SAGE Demo

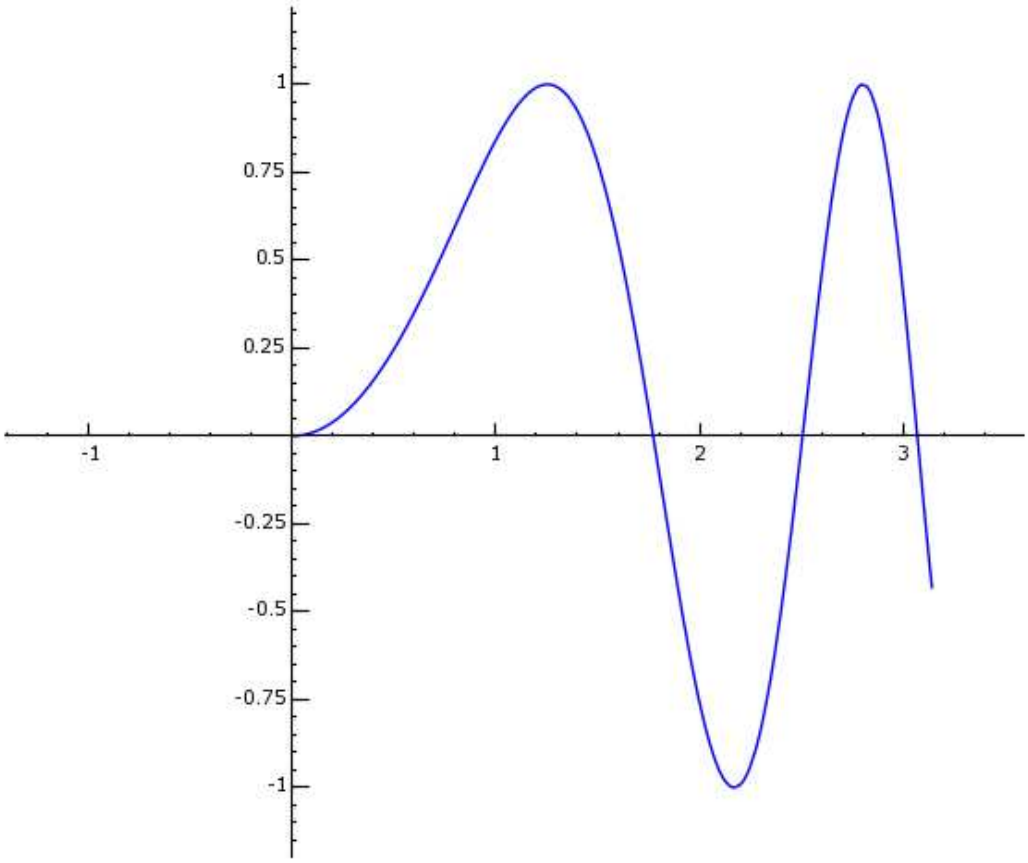
Sage Demo

Calculus

```
2 + 3
```

5

```
x = var('x')
show(plot(sin(x^2), 0, pi))
```




```
a = integrate(sin(x^2),x); a
```

```
sqrt(pi)*((sqrt(2)*I + sqrt(2))*erf((sqrt(2)*I + sqrt(2))*x/2) +
(sqrt(2)*I - sqrt(2))*erf((sqrt(2)*I - sqrt(2))*x/2))/8
```

```
show(a)
```

$$\frac{\sqrt{\pi} \cdot \left(\left(\sqrt{2} \cdot i + \sqrt{2}\right) \cdot \left(\mathrm{erf}\left(\frac{\left(\sqrt{2} \cdot i + \sqrt{2}\right) \cdot x}{2}\right)\right) + \left(\sqrt{2} \cdot i - \sqrt{2}\right) \cdot \left(\mathrm{erf}\left(\frac{\left(\sqrt{2} \cdot i - \sqrt{2}\right) \cdot x}{2}\right)\right)\right)}{8}$$

```
latex(a)
```

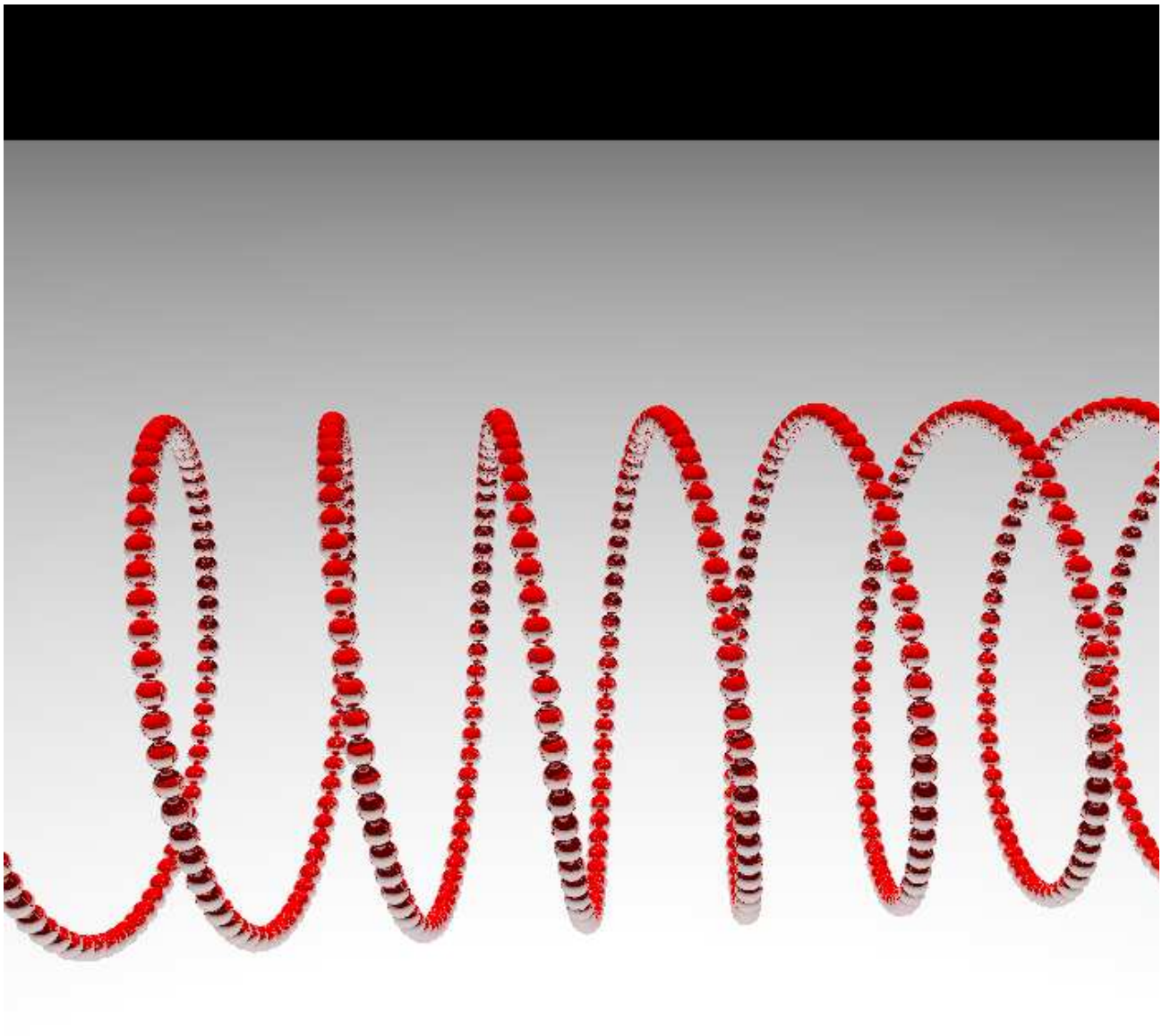
```
\frac{{\sqrt{ \pi } \cdot \left( {\left( {\sqrt{ 2 } \cdot i} +
\sqrt{ 2 } \right) \cdot \left( \text{erf} \left( \frac{{\left(
{\sqrt{ 2 } \cdot i} + \sqrt{ 2 } \right) \cdot x}}{2} \right)
\right)} + {\left( {\sqrt{ 2 } \cdot i} - \sqrt{ 2 } \right) \cdot
\left( \text{erf} \left( \frac{{\left( {\sqrt{ 2 } \cdot i} - \sqrt
```

2 } \right) \cdot x}}{2} \right) \right)} \right)}}{8}

```
var('a,b,c,X')
s = solve(a*X^2 + b*X + c == 0, X)
show(s[0])
```

$$X = \frac{-\sqrt{b^2 - 4 \cdot a \cdot c} - b}{2 \cdot a}$$

```
t = Tachyon(xres=800,yres=800, camera_center=(2,5,2),
look_at=(2.5,0,0))
t.light((0,0,100), 1, (1,1,1))
t.texture('r', ambient=0.1, diffuse=0.9, specular=0.5,
opacity=1.0, color=(1,0,0))
for i in srange(0,50,0.1):
   t.sphere((i/10,sin(i),cos(i)), 0.05, 'r')
t.texture('white', color=(1,1,1), opacity=1, specular=1,
diffuse=1)
```

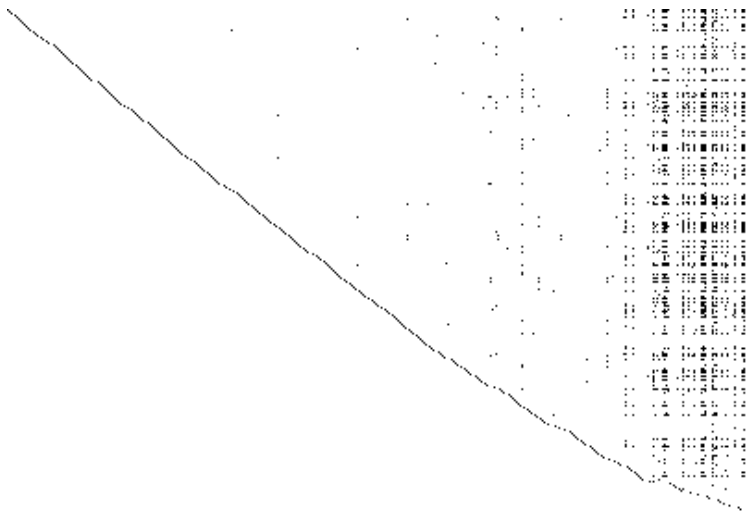


Linear Algebra

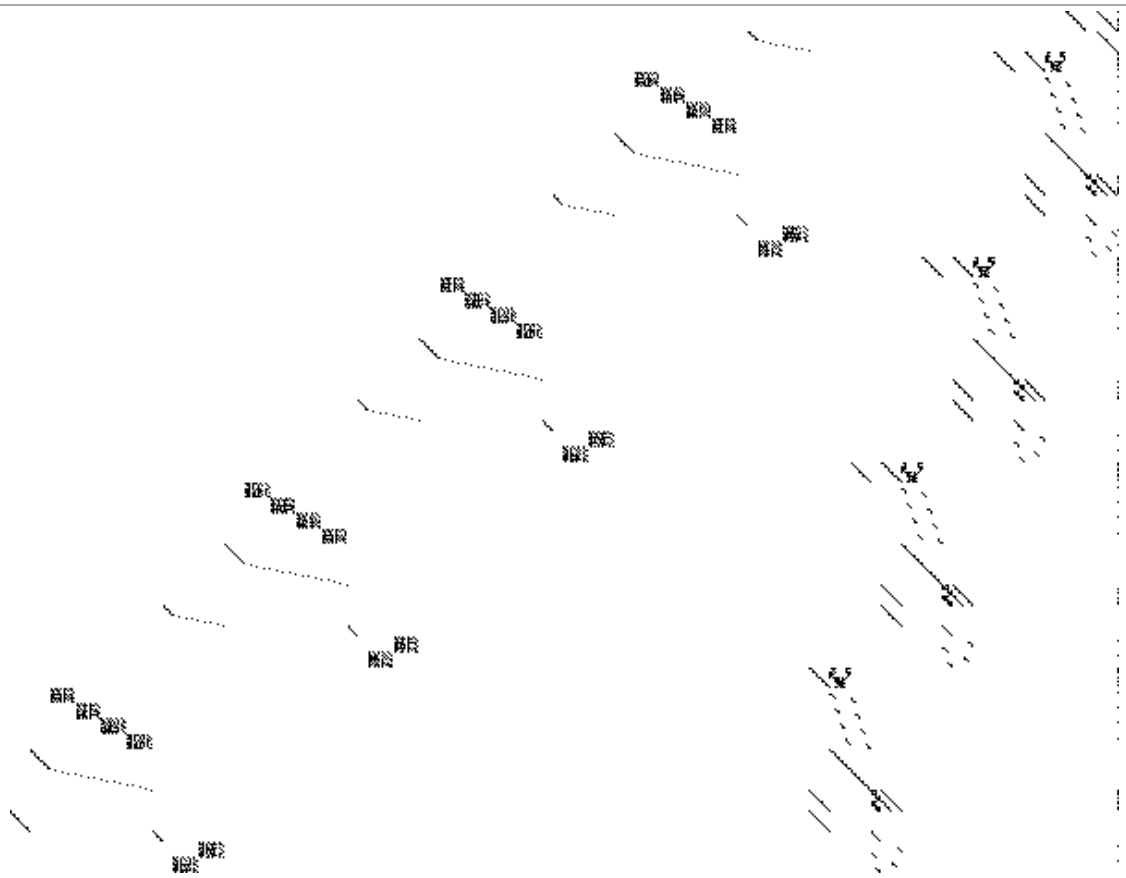
```
A = random_matrix(GF(127),3000,3000,density=1.0/3000,sparse=True)
time A.echelonize()
A.rank()
```

```
Time: CPU 0.12 s, Wall: 0.13 s
1616
```

```
n = 300
A = random_matrix(GF(127),n,n+100,sparse=True,density=2/n)
A.echelonize()
A.visualize_structure()
```



```
sr = mq.SR(4,2,2,4,gf2=True, allow_zero_inversions=True)
F,s = sr.polynomial_system()
A,v = F.coefficient_matrix()
A.visualize_structure(maxsize=600)
```



Factoring

```
time factor(next_prime(2^40) * next_prime(2^300),verbose=0)
```

```
1099511627791 *
20370359763344860862684456884093781610514683936659362506361404493548129976333670618339753 3
CPU time: 3.84 s,  Wall time: 3.88 s
```

```
time ecm.factor(next_prime(2^40) * next_prime(2^300))
```

```
[1099511627791,
```

```
2037035976334486086268445688409378161051468393665936250636140449354
8129976333670618339753 3]
CPU time: 0.19 s,  Wall time: 0.61 s
```

```
v,t = qsieve(next_prime(2^90)*next_prime(2^91),time=True)
print v, t[:4]
```

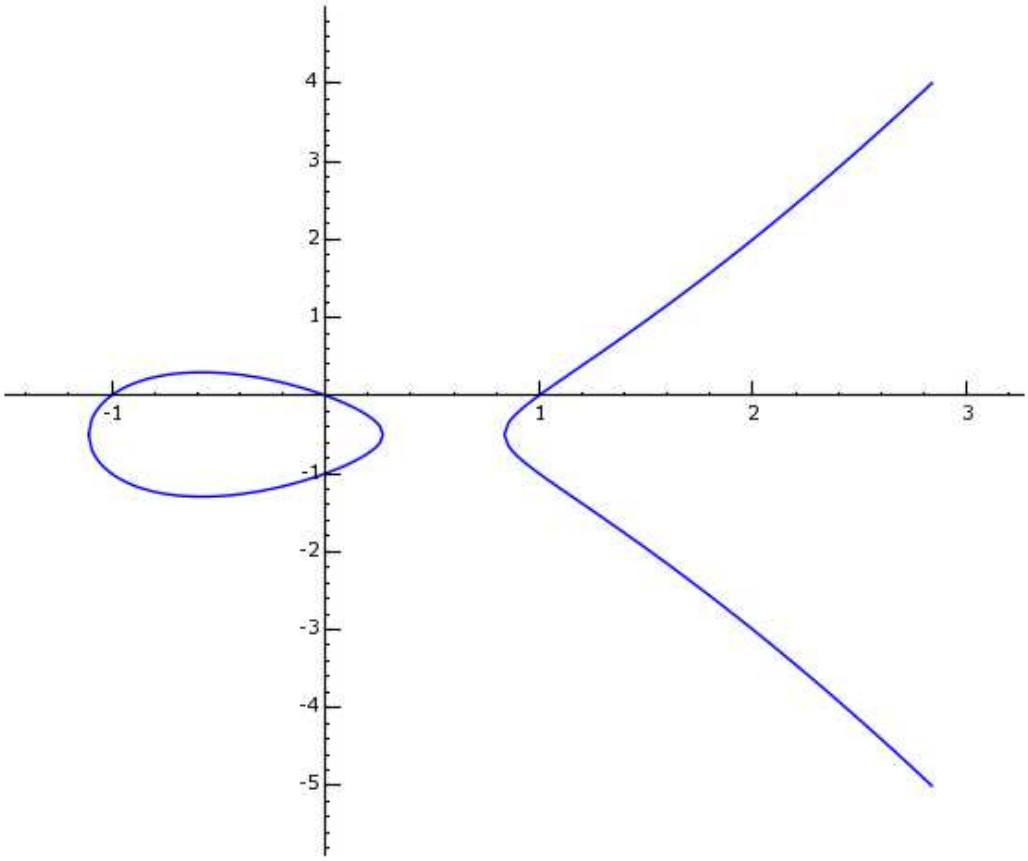
```
[1237940039285380274899124357, 2475880078570760549798248507] 3.53
```

Elliptic Curves

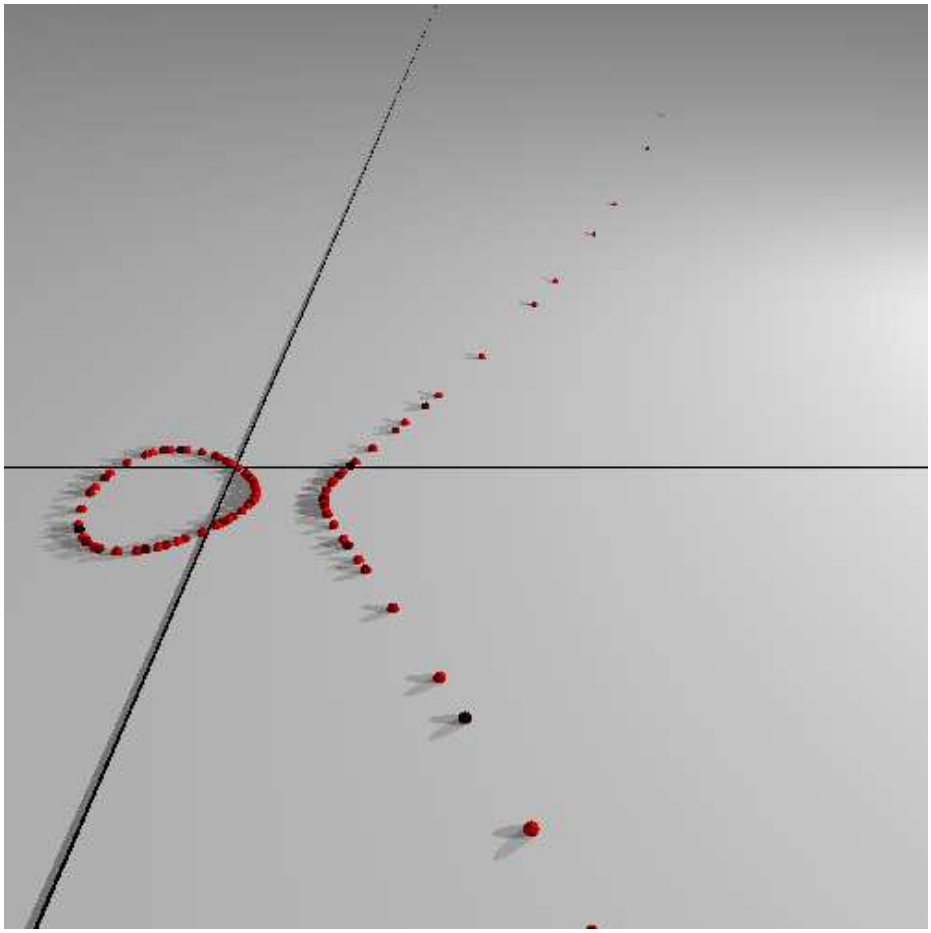
```
e = EllipticCurve("37a") # Cremona Label
show(e)
```

$$y^2 + y = x^3 - x$$

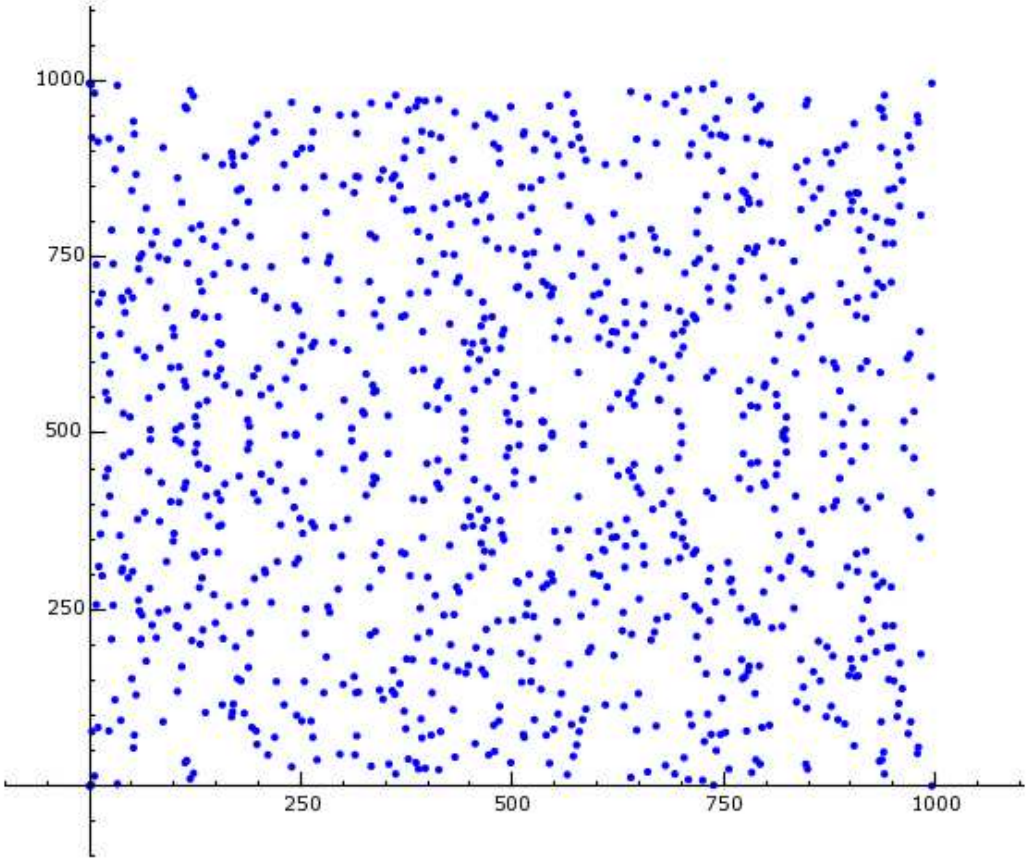
```
show(plot(e))
```



```
t = Tachyon(xres=500, yres=500, camera_center=(2,7,4),
look_at=(2,0,0), raydepth=4)
t.light((10,3,2), 1, (1,1,1))
t.light((10,-3,2), 1, (1,1,1))
t.texture('black', color=(0,0,0))
t.texture('red', color=(1,0,0))
t.texture('grey', color=(.9,.9,.9))
t.plane((0,0,0),(0,0,1),'grey')
t.cylinder((0,0,0),(1,0,0),.01,'black')
t.cylinder((0,0,0),(0,1,0),.01,'black')
E = EllipticCurve('37a')
P = E([0,0])
Q = P
n = 100
for i in range(n):
   Q = Q + P
   c = i/n + .1
   t.texture('r%s'%i,color=(float(i/n),0,0))
   t.sphere((Q[0], -Q[1], .01), .04, 'r%s'%i)

t.show()
```

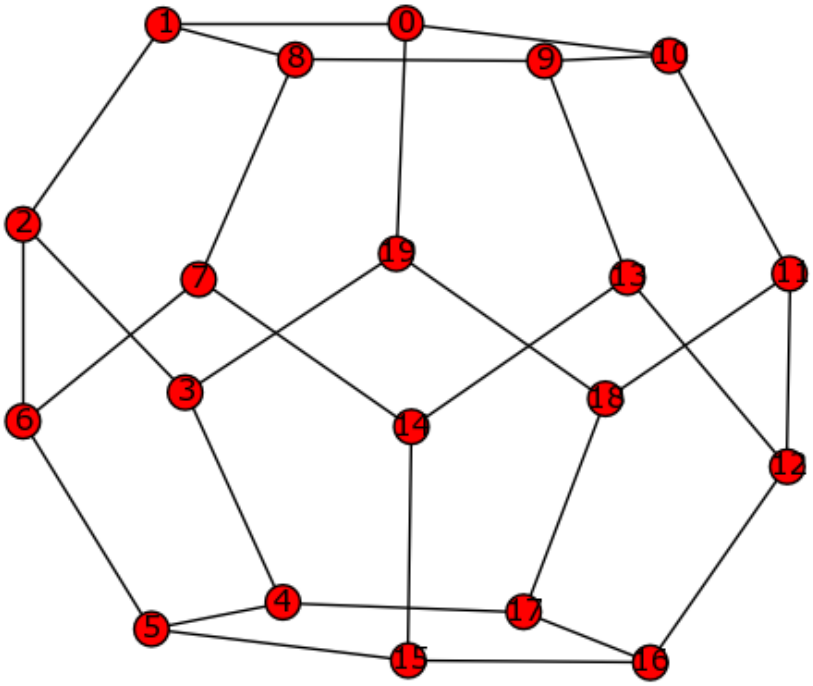


```
E = e.change_ring(GF(997))
show(E.plot())
```

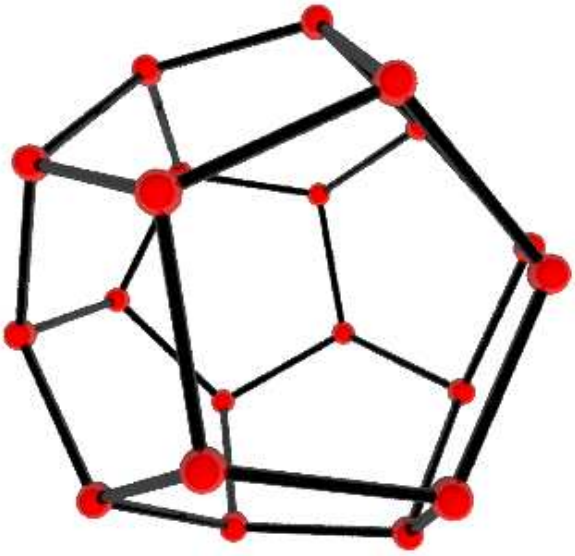


Graph Theory

```
D = graphs.DodecahedralGraph()
D.show()
```




```
D.show3d()
```



```
gamma = SymmetricGroup(20).random_element()
E = D.copy()
E.relabel(gamma)
D.is_isomorphic(E) # only free implementation is in Sage!
```

```
True
```

```
D.radius()
```

```
5
```

Multivariate Polynomial Rings

```
P.<x,y,z> = PolynomialRing(QQ,3)
p = (x + y + z + 1)^20 # the Fateman fastmult benchmark
q = p + 1
time r = p*q
```

```
Time: CPU 1.20 s, Wall: 1.21 s
```

```
P.<x,y,z> = PolynomialRing(GF(32003),3)
p = (x + y + z + 1)^20 # the Fateman fastmult benchmark
q = p + 1
time r = p*q
```

```
Time: CPU 0.24 s, Wall: 0.24 s
```

```
P = PolynomialRing(GF(32003),10,'x')
I = sage.rings.ideal.Cyclic(P,7)
t = cputime()
gb = I.groebner_basis('libsingular:std')
print 'Sage/Singular', cputime(t)

I = sage.rings.ideal.Cyclic(P,7)
t = magma.cputime()
gb = I.groebner_basis('magma:GroebnerBasis')
print 'MAGMA', magma.cputime(t)
```

```
Sage/Singular 2.330645
MAGMA 0.5
```

Combinatorics

```
C = Combinations(range(5)); C
```

```
Combinations of [0, 1, 2, 3, 4]
```

```
C.list()
```

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

```
C.unrank(10)
```

```
[1, 2]
```

Interfaces

```
%gap
a := 1;
for i in [1..100] do if IsPrime(i) then a:=a+1; else a:=a+2; fi;
```

```
1
176
```

```
%singular
int a = 1; int i = 1;
for(i=1; i<=100; i=i+1) { if(prime(i) == i){  a=a+1; } else
{a=a+2;} };
```

```
176
```

```
%gp
a=1; for(X=1,100,if(isprime(X),a+=1,a+=2)); a
```

```
176
```

```
a= gap(1) # or gp(1), magma(1), singular(1)
for i in range(100):
    if gap(i+1).IsPrime():
        a+=1
    else:
        a+=2
a
```

```
176
```

```
a = gap(176)
type(a)
```

```
<class 'sage.interfaces.gap.GapElement'>
```

```
pari(a).factor()
```

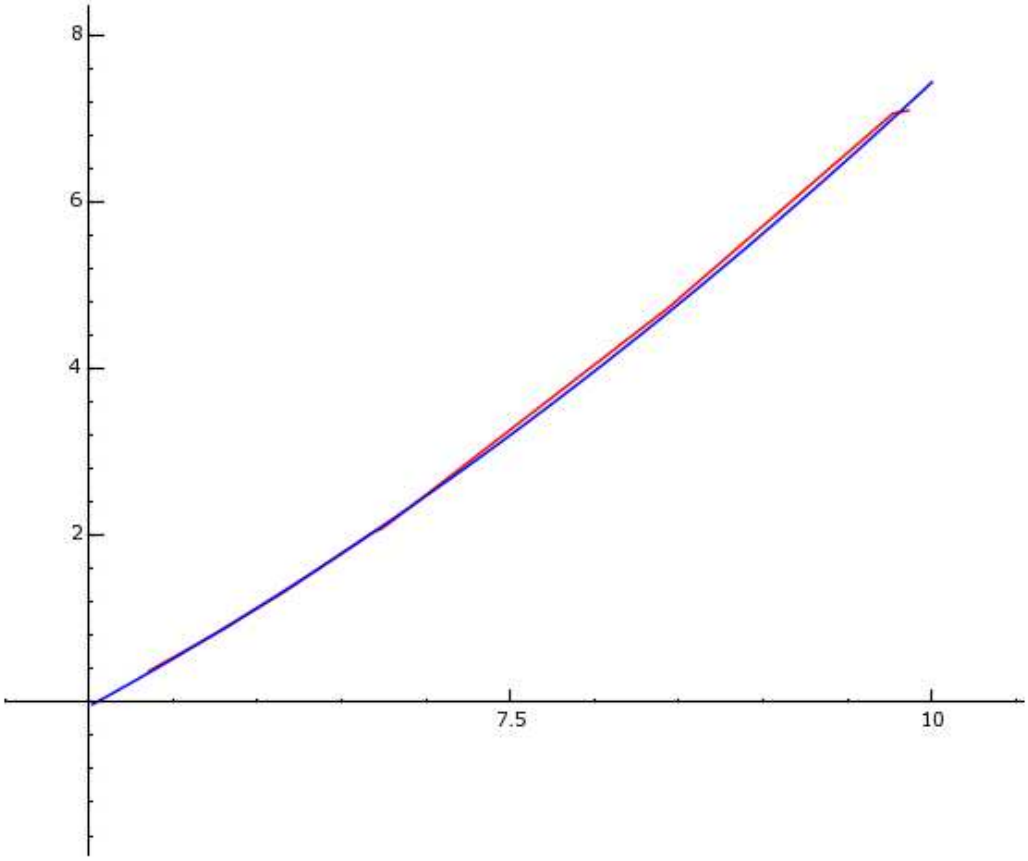
```
[2, 4; 11, 1]
```

Numerical Analysis: Curve Fitting

```
import scipy.linalg.basic

xdata = [5.357, 5.457, 5.797, 5.936, 6.161, 6.697, 6.731, 6.775,
8.442, 9.769, 9.861]
ydata = [0.376, 0.489, 0.874, 1.049, 1.327, 2.054, 2.077, 2.138,
4.744, 7.068, 7.104]
matrix = []

for x in xdata:
    matrix.append([1.0, x, x*x]) # for y = a + bx + cx^2

coeffs = scipy.linalg.basic.lstsq(matrix, ydata)[0]

# plotting etc

print "scipy.linalg.basic.lstsq curve fitting example"
print "fitting data to quadratic equation y = a + bx + cx^2"

print "yields:  x data     y data    calc value   error"
for i in range(len(xdata)):
    ycalc = coeffs[0] + coeffs[1] * xdata[i] + coeffs[2] *
xdata[i] * xdata[i]
    error = ycalc - ydata[i]
    print "         % .3f    % .3f      % .3f    % .3f" %
(xdata[i], ydata[i],
ycalc, error)
print

x = var('x')
f1 = line(zip(xdata,ydata), rgbcolor=(1,0,0))
```

```
scipy.linalg.basic.lstsq curve fitting example
```

```
fitting data to quadratic equation y = a + bx + cx^2
yields:  x data     y data    calc value   error
          5.357     0.376       0.355    -0.021
          5.457     0.489       0.472    -0.017
          5.797     0.874       0.882     0.008
          5.936     1.049       1.055     0.006
          6.161     1.327       1.341     0.014
          6.697     2.054       2.054     0.000
          6.731     2.077       2.101     0.024
          6.775     2.138       2.162     0.024
          8.442     4.744       4.688    -0.056
          9.769     7.068       7.010    -0.058
          9.861     7.104       7.181     0.077
```



Online Database: Sequences of Integers

```
for sq in sloane_find([2,3,5,7], 2):
  print sq[0], sq[1]
```

```
Searching Sloane's online database...
40 The prime numbers.
41 a(n) = number of partitions of n (the partition numbers).
```