817812504576000*x^10 -
2586265709980233335423586277696550284926725287919344979019745
2
9807354439554222744939743165418658781462528*x^9 -
7435953976875041886867925908087784268653431450648222615564489
2
3868646848937767581548075372424851457160731688 96*x^8 -
5559018977985786766645284095767397156234669279490549262220755
6
5586923806817978471063076508625296547660826882867 2000*x^7 -
1903333063107658888365127669217562989616807120239121760197449
6
6408640401662509122562871392932842571735553076625408000000*x^
6
3580871906354256462944065200534302942917475186995717661957184
4
9902463215199534743377557525272371230188559921473126400000000
0
4142495387628541664589305557979760132186417661722660534128479
6
5684939144550788797176225201694582354926449085068083200000000
0
^4 -
2987284460437993856939692657470295917161891109246904372073290
5
4883960083540100214771626254858225464892456447690932224000000
0
00*x^3 -
```

```
122309740785524793287413112615004557872812233260200722552275 23
679102056012346287364746938262081934091986177662365204480000 0C
000000*x^2 +
831580969190747761079569841968066176011487148098794417796670 3
120368215317750626120041087710276824126822471317638348800000 0C
00000000*x -
252756652251178609806279149376365836679273686302568260052575 75
592523661781114653987166412127435849632137257424624025600000 0C
00000000000, 1>,
<201450365335153895726989989332226170465725355447744573776157 7
062027954783460040033356416*x^15 -
717669626060388500090907251246194087798518372548527971466882 99
760277037690000818136247343652290734 08*x^14 +
180994279434160406820444169021572756873214998582552668218904 92
707289512359748919991742505240446267096707891 2*x^13 +
115853198039385723128162767582560246943274593856864860014335 84
340868418549860228678557761331897450622504019551846 40*x^12 +
207299750489575450788018225633157275526984055904561125376979 87
602335894825891989862156496102309804006794551232817413488 64*x^
116905490789123716294113446379520887381255492482553109470197 07
102472455062202018579338778278695984658102098749333820623395 22
10 +
224431573887055157805252672718695077222642473423521676050367 05
315258241449986688786808916361216805511032746082038263067053 65
56*x^9 +
392652231777766843141365189480774248410906216724497021915225 65
731101103686218381319655519229438709144398006113319033160206 46
5164800*x^8 -
103837393761338227089832182182330218644476306589609911278016 91
254333206436479897356067210021719356874546207654529219649468 27
9226496000000*x^7 -
127731825395857166354940703638834611846068657946706503914078 66
375502683699665339068579131934873285648458055705010559455351 84
398684262400000000*x^6 -
504697236883839758664971200736941513237138973461958779422774 15
432576761049030113233811883481781887182154478411596869663560 41
748001853440000000000*x^5 -
792523870044329673917061984584142921768904490954259541640827 18
582160169979113685338465318702034625428047242810166467694139 51
869771018240000000000000000*x^4 -
472307383820839996860733518170467439641304943583811752986409 29
804958854810208018269452013379441950001790136199844352491753 25
937673825484800000000000000000*x^3 +
695914027734778788489763847130447540102124249967188207155373 67
403578612670230699521510520545390489800675208086120491373493 17
625216139264000000000000000000000*x^2 -
332850026424769025849072799130070798976246033016559072781178 39
515873603664554716423718360115345100390774217640502035491699 07
355513014681600000000000000000000*x +
681177754790413250922102743815175613955416677043078265545410 29
025145445080681048925900484283193170571751560198945352909164 35
372209065984000000000000000000000000,
655990569920358508673394 9>,
<156606499455236502995279895629796917152383596516390611594133 4
```

```
474464321849080105692332032*x^15 -
556358684359614786967985881984385291801802613282700560900286595
059002685671059691976287684784926699 52*x^14 +
157426376636039476188592947477737474543273972705169571336355195
435152394035514331155903098505118879385052774 4*x^13 +
939275026072738455465899674794177200234027220034160842002726588
414802582753573456588141585950702194594367458508800 0*x^12 +
152056212560741538499767972222437266841919802853781226913870088
407663247965513073258756103386839325577774885872976862904 32*x^
666594662254826531119265380041213162963387292263658887143951466
065146882028167710625814685718079801483110596930915197821072170
0 +
801840555394085824037121537811347118641324833129191959329034966
075315248844318623733920697493136709899174246339766041522578258
8*x^9 -
602539178207848597903781666779409440633264064513618498789850838
700063213105112598635372989149618374954604196830028707040364848
046848*x^8 -
125345073159441043187322379867782223339334580713438712116875433
057452320353395657670518439099721512928827322880647019954224844
5492229120000*x^7 -
126294878197237598817187738399270788425066914048750163086733644
106106596998064572111607094212450062082646747030145919508762799
798360211456000000*x^6 -
465637156186100538815393309209104467440053594687304630005160122
235408983642985301077117953388665027885306066042834733778716114
493087481856000000000*x^5 -
681372681481966827148360997985606690571036278973128323628823299
309165407774902688406743577261667096200728841851997123699020882
250050936832000000000000*x^4 -
339822892269241355816208429950551229332706130143428161540794195
815861138122323851042008932992783533768063423142581903368931336
783762606948352000000000000000*x^3 +
125829682869604371150333284657095602170632528645257895644630886
139695145352963214751497406061524769318541374380100127000027277
179494177112064000000000000000000*x^2 -
714299400309218383265647783367840725131325145918175495211076900
218671706321745386742315899336916630736190695034419462707331250
150708346978304000000000000000000000*x +
183656232405662449254293582475561226558898939514847789800803330
252739464977678414109694541741347718191553931348217317396279310
495068424175616000000000000000000000,
196797170976107552602018 47>
]
```

Invariants of genus 2 curves

```
%magma
frob_seq := QuarticCMFieldOrdinaryWeilNumbers(K,3); frob_seq;
```

```
[
1/5*(10*t^3 - 23*t^2 + 215*t - 48)
]
```

```
%magma
chi := MinimalPolynomial(frob_seq[1]); chi;
```

```
x^4 - 210*x^3 + 21502*x^2 - 1377810*x + 43046721
```

```
%magma
FF<t> := FiniteField(3,8);
IgusaInvariantsSequences(DBG2,chi);
```

```
[
[
[ 1, t^4527, t^3539, t^6402, t^4026 ],
[ 1, t^461, t^4057, t^6086, t^5518 ],
[ 1, t^1383, t^5611, t^5138, t^3434 ],
[ 1, t^4149, t^3713, t^2294, t^3742 ],
[ 1, t^5887, t^4579, t^322, t^4666 ],
[ 1, t^4541, t^617, t^966, t^878 ],
[ 1, t^503, t^1851, t^2898, t^2634 ],
[ 1, t^1509, t^5553, t^2134, t^1342 ]
]
]
[
Maximal Order of Equation Order with defining polynomial x^4 -
210*x^3 + 21502*x^2 - 1377810*x + 43046721 over ZZ
]
[
[]
]
[
[]
]
[ 5, 27, 151 ]
[ 8 ]
[ 1 ]
```

```
%magma
IgLIX := IgusaLIXInvariants(DBIX,DAB)[1];
JJ_seq := IgusaLIXToIgusaInvariants(IgLIX,FF : LiftingPrecision
:= 512);
JJ_seq;
```

```
[
[ 1, t^4527, t^3539, t^6402, t^4026 ],
[ 1, t^461, t^4057, t^6086, t^5518 ],
[ 1, t^1383, t^5611, t^5138, t^3434 ],
[ 1, t^5887, t^4579, t^322, t^4666 ],
[ 1, t^1509, t^5553, t^2134, t^1342 ],
[ 1, t^4149, t^3713, t^2294, t^3742 ],
[ 1, t^503, t^1851, t^2898, t^2634 ],
[ 1, t^4541, t^617, t^966, t^878 ]
]
```

Mestre's algorithm

Mestre's algorithm (reconstruction of a genus 2 curve from Igusa invariants) has been extended to

finite fields of characteristic 2 and 3.

```
%magma
JJ := JJ_seq[1];
C := HyperellipticCurveFromIgusaInvariants(JJ);
J2 := IgusaInvariants(C);
IgusaToNormalizedIgusaInvariants(J2);
```

```
[ 1, t^4527, t^3539, t^6402, t^4026 ]
```

Conversely, the Igusa class invariants were constructed using canonical lifting of the invariants of an ordinary curve over a finite field.

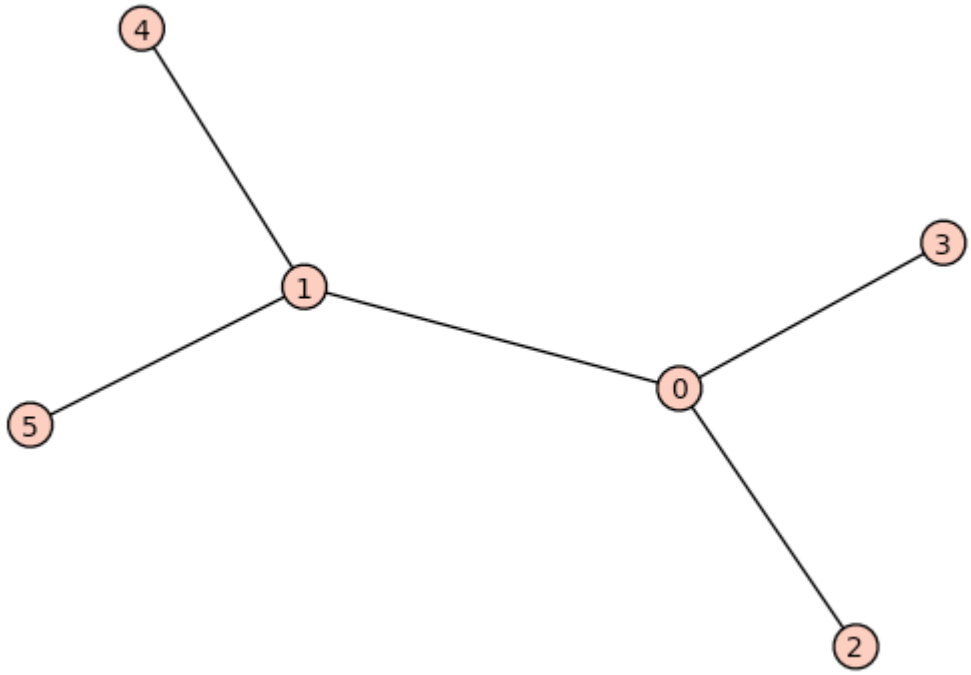
```
%magma
CanonicalLiftAbsoluteIgusaInvariants(C,256);
```

```
[
-96371175509799312765850990978286207052447309130804114934201720
17465078338738842449453674460965691350592331016204491 8*$.1^7 -
10650794707408917155898592713515786325622109919934295658413476
65502109843187729276732058055570260195770469193108962 0*$.1^6 -
63135006716367772731979862386034388424000662853155563136953241
49692767501509724531072755569807184899644312776522732 2*$.1^5 +
43727836848541426366045048637082593144338125501378754912564355
40867219086279040533903644697012937616974987991799720 9*$.1^4 +
24329346054209682075016622914892076337133704068934033522380294
83768025013225502319329683508550428867532583046179644 2*$.1^3 -
26972375684917094116433129326912312519506747072951429979990006
12431249236720358730247088418500235032455884779379753*$.1^2 +
43436634050596213896912009288769172649766288146042239788297571
42386307270102496517746641882039133440948755660592512 5*$.1 -
43561291733105919005443873810387073345402467955574449770377423
09099824482423153065309674212688024291040347492949181 9,
70734251729884720348158100172063928155826960832518436677700631
00645473967772034782597370365027070833432268173887048*$.1^7 +
24754628566541876815156239657638681154034567509703598932374395
37004951581958819657484941029146187164632648017494846 4*$.1^6 +
66484956130767123507481252385892148684180977454401744191751124
55816469375397577617228914524275914157631719002772432 1*$.1^5 -
28186801677208323511505117735374585421258414649248752154181636
32179971046970508014920744191880837290112075318386756 7*$.1^4 +
47114994129303865372033662695872032015798631662188171271062045
29342341189325478185987070992376445655071795664526424 7*$.1^3 +
34731607100368582449232653739863882962754693955808264863419042
33148384208510276118500302907754754273786087215198602 2*$.1^2 +
56570889279001728622507347915557607989790903884104989015039220
06645644630445443360897485133100248783405057273651747 0*$.1 -
31086101225949229345802390481941662307326343093157367258865762
59384800901090837128514799490690002173008779437503696 0,
59738286440549335361973913585162020172384138998729595588864330
62153489206576654154761848195285728568894726357791672 9*$.1^7 -
17699825828368372882028571219646005781375543877568233405198202
33000798166967463056675438399370387392786880688488905*$.1^6 -
47237515924102001058120147339773378894220563641690159540245820
```

```
679163132413204338437448016416021898582060098005842511*$.1^5 -
188227456646412385722730931016989266998231080539196093029758955
672135437197586345540686851784995422451881771163362813*$.1^4 +
294974809017084618104515832019301084585044160613310413255767775
246947348304036218966721880922602505972234439657249364*$.1^3 +
489554911877319929187707014996843207833121268230395023561451677
239145109695795451742333849256534183124219630425802095*$.1^2 -
461185487676919893757362222376158935431705390123226305194354800
177172887960093099709322213345076197066802155592310141*$.1 -
235855867886460711740010600711454108045336521021045938141788544
501020359753447352842409190410398686753651048145031755 ]
```

Isogeny graphs in genus 2

```
%sage
magma.eval("jj := IgusaToAbsoluteIgusaInvariants(JJ);")
magma.eval("A, X :=
AbsoluteIgusaInvariantIsogenyAdjacencyMatrix(jj,2);")
G = Graph(magma("A;").sage())
G.plot(scaling_term=0.005)
```



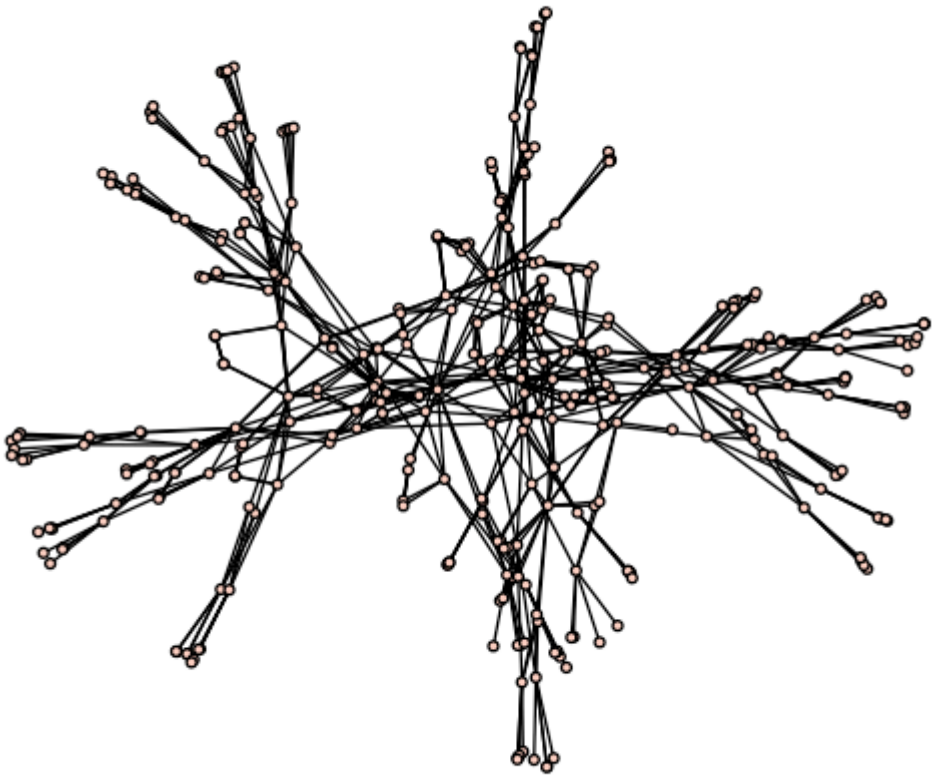

```
%magma
P<x> := PolynomialRing(ZZ);
chi := x^4 - 12*x^3 - 170*x^2 - 8748*x + 531441;
JJ := IgusaInvariantsSequences(DBG2,chi)[1][1];
jj := IgusaToAbsoluteIgusaInvariants(JJ);
```

```
%sage
```

```
magma.eval("A, X :=
AbsoluteIgusaInvariantIsogenyAdjacencyMatrix(jj,2);")
G = Graph(magma("A;").sage())
G.num_verts()
```

```
366
```

```
G.plot(scaling_term=0.001, vertex_size=10,
vertex_labels=False)
```



```
leaves = []
for v in G.vertices():
   if len(G.neighbors(v)) == 1:
       leaves.append(v)
G.delete_vertices(leaves)
G.plot(scaling_term=0.001, vertex_size=10,
vertex_labels=False)
```

Endomorphism rings

```
%magma
C := HyperellipticCurveFromIgusaInvariants(JJ);
J := Jacobian(C);
EndomorphismRing(J);
```

```
Transformation of Order over
Equation Order with defining polynomial x^4 + 12*x^3 - 170*x^2
8748*x + 531441 over ZZ
Transformation Matrix:
[  1   0   0   0]
[  0   2   0   0]
[  0   0   8   0]
[  0   0   0 352]
[ 2, 8, 352 ]
```

Invariants of genus 3 curves

The invariants of Shioda for genus 3 hyperelliptic curves, and of Dixmier, extended by Ohno, are implemented for generic genus 3 curves (plane quartics).

```
%magma
P<x> := PolynomialRing(QQ);
```

```
C := HyperellipticCurve(x^7 + 3*x + 1);
SS, ii := ShiodaInvariants(C);
print C;
print SS;
print ii;
```

```
Hyperelliptic Curve defined by y^2 = x^7 + 3*x + 1 over Ratior
Field
[ -3/4, 0, 3/512, 0, 3/16384, 1/512, 81/4587520, 1/8192,
81/146800640 ]
[ 2, 3, 4, 5, 6, 7, 8, 9, 10 ]
```

```
%magma
PP<X,Y,Z> := ProjectiveSpace(QQ,2);
C1 := Curve(PP,X^3*Y + Y^3*Z + Z^3*X);
D1, ii := DixmierOhnoInvariants(C1);
d1 := [ D1[i]/D1[1]^ii[i] : i in [1..#D1] ];
C2 := Curve(PP,X^4+Y^4+Z^4);
D2, ii := DixmierOhnoInvariants(C2);
d2 := [ D2[i]/D2[1]^ii[i] : i in [1..#D2] ];
print C1;
print "D1:", D1;
print "d1:", d1;
print C2;
print "D2:", D2;
print "d2:", d2;
```

```
Curve over Rational Field defined by
X^3*Y + Y^3*Z + X*Z^3
D1: [ 9, -729, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, -823543 ]
d1: [ 1, -9, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, -823543/387420489 ]
Curve over Rational Field defined by
X^4 + Y^4 + Z^4
D2: [ 144, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, -1099511627776 ]
d2: [ 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, -16/387420489 ]
```

```
%magma
C3 := Curve(PP,X^4 + 2*X^3*Y + 3*X^2*Y^2 + 2*X*Y^3 + 18*X*Y*Z^2
+ 9*Y^2*Z^2 - 9*Z^4);
D3, ii := DixmierOhnoInvariants(C3);
d3 := [ D3[i]/D3[1]^ii[i] : i in [1..#D3] ];
print C3;
print "D3:", D3;
print "d3:", d3;
```

```
Curve over Rational Field defined by
X^4 + 2*X^3*Y + 3*X^2*Y^2 + 2*X*Y^3 + 18*X*Y*Z^2 + 9*Y^2*Z^2 -
D3: [ 1296, -15116544, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
-219254597601400764204318 72 ]
d3: [ 1, -9, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, -823543/387420489 ]
```

$Pic^0(C)$ for general curves

This is a generic wrapper for Florian Hess' Riemann-Roch spaces in Magma. These examples are genus one, but the construction is for arbitrary genus!

```
%magma
PP<X,Y,Z> := ProjectiveSpace(QQ,2);

C := Curve(PP,X^3+Y^3+Z^3);

P := C![-1,1,0];

J := PicardGroup(C,P);

Q := J!C![0,-1,1];
[ i*Q : i in [1..3] ];
```

```
[
(X, Y + Z),
(X + Z, Y),
0
]
```

Here we work with a twist of rank 2.

```
%magma
C := Curve(PP,X^3+Y^3+19*Z^3);
P := C![-1,1,0];

J := PicardGroup(C,P);

Q := J!C![-3,2,1];
[ i*Q : i in [1..3] ];
```

```
[
(X + 3*Z, Y - 2*Z),
(X + 92/35*Z, Y + 33/35*Z),
(X + 9613/10386*Z, Y + 27323/10386*Z)
]
```

```
%magma
R := J!C![-3/2,-5/2,1];

[ i*Q + j*R : i, j in [-2..2] ];
```

```
[
(X + 1502783/670397*Z, Y + 1325880/670397*Z),
(X + 2395/201*Z, Y - 2386/201*Z),
(X + 33/35*Z, Y + 92/35*Z),
(X + 109/31*Z, Y - 90/31*Z),
(X - 1025/1533*Z, Y + 4112/1533*Z),
(X - 1322/4983*Z, Y + 13301/4983*Z),
(X + 36/13*Z, Y - 17/13*Z),
(X - 2*Z, Y + 3*Z),
(X + 8/3*Z, Y + 1/3*Z),
(X - 594/103*Z, Y + 613/103*Z),
```

```
(X - 831/196*Z, Y + 895/196*Z),
(X + 5/2*Z, Y + 3/2*Z),
0,
(X + 3/2*Z, Y + 5/2*Z),
(X + 895/196*Z, Y - 831/196*Z),
(X + 613/103*Z, Y - 594/103*Z),
(X + 1/3*Z, Y + 8/3*Z),
(X + 3*Z, Y - 2*Z),
(X - 17/13*Z, Y + 36/13*Z),
(X + 13301/4983*Z, Y - 1322/4983*Z),
(X + 4112/1533*Z, Y - 1025/1533*Z),
(X - 90/31*Z, Y + 109/31*Z),
(X + 92/35*Z, Y + 33/35*Z),
(X - 2386/201*Z, Y + 2395/201*Z),
(X + 1325880/670397*Z, Y + 1502783/670397*Z)
]
```

Alternative models for elliptic curves in Sage

These examples are twist of Hessian curves, which can be easily implemented in Sage directly:

```
%sage
k = FiniteField(101)
E = HessianCurve(k(0))
E
```

```
Hessian curve over Finite Field of size 101 defined by x^3 + y
z^3 = 0
```

```
%sage
P = E.random_point()
[ n*P for n in range(32) ]
```

```
[(0 : 100 : 1), (96 : 45 : 1), (4 : 84 : 1), (46 : 3 : 1), (13
1), (32 : 16 : 1), (48 : 26 : 1), (10 : 82 : 1), (53 : 31 : 1)
: 73 : 1), (91 : 89 : 1), (40 : 30 : 1), (60 : 51 : 1), (70 :
1), (11 : 33 : 1), (76 : 21 : 1), (20 : 92 : 1), (0 : 100 : 1)
56 : 1), (95 : 77 : 1), (34 : 49 : 1), (81 : 43 : 1), (19 : 2
(35 : 64 : 1), (85 : 42 : 1), (88 : 18 : 1), (18 : 88 : 1), (4
: 1), (64 : 35 : 1), (2 : 19 : 1), (43 : 81 : 1), (49 : 34 : 1
```

$\mathrm{Pic}^0(C)$ for singular hyperelliptic curves

```
%magma
P7<x> := PolynomialRing(FiniteField(7));
C := SingularHyperellipticCurve(x*(x^4+x+1)^2);
J := SingularPicardGroup(C);
P := Random(J);
[ n*P : n in [1..16] ];
```

```
[
(x^3 + 6*x^2 + 2*x, y + 5*x^2 + 2*x),
(x^4 + 5*x^3 + 5*x^2 + 3*x + 4, y + 4*x^3 + 5*x^2 + 6*x + 1),
```

```
    (x^3 + x^2 + 2*x + 3, y + 4*x^2 + 3*x + 4),
    (x^4 + x^3 + 2*x^2 + 3*x, y + 2*x^3 + 3*x^2 + 6*x),
    (x^4 + 6*x^3 + x + 4, y + 3*x^3 + 2*x^2 + 2*x + 5),
    (x^4 + 5*x^3 + 5*x + 4, y + 5*x^3 + 2*x^2 + 2),
    (x^4 + 5*x^2 + 6*x + 1, y + x^3 + 5*x + 2),
    (x^3 + x^2 + 4*x + 6, y + 2*x^2 + 5*x + 1),
    (x^4 + 3*x^3 + 5*x^2 + 4*x + 2, y + 6*x^3 + x^2 + 3*x + 4),
    (x^4 + 3*x^3 + 6*x + 4, y + 3*x^3 + x^2 + 2*x + 4),
    (x^3 + 4*x^2 + 2*x, y + x^2 + 2*x),
    (x^4 + 3*x^3 + 6*x + 4, y + 5*x^3 + 5*x),
    (x^4 + 4*x^3 + 2*x^2 + 2, y + 6*x^3 + 4*x^2 + 5*x + 5),
    (x^4 + 6*x, y + 5*x^3 + 6*x^2),
    (x^3 + 6, y + 5*x^2 + 3*x + 3),
    (x^4 + 6*x^2 + 2*x + 4, y + 5*x^3 + 6*x^2)
    ]
```

```
%magma
(7^4-1)*P;
(7^2-1)*P;
(7^2+1)*P;
```

```
    (1)
    (x^3 + x^2 + 5, y + x + 2)
    (x^4 + 2*x^3 + x + 1, y + 6*x^3 + x^2 + x + 1)
```

Exercise: Create an spkg for ECHIDNA and provide comments!