Introduction to Sage

Sébastien Labbé ⟨slabqc@gmail.com⟩

Sage Days 38

Centre de recherches mathématiques de Montréal
May 7th 2012

"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)

(il a fondé GAP en 1986)

"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

Why You Do Not Usually Need to Know about Internals

« You should realize at the outset that while knowing about the internals of Mathematica may be of intellectual interest, it is usually much less important in practice than you might at first suppose. [...] For the internals of Mathematica are quite complicated, [...], it is usually extremely difficult to reach a reliable conclusion about how the detailed implementation of this algorithm will actually behave in particular circumstances. »

—Mathematica Tutorial

`http://reference.wolfram.com/mathematica/tutorial/WhyYouDoNotUsuallyNeedToKnowAboutInternals.html`

Outline

Plan

Beginning of Sage

Existing commercial softwares :

- Maple, Waterloo Maple Inc., Maplesoft, since 1985.
- Matematica, Wolfram Research, since 1988.
- Magma, University of Sydney, since 1990
- Matlab, MathWorks, since 1989

Free existing softwares :

- Maxima, William Schelter et coll., since 1967
- GAP, GAP Group, since 1986
- Singular, University of Kaiserslautern, since 1984
- PARI/GP, since 1985

Beginning of Sage

- *1999-2005.* William Stein write over 25000 lines of Magma code for its own research, but had some serious concerns :
  - Magma did not allow users to define their own types (or classes)
  - No exception handling
  - No "eval statements“
  - No way for users to write compiled code
  - Magma was closed source
  - Magma is not free

« I sat down on that park bench, and realized what a dangerous path I was taking in giving up so much freedom so early in my career. I resolved at that moment not to do it. At that moment I started designing what would eventually become Sage. »

— William Stein

Mathematical Software and Me : A Very Personal Recollection, 2009

Beginning of Sage

- *Jan. 2005.* William Stein creates Sage.
- *Feb. 2005.* Sage version 0.1 : gathers 5 software : PARI, Maxima, Python, Singular et GAP.
- David Kohel in Australia, David Joyner, in Maryland, joined the project.
- *Dec. 2005.* William receives an email from Magma leader.

Beginning of Sage

Date: Mon, 19 Dec 2005 16:54:09 -0800
From: "John Cannon" <john@maths.usyd.edu.au>
Subject: Magma calculator
William,

This is to formally advise you that your permission to run a general-purpose calculator based on Magma ends on Dec 31, 2005. This was originally set up at your request so students in your courses at Harvard could have easy access to Magma.

Please confirm receipt of this letter.
Wishing you a happy Christmas,
John

Beginning of Sage

« This single email seriously scared me. Though I was working on Sage very hard for nearly a year at this point, I honestly didn't then expect Sage to really be able to replace Magma for me. [...] What John made crystal clear to me was that he could destroy my entire longterm plans on a whim. I looked around for other options, and there just weren't any. Sage had to succeed. »

— William Stein

Mathematical Software and Me : A Very Personal Recollection, 2009

Beginning of Sage

- *Feb. 2006.* Sage version 1.0 is released ; first "annual" Sage Days are organised : Sage Days 1.
- *Dec. 2007.* Sage gets slashdoted :

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

Beginning of Sage

Year	Releases	Functions	Contributors
2007	≈ 60	17947	
2008	23	22034	≈ 100
2009	14	23763	
2010	16	26592	219
2011	5	28462	235
2012	1	28573	245

Plan

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.

Sage is an *open source* software

Sage is distributed under the terms of the GNU General Public License version 2 (GPLv2) which guarentees four types of freedom :

- The freedom to use the software (it is free).
- The freedom to read the source code.
- The freedom to improve the software.
- The freedom to redistribute the modified software to anyone.

Sage is an *open source* software

Sage is distributed under the terms of the GNU General Public License version 2 (GPLv2) which guarentees four types of freedom :

- The freedom to use the software (it is free).
- The freedom to read the source code.
- The freedom to improve the software.
- The freedom to redistribute the modified software to anyone.

Sage is an *open source* software

Sage is distributed under the terms of the GNU General Public License version 2 (GPLv2) which guarentees four types of freedom :

- The freedom to use the software (it is free).
- The freedom to read the source code.
- The freedom to improve the software.
- The freedom to redistribute the modified software to anyone.


Sage is an *open source* software

Sage is distributed under the terms of the GNU General Public License version 2 (GPLv2) which guarentees four types of freedom :

- The freedom to use the software (it is free).
- The freedom to read the source code.
- The freedom to improve the software.
- The freedom to redistribute the modified software to anyone.



Sage is an *open source* software

Sage is distributed under the terms of the GNU General Public License version 2 (GPLv2) which guarentees four types of freedom :

- The freedom to use the software (it is free).
- The freedom to read the source code.
- The freedom to improve the software.
- The freedom to redistribute the modified software to anyone.




Sage is . . .

a *distribution* of softwares

Sage is a *distribution* of open source softwares

These softwares are included in Sage :

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 open source softwares

These softwares are included in Sage :

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 open source softwares

These softwares are included in Sage :

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 open source softwares

These softwares are included in Sage :

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 open source softwares

These softwares are included in Sage :

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 open source softwares

These softwares are included in Sage :

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

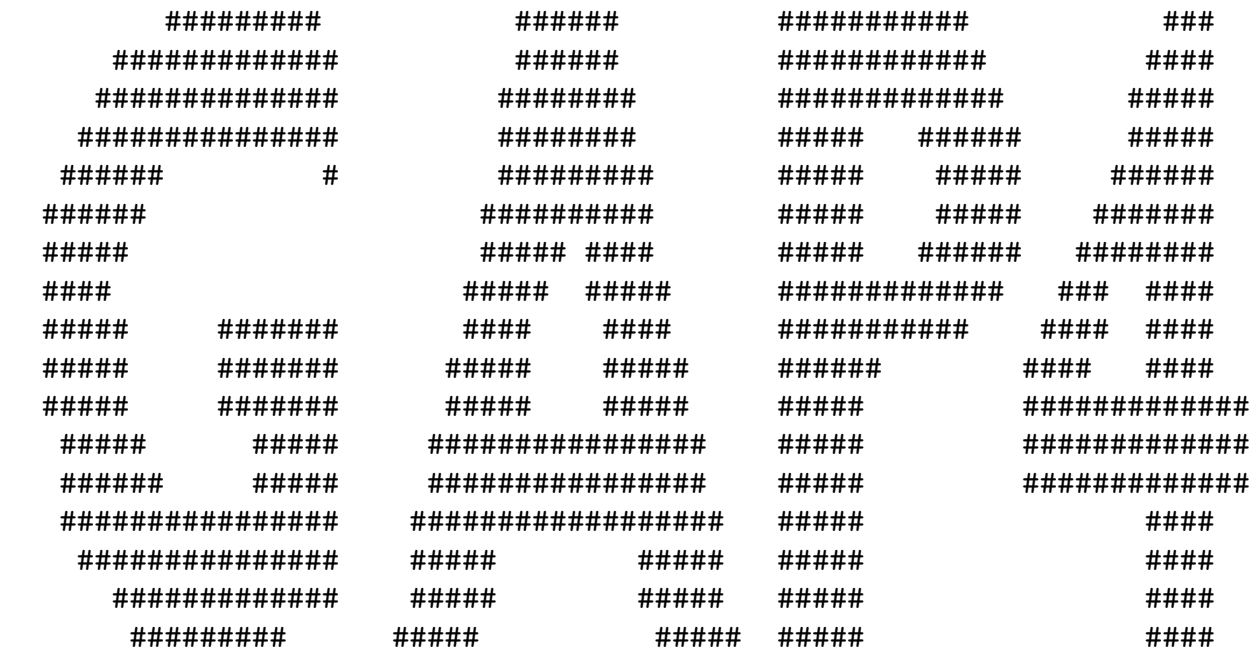
Sage is a *distribution* of open source softwares

These softwares are included in Sage :

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 *combines* software.

Sage *combines* software

[This examples are from a presention of William Stein]

Sage *combines* software

[This examples are from a presention of William Stein]

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

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

Sage *combines* software

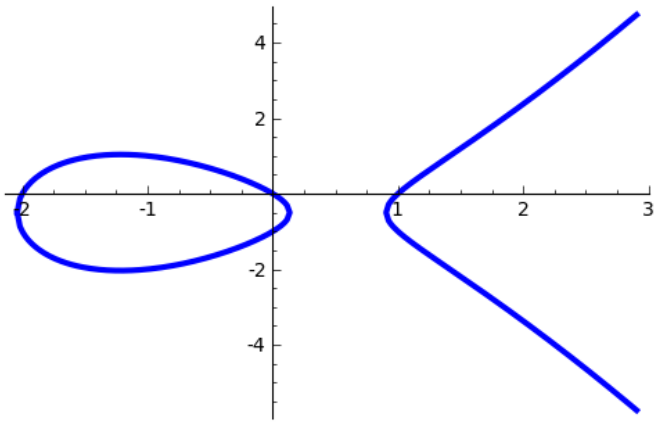
[This examples are from a presention of William Stein]

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

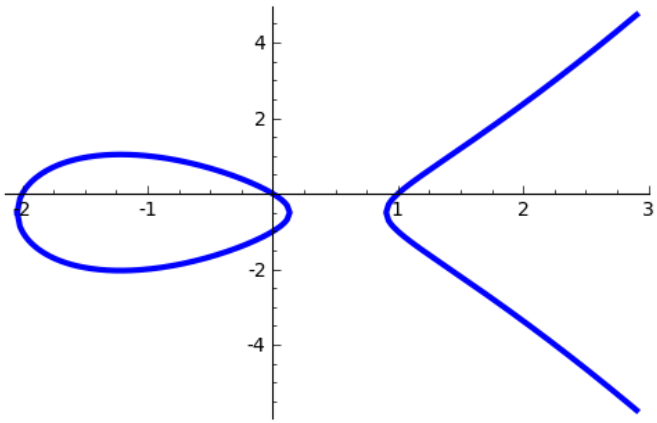
[This examples are from a presention of William Stein]

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()
Abelian Group isomorphic to Z/3 + Z/3
Defined on 2 generators
```

Sage *combines* software

Let's integrate $\int \cos(x^2)dx$:

```
sage: integrate(cos(x^2), x)
-1/8*((I + 1)*sqrt(2)*erf((1/2*I - 1/2)*sqrt(2)*x) +
(I - 1)*sqrt(2)*erf((1/2*I + 1/2)*sqrt(2)*x))*sqrt(pi)
```

Software used for this computation :

```
sage: from sage.misc.citation import get_systems
sage: get_systems("integrate(cos(x^2), x)")
['MPFI', 'ginac', 'GMP', 'Maxima']
```

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

(started Sage)

Sage Important Philosophy : Elements have parent

```
sage: m = matrix(ZZ, 3, [2,3,4,2,4,6,8,8,5])
sage: m.parent()
Full MatrixSpace of 3 by 3 dense matrices over Integer Ring
sage: m.echelon_form()
[2 0 1]
[0 1 2]
[0 0 3]

sage: m = matrix(QQ, 3, [2,3,4,2,4,6,8,8,5])
sage: m.parent()
Full MatrixSpace of 3 by 3 dense matrices over Rational Fiel
sage: m.echelon_form()
[1 0 0]
[0 1 0]
[0 0 1]
```

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 !
- Easy to write and read :

math : $\left\{ 17x \middle| x \in \{0, 1, ..., 9\} \text{ et } x \text{ est impair} \right\}$

python :
```
[17*x for x in range(10) if x % 2 == 1]
```

Sage uses *Python*

« Google has made no secret of the fact they use Python a lot for a number of internal projects. Even knowing that, once I was an employee, I was amazed at how much Python code there actually is in the Google source code system. »

— Guido van Rossum

(Python creator)

Plan

Developer Map

There are currently $244 + 1$ contributors
in $165 + 1$ different places from all around the world.

Working together with internet

Development :

- `http://groups.google.com/group/sage-devel`
- `http://trac.sagemath.org/sage_trac/`
- `http://wiki.sagemath.org/`

Support :

- `http://groups.google.com/group/sage-support`
- `http://ask.sagemath.org/questions/`
- `http://wiki.sagemath.org/`

Other Google Group include : `sage-combinat-devel`, `sage-release` and `sage-edu`.

irc-channel

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

Sage Days

Sage Days are workshops aiming at

- fix bugs and develop new functionalities
- attract new users and developers

Dozen of workshops are organized every year all around the world.

Sage Days in 2010

- 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 in 2011

- Sage Days 27 : Seattle, USA (January 2011)
- Sage Days 28 : Orsay, France (January 2011)
- Sage Days 29 : Seattle, USA (March 21-25, 2011)
- Sage Days 30 : Halifax, Canada (May 2-6, 2011)
- Sage Days 31 : Seattle, USA (June 11-18, 2011)
- Sage Education Days 3 : Seattle, USA (June 16-18, 2011)
- Sage Days 32 : Seattle, USA (August 21-25, 2011
- Sage Days 33 : Women in Sage, Seattle (September 2011)
- Sage Days 34 : Kaiserslautern, Germany (Sept. 2011)
- Sage Days 34.5 : Bobo Dioulasso, Burkina Faso (Nov. 2011)
- Sage Days 35 : Warwick UK (December 2011)

Sage Days in 2012

- Sage Days 35.5 : Wenham, MA (January 2012)
- Sage-Combinat Days 35.75 : France (February 2012)
- Sage Days 36 : UC San Diego, (February 2012)
- Review Days 2 : March 17-22, Seattle WA.
- Sage Days 36.5 : April 17-22, Seattle WA.
- Sage Days 37 : 28 April 2012, Seoul, South Korea.
- Sage Days 38 : May 7-11 2012, Montréal, Canada.
- Bug Days 19 : May 24-29, Seattle WA.
- Sage Days 39 : May 26-27 2012, Fukuoka, Japan.
- Sage Days 41 : June 11-15, 2012, Seattle.
- Sage Edu Days 4 : June 13-15, 2012, Seattle.
- SAGE-Combinat Days 40 : July 9-13, Minneapolis, MN.
- Sage Days 42 : July 15-19, 2012, Seattle, WA.

Plan

Some functionalities

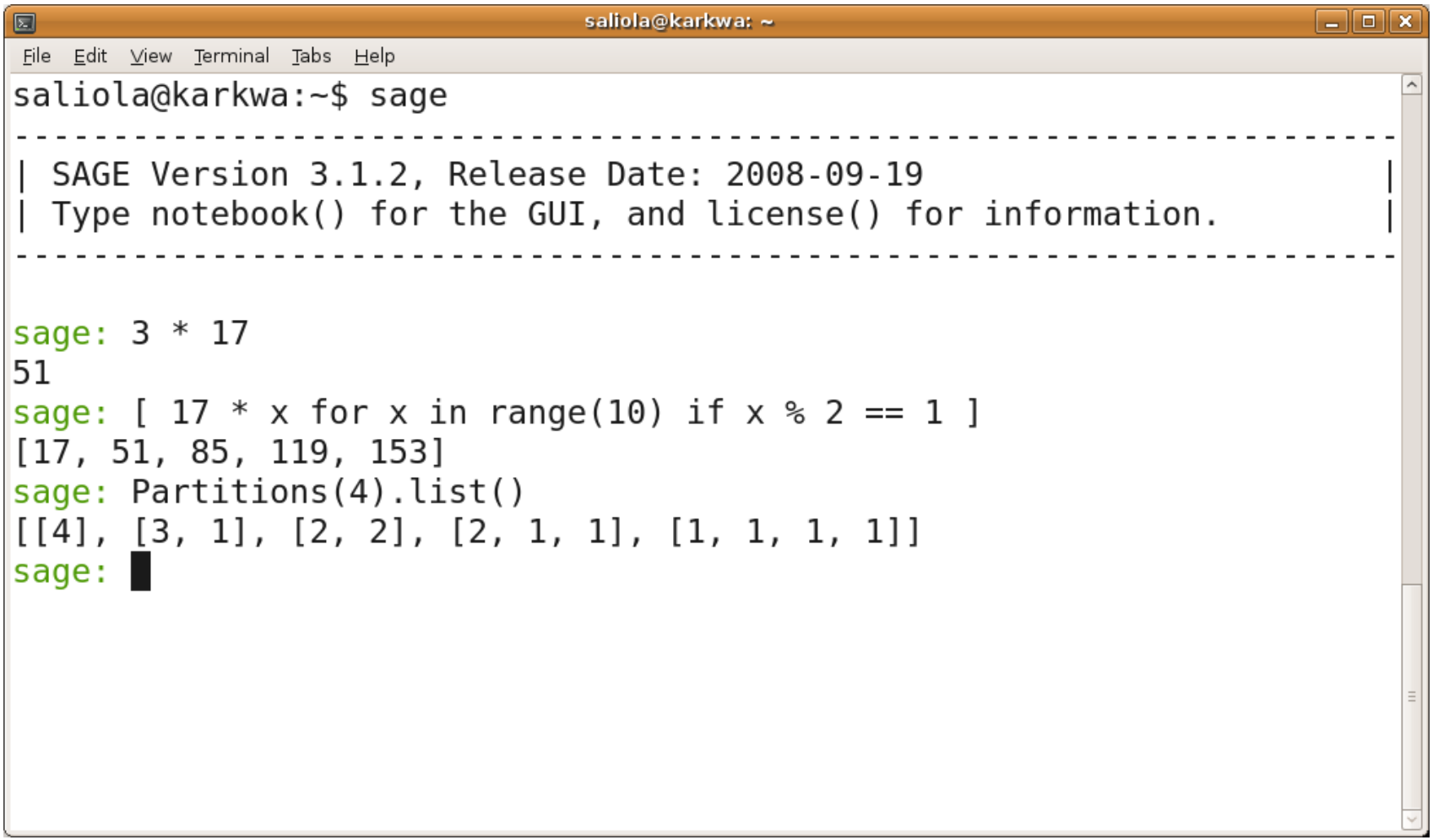
Interfaces :

- Command line
- Sage Notebook (local)
- Sage Notebook (internet)
- New : Sage in a web page (by Jason Grout, USA)

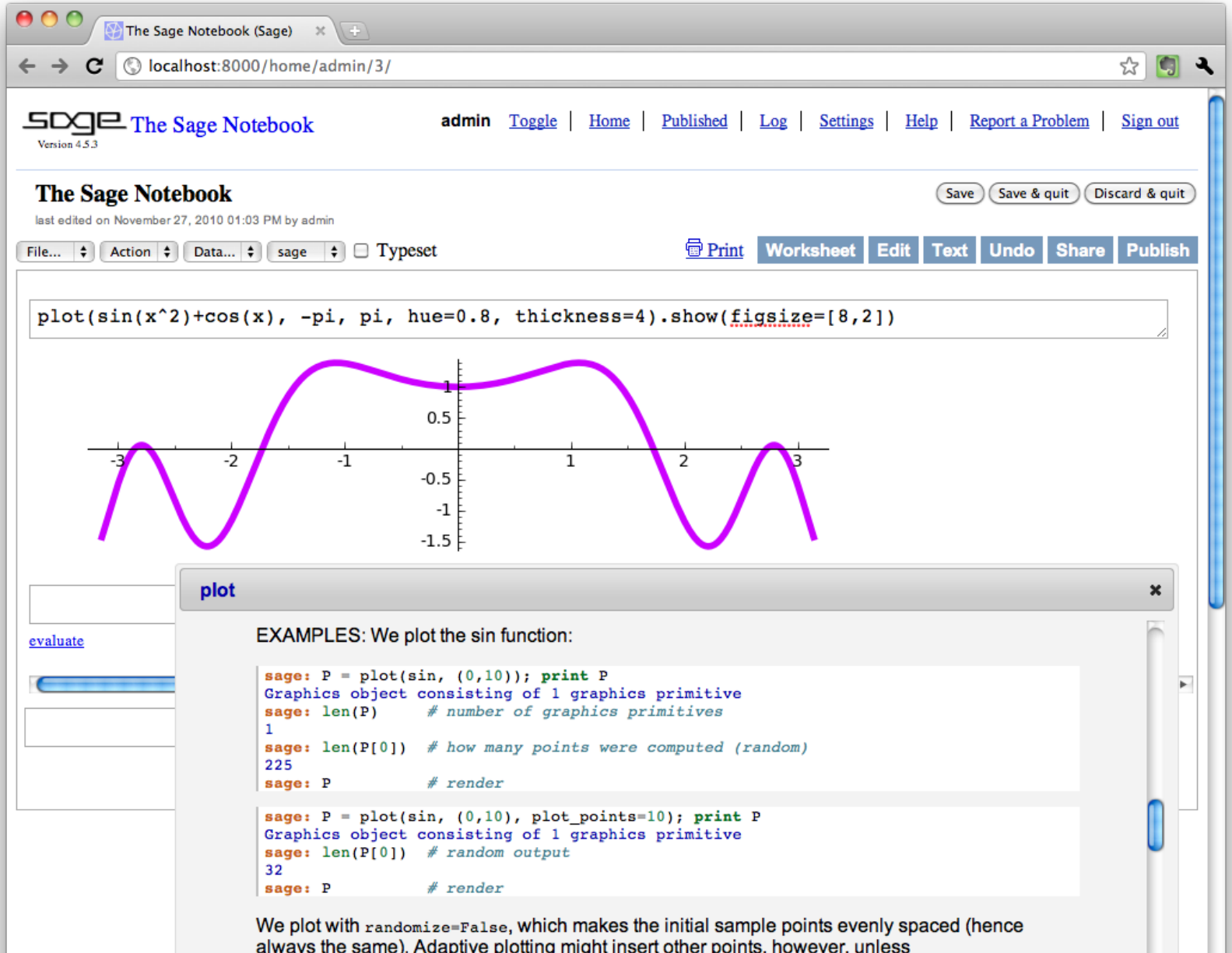
Interesting features :

- Sagetex (Dan Drake, Corée du Sud)
- Graph Editor (Radoslav Kirov, États-Unis)
- Interact and animations
- Cython : translates Python code $\mapsto$ C code.

Command line interface

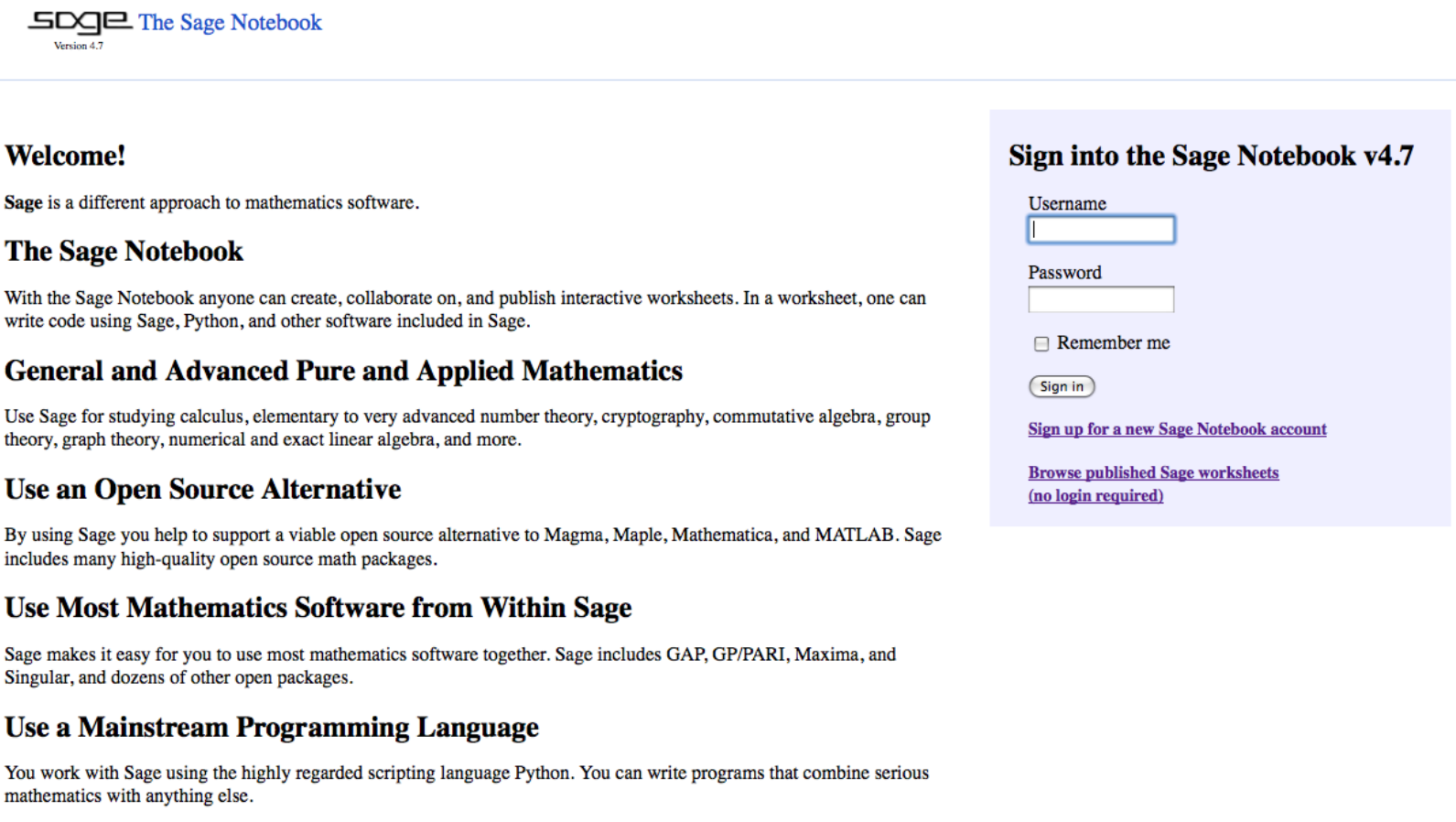


Notebook interface (local)



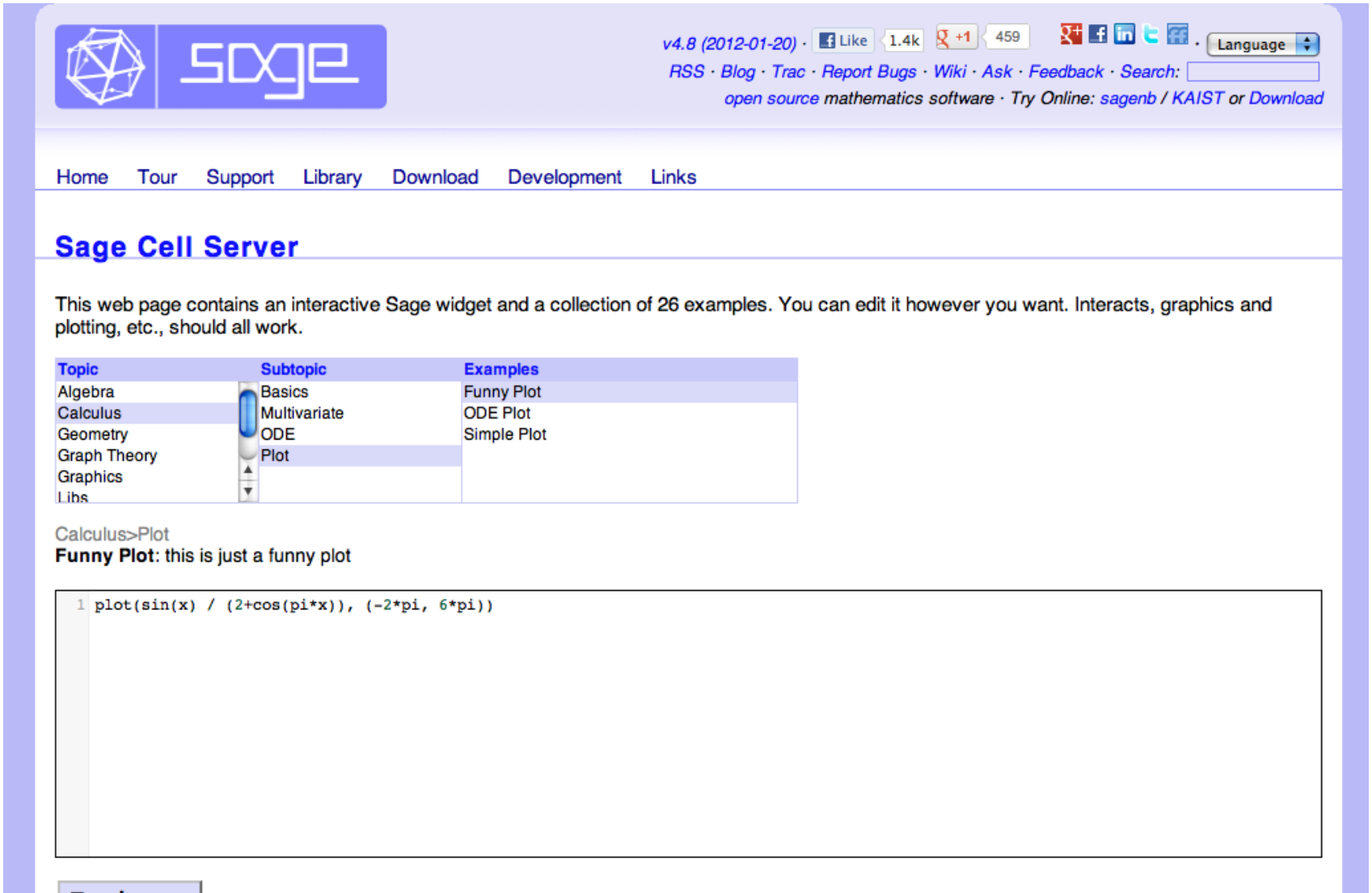
Notebook interface (internet)

http://sagenb.org/



Single Cell Server

http://www.sagemath.org/eval.html



LaTeX

In this LaTeXfile, I write

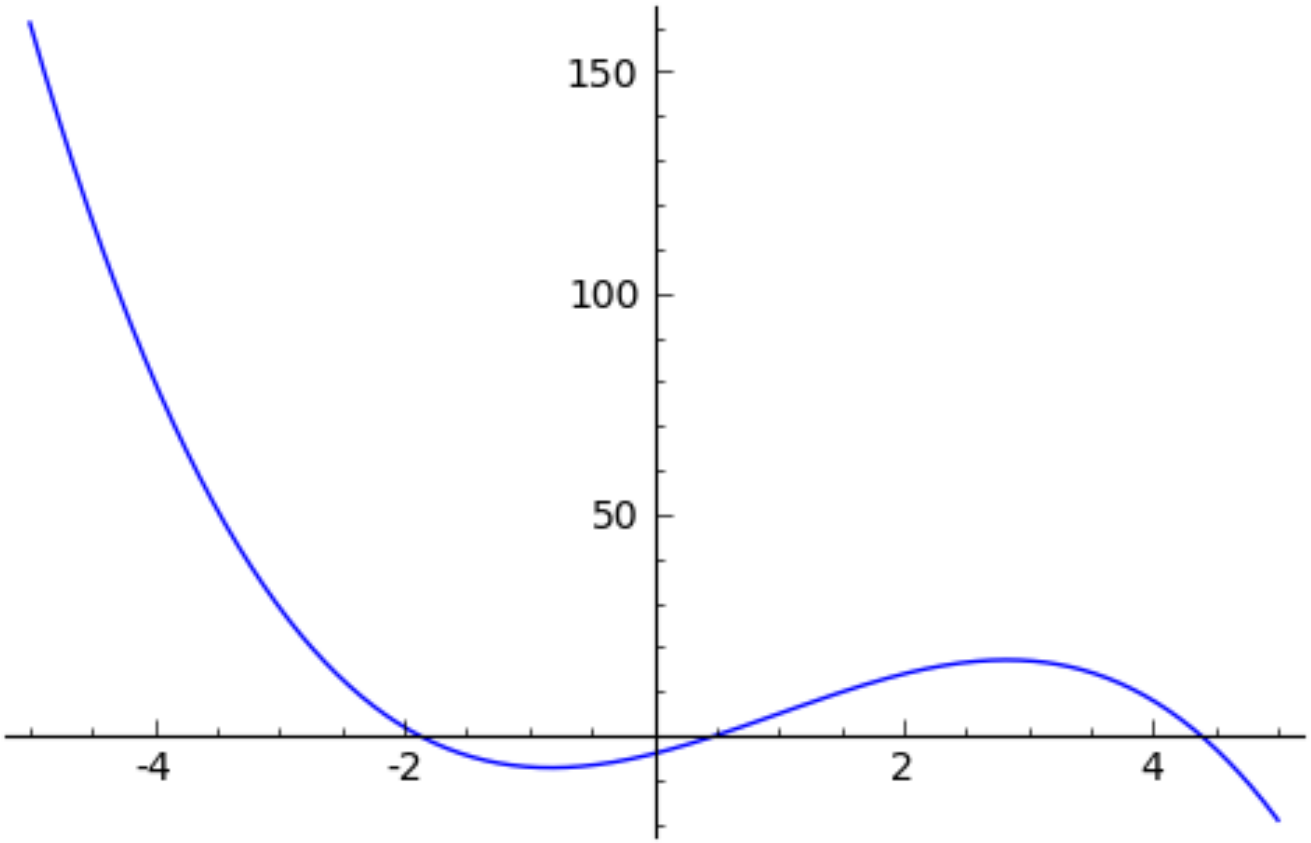
```
\sageplot{plot(-x^3+3*x^2+7*x-4,-5,5)}
```

LaTeX

In this LaTeXfile, I write

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

and this is replaced by :



LaTeX

In this LaTeXfile :

```
\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 :

1	2	4	8
3	5	6	
7			

1	4	6	7
2	5	8	
3			

LaTeX

It got replaced with :

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

1	2	4	8
3	5	6	
7			

1	4	6	7
2	5	8	
3			

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

For more informations

`http://sagemath.org`

For more information

- `http://sagemath.org`
- `http://sagemath.org/doc`
- `http://sagemath.org/help.html`
- `http://wiki.sagemath.org`

Ressources in French :

- `http://sagemath.org/fr`
- `http://sagemath.org/fr/html/tutorial`

Le livre Calcul mathématique avec Sage :

- `http://sagebook.gforge.inria.fr/`

Let's start !

Commençons !