Sage : mathematics software based on Python

Sébastien Labbé ⟨slabqc@gmail.com⟩
Franco Saliola ⟨saliola@gmail.com⟩

Département de Mathématiques, UQÀM

29 novembre 2010

Outline

What is Sage?

Sage is . . .

a *distribution* of software

Sage is a *distribution* of software

When you install Sage, you get:

ATLAS	Automatically Tuned Linear Algebra Software
BLAS	Basic Fortan 77 linear algebra routines
Bzip2	High-quality data compressor
Cddlib	Double Description Method of Motzkin
Common Lisp	Multi-paradigm and general-purpose programming lang.
CVXOPT	Convex optimization, linear programming, least squares
Cython	C-Extensions for Python
F2c	Converts Fortran 77 to C code
Flint	Fast Library for Number Theory
FpLLL	Euclidian lattice reduction
FreeType	A Free, High-Quality, and Portable Font Engine

Sage is a *distribution* of software

When you install Sage, you get:

G95	Open source Fortran 95 compiler
GAP	Groups, Algorithms, Programming
GD	Dynamic graphics generation tool
Genus2reduction	Curve data computation
Gfan	Gröbner fans and tropical varieties
Givaro	C++ library for arithmetic and algebra
GMP	GNU Multiple Precision Arithmetic Library
GMP-ECM	Elliptic Curve Method for Integer Factorization
GNU TLS	Secure networking
GSL	Gnu Scientific Library
JsMath	JavaScript implementation of LaTeX

Sage is a *distribution* of software

When you install Sage, you get:

IML	Integer Matrix Library
IPython	Interactive Python shell
LAPACK	Fortan 77 linear algebra library
Lcalc	L-functions calculator
Libgcrypt	General purpose cryptographic library
Libgpg-error	Common error values for GnuPG components
Linbox	C++ linear algebra library
Matplotlib	Python plotting library
Maxima	computer algebra system
Mercurial	Revision control system
MoinMoin	Wiki

Sage is a *distribution* of software

When you install Sage, you get:

MPFI	Multiple Precision Floating-point Interval library
MPFR	C library for multiple-precision floating-point computations
ECLib	Cremona's Programs for Elliptic curves
NetworkX	Graph theory
NTL	Number theory C++ library
Numpy	Numerical linear algebra
OpenCDK	Open Crypto Development Kit
PALP	A Package for Analyzing Lattice Polytopes
PARI/GP	Number theory calculator
Pexpect	Pseudo-tty control for Python
PNG	Bitmap image support

Sage is a *distribution* of software

When you install Sage, you get:

PolyBoRi	Polynomials Over Boolean Rings
PyCrypto	Python Cryptography Toolkit
Python	Interpreted language
Qd	Quad-double/Double-double Computation Package
R	Statistical Computing
Readline	Line-editing
Rpy	Python interface to R
Scipy	Python library for scientific computation
Singular	fast commutative and noncommutative algebra
Scons	Software construction tool
SQLite	Relation database

Sage is a *distribution* of software

When you install Sage, you get:

Sympow	L-function calculator
Symmetrica	Representation theory
Sympy	Python library for symbolic computation
Tachyon	lightweight 3d ray tracer
Termcap	for writing portable text mode applications
Twisted	Python networking library
Weave	Tools for including C/C++ code within Python
Zlib	Data compression library
ZODB	Object-oriented database

Sage is a *distribution* of software

When you install Sage, you get:

Sympow	L-function calculator
Symmetrica	Representation theory
Sympy	Python library for symbolic computation
Tachyon	lightweight 3d ray tracer
Termcap	for writing portable text mode applications
Twisted	Python networking library
Weave	Tools for including C/C++ code within Python
Zlib	Data compression library
ZODB	Object-oriented database

. . . and more!

```
> sage -gap

          #########          ######        ###########          ###
       #############         ######        ############        ####
      ##############        ########       #############      #####
     ###############        ########       #####   ######     #####
    ######        #         #########      #####    #####    ######
   ######                  ##########      #####    #####   #######
   #####                   ##### ####      #####   ######  ########
   ####                   #####  #####     #############  ###  ####
   #####    #######       ####    ####     ###########   ####  ####
   #####    #######      #####    #####    ######       ####   ####
   #####    #######      #####    #####    #####        #############
    #####      #####    ################   #####        #############
    ######     #####    ################   #####        #############
    ################   ##################  #####                ####
     ###############   #####        #####  #####                ####
       #############   #####        #####  #####                ####
        #########     #####          #####  #####                ####

   Information at:  http://www.gap-system.org
   Try '?help' for help. See also  '?copyright' and  '?authors'

  Loading the library. Please be patient, this may take a while.
GAP4, Version: 4.4.10 of 02-Oct-2007, x86_64-unknown-linux-gnu-gcc

gap>
```

```
> sage -singular

                    SINGULAR                                 /  Development
 A Computer Algebra System for Polynomial Computations   /   version 3-1-0
                                                       0<
     by: G.-M. Greuel, G. Pfister, H. Schoenemann        \   Mar 2009
FB Mathematik der Universitaet, D-67653 Kaiserslautern    \
>
```

```
> sage -maxima

Maxima 5.16.3 http://maxima.sourceforge.net
Using Lisp ECL 9.4.1
Distributed under the GNU Public License. See the file COPYING.
Dedicated to the memory of William Schelter.
The function bug_report() provides bug reporting information.
(%i1)
```

```
> sage -gp

                GP/PARI CALCULATOR Version 2.3.3 (released)
         amd64 running linux (x86-64/GMP-4.2.1 kernel) 64-bit version
          compiled: Jul 10 2009, gcc-4.3.2 (Ubuntu 4.3.2-1ubuntu12)
               (readline v5.2 enabled, extended help available)

                    Copyright (C) 2000-2006 The PARI Group

PARI/GP is free software, covered by the GNU General Public License, and
comes WITHOUT ANY WARRANTY WHATSOEVER.

Type ? for help, \q to quit.
Type ?12 for how to get moral (and possibly technical) support.

parisize = 8000000, primelimit = 500000
?
```

```
> sage -R

R version 2.6.1 (2007-11-26)
Copyright (C) 2007 The R Foundation for Statistical Computing
ISBN 3-900051-07-0

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

>
```

Sage is . . .

a distribution of software

for *mathematics research*

Sage is software for *mathematics research*

Algebra	GAP, Maxima, Singular, ...
Exact linear algebra	Linbox, IML, ...
Numerical linear algebra	GSL, Scipy, Numpy, ...
Arbitrary precision arithmetic	GMP, MPFR, MPFI, NTL, ...
Calculus	Maxima, Sympy, ...
Combinatorics	Symmetrica, *-combinat, ...
Algebraic geometry	Singular, ...
Arithmetic geometry	PARI, NTL, mwrank, ecm, ...
Graph theory	NetworkX, ...
Group theory	GAP, ...
⋮	⋮

Sage uses *Python*
as its programming language.

Sage uses *Python*

- Sage $\approx$ Python + a huge Python library
- Sage may be the first successful math software system to not invent its own new language just for mathematics.
- Tens of thousands of third party Python packages are immediately available for use with Sage!

Sage *combines* the power of many existing software.

Sage *combines* software

[This example is from a talk by William Stein]

Sage *combines* software

[This example is from a talk by William Stein]

Construct an elliptic curve using *John Cremona's table*:

```
sage: E = EllipticCurve('389a')
```

Sage *combines* software

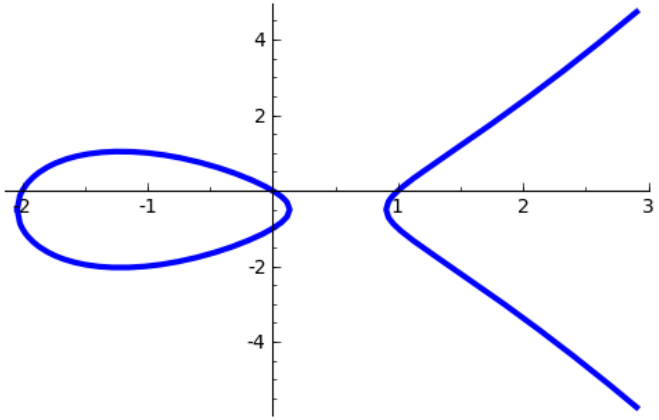
[This example is from a talk by William Stein]

Construct an elliptic curve using *John Cremona's table*:

```
sage: E = EllipticCurve('389a')
```

Use *matplotlib* to plot it:

```
sage: plot(E,thickness=3)
```



Sage *combines* software

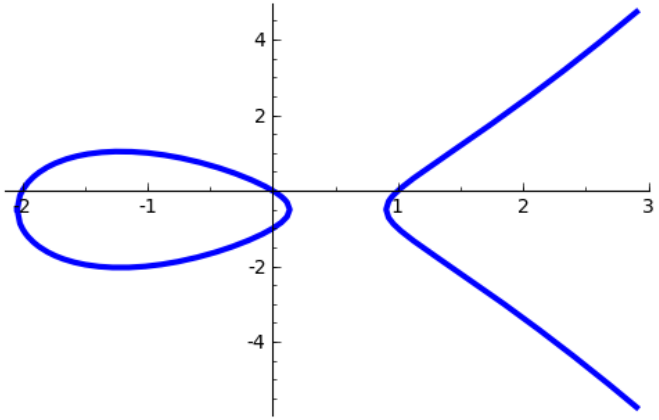
[This example is from a talk by William Stein]

Construct an elliptic curve using *John Cremona's table*:

```
sage: E = EllipticCurve('389a')
```

Use *matplotlib* to plot it:

```
sage: plot(E,thickness=3)
```



mwrank to do a 2-descent:

```
sage: E.mwrank()
Curve [0,1,1,-2,0] :     Rank = 2
```

Sage *combines* software

PARI to compute Fourier coefficients a_n:

```
sage: E.anlist(15)
[0, 1, -2, -2, 2, -3, 4, -5, 0, 1, 6, -4, -4, -3, 10, 6
```

Sage *combines* software

PARI to compute Fourier coefficients a_n:

```
sage: E.anlist(15)
[0, 1, -2, -2, 2, -3, 4, -5, 0, 1, 6, -4, -4, -3, 10, 6
```

lcalc to compute zeros in the critical strip of the L-series:

```
sage: E.lseries().zeros(5)
[0.000000000, 0.000000000, 2.87609907, 4.41689608, 5.79
```

Sage *combines* software

PARI to compute Fourier coefficients a_n:

```
sage: E.anlist(15)
[0, 1, -2, -2, 2, -3, 4, -5, 0, 1, 6, -4, -4, -3, 10, 6
```

lcalc to compute zeros in the critical strip of the L-series:

```
sage: E.lseries().zeros(5)
[0.000000000, 0.000000000, 2.87609907, 4.41689608, 5.79
```

sympow to compute the modular degree:

```
sage: E.modular_degree()
40
```

Sage *combines* software

PARI to compute Fourier coefficients a_n:

```
sage: E.anlist(15)
[0, 1, -2, -2, 2, -3, 4, -5, 0, 1, 6, -4, -4, -3, 10, 6
```

lcalc to compute zeros in the critical strip of the L-series:

```
sage: E.lseries().zeros(5)
[0.000000000, 0.000000000, 2.87609907, 4.41689608, 5.79
```

sympow to compute the modular degree:

```
sage: E.modular_degree()
40
```

Magma to compute the rank of the 3-selmer group:

```
sage: magma(E).ThreeSelmerGroup()
```

Sage *combines* software

"We implement all conversion routines, instead of expecting upstream to do it: we make them communicate with Sage, whether they want to or not. Resistence is futile."

—William Stein

Sage is also *new code*: provides new or improved functionality not previously available.

Sage is *open-source software*

Sage is *open-source software*

"You can read Sylow's Theorem and its proof in Huppert's book in the library . . . then you can use Sylow's Theorem for the rest of your life free of charge, but for many computer algebra systems license fees have to be paid regularly . . . .

With this situation two of the most basic rules of conduct in mathematics are violated: In mathematics information is passed on free of charge and everything is laid open for checking."

—J. Neubüser (1993)

(started GAP in 1986)

Sage is *open-source software*

"I think, fundamentally, open source does tend to be more stable software. It's the right way to do things. I compare it to science versus witchcraft.

In science, the whole system builds on people looking at other people's results and building on top of them.

In witchcraft, somebody had a small secret and guarded it—but never allowed others to really understand it and build on it."

—Linus Torvalds

Mission

The Sage Project aims to create a viable high-quality and open-source alternative to Magma, Maple, Mathematica, Matlab and MuPAD, and to foster a friendly community of users and developers.

High-quality code and documentation

All new code is:

- rigorously tested
- well documented
- peer-reviewed

New releases every 3–4 weeks

Some history of the Sage project

- *1999-2005.* William Stein wrote over 25,000 lines of Magma code for his research. Decided that Magma was a bad long term investment since he couldn't see or modify the internals.
- *Jan. 2005.* William Stein started Sage.
- *Feb. 2005.* Sage version 0.1: a Python library gluing together PARI, Maxima, Python, Singular e GAP.
- *Feb. 2006.* Sage version 1.0 released; and the "first annual" Sage Days workshop.

Some history of the Sage project

- *Nov. 2007.* Sage won first place in Les Trophées du Libre competition (honours the best existing free software)
- *Dec. 2007.* Sage gets slashdotted:

IT: Open Source 'Sage' Takes Aim at High End Math Software

Posted by CmdrTaco on Saturday December 08 2007, @12:19PM from the that'll-take-awhile dept.

coondoggie writes

> "A new open source mathematics program is looking to push aside commercial software commonly used in mathematics education, in large government laboratories and in math-intensive research. The program's backers say the software, called Sage, can do anything from mapping a 12-dimensional object to calculating rainfall patterns under global warming."

software it sage octave math story

Sage Days!

- Intensive 5-day workshops to develop and implement new features and to attract new users and developers.
- Sufficiently novel algorithms are submitted for publication in academic journals.

There have been over 30 Sage Days!

Sage Days 2010–2011

- Sage Days 19: Seattle, USA (January 2010)
- Sage Days 20: Marseille, France (February 2010)
- Sage Days 20.25: Montreal, Canada (March 2010)
- Sage Days 20.5: Toronto, Canada (May 2010)
- Sage Days 21: Seattle, USA (June 2010)
- Sage-Combinat/Chevie: France (June 2010)
- Sage Days 22: Berkeley, USA (July 2010)
- Sage Days 23: Leiden, Netherlands (July 2010)
- Sage Days 23.5: Kaiserslautern, Germany (July 2010)
- Sage Days 24: Linz, Austria (July 2010)
- Sage Days 25: Mumbai, India (August 2010)
- Sage Days 25.5: Montreal, Canada (September 2010)
- Sage Days 26: Seattle, USA (December 2010)
- Sage Days 27: Seattle, USA (January 2011)
- Sage Days 28: Orsay, France (January 2011)

Sage Community

mailing lists

- `sage-devel` for development
- `sage-combinat-devel` for combinatorics
- `sage-windows` for Microsoft Windows Port
- `sage-nt` for number theory
- `sage-finance` for finance
- `sage-flame` for flame wars
- `sage-release` for releases
- `sage-edu` for education
- `sage-grid` for scientific grid computing

irc-channel

- `#sage-devel` on `freenode.net`

Map of contributors to the Sage project

There are currently **219** contributors
in **147** different places from all around the world.

Some useful features

Sage as a Python library

script.py:

```
from sage.all import *

x = var('x')
f = x**2 + 3*x + 1
print diff(f, x)
```

execution and output:

```
> sage -python script.py
2*x + 3
```

LaTeX

In this LaTeX file, I typed:

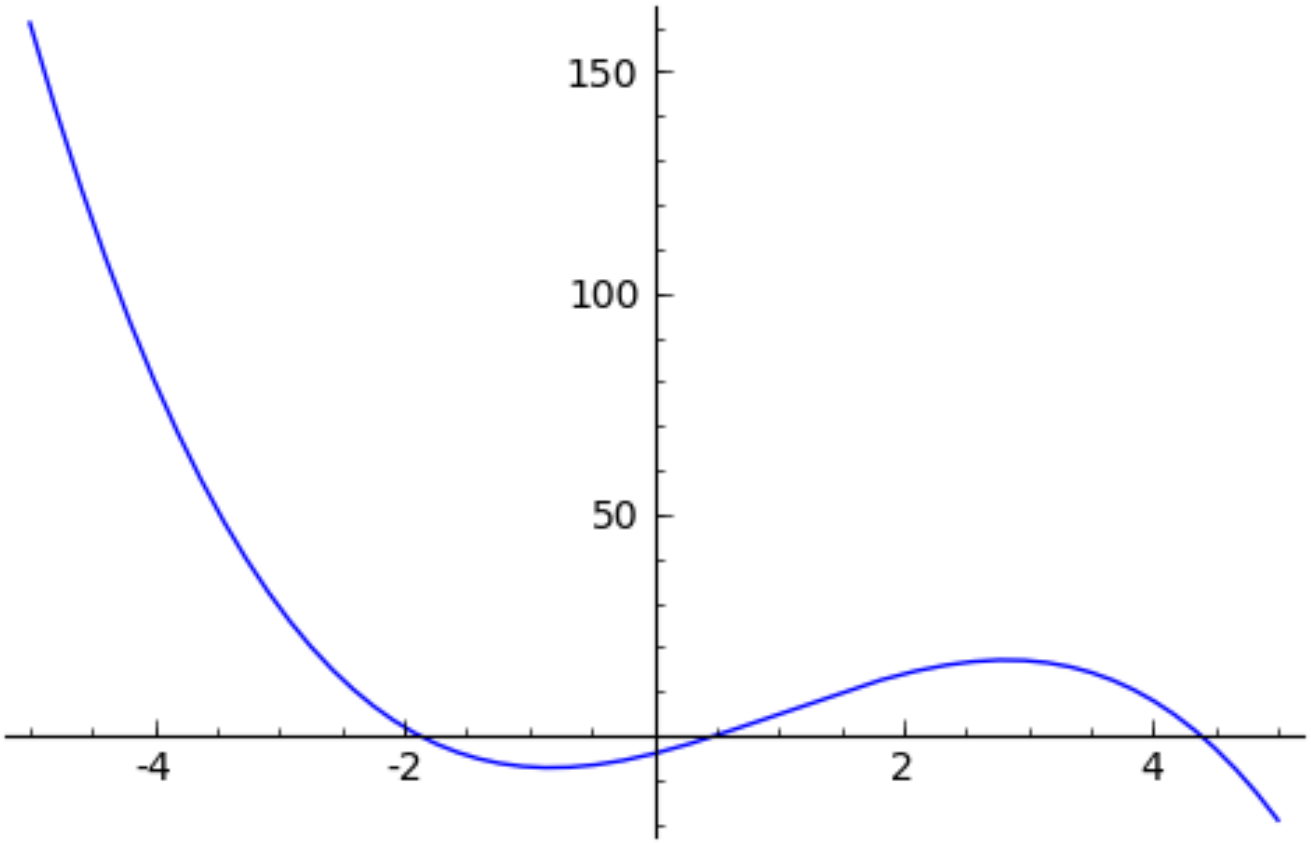
```
\sageplot{plot(-x^3+3*x^2+7*x-4,-5,5)}
```

LaTeX

In this LaTeX file, I typed:

```
\sageplot{plot(-x^3+3*x^2+7*x-4,-5,5)}
```

and it got replaced by:



LaTeX

In this LaTeX file:

```
\begin{sagesilent}
  sigma = Permutation([7,3,1,5,2,6,8,4])
  P, Q = sigma.robinson_schensted()
\end{sagesilent}

Let $\sigma = \sage{sigma}$. The
Robinson-Schensted-Knuth algorithm produces
the tableaux:
    \[\sage{P} \quad \sage{Q}\]
```

LaTeX

It got replaced with:

Let $\sigma =$ `[7, 3, 1, 5, 2, 6, 8, 4]`. The Robinson-Schensted-Knuth algorithm produces the tableaux:

$$\begin{array}{|c|c|c|c|}\hline 1 & 2 & 4 & 8 \\ \hline 3 & 5 & 6 \\ \cline{1-3} 7 \\ \cline{1-1}\end{array} \qquad \begin{array}{|c|c|c|c|}\hline 1 & 4 & 6 & 7 \\ \hline 2 & 5 & 8 \\ \cline{1-3} 3 \\ \cline{1-1}\end{array}$$

LaTeX

It got replaced with:

Let $\sigma =$ `[7, 3, 1, 5, 2, 6, 8, 4]`. The Robinson-Schensted-Knuth algorithm produces the tableaux:

$$\begin{array}{|c|c|c|c|}\hline 1 & 2 & 4 & 8 \\ \hline 3 & 5 & 6 \\ \cline{1-3} 7 \\ \cline{1-1}\end{array} \qquad \begin{array}{|c|c|c|c|}\hline 1 & 4 & 6 & 7 \\ \hline 2 & 5 & 8 \\ \cline{1-3} 3 \\ \cline{1-1}\end{array}$$

This is done with the *sagetex* package for LaTeX, written by Dan Drake. Of course, the package is included with Sage.

Command line interface

saliola@karkwa: ~

File Edit View Terminal Tabs Help

```
saliola@karkwa:~$ sage
----------------------------------------------------------------------
| SAGE Version 3.1.2, Release Date: 2008-09-19                       |
| Type notebook() for the GUI, and license() for information.        |
----------------------------------------------------------------------

sage: 3 * 17
51
sage: [ 17 * x for x in range(10) if x % 2 == 1 ]
[17, 51, 85, 119, 153]
sage: Partitions(4).list()
[[4], [3, 1], [2, 2], [2, 1, 1], [1, 1, 1, 1]]
sage:
```

Notebook interface

The Sage Notebook (Sage)

localhost:8000/home/admin/3/

The Sage Notebook
Version 4.5.3

admin Toggle | Home | Published | Log | Settings | Help | Report a Problem | Sign out

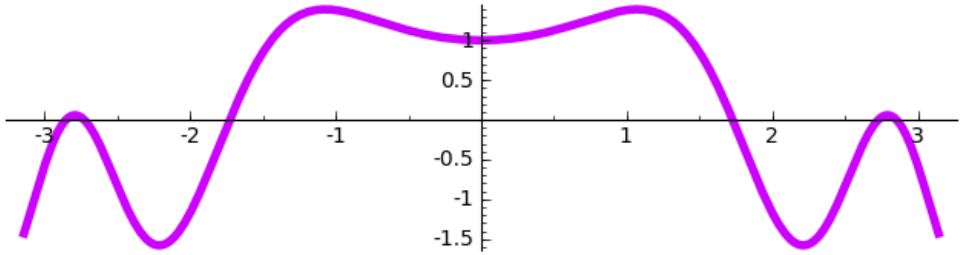
The Sage Notebook

last edited on November 27, 2010 01:03 PM by admin

Save | Save & quit | Discard & quit

File... | Action | Data... | sage | Typeset | Print | Worksheet | Edit | Text | Undo | Share | Publish

```
plot(sin(x^2)+cos(x), -pi, pi, hue=0.8, thickness=4).show(figsize=[8,2])
```



evaluate

plot

EXAMPLES: We plot the sin function:

```
sage: P = plot(sin, (0,10)); print P
Graphics object consisting of 1 graphics primitive
sage: len(P)      # number of graphics primitives
1
sage: len(P[0])   # how many points were computed (random)
225
sage: P           # render
```

```
sage: P = plot(sin, (0,10), plot_points=10); print P
Graphics object consisting of 1 graphics primitive
sage: len(P[0])   # random output
32
sage: P           # render
```

We plot with `randomize=False`, which makes the initial sample points evenly spaced (hence always the same). Adaptive plotting might insert other points, however, unless

Cython

- Cython is a programming language based on Python
- easily write C extensions for Python:

  translates sourcecode to optimized C code,
  which compiles as Python extension modules

- Cython allows you to:
  - declare C types for variables and class functions
  - use external C/C++ libraries from within Python
  - write extremely fast code!
- two major use cases:
  - extend CPython with fast compiled modules
  - interfacing Python code with external C/C++ libraries