seminar

An Introduction to Sage

Marco Varisco

Algebra/Topology Seminar, February 7, 2013

- Sage as a (smart) calculator
- A simple algorithm
- Some calculus and pretty pictures
- Solving equations
- Playing with finite groups
- Minimal free resolutions of monomial ideals

Sage as a *(smart)* calculator

```
2^16
```

```
65536
```

```
2^160
```

```
1461501637330902918203684832716283019655932542976
```

```
sin(pi/3)
```

```
1/2*sqrt(3)
```

```
numerical_approx(sin(pi/3)); n(sin(pi/3)); sin(pi/3).n()
```

```
0.866025403784439
0.866025403784439
0.866025403784439
```

```
n(sin(pi/3), digits=50)
```

```
0.86602540378443864676372317075293618347140262690519
```

```
sin(pi/3).n(digits=3)
```

```
0.866
```

```
a = 12222
```

```
a
```

```
12222
```

```
factor(a)
```

```
2 * 3^2 * 7 * 97
```

```
a.factor()
```

```
2 * 3^2 * 7 * 97
```

```
a.prime_divisors()
```

```
[2, 3, 7, 97]
```

```
pd = _
pd
```

```
[2, 3, 7, 97]
```

```
pd[0]
```

```
2
```

```
pd.append('text')
pd
```

```
[2, 3, 7, 97, 'text']
```

```
a % 55
```

```
12
```

```
a.inverse_mod(55)
```

```
23
```

```
a.inverse_mod(56)
```

```
Traceback (click to the left of this block for traceback)
...
ZeroDivisionError: Inverse does not exist.
```

```
A = mod(a, 55)
```

```
type(a); type(A)
```

```
<type 'sage.rings.integer.Integer'>
<type 'sage.rings.finite_rings.integer_mod.IntegerMod_int'>
```

```
A^-1
```

```
23
```

```
a^-1
```

```
1/12222
```

```
A.multiplicative_order()
```

```
4
```

A simple algorithm

```
def EA(a, b):
    while b!=0:
        r = a%b
        a = b
        b = r
```

```
    return a
```

```
EA(12222, 55)
```

```
    1
```

```
EA(12222, 550)
```

```
    2
```

```
def EA(a, b, show=False):
    while b!=0:
        r = a%b
        if show: print (a,b,r)
        a = b; b = r
    if a<0: a = -a
    return a
```

```
EA(12222, 550, show=True)
```

```
    (12222, 550, 122)
    (550, 122, 62)
    (122, 62, 60)
    (62, 60, 2)
    (60, 2, 0)
    2
```

```
EA(12222, 550) == gcd(12222, 550)
```

```
    True
```

```
xgcd(12222, 550)
```

```
    (2, -9, 200)
```

```
d, s, t = xgcd(12222, 550)
```

```
d == s*12222 + t*550
```

```
    True
```

Some calculus and pretty pictures

```
f = atan(sqrt(x))
f
```

```
    arctan(sqrt(x))
```

```
type(f)
```

```
    <type 'sage.symbolic.expression.Expression'>
```

```
If = f.integrate(x)
If
```

```
    x*arctan(sqrt(x)) - sqrt(x) + arctan(sqrt(x))
```

```
show(If)
```

$$x \arctan\left(\sqrt{x}\right) - \sqrt{x} + \arctan\left(\sqrt{x}\right)$$

```
latex(If)
```

```
x \arctan\left(\sqrt{x}\right) - \sqrt{x} +
\arctan\left(\sqrt{x}\right)
```

At this point you should check the **Typeset** box at the top of this worksheet.

```
DIf = If.differentiate(x)
f - DIf
```

$$-\frac{\sqrt{x}}{2\,(x+1)} + \frac{1}{2\,\sqrt{x}} - \frac{1}{2\,(x+1)\,\sqrt{x}}$$

```
(f - DIf).simplify_full()
```

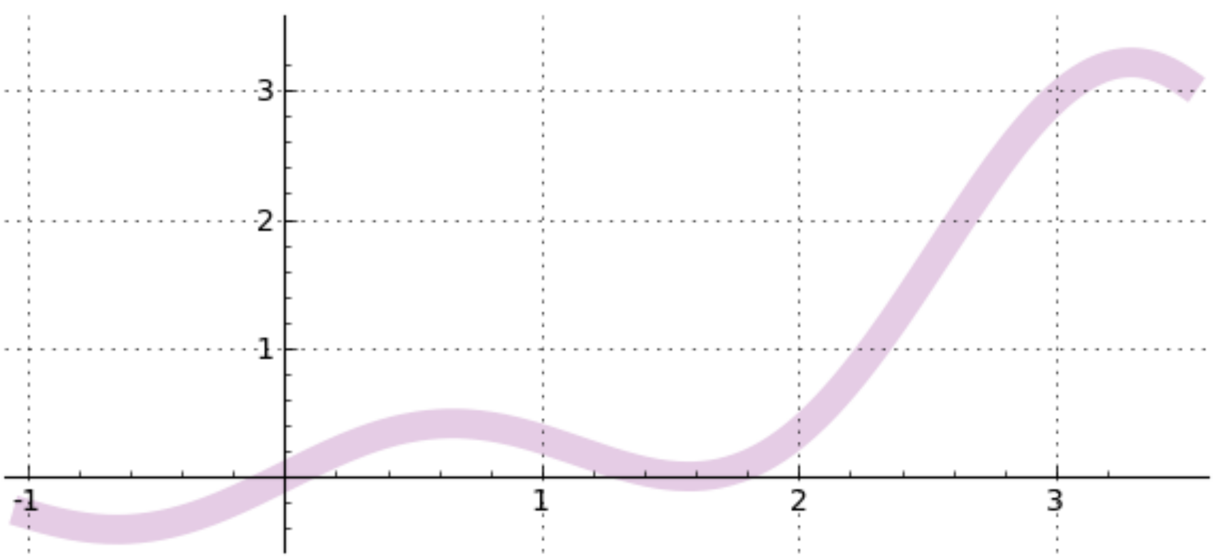
$$0$$

```
f = x*cos(x)^2
f
```

$$x \cos\left(x\right)^{2}$$

```
plot(f)
```

```
p = plot(f, xmin=-1, xmax=3.5, ymin=-0.5, ymax=3.5, aspect_ratio=.5,
gridlines=True, color='purple', thickness=10, alpha=0.2, figsize=6)
p
```




```
p.save('plot.pdf')
```

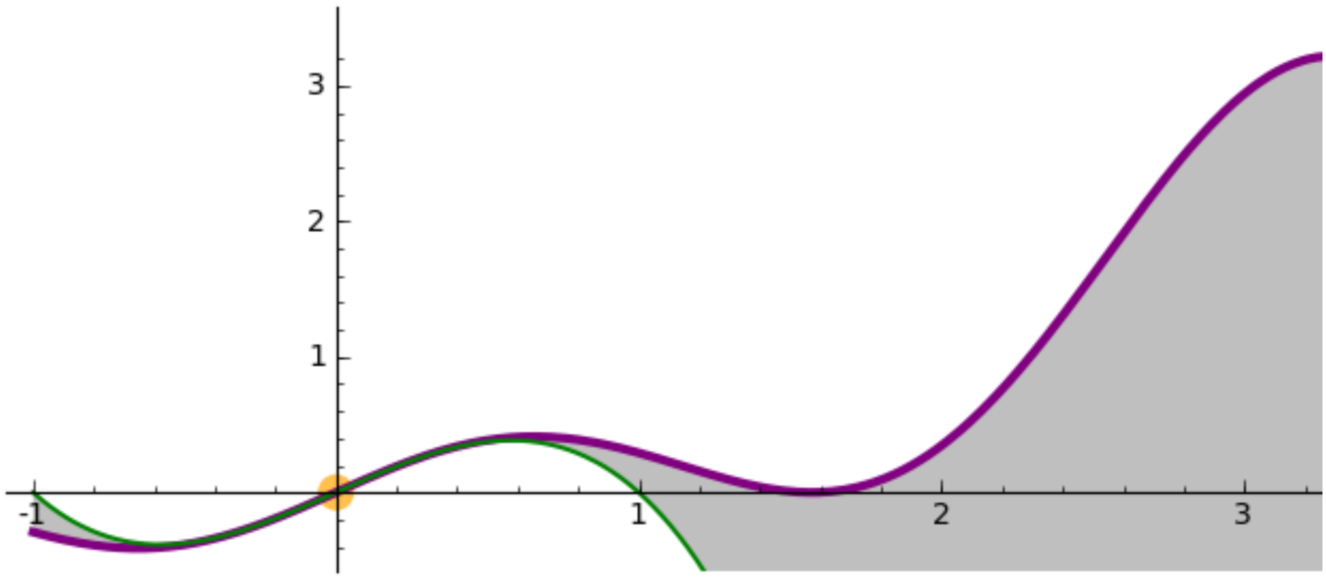
plot.pdf

```
plotf = plot(f, (x,-1,3.5), color='purple', thickness=3)
origin = point((0,0), color='orange', alpha=.7, size=150)
label = 'MacLaurin polynomial of $%s$ of degree'%latex(f)
```

```
@interact
def foo(j=slider(0, 20, 1, default=3, label=label)):
    Tjf = f.taylor(x, 0, j)
    plotTjf = plot(Tjf, (x,-1,3.5), color='green', thickness=1.5,
fill=f)
    html('$%s$'%latex(Tjf))
    show(plotf + plotTjf + origin, ymin=-0.5, ymax=3.5, figsize=
[7,3])
```

MacLaurin polynomial of $x\cos(x)^2$ of degree 3

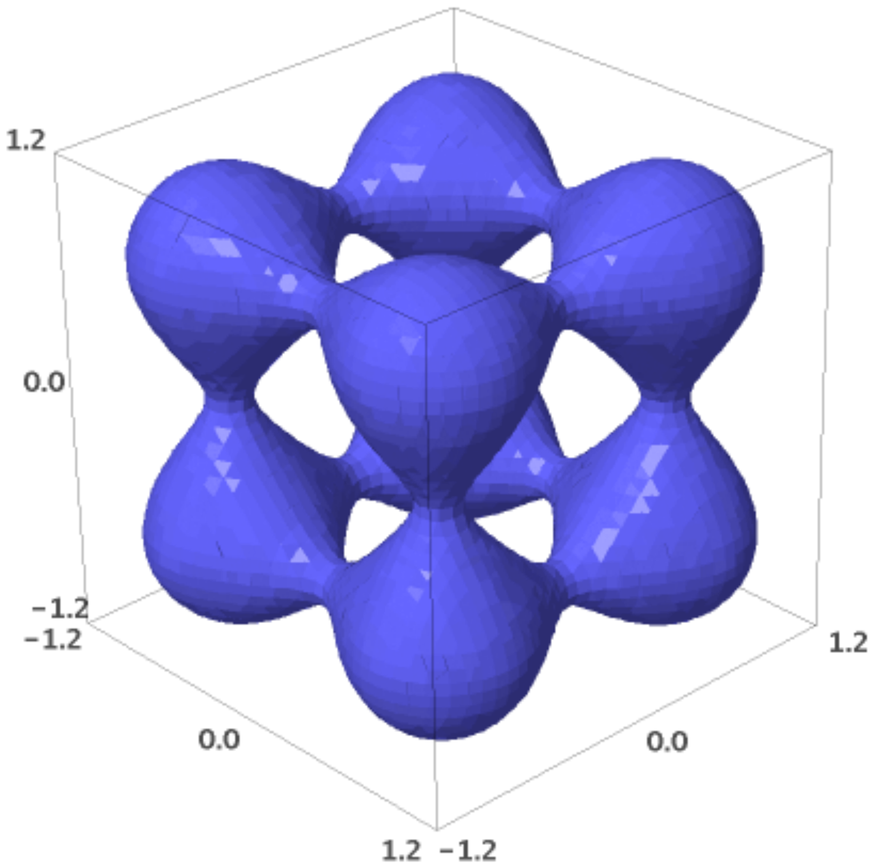
$-x^3 + x$




```
frames = []
for j in range(-1, 21, 2):
    Tjf = f.taylor(x, 0, j)
    plotTjf = plot(Tjf, (x,-1,3.5), color='green', thickness=1.5,
fill=f)
    t = text('$%s$'%latex(Tjf), (3,-0.8), color='black',
horizontal_alignment='right')
    frames.append(t + plotf + plotTjf + origin)
a = animate(frames, ymin=-0.5, ymax=3.5, figsize=[7,3])
a.show(delay=40, iterations=4)
```

```
var('x, y, z')
f = x^2 + y^2 + z^2 + cos(4*x) + cos(4*y) + cos(4*z)
c = 0.2
implicit_plot3d(f==c, (x, -1.2, 1.2), (y, -1.2, 1.2), (z, -1.2, 1.2))
```

Sleeping... Make Interactive




```
implicit_plot3d(f==c, (x, -1.2, 1.2), (y, -1.2, 1.2), (z, -1.2, 1.2),
opacity=2/3) + dodecahedron((0,0,0), 1/2, color="purple",
opacity=2/3)
```

Solving equations

```
solve(x^3 + 6*x == 20, x)
```

$$[x = (-3i - 1), x = (3i - 1), x = 2]$$

```
solve(x^4 + 6*x == 20, x)[0]
```

$$x = -\frac{1}{2}\sqrt{\frac{3\left(\frac{2}{9}\sqrt{3}\sqrt{130187}+18\right)^{\left(\frac{2}{3}\right)}-80}{\left(\frac{2}{9}\sqrt{3}\sqrt{130187}+18\right)^{\frac{1}{3}}}}\sqrt{\frac{1}{3}} - \frac{1}{2}\sqrt{-\left(\frac{2}{9}\sqrt{3}\sqrt{130187}+18\right)^{\left(\frac{1}{3}\right)} + \frac{36\sqrt{\frac{1}{3}}}{\sqrt{\frac{3\left(\frac{2}{9}\sqrt{3}\sqrt{130187}+18\right)^{(\cdot}}{\left(\frac{2}{9}\sqrt{3}\sqrt{130187}+18\right)}}}}$$

```
solve(x^5 + 6*x == 20, x)
```

$[0 = x^5 + 6x - 20]$

```
find_root(x^5 + 6*x == 20, 0, 1)
```

```
Traceback (click to the left of this block for traceback)
...
RuntimeError: f appears to have no zero on the interval
```

```
find_root(x^5 + 6*x == 20, 0, 2)
```

1.59778267898

```
var('x, y, z')
solve([x + 3*y - 2*z == 5, 3*x + 5*y + 6*z == 7], x, y, z)
```

$[[x = -7r_1 - 1, y = 3r_1 + 2, z = r_1]]$

```
A = matrix([[1, 3, -2], [3, 5, 6]])
v = vector([5, 7])
A.solve_right(v)
```

$(-1, 2, 0)$

```
A.right_kernel()
```

$\mathrm{RowSpan}_{\mathbf{Z}}\begin{pmatrix} 7 & -3 & -1 \end{pmatrix}$

```
Av = A.augment(v)
Av.echelon_form()
```

$$\begin{pmatrix} 1 & 3 & -2 & 5 \\ 0 & 4 & -12 & 8 \end{pmatrix}$$

```
type(Av)
```

<type 'sage.matrix.matrix_integer_dense.Matrix_integer_dense'>

```
Av = Av.change_ring(QQ)
type(Av); Av.echelon_form()
```

<type 'sage.matrix.matrix_rational_dense.Matrix_rational_dense'>

$$\begin{pmatrix} 1 & 0 & 7 & -1 \\ 0 & 1 & -3 & 2 \end{pmatrix}$$

Playing with finite groups

Unckeck the **Typeset** box now, please.

```
S4 = SymmetricGroup(4)
```

```
S4
```

```
Symmetric group of order 4! as a permutation group
```

```
show(S4)
```

$\langle (1,2,3,4), (1,2) \rangle$

```
S4.conjugacy_classes_representatives()
```

```
[(), (1,2), (1,2)(3,4), (1,2,3), (1,2,3,4)]
```

```
len(S4.conjugacy_classes_representatives()), len(S4.subgroups()),
len(S4.conjugacy_classes_subgroups()), len(S4.normal_subgroups())
```

```
(5, 30, 11, 4)
```

```
u = S4( (1,2,3,4) )
v = S4( (2,4) )
w = S4( ((1,2),(3,4)) )
```

```
u.order(), u*v, w==u^-1*v
```

```
(4, (1,4)(2,3), True)
```

```
G = S4.subgroup([u, v])
```

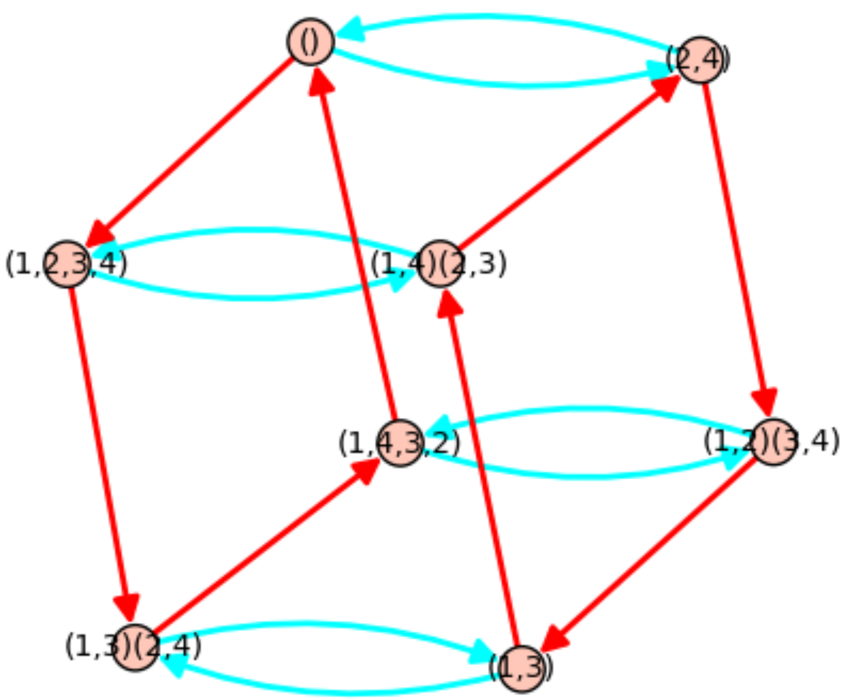
```
G.order(), G.is_abelian(), G.center().order(),
G.is_isomorphic(DihedralGroup(4))
```

```
(8, False, 2, True)
```

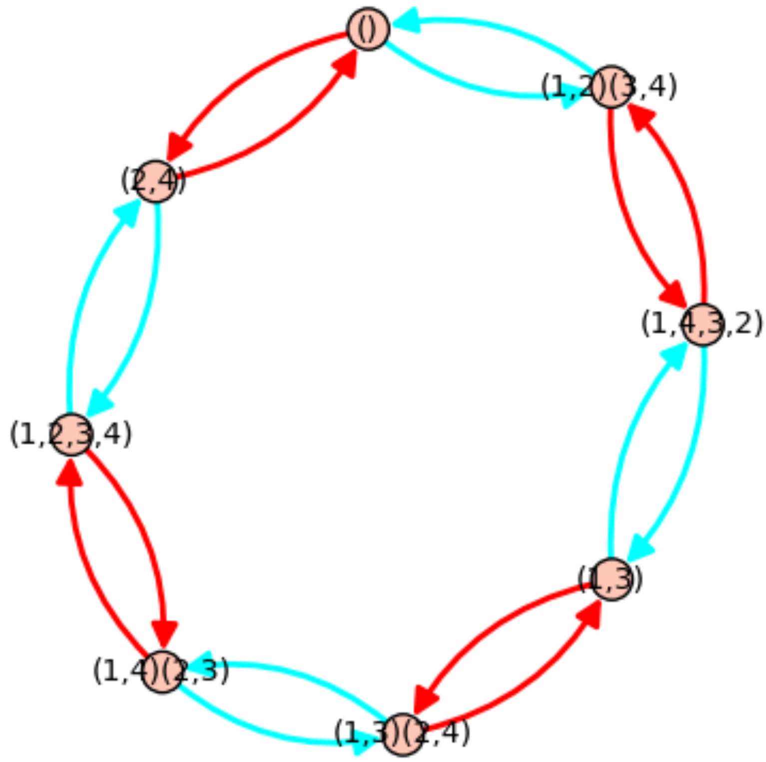
```
(1,4) in G, (1,3) in G
```

```
(False, True)
```

```
G.cayley_graph(generators=[u, v]).show(color_by_label=True)
```




```
G.cayley_graph(generators=[v, w]).show(color_by_label=True)
```




```
T = G.cayley_table()
T
```

```
 *  a b c d e f g h
  +----------------
 a| a b c d e f g h
 b| b a d c f e h g
 c| c g a e d h b f
 d| d h b f c g a e
 e| e f g h a b c d
 f| f e h g b a d c
 g| g c e a h d f b
 h| h d f b g c e a
```

```
T.translation()
```

```
{'a': (), 'c': (1,2)(3,4), 'b': (2,4), 'e': (1,3), 'd': (1,2,3,4),
'g': (1,4,3,2), 'f': (1,3)(2,4), 'h': (1,4)(2,3)}
```

```
from sage.matrix.operation_table import OperationTable
def commutator(h, g): return h*g*h^-1*g^-1
OperationTable(G, commutator)
```

```
 .  a b c d e f g h
  +----------------
 a| a a a a a a a a
 b| a a f f a a f f
 c| a f a f f a f a
 d| a f f a f a a f
 e| a a f f a a f f
 f| a a a a a a a a
```

```
g| a f f a f a a f
h| a f a f f a f a
```

Minimal free resolutions of monomial ideals

```
R.<a,b,c,d,e> = PolynomialRing(QQ)
I = R.ideal([a*c, b*d, a*e, d*e])
R; I
```

```
Multivariate Polynomial Ring in a, b, c, d, e over Rational Field
Ideal (a*c, b*d, a*e, d*e) of Multivariate Polynomial Ring in a, b,
c, d, e over Rational Field
```

```
I.syzygy_module()
```

```
[  -e    0    c    0]
[   0   -e    0    b]
[   0    0   -d    a]
[-b*d  a*c    0    0]
```

```
singular.mres(I, 0)
```

```
[1]:
   _[1]=d*e
   _[2]=a*e
   _[3]=b*d
   _[4]=a*c
[2]:
   _[1]=c*gen(2)-e*gen(4)
   _[2]=b*gen(1)-e*gen(3)
   _[3]=a*gen(1)-d*gen(2)
   _[4]=a*c*gen(3)-b*d*gen(4)
[3]:
   _[1]=a*c*gen(2)-b*c*gen(3)-b*d*gen(1)+e*gen(4)
[4]:
   _[1]=0
[5]:
   _[1]=gen(1)
```

And don't forget the SageTeX example! ∞