

Pourquoi Sage est-elle si cool?

Franco V. Saliola
<saliola@gmail.com>

Institut Gaspard Monge
Université de Marne-la-Vallée

18 November 2008

Sage est un *distribution* de logiciel à code source ouvert.

Sage est un *distribution* de logiciel à code source ouvert.

Logiciels inclus dans Sage :

ATLAS	Automatically Tuned Linear Algebra Software
BLAS	Basic Fortran 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 est un *distribution* de logiciel à code source ouvert.

Logiciels inclus dans 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 est un *distribution* de logiciel à code source ouvert.

Logiciels inclus dans Sage :

IML	Integer Matrix Library
IPython	Interactive Python shell
LAPACK	Fortan 77 linear algebra library
Lcalc	L-functions calculator
Libcrypt	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 est un *distribution* de logiciel à code source ouvert.

Logiciels inclus dans 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 est un *distribution* de logiciel à code source ouvert.

Logiciels inclus dans 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 est un *distribution* de logiciel à code source ouvert.

Logiciels inclus dans 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 est un *distribution* de logiciel à code source ouvert.

Logiciels inclus dans 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

... ainsi que d'autres progiciel optionnel

Sage est un distribution de logiciel mathématiques.

« Le but général de Sage est de créer une alternative libre viable à Maple, Mathematica, Magma et MATLAB [et MuPAD]. »

Sage est un distribution de logiciel mathématiques.

« Le but général de Sage est de créer une alternative libre viable à Maple, Mathematica, Magma et MATLAB [et MuPAD]. »

Algèbre	GAP, Maxima, Singular
Géométrie algébrique	Singular, Macaulay2 (optionnel)
Arithmétique en précision arbitraire	GMP, MPFR, MPFI, NTL, ...
Géométrie arithmétique	PARI, NTL, mwrnk, ecm, ...
Calcul	Maxima, Sympy
Mathématiques combinatoires	Symmetrca, MuPAD-Combinat*
Algèbre linéaire exacte	Linbox, IML
Théorie des graphes	NetworkX
Théorie des groupes	GAP
Algèbre linéaire numérique	GSL, Scipy, Numpy

Sage est un distribution de logiciel à code source ouvert.

“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)

Un petit histoire

- *1999-2005.* William Stein wrote over 25,000 lines of Magma code for his research; decided Magma was a bad long-term investment because he couldn't see or change the internals.
- *Feb. 2005.* SAGE-0.1, a Python library that glued together open source programs PARI, Maxima, Python, Singular, GAP.
- Feb. 2006. SAGE-1.0 (& Sage Days 1 workshop)
- *2007–Today.* Several workshops. Sage Days 10 in Nancy in October, Sage Days 11 last week. (Not counted by positive integers only!)
- *Money:* NSF grants (funds a postdoc, workshops, hardware); Google; Microsoft Research (native Windows port);
- *Current Version.* Sage-3.2.

La langage de programmation de Sage est Python

Python est un langage de programmation très puissant, moderne, et interprété.

La langage de programmation de Sage est Python

Python est un langage de programmation très puissant, moderne, et interprété.

- *interprété* signifie que se marche comme Maple, MuPAD, ...

```
python: x = 17
```

```
python: x
```

```
17
```

```
python: 3*x
```

```
51
```

La langage de programmation de Sage est Python

Python est un langage de programmation très puissant, moderne, et interprété.

- *interprété* signifie que se marche comme Maple, MuPAD, ...

```
python: x = 17
```

```
python: x
```

```
17
```

```
python: 3*x
```

```
51
```

- Elle est facile à apprendre; beaucoup de documentation libre:
 - Plongez au coeur de Python (<http://diveintopython.adrahon.org>)
 - Tutoriel Python (<http://www.ceramiko.ch/python/main.html>)
 - Tutoriel Sage (<http://wiki.sagemath.org/i18n/French>)

La langage de programmation de Sage est Python

La langage de programmation de Sage est Python

- Facile à lire et à écrire:

math: $\{17x \mid x \in \{0, 1, \dots, 9\} \text{ and } x \text{ is odd}\}$

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

La langage de programmation de Sage est Python

- Facile à lire et à écrire:

math: $\{17x \mid x \in \{0, 1, \dots, 9\} \text{ and } x \text{ is odd}\}$

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

- Beaucoup des *bibliothèques* pour Python: base de données, graphiques, réseau, ...

La langage de programmation de Sage est Python

- Facile à lire et à écrire:

math: $\{17x \mid x \in \{0, 1, \dots, 9\} \text{ and } x \text{ is odd}\}$

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

- Beaucoup des *bibliothèques* pour Python: base de données, graphiques, réseau, ...
- Facile à utiliser des *bibliothèques C/C++* avec Python.

La langage de programmation de Sage est Python

- Facile à lire et à écrire:

math: $\{17x \mid x \in \{0, 1, \dots, 9\} \text{ and } x \text{ is odd}\}$

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

- Beaucoup des *bibliothèques* pour Python: base de données, graphiques, réseau, ...
- Facile à utiliser des *bibliothèques C/C++* avec Python.
- *Cython*: code Python \mapsto code C.

La langue de programmation de Sage est 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
(créateur de Python)

On peut utiliser Sage avec \LaTeX

On peut utiliser Sage avec \LaTeX

dans \LaTeX :

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

Soit $\text{\sigma} = \text{\sage{sigma}}$. L'algorithme de Robinson-Schensted donne les tableau:

```
\[\text{\sage{P}} \quad \text{\sage{Q}}\]
```


On peut utiliser Sage avec \LaTeX

dans \LaTeX :

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

Soit $\sigma = \text{\sage{sigma}}$. L'algorithme de Robinson-Schensted donne les tableaux:

```
\[\sage{P} \quad \sage{Q}\]
```

dans le document:

Soit $\sigma = 73152684$. L'algorithme de Robinson-Schensted donne les tableaux:

1	2	4	8
3	5	6	
7			

1	4	6	7
2	5	8	
3			

On peut utiliser Sage avec \LaTeX

dans \LaTeX :

Volià un arbre:

```
\sageplot{Graph({0:[1,2,3], 2:[5]}).plot()}
```

On peut utiliser Sage avec \LaTeX

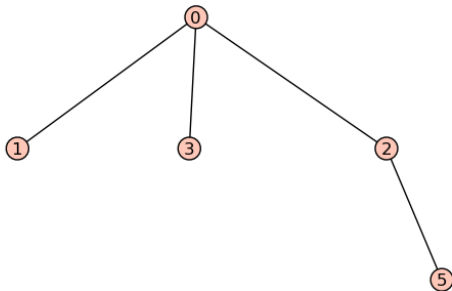
dans \LaTeX :

Volià un arbre:

```
\sageplot{Graph({0:[1,2,3], 2:[5]}).plot()}
```

dans le document:

Volià un arbre:



On peut utiliser Sage avec \LaTeX

dans \LaTeX :

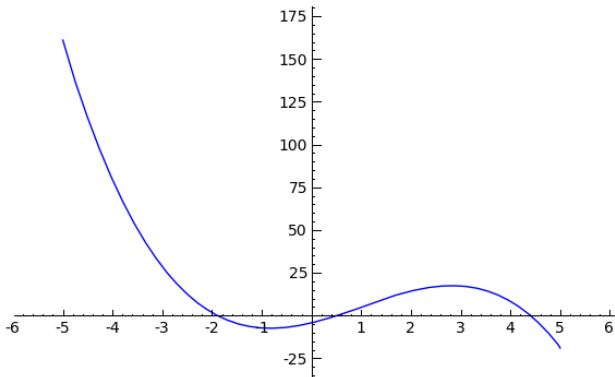
```
\sageplot{plot(-x^3+3*x^2+7*x-4,-5,5)}
```

On peut utiliser Sage avec \LaTeX

dans \LaTeX :

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

dans le document:



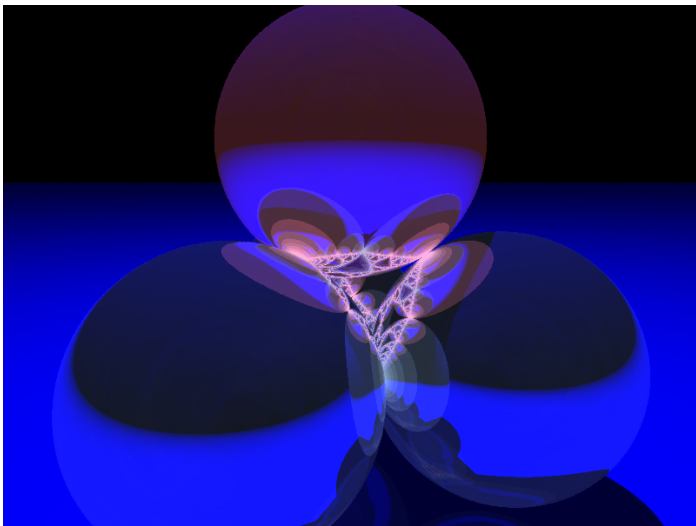
On peut utiliser Sage avec L^AT_EX

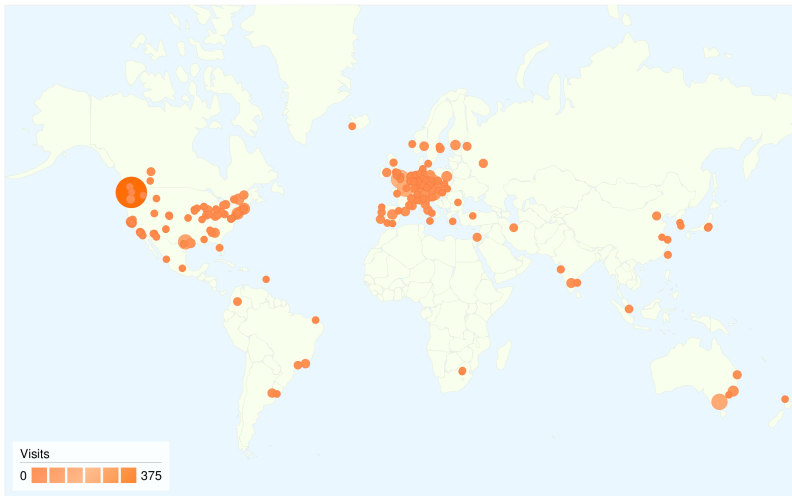
dans L^AT_EX:

```
\begin{sagesilent}
  t6 = Tachyon(camera_center=(1.25,-3.85,0.65), xres=800, yres=600,
              raydepth=12, aspectratio=.75, antialiasing=True)
  t6.light((0.02,0.012,0.001), 0.01, (1,0,0))
  t6.light((0,0,10), 0.01, (0,0,1))
  t6.texture('s', color=(.8,1,1), opacity=.9, specular=.95, \
            diffuse=.3, ambient=0.05)
  t6.texture('p', color=(0,0,1), opacity=1, specular=.2)
  t6.sphere((-1,-.57735,-0.7071),1,'s')
  t6.sphere((1,-.57735,-0.7071),1,'s')
  t6.sphere((0,1.15465,-0.7071),1,'s')
  t6.sphere((0,0,0.9259),1,'s')
  t6.plane((0,0,-1.9259),(0,0,1),'p')
\end{sagesilent}
\sageplot{t6}
```

On peut utiliser Sage avec \LaTeX

dans le document:





11,439 visits came from 2,859 cities

Plusieurs façons d'utiliser Sage

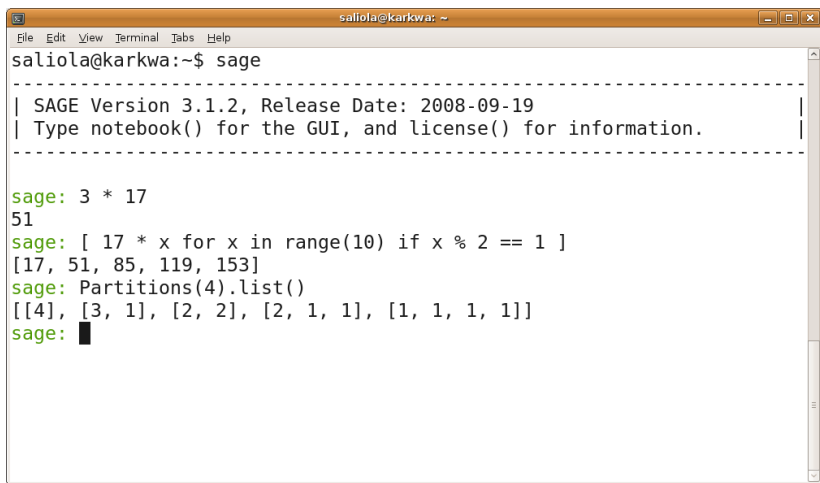
- Comme un bibliothèque pour les script Python:

```
#!/usr/bin/env sage -python
```

```
import sys  
from sage.all import *
```

Plusieurs façons d'utiliser Sage

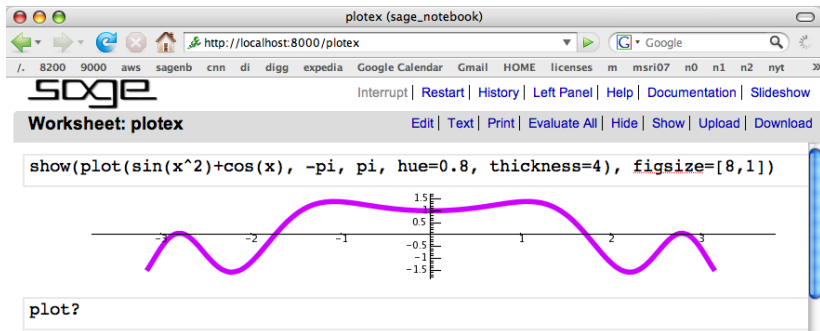
- à partir de la ligne de commande:



```
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: █
```

Plusieurs façons d'utiliser Sage

Interface graphique « notebook » (essayer en ligne: sagenb.org)



The screenshot shows a web browser window titled "plotex (sage_notebook)". The address bar shows "http://localhost:8000/plotex". The page header includes the Sage logo and navigation links: "Interrupt | Restart | History | Left Panel | Help | Documentation | Slideshow". Below the header, there is a "Worksheet: plotex" section with links: "Edit | Text | Print | Evaluate All | Hide | Show | Upload | Download". The main content area contains a code cell with the following code:

```
show(plot(sin(x^2)+cos(x), -pi, pi, hue=0.8, thickness=4), figsize=[8,1])
```

Below the code, a plot is displayed. The plot shows a smooth, wavy curve in a magenta color. The x-axis ranges from approximately -3.14 to 3.14, with major ticks at -3, -2, -1, 1, 2, and 3. The y-axis ranges from -1.5 to 1.5, with major ticks at -1.5, -1, -0.5, 0.5, 1, and 1.5. The curve starts at approximately (-3.14, -1.5), reaches a local minimum at x ≈ -2.5, crosses the x-axis at x ≈ -1.5, reaches a local maximum at x ≈ -0.5, crosses the x-axis again at x ≈ 1.5, reaches a local minimum at x ≈ 2.5, and ends at approximately (3.14, -1.5).

Below the plot, there is a text input field containing the text "plot?".

Type: `<type 'instance'>`
Definition: `plot([noargspec])`
Docstring:

Use plot by writing

```
plot(X, ...)
```

where X is a SAGE object that either is callable and returns numbers that can be coerced to floats, or has a plot method that returns a GraphicPrimitive object.

Type plot.options for a dictionary of the default options for plots. You can change this to change

DEMO

- Getting help
- Interfaces: GAP, Maple, MuPAD, Magma, ...
- Combinatorics: permutations, symmetric functions, ...
- Cython
- Words