SD15 - MSR talk

What does Sage do?

Craig Citro

http://www.sagemath.org/

Sage is a free and open-source math software package that aims to be a viable alternative to Magma, Mathematica, Maple, and Matlab.

Sage:

- solves interesting problems quickly.
- makes experimentation easy.
- interfaces with everything under the sun.
- makes creating and interacting with graphics simple.
- makes sharing resources and collaborating natural.
- shows you every line of source code.
- local or on the Web – no difference.
- is completely and totally FREE.

A Lightning-Fast Tour of Sage

Some Basics ...

```
2+2
```

```
4
```

```
factor(2009)
```

```
7^2 * 41
```

```
random_matrix(RDF,50).determinant()
```

```
8.11640778139e+19
```

```
f = x^7+1
roots = f.roots(ring=CDF)
roots
```

```
[(-1.0 + 2.22044604925e-16*I, 1), (-0.623489801859 -
0.781831482468*I, 1), (-0.623489801859 + 0.781831482468*I, 1),
(0.222520933956 - 0.974927912182*I, 1), (0.222520933956 +
0.974927912182*I, 1), (0.900968867902 - 0.433883739118*I, 1),
(0.900968867902 + 0.433883739118*I, 1)]
```

```
preparse('R.<x> = ZZ[]')
```

```
"R = ZZ['x']; (x,) = R._first_ngens(1)"
```

```
show(roots)
```

$$[(-1.0 + 2.22044604925 \times 10^{-16}i, 1),\ (-0.623489801859 - 0.781831482468i, 1),\ (-0.623489801859 + 0.781831482468i, 1),\ (0.222520933956 - 0.974927912182i, 1),\ (0.222520933956 + 0.974927912182i, 1),\ (0.900968867902 - 0.433883739118i, 1),\ (0.900968867902 + 0.433883739118i, 1)]$$

```
show(taylor(e^x, x, 0, 20))
```

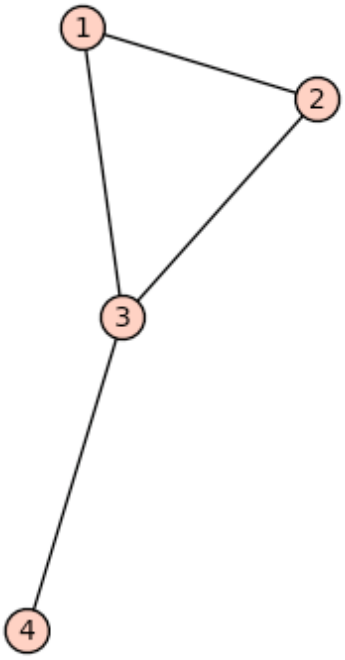
$$1 + x + \frac{x^2}{2} + \frac{x^3}{6} + \frac{x^4}{24} + \frac{x^5}{120} + \frac{x^6}{720} + \frac{x^7}{5040} + \frac{x^8}{40320} + \frac{x^9}{362880} + \frac{x^{10}}{3628800} + \frac{x^{11}}{39916800} + \frac{x^{12}}{479001}$$

Graph Theory

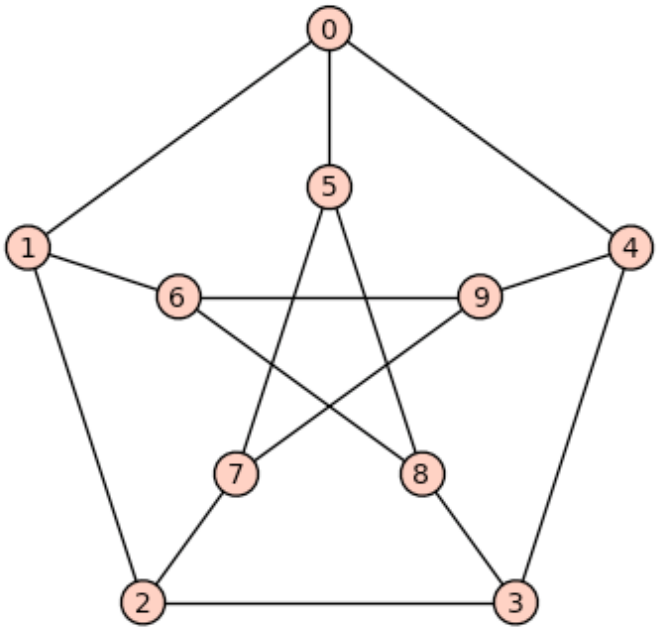
```
G = Graph({1:[2,3], 2:[], 3:[1,2,4], 4:[]})
G
```

```
Graph on 4 vertices
```

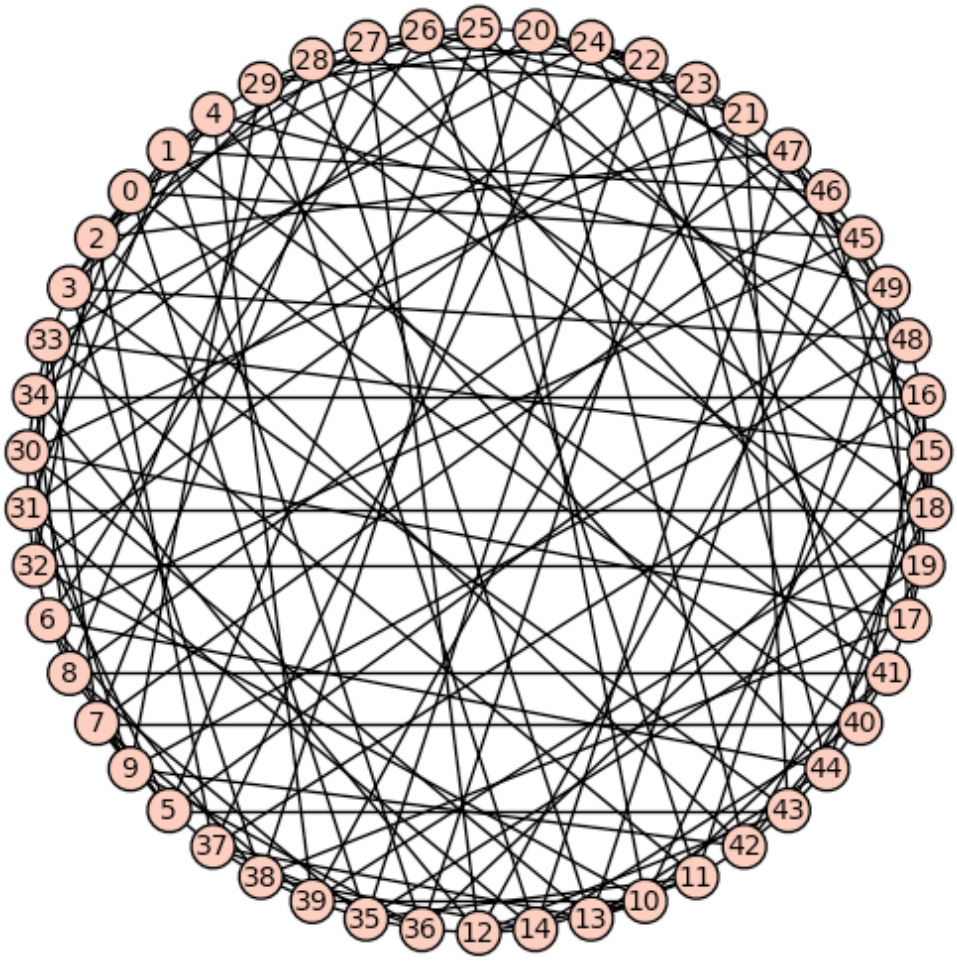
```
show(G)
```




```
show(graphs.PetersenGraph())
```




```
HS = graphs.HoffmanSingletonGraph()
HS.plot()
```




```
time HS.automorphism_group(order=True, return_group=False)
```

```
252000
Time: CPU 0.01 s, Wall: 0.01 s
```

```

```

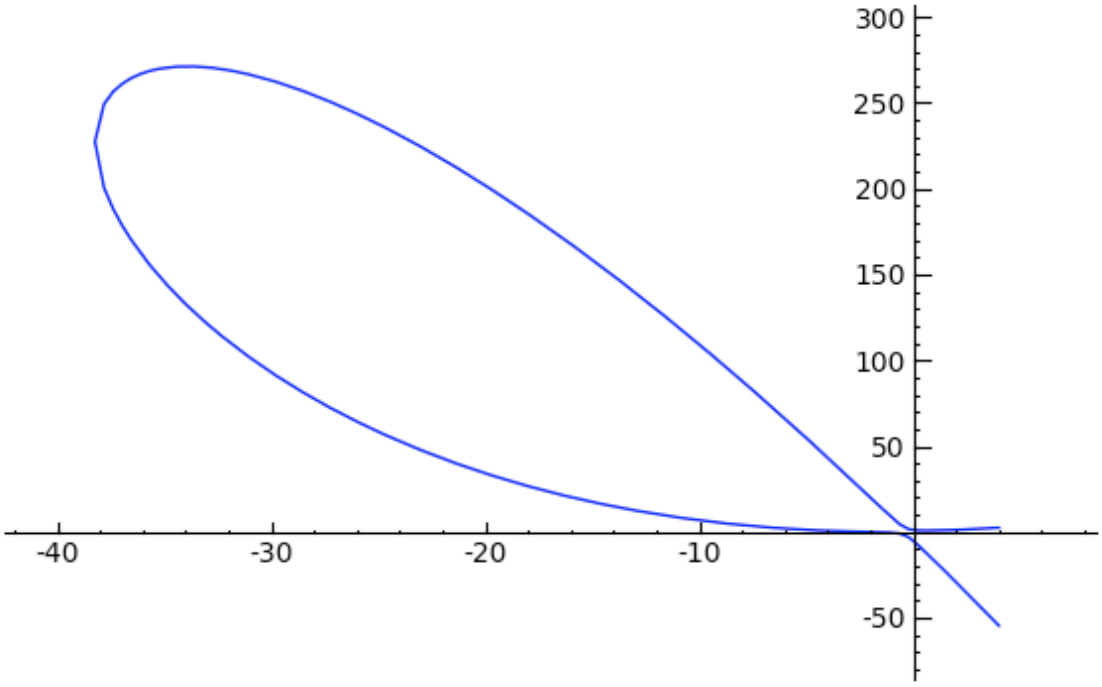
Elliptic Curves!

```
E = EllipticCurve([12,3,4,5,6])
E.global_minimal_model??
```

```
Elliptic Curve defined by y^2 + 12*x*y + 4*y = x^3 + 3*x^2 + 5*x + 6 c
```

```
# use tab completion and introspection!
```

```
show(plot(E, xmax=4))
```




```
factor(E.conductor())
```

```
2^4 * 5 * 11 * 13 * 277
```

```
time E.rank()
```

```
2
Time: CPU 0.01 s, Wall: 0.23 s
```

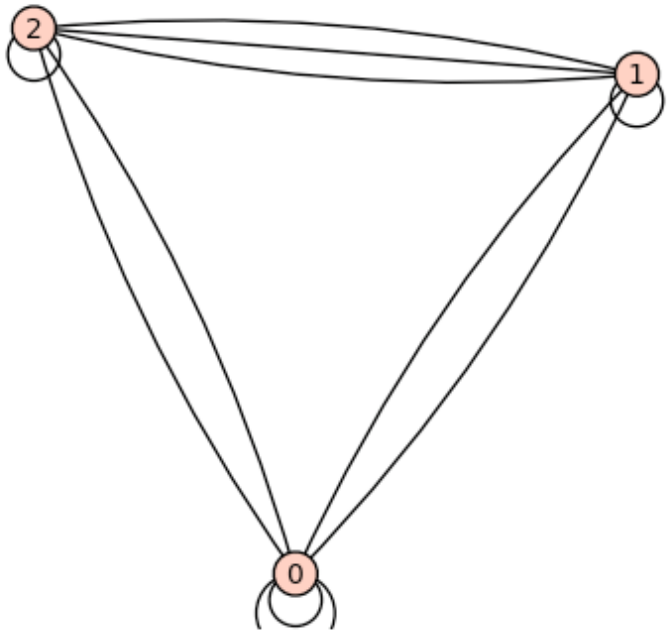
```
time E.gens()
```

```
[(-6 : 66 : 1), (-2 : 20 : 1)]
Time: CPU 0.11 s, Wall: 0.86 s
```

```
S = SupersingularModule(37)
G = Graph(S.hecke_matrix(5))
G.girth()
```

```
1
```

```
show(G)
```



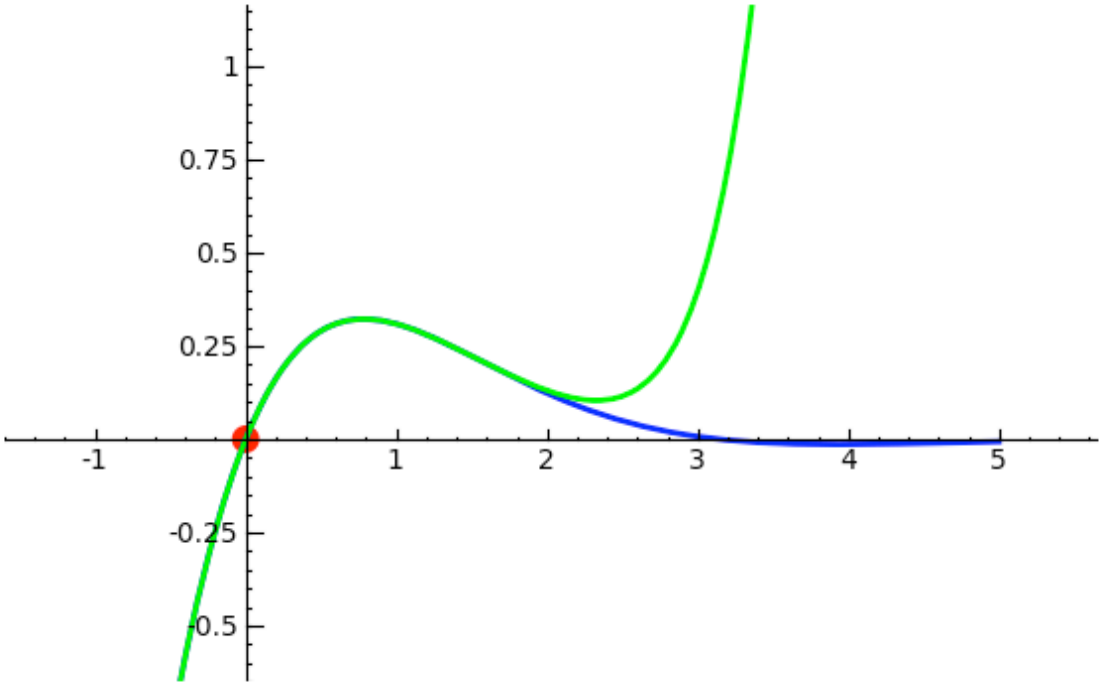
Experiment with Data Interactively ...

```
var('x')
x0   = 0
f(x) = sin(x)*e^(-x)
p    = plot(f,-1,5, thickness=2)
dot  = point((x0,f(x0)),pointsize=80,rgbcolor=(1,0,0))
@interact
def _(order=(1..12)):
 ft = f.taylor(x,x0,order)
 pt = plot(ft,-1, 5, color='green', thickness=2)
 html('$f(x)\;=\;%s$'%latex(f))
 html('$\hat{f}(x;%s)\;=\;%s+\mathcal{O}(x^{%s})$'%
(x0,latex(ft),order+1))
 show(dot + p + pt, ymin = -.5, ymax = 1)
```

order

$f(x) = x \mapsto e^{-x}\sin(x)$

$\hat{f}(x;0) = x \mapsto x - x^2 + \frac{x^3}{3} - \frac{x^5}{30} + \frac{x^6}{90} - \frac{x^7}{630} + \frac{x^9}{22680} + \mathcal{O}(x^{10})$




```
var('x y')
var('xx yy', ns=1)
G = sin(xx^2 + yy^2) * cos(yy) * exp(-0.5*(xx^2+yy^2))
def F(x,y):
   return G.subs(xx=x).subs(yy=y)
plotF = plot3d(F, (0.4, 2), (0.4, 2), adaptive=True, color='blue')
@interact
def _(x0=(0.5,1.5), y0=(0.5, 1.5),
     order=(1..10)):
   F0 = float(G.subs(xx=x0).subs(yy=y0))
   P = (x0, y0, F0)
   dot = point3d(P, size=15, color='red')
   plot = dot + plotF
   approx = F0
   for n in range(1, order+1):
       for i in range(n+1):
           if i == 0:
               deriv = G.diff(yy, n)
           elif i == n:
               deriv = G.diff(xx, n)
           else:
               deriv = G.diff(xx, i).diff(yy, n-i)
           deriv = float(deriv.subs(xx=x0).subs(yy=y0))
           coeff = binomial(n, i)/factorial(n)
           approx += coeff * deriv * (x-x0)^i * (y-y0)^(n-i)
   plot += plot3d(approx, (x, 0.4, 1.6),
            (y, 0.4, 1.6), color='red', opacity=0.7)
```

```
    html('$F(x,y) = e^{-(x^2+y^2)/2} \\cos(y) \\sin(x^2+y^2)$')
    show(plot)
```

x0

y0

order

$F(x,y) = e^{-(x^2+y^2)/2}\cos(y)\sin(x^2+y^2)$

Features Galore ...

```
# interfaces
(gp(2) + gap(5)) * singular(7)
```

```
    49
```

```
%r

c(1:10)
d <- c(1:15)
mean(d)
```

```
     [1]  1  2  3  4  5  6  7  8  9 10

    [1] 8
```

```
M = random_matrix(ZZ,50)
```

```
time M*M
```

```
    50 x 50 dense matrix over Integer Ring
    Time: CPU 0.01 s, Wall: 0.01 s
```

```
timeit('M*M')
```

```
    125 loops, best of 3: 6.24 ms per loop
```

```
x = 3
y = 5
timeit('x*y')
```

```
    625 loops, best of 3: 259 ns per loop
```

```
# easy naive parallelism ... no licenses required

ls = [2^n-1 for n in [190..200]]
```

```
time v = [ factor(x) for x in ls ]
```

```
Time: CPU 5.99 s, Wall: 6.81 s
```

```
@parallel(2)
def f_para(n):
   return factor(n)

time v = list(f_para(ls))
```

```
Time: CPU 0.03 s, Wall: 3.86 s
```

```

```

The Sage Notebook is the web-based interface to Sage that I'm using to give this talk -- and that I used to write it. The Notebook has a number of important features, including:

- WYSIWYG HTML editor
- easily usable over a network
- jsMath integration
- LaTeX integration

Hi my name is **Craig**

Unknown control sequence '\operatorname'

```
html('$$ \\textbf{Conjecture: } \\operatorname{Re}(s) > 0, \\zeta(s) =
0 \\longrightarrow \\operatorname{Re}(s) = \\frac{1}{2} $$')
```

Unknown control sequence '\operatorname'

```
M = random_matrix(ZZ,3)
```

```
show(M)
```

$$\begin{pmatrix} -1 & 16 & 3 \\ -1 & -1 & -4 \\ -1 & -1 & -8 \end{pmatrix}$$

```
E
```

```
Elliptic Curve defined by y^2 + 12*x*y + 4*y = x^3 + 3*x^2 + 5*x + 6
over Rational Field
```

```
latex(E)
```

```
y^2 + 12xy + 4y = x^3 + 3x^2 + 5x + 6
```

```
show(E)
```

$$y^2 + 12xy + 4y = x^3 + 3x^2 + 5x + 6$$

```
show(random_matrix(ZZ,4))
```

$$\begin{pmatrix} 0 & 0 & 0 & 7 \\ -7 & -1 & 1 & 1 \\ -1 & 1 & -2 & 1 \\ 0 & -1 & -1 & -1 \end{pmatrix}$$

```

```

Cython

```
%python
def mysum(N):
   s = int(0)
   for k in range(1,N):
       s += k
   return s
```

```
time mysum(10^6)
```

```
499999500000L
Time: CPU 0.26 s, Wall: 0.27 s
```

```
%cython
def mysum_cython(N):
   cdef int k
   cdef long long s = 0

   for k in range(N):
       s += k
   return s
```

__Users_cr...6_code_sage66_spyx.c __Users_cr...ode_sage66_spyx.html

```
time mysum_cython(10^6)
```

```
499999500000L
Time: CPU 0.00 s, Wall: 0.00 s
```

```

```

Questions?