

2011-02-04-sqrt5_talk_demo

This worksheet requires:

1. This worksheet uses the patch from [trac 9402 for \$L\$ -series](#).
2. It also uses [Purple Sage](#).

```
import psage.modform.hilbert.sqrt5 as H
F = H.tables.F
a = F.gen()
```

```
time S = H.HilbertModularForms(5*a-2); S
```

```
Time: CPU 0.04 s, Wall: 0.04 s
```

```
Hilbert modular forms of dimension 2, level 5*a-2 (of no  
over QQ(sqrt(5))
```

```
T2 = S.hecke_matrix(F.factor(2)[0][0]); T2
```

```
[0 5]
[3 2]
```

```
T2.charpoly().factor()
```

```
(x - 5) * (x + 3)
```

Finding elliptic curves of norm conductor 199...

```
time S = H.HilbertModularForms(3*a+13); S
```

```
Time: CPU 0.04 s, Wall: 0.04 s
```

```
Hilbert modular forms of dimension 4, level 3*a+13 (of no  
over QQ(sqrt(5))
```

```
T2 = S.hecke_matrix(F.factor(2)[0][0]); T2
```

```
[0 4 1 0]
[4 0 1 0]
[1 1 2 1]
[0 0 3 2]
```

```
T2.charpoly().factor()
```

```
(x - 5) * (x - 3) * x * (x + 4)
```

```
E = EllipticCurve([0,-a-1,1,a,0]);
k = F.factor(2)[0][0].residue_field()
k.cardinality() + 1 - E.change_ring(k).cardinality()
```

-4

```
Z = S.elliptic_curve_factors(); Z
```

```
[
  Isogeny class of elliptic curves over QQ(sqrt(5)) attache
  number 0 in Hilbert modular forms of dimension 4, level 3
  norm 199=199) over QQ(sqrt(5)),
  Isogeny class of elliptic curves over QQ(sqrt(5)) attache
  number 1 in Hilbert modular forms of dimension 4, level 3
  norm 199=199) over QQ(sqrt(5)),
  Isogeny class of elliptic curves over QQ(sqrt(5)) attache
  number 2 in Hilbert modular forms of dimension 4, level 3
  norm 199=199) over QQ(sqrt(5))
]
```

```
A = Z[0]; A.aplist(100)
```

```
[-4, -2, -3, 5, -3, 0, 2, -7, 6, -6, -4, -4, -3, 12, 0, 3
-12, -3, -10, -10, 0, 12]
```

```
k = F.factor(3)[0][0].residue_field()
k.cardinality() + 1 - E.change_ring(k).cardinality()
```

-2

A Bigger Example

```
N = F.factor(100019)[0][0]; N
```

```
Fractional ideal (65*a + 292)
```

```
time S = H.HilbertModularForms(N); S
```

Time: CPU 0.27 s, Wall: 0.27 s
Hilbert modular forms of dimension 1667, level $65a+292$ ($100019=100019$) over $\mathbb{Q}(\sqrt{5})$

```
time T5 = S.hecke_matrix(F.factor(5)[0][0])
```

Time: CPU 0.13 s, Wall: 0.13 s

```
len(T5.nonzero_positions())/1667.0^2
```

0.00359460201540976

```
T5.visualize_structure(maxsize=4096)
```



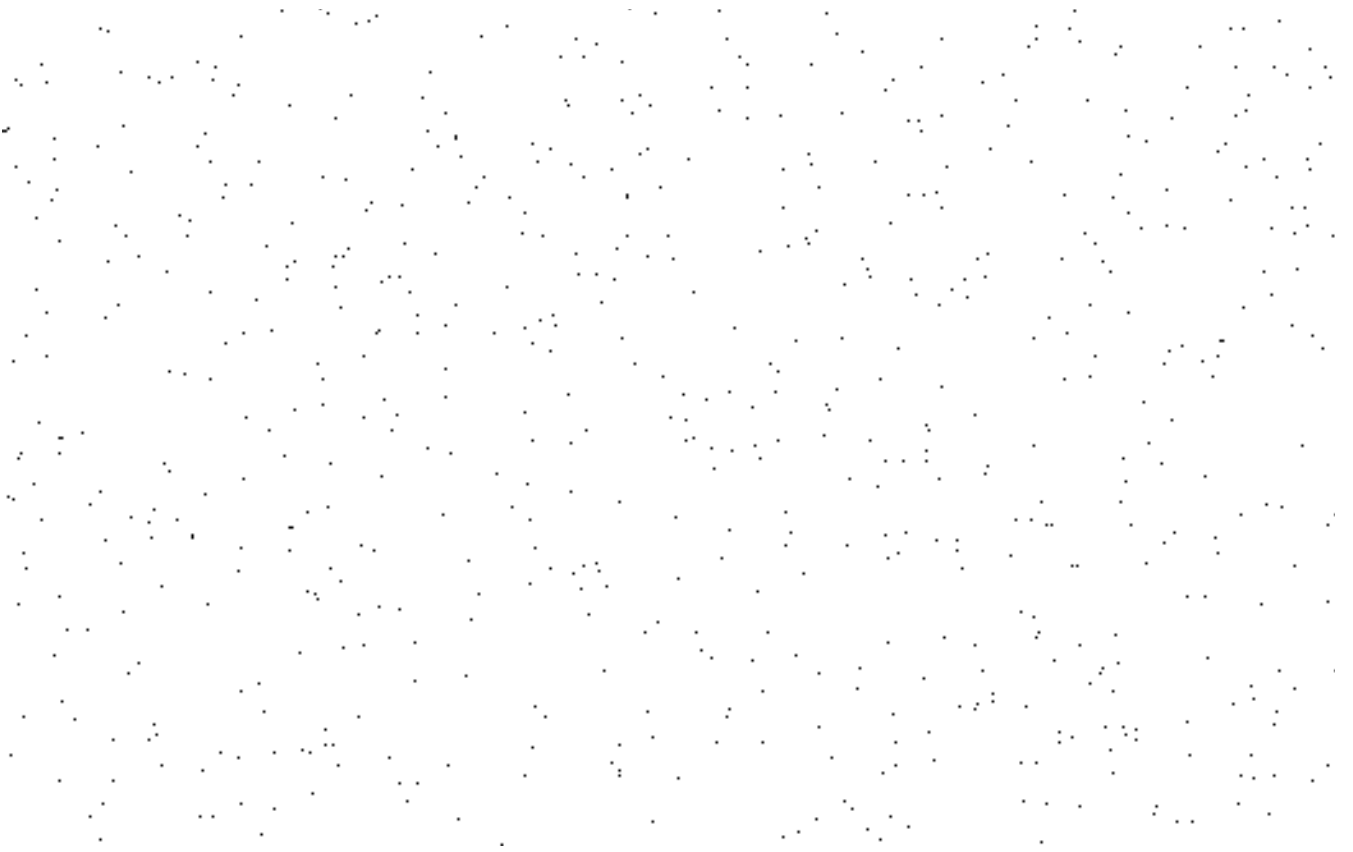
1. The first step in the process of identifying a problem is to recognize that a problem exists. This is often done by comparing current performance with a desired state or goal. For example, a manager might notice that sales are declining or that customer satisfaction is low. Once a problem is identified, the next step is to define it more precisely. This involves determining the scope of the problem, its causes, and its effects. A clear definition of the problem is essential for developing an effective solution.

2. The second step is to analyze the problem. This involves gathering information about the problem and its context. This can be done through interviews, surveys, or data analysis. The goal is to understand the underlying causes of the problem and to identify any constraints or resources that may affect the solution. A thorough analysis is necessary to ensure that the solution addresses the root cause of the problem rather than just the symptoms.

3. The third step is to generate potential solutions. This involves brainstorming ideas and evaluating them based on their feasibility, effectiveness, and cost. It is important to consider a wide range of options and to evaluate them against the criteria of the problem. This step often involves collaboration with others who may have different perspectives or expertise. The goal is to identify a solution that is both practical and effective.

4. The fourth step is to implement the chosen solution. This involves developing a plan of action, allocating resources, and executing the plan. It is important to monitor progress and to make adjustments as needed. Implementation is often the most challenging part of the process, as it requires coordination and communication with others. The goal is to ensure that the solution is implemented correctly and that it achieves the desired results.

5. The final step is to evaluate the results of the solution. This involves comparing the actual performance with the desired state and determining whether the problem has been solved. If the problem has not been solved, the process may need to be repeated. Evaluation is essential to ensure that the solution is effective and to learn from the experience for future problems. It is important to document the results and to share them with others who may be interested in the process.



```
# 7 is an inert prime -- norm 49  
time T7 = S.hecke_matrix(F.factor(7)[0][0])
```

```
Time: CPU 0.00 s, Wall: 0.00 s
```

```
# 11 is split  
time T11 = S.hecke_matrix(F.factor(11)[0][0])
```

```
Time: CPU 0.22 s, Wall: 0.22 s
```

```
# 13 is inert -- norm 169  
time T13 = S.hecke_matrix(F.factor(13)[0][0])
```

```
Time: CPU 13.25 s, Wall: 13.51 s
```

Example Curve: Norm Conductor 31

```
a = F.0
```

```
E = EllipticCurve([1,a+1,a,a,0]); show(E)
```

$$y^2 + xy + ay = x^3 + (a + 1)x^2 + ax$$

```
E.j_invariant()
```

```
-106208/31*a + 51455/31
```

```
E.torsion_subgroup()
```

```
Torsion Subgroup isomorphic to Z/8 associated to the Elli  
defined by y^2 + x*y + a*y = x^3 + (a+1)*x^2 + a*x over N  
in a with defining polynomial x^2 - x - 1
```

```
E.conductor()
```

```
Fractional ideal (5*a - 2)
```

```
E.local_data(E.conductor())
```

```
Local data at Fractional ideal (5*a - 2):  
Reduction type: bad non-split multiplicative  
Local minimal model: Elliptic Curve defined by y^2 + x*y  
+ (a+1)*x^2 + a*x over Number Field in a with defining po  
x^2 - x - 1  
Minimal discriminant valuation: 1  
Conductor exponent: 1  
Kodaira Symbol: I1  
Tamagawa Number: 1
```

```
time L = E.lseries().dokchitser(53)
```

```
Time: CPU 43.54 s, Wall: 46.52 s
```

```
L(1)
```

```
0.359928959498039
```

```
phi = F.embeddings(RR)
```

```
Omega0 = E.period_lattice(phi[0]); Omega0
```

```
Period lattice associated to Elliptic Curve defined by y^  
a*y = x^3 + (a+1)*x^2 + a*x over Number Field in a with d  
polynomial x^2 - x - 1 with respect to the embedding Ring  
From: Number Field in a with defining polynomial x^2 -  
To: Algebraic Real Field  
Defn: a |--> -0.618033988749895?
```

```
Omega0.real_period()
```

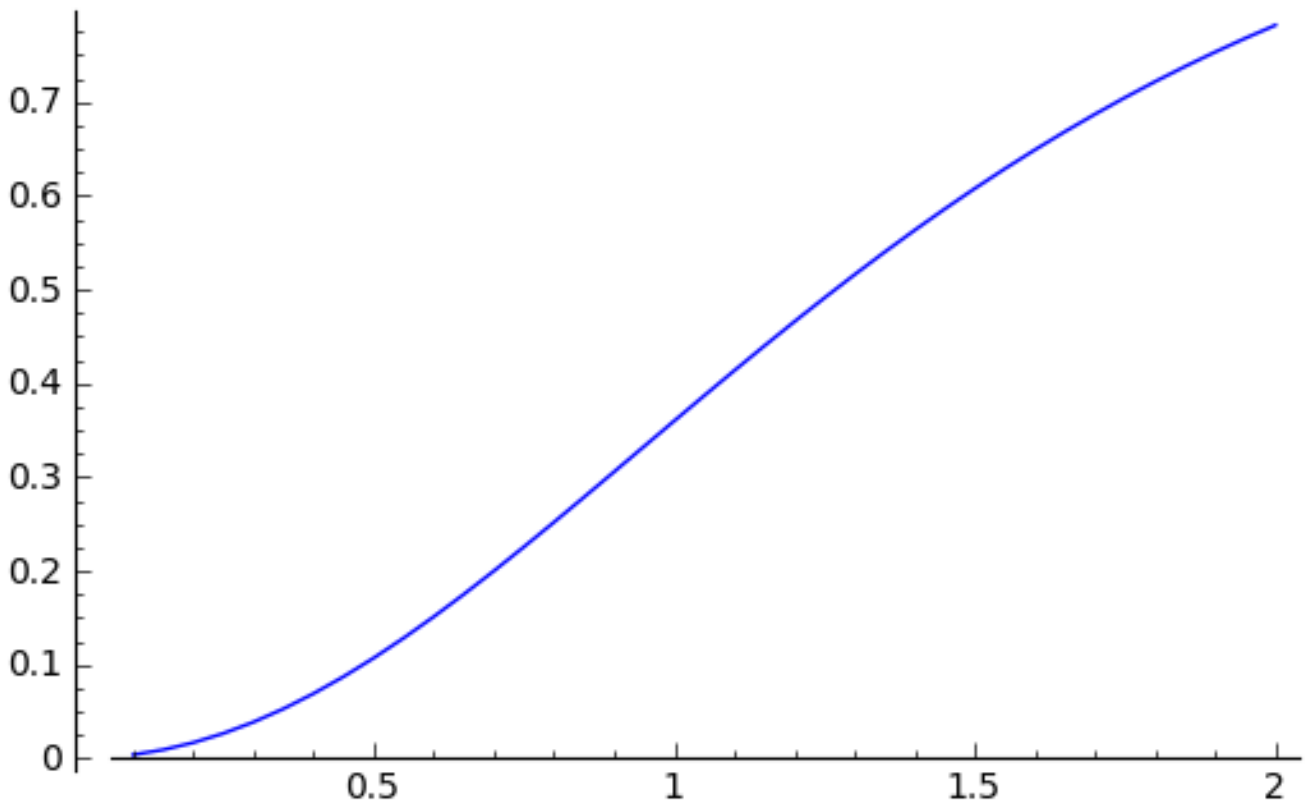
3.05217315335726

```
Omega1 = E.period_lattice(phi[1]); Omega1.real_period()
```

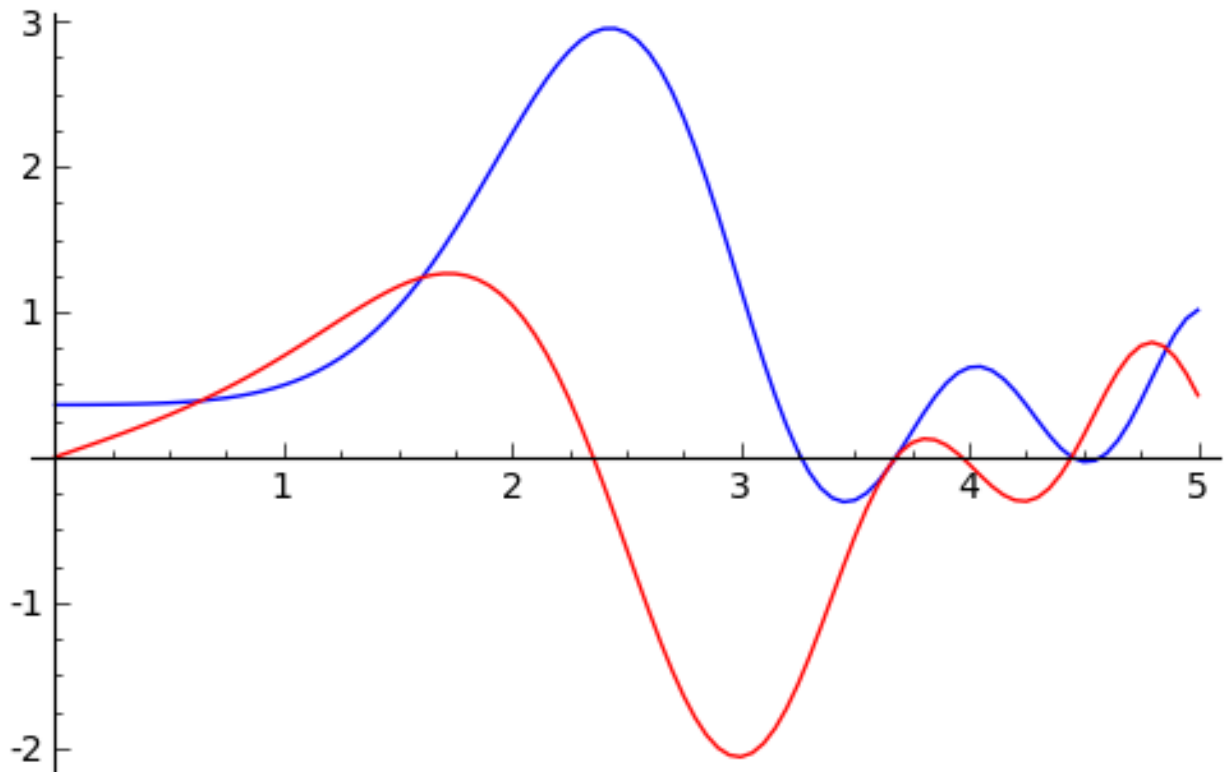
8.43805988789973

```
(  
  (math.sqrt(F.disc()) * L(1) * E.torsion_order()^2 ) /  
  (Omega0.real_period()*Omega1.real_period() *  
E.tamagawa_product_bsd())  
)
```

```
Lplot = line([(s, L(s)) for s in [.1,.15, .., 2]]); Lplot
```



```
set_verbose(-2)  
c = (line([(s, L(1+I*s).real()) for s in [0,0.05,..,5]]) +  
  line([(s, L(1+I*s).imag()) for s in [0,0.05,..,5]],  
color='red'))  
c.show()  
c.save('critical_zero-31.pdf')
```



[critical_zero-31.pdf](#)

```
find_root(lambda s:L(1+I*s).real(), 3.5,4)
```

```
3.678991475792357
```

```
L(1 + 3.67899147*I)
```

```
-1.39082483086184e-8 - 1.11224155167065e-8*I
```

Example Curve: Norm Conductor 199 (rank 1)

```
E = EllipticCurve([0,-a-1,1,a,0]); show(E)
```

$$y^2 + y = x^3 + (-a - 1)x^2 + ax$$

```
E.j_invariant()
```

```
-524288/199*a + 622592/199
```

```
E.torsion_subgroup()
```


Torsion Subgroup isomorphic to $\mathbb{Z}/3$ associated to the Elli defined by $y^2 + y = x^3 + (-a-1)x^2 + ax$ over Number F with defining polynomial $x^2 - x - 1$

```
E.conductor()
```

```
Fractional ideal (3*a + 13)
```

```
E.tamagawa_numbers()
```

```
[1]
```

```
E.local_data()
```

```
[Local data at Fractional ideal (3*a + 13):  
Reduction type: bad split multiplicative  
Local minimal model: Elliptic Curve defined by  $y^2 + y = (-a-1)x^2 + ax$  over Number Field in a with defining pol  
- x - 1  
Minimal discriminant valuation: 1  
Conductor exponent: 1  
Kodaira Symbol: I1  
Tamagawa Number: 1]
```

```
E.gens()
```

```
[(0 : 0 : 1)]
```

```
E.regulator_of_points(E.gens())
```

```
0.154308568543030
```

```
time L = E.lseries().dokchitser()
```

```
Time: CPU 55.21 s, Wall: 55.56 s
```

```
L(1)
```

```
0
```

```
L.derivative(1,1)
```

```
0.657814883009960
```

```
phi = F.embeddings(RR)  
Omega0 = E.period_lattice(phi[0]); O0 =  
Omega0.real_period()  
Omega1 = E.period_lattice(phi[1]); O1 =  
Omega1.real_period()  
O0, O1
```

```
(3.53489274657737, 6.06743219455559)
```

```
print (
    math.sqrt(F.disc()) * L.derivative(1,1) *
    E.torsion_order()^2 /
        (Omega0.real_period()*Omega1.real_period())*
    E.tamagawa_product_bsd()
        *E.regulator_of_points(E.gens()))
)
```

4.000000000000002

Example Curve: Norm Conductor 1831

```
E = EllipticCurve([0,-a,1,-a-1,2*a+1]); show(E)
```

$$y^2 + y = x^3 + (-a)x^2 + (-a - 1)x + (2a + 1)$$

```
E.conductor()
```

Fractional ideal (7*a + 40)

```
E.torsion_subgroup()
```

Torsion Subgroup isomorphic to Trivial group associated to
 Elliptic Curve defined by $y^2 + y = x^3 + (-a)x^2 + (-a - 1)x + (2a + 1)$
 over Number Field in a with defining polynomial x

```
E.tamagawa_numbers()
```

[1]

```
E.local_data()
```

[Local data at Fractional ideal (7*a + 40):
Reduction type: bad non-split multiplicative
Local minimal model: Elliptic Curve defined by $y^2 + y = (-a)x^2 + (-a-1)x + (2a+1)$ over Number Field in a with polynomial $x^2 - x - 1$
Minimal discriminant valuation: 1
Conductor exponent: 1
Kodaira Symbol: I1
Tamagawa Number: 1]

```
show(E.gens())
```

$$\left[(0 : -a - 1 : 1), \left(-\frac{3}{4}a + \frac{1}{4} : -\frac{5}{4}a - \frac{5}{8} : 1 \right) \right]$$

```
reg = E.regulator_of_points(E.gens()); reg  
0.191946627694056
```

```
time L = E.lseries().dokchitser()  
Time: CPU 205.79 s, Wall: 207.57 s
```

```
L(1)  
-2.31497738102376e-20
```

```
L.derivative(1,1)  
7.76867974285369e-22
```

```
L.derivative(1,2)  
2.88288222151816
```

```
phi = F.embeddings(RR)  
Omega0 = E.period_lattice(phi[0]); O0 =  
Omega0.real_period()  
Omega1 = E.period_lattice(phi[1]); O1 =  
Omega1.real_period()  
O0, O1  
(3.75830925418163, 5.02645072067941)
```

```
print (  
    math.sqrt(F.disc()) * (L.derivative(1,2)/2) *  
    E.torsion_order()^2 /  
    (Omega0.real_period()*Omega1.real_period())*
```

```
E.tamagawa_product_bsd()  
    *E.regulator_of_points(E.gens()))  
)
```

0.8888888888888870

8/9.0

0.8888888888888889