Sage: an open-source mathematics software

Paul Zimmermann

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE | INRIA
centre de recherche NANCY - GRAND-EST

2nd SCIEnce Workshop
Palaiseau, France
January 19, 2009

Disclaimer

- I'm not the best person to talk about Sage

Disclaimer

- I'm not the best person to talk about Sage
- I'm biased towards Sage for the following reasons:

Disclaimer

- I'm not the best person to talk about Sage
- I'm biased towards Sage for the following reasons:
  - it is open-source

Disclaimer

- I'm not the best person to talk about Sage
- I'm biased towards Sage for the following reasons:
  - it is open-source
  - it uses some of the software tools I've contributed to

Disclaimer

- I'm not the best person to talk about Sage
- I'm biased towards Sage for the following reasons:
  - it is open-source
  - it uses some of the software tools I've contributed to
  - its developers give feedback to the "upstream" packages

Disclaimer

- I'm not the best person to talk about Sage
- I'm biased towards Sage for the following reasons:
  - it is open-source
  - it uses some of the software tools I've contributed to
  - its developers give feedback to the "upstream" packages
- I will do my best to be objective

Plan

1. Sage: a computer algebra system
2. Sage: a bridge between different tools
3. Sage: an active community

A Computer Algebra System

Sage Basics

```
bash-3.00$ sage
----------------------------------------------------------------------
| Sage Version 3.2.3, Release Date: 2009-01-05                       |
| Type notebook() for the GUI, and license() for information.        |
----------------------------------------------------------------------
sage: 17+42
59
```

Sage Basics

```
bash-3.00$ sage
------------------------------------------------------------------------
| Sage Version 3.2.3, Release Date: 2009-01-05                         |
| Type notebook() for the GUI, and license() for information.          |
------------------------------------------------------------------------
sage: 17+42
59
sage: l=[x,1,'a',x]
sage: l.<tab>
l.append   l.extend   l.insert   l.remove   l.sort
l.count    l.index    l.pop      l.reverse
sage: l.count?
Type:           builtin_function_or_method
Base Class:     <type 'builtin_function_or_method'>
String Form:    <built-in method count of list object at 0xb636906c>
Namespace:      Interactive
Docstring:
    L.count(value) -> integer -- return number of occurrences of value
Class Docstring:
    <attribute '__doc__' of 'builtin_function_or_method' objects>
```

Sage Basics (continued)

```
sage: l.count(x)
2
sage: len(l)
4
sage: l[0:4]
[x, 1, 'a', x]
```

Classes of Objects

```
sage: R = RealIntervalField(42)
sage: a = R((exp(1), pi))
sage: a
3.?
sage: a.lower(), a.upper()
(2.71828182845, 3.14159265360)
sage: b = sin(a)
sage: b.lower(), b.upper()
(-3.30279907449e-13, 0.410781290504)
sage: (a*b).diameter()
1.29050748448
```

Calculus

```
sage: eq = x^5+x+1 == 0
sage: sol = solve(eq); sol
[x == (sqrt(3)*I/2 - 1/2)/(9*(sqrt(23)/(6*sqrt(3)) - 25/54)^(1/3))
    + (sqrt(23)/(6*sqrt(3)) - 25/54)^(1/3)*(-sqrt(3)*I/2 - 1/2) + 1/3,
x == (sqrt(23)/(6*sqrt(3)) - 25/54)^(1/3)*(sqrt(3)*I/2 - 1/2)
+ (-sqrt(3)*I/2 - 1/2)/(9*(sqrt(23)/(6*sqrt(3)) - 25/54)^(1/3)) + 1/3,
x == (sqrt(23)/(6*sqrt(3)) - 25/54)^(1/3) + 1/(9*(sqrt(23)/(6*sqrt(3))
   - 25/54)^(1/3)) + 1/3,
x == (-sqrt(3)*I - 1)/2,
x == (sqrt(3)*I - 1)/2]
sage: [s.rhs().n() for s in sol]

[0.877438833123343 + 0.744861766619737*I,
 -0.754877666246686 - 4.16333634234434e-17*I,
 0.877438833123343 - 0.744861766619737*I,
 -0.500000000000000 - 0.866025403784439*I,
 -0.500000000000000 + 0.866025403784439*I]

sage: eq.substitute(x=sol[0].rhs()).rhs().n()
4.52970994047064e-14 + 4.21468415723325e-14*I
```

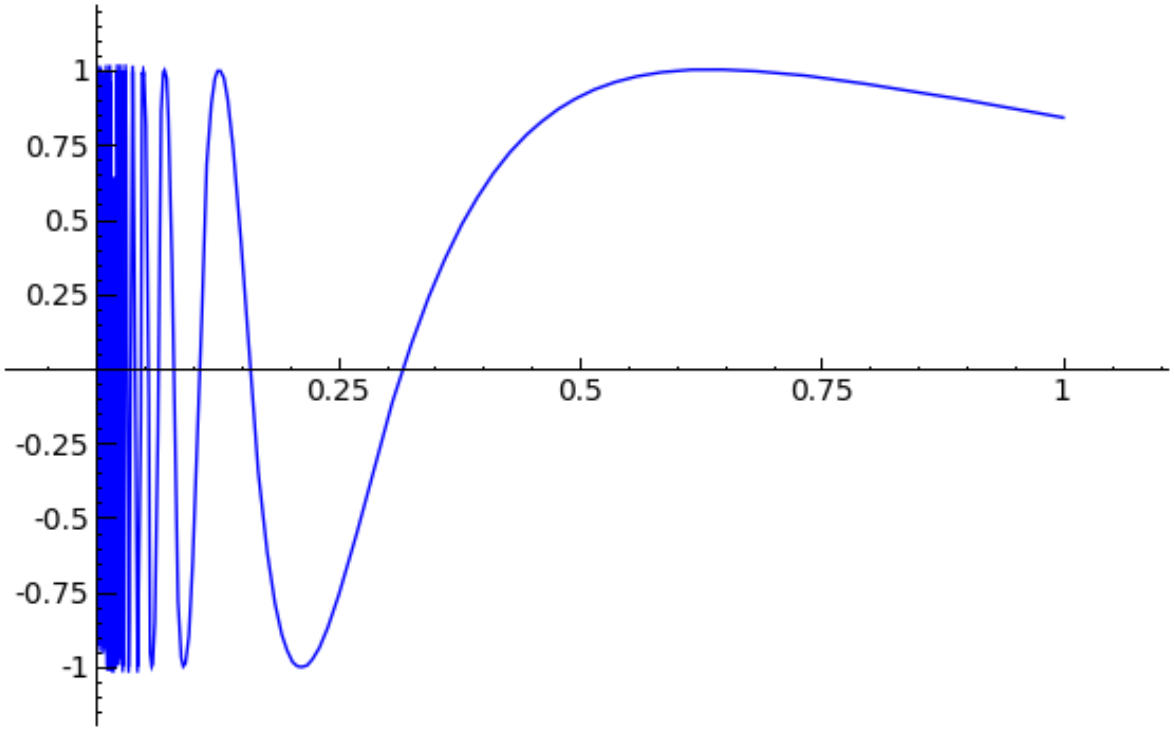
```
sage: [eq.substitute(x=s.rhs()).rhs().n() for s in sol]

[4.52970994047064e-14 + 4.21468415723325e-14*I,
 1.70974345792274e-14 - 1.91727504138695e-16*I,
 4.41868763800812e-14 - 4.46309655899313e-14*I,
 1.99840144432528e-15 + 6.66133814775094e-16*I,
 1.99840144432528e-15 - 6.66133814775094e-16*I]
```

```
sage: diff(e^e^e^e^x, x, 4)
(((e^x + 1)*e^(e^x + x) + e^x)*e^(e^e^x + e^x + x)
+ (e^(e^x + x) + e^x + 1)^2*e^(e^e^x + e^x + x)
+ e^(e^x + 2*x) + (e^x + 1)^2*e^(e^x + x) + e^x)
*e^(e^e^e^x + e^e^x + e^x + x) + 3*(e^(e^e^x + e^x
+ x) + e^(e^x + x) + e^x + 1)*((e^(e^x + x) + e^x
+ 1)*e^(e^e^x + e^x + x) + (e^x + 1)*e^(e^x + x)
+ e^x)*e^(e^e^e^x + e^e^x + e^x + x) + (e^(e^e^x
+ e^x + x) + e^(e^x + x) + e^x + 1)^3*e^(e^e^e^x
+ e^e^x + e^x + x)
sage: integrate(integrate(integrate(integrate(_,
      x), x), x), x)
e^e^e^e^x
```

```
sage: e4 = diff(e^e^e^e^x, x, 4)
sage: f = lambda e: integrate(e,x)
sage: f(f(f(f(e4))))
e^e^e^e^x
```

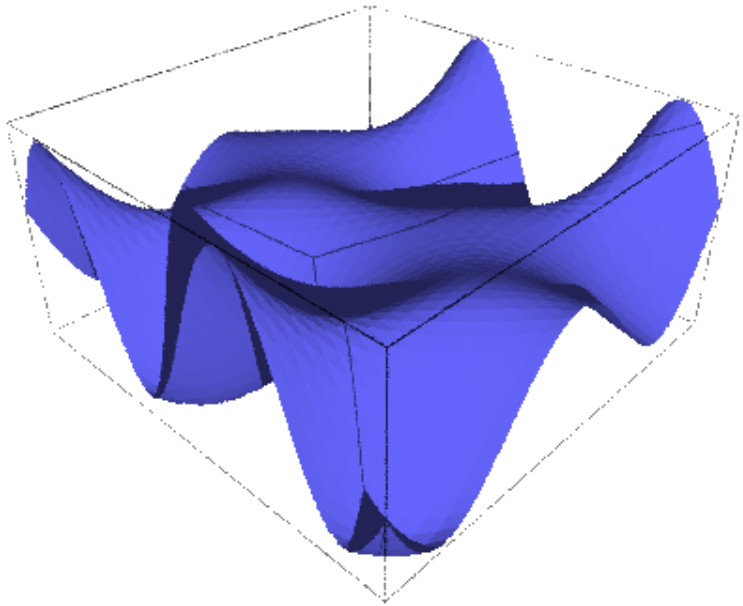
Graphics

```
sage: plot(sin(1/x), (x, 0, 1))
```



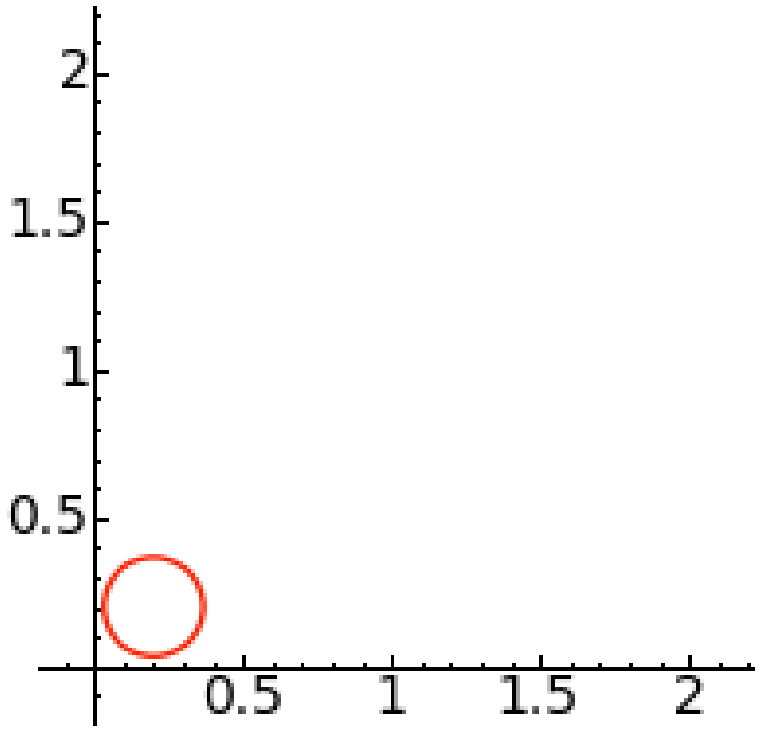
Graphics

```
sage: var('y');
sage: P=plot3d(sin(x-y)*y*cos(x),(x,-3,3),(y,-3,3),
sage: show(P, viewer='tachyon')
```



Graphics

```
sage: a = animate([circle((i,i), 1-1/(i+1), hue=i/10) for i in
      srange(0,2,0.2)], xmin=0,ymin=0,xmax=2,ymax=2,figsize=[2,2])
```



Basic Rings

```
sage: a = ZZ(17)
sage: b = QQ(3/4)
sage: c = a+b
sage: type(c)
<type 'sage.rings.rational.Rational'>
sage: d = RR(pi)
```

Polynomials

```
sage: P.<x> = PolynomialRing(GF(17))
sage: p = P.random_element(); p
16*x^2 + 6*x + 9
sage: p^3
16*x^6 + x^5 + 4*x^4 + 11*x^3 + 15*x^2 + 13*x + 15
sage: p.roots()
[(4, 1), (2, 1)]
```

Matrices

```
sage: m = Matrix(P,2,2)
sage: m.randomize(); m

[13*x^2 + 12*x + 15      11*x^2 + x + 8]
[          4*x + 14  16*x^2 + 11*x + 1]
sage: m.rank()
...
NotImplementedError: echelon form over Univariate
Polynomial Ring in x over Ring of integers modulo
17 not yet implemented
sage: m.det()
4*x^4 + 2*x^3 + 6*x^2 + 12*x + 5
sage: factor(_)
(4) * (x^4 + 9*x^3 + 10*x^2 + 3*x + 14)
```

Useful Commands

```
sage: search_src ("integration", extra1="numerical")
calculus/calculus.py:
   numerical integration using the GSL C library.  It is po
functions/piecewise.py:
   Riemann sums in numerical integration based on a subdivi
functions/transcendental.py:
   from sage.gsl.integration import numerical_integral
gsl/all.py:
   from integration import numerical_integral
interfaces/maxima.py:
   Note that GP also does numerical integration, and can do
ext/fast_eval.pyx:
   For many applications such as numerical integration, di
gsl/integration.pyx:
   that implements numerical integration using Maxima.  It
```

```
sage: *integra*?
exponential_integral_1
half_integral_weight_modform_basis
integral
integral_closure
integral_numerical
integrate
is_integrally_closed
numerical_integral
```

```
sage: numerical_integral(sin(1/x), 0, 1)
(0.50407021996807966, 0.00012692441400448127)
sage: numerical_integral(sin(1/x), 0, 1, max_points=100000)
(0.50406704976674 91, 9.9983194154764902e-07)
```

Programming in Sage

```
sage: def isprime(p):
    i = 2
    while i*i <= p:
        if p % i == 0:
            return False
        i = i + 1
    return True
for p in range(1000):
    if isprime(p) <> is_prime(p):
        raise ValueError, p
```

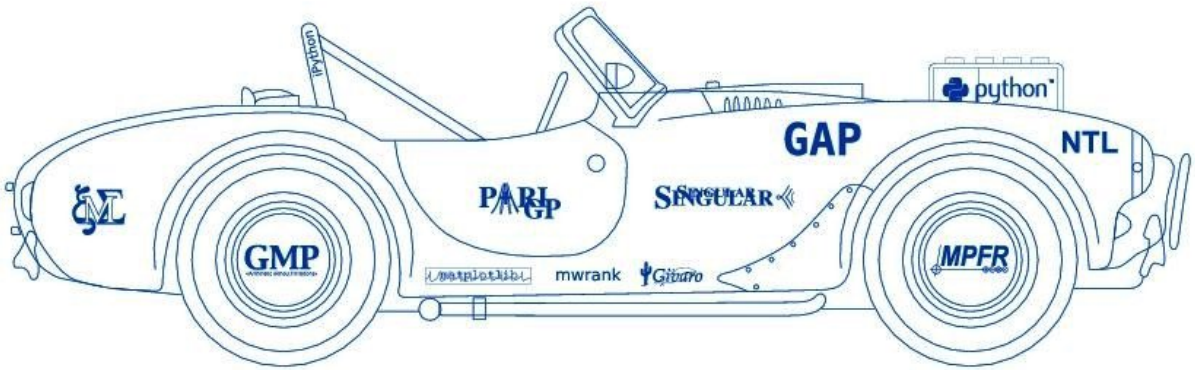
A useful command when programming:

```
sage:  attach xxx.sage
```

A Bridge Between Different Tools

SAGE

Building »The Car«



»Every free computer algebra system I've tried has reinvented many times the wheel without being able to build the car.«

`http://www.sagemath.org/links-components.html`:

ATLAS, BLAS, Bzip2, Cddlib, Common Lisp, CVXOPT, Cython, mwrank, F2c, Flint, FpLLL, FreeType, G95, GAP, GD, Genus2reduction, Gfan, Givaro, GMP, GMP-ECM, GNU TLS, GSL, JsMath, IML, IPython, LAPACK, Lcalc, Libgcrypt, Libgpg-error, Linbox, M4RI, Matplotlib, Maxima, Mercurial, MoinMoin Wiki, MPFI, MPFR, ECLib, NetworkX, NTL, Numpy, OpenCDK, PALP, PARI/GP, Pexpect, PNG, PolyBoRi, PyCrypto, Python, Qd, R, Readline, Rpy, Scipy, Singular, Scons, SQLite, Sympow, Symmetrica, Sympy, mpmath, Tachyon, Termcap, Twisted, Weave, Zlib, ZODB.

```
sage: P.<x> = PolynomialRing(GF(17))
sage: m = Matrix(P,2,2)
sage: m.randomize(); m

[  8*x^2 + 7*x + 4 2*x^2 + 16*x + 11]
[          7*x + 4  16*x^2 + 5*x + 9]
sage: m.change_ring(FractionField(P)).rank()
2
```

```
sage: z = gp(m)
sage: z
[Mod(8, 17)*x^2 + Mod(7, 17)*x + Mod(4, 17),
 Mod(2, 17)*x^2 + Mod(16, 17)*x + Mod(11, 17);
 Mod(7, 17)*x + Mod(4, 17),
 Mod(16, 17)*x^2 + Mod(5, 17)*x + Mod(9, 17)]
sage: type(z)
<class 'sage.interfaces.gp.GpElement'>
sage: z.matrank()
2
```

```
sage: gp.eval('intnum(x=1,[1],sin(x)/x^2)')
'0.50720744201747388836088625 13'
sage: res = eval(_); res
0.5072074420174739 2

sage: s = gp.eval('Mod(8,17)'); s
'Mod(8, 17)'
sage: a = eval(s); a
8
sage: a.parent()
Ring of integers modulo 17
```

Command-Line Interface

```
sage: %gp

  --> Switching to GP/PARI interpreter <--

''
gp: a = factorint(2^128+1)


[59649589127497217 1]

[5704689200685129054721 1]

gp: quit

  --> Exiting back to SAGE <--
```

```
sage: m = gp('a'); m
[5964958912749721 7, 57046892006851290547211]
sage: m * m
[3558073483084938890843851471799810,
    5964958912749721 8;
340282366920938469168063808116897266178,
    5704689200685129054722]
```

Using two different packages

```
sage: %maple
maple: m := linalg[hilbert](5);
m := matrix([[1, 1/2, 1/3, 1/4, 1/5], [1/2, 1/3, 1/4,
1/5, 1/6], [1/3, 1/4, 1/5, 1/6, 1/7], [1/4, 1/5, 1/6,
1/7, 1/8], [1/5, 1/6, 1/7, 1/8, 1/9]])
maple: d := linalg[charpoly](m, x);
d := x^5-563/315*x^4+735781/2116800*x^3-852401/
222264000*x^2+61501/53343360000*x-1/266716800000
maple: quit
sage: d = maple('d')
sage: d = gp(d)
sage: d.polroots()
[0.00000328792877217186295711500476 0 + 0.E-28*I,
0.00030589804015119172687949784 07 + 0.E-28*I,
0.01140749162341980655945145887 + 0.E-28*I,
0.20853421861101333590500251 01 + 0.E-28*I,
1.56705069109823079553301100 6 + 0.E-28*I]~
```

An Active Community

The Sage Web Page

`sagemath.org`

French mirror: `sagemath.fr`

The `trac` server

`http://trac.sagemath.org/sage_trac`

The Review Process

1. bug reported by say A
2. a patch is proposed by say B (might be A)

   `[with patch, needs review]`
3. a review is done by C (different from B)

   `[with patch, with negative review]`

   `[with patch, with positive review]`

Fix a Bug and Rebuild Sage

Submit a Patch

Cf `calculus.py`:
line 1372: algorithim → algorithm
Line 1389: suitible → suitable
Line 6551: tahn → tanh

The Sage Days

http://wiki.sagemath.org/SageDays

Bug and Doc Days

Coding Sprints

Now Try Yourself!