

Introduction à



Marc MEZZAROBBA

CNRS, LIP6, Université Paris 6, équipe Pequan

Qu'est-ce que Sage ?

« *Une alternative libre viable à Magma, Maple, Mathematica et Matlab* »

« *Construire la voiture au lieu de réinventer la roue* »



Essayer Sage



<http://sagemath.org/>
GNU GPL



<http://cloud.sagemath.com/>



<http://ask.sagemath.org/>



Calcul mathématique avec Sage



A. Casamayou
N. Cohen
G. Connan
T. Dumont
L. Fousse
F. Maltey
M. Meulien
M. Mezzarobba
C. Pernet
N. M. Thiéry
P. Zimmermann

<http://sagebook.gforge.inria.fr/>























Qu'est-ce que Sage ?

- 1 Une distribution
- 2 Une bibliothèque Python
- 3 Un système interactif



Une distribution

```
$ wget http://mirror/sage/src/sage-6.7.tar.gz
&& tar xfz sage-6.7.tar.gz && cd sage-6.7 && make
$ ./sage
```

ATLAS •  cddlib • cephes • cliquer • CVXOPT
 Python • eclib • GMP-ECM • FFLAS-FPACK
fpll • FLINT •  GAP • Gfan •  Ginac • GLPK
GSL • IML • IP[y]:thon •  Jinja • Jmol
 LAPACK • lcalc •  LinBox •  matplotlib
 Maxima • M4RI • MPC • MPFI •  MPFR
mpmath • networkx • NTL •  NumPy • PALP
PARI •  PolyBoRi •  PPL •  Ginac • pynac •  python
 R •  SciPy •  SINGULAR • symmetrica
sympow •  tachyon •  Tornado • + d'autres...

Une bibliothèque Python

<code>sage.algebras</code>	<code>sage.logic</code>
<code>sage.calculus</code>	<code>sage.matrix</code>
<code>sage.categories</code>	<code>sage.modular</code>
<code>sage.coding</code>	<code>sage.modules</code>
<code>sage.combinat</code>	<code>sage.monoids</code>
<code>sage.crypto</code>	<code>sage.numerical</code>
<code>sage.databases</code>	<code>sage.parallel</code>
<code>sage.finance</code>	<code>sage.plot</code>
<code>sage.functions</code>	<code>sage.rings</code>
<code>sage.geometry</code>	<code>sage.sat</code>
<code>sage.graphs</code>	<code>sage.schemes</code>
<code>sage.groups</code>	<code>sage.sets</code>
<code>sage.homology</code>	<code>sage.stats</code>
<code>sage.interfaces</code>	<code>sage.symbolic</code>
<code>sage.lfunctions</code>	<code>...</code>

- S'appuie sur les logiciels tiers embarqués
- \simeq 460 000 lignes de code spécifique (+ 900 k de doc, tests, ...)



Une bibliothèque Python

<code>sage.algebras</code>	<code>sage.logic</code>
<code>sage.calculus</code>	<code>sage.matrix</code>
<code>sage.categories</code>	<code>sage.modular</code>
<code>sage.coding</code>	<code>sage.modules</code>
<code>sage.combinat</code>	<code>sage.monoids</code>
<code>sage.crypto</code>	<code>sage.numerical</code>
<code>sage.databases</code>	<code>sage.parallel</code>
<code>sage.finance</code>	<code>sage.plot</code>
<code>sage.functions</code>	<code>sage.rings</code>
<code>sage.geometry</code>	<code>sage.sat</code>
<code>sage.graphs</code>	<code>sage.schemes</code>
<code>sage.groups</code>	<code>sage.sets</code>
<code>sage.homology</code>	<code>sage.stats</code>
<code>sage.interfaces</code>	<code>sage.symbolic</code>
<code>sage.lfunctions</code>	<code>...</code>

- S'appuie sur les logiciels tiers embarqués
- \simeq 460 000 lignes de code spécifique (+ 900 k de doc, tests, ...)



Une bibliothèque Python

<code>sage.algebras</code>	<code>sage.logic</code>
<code>sage.calculus</code>	<code>sage.matrix</code>
<code>sage.categories</code>	<code>sage.modular</code>
<code>sage.coding</code>	<code>sage.modules</code>
<code>sage.combinat</code>	<code>sage.monoids</code>
<code>sage.crypto</code>	<code>sage.numerical</code>
<code>sage.databases</code>	<code>sage.parallel</code>
<code>sage.finance</code>	<code>sage.plot</code>
<code>sage.functions</code>	<code>sage.rings</code>
<code>sage.geometry</code>	<code>sage.sat</code>
<code>sage.graphs</code>	<code>sage.schemes</code>
<code>sage.groups</code>	<code>sage.sets</code>
<code>sage.homology</code>	<code>sage.stats</code>
<code>sage.interfaces</code>	<code>sage.symbolic</code>
<code>sage.lfunctions</code>	<code>...</code>

- S'appuie sur les logiciels tiers embarqués
- \simeq 460 000 lignes de code spécifique (+ 900 k de doc, tests, ...)



Une bibliothèque Python

`sage.algebras`

`sage.calculus`

`sage.categories`

`sage.coding`

`sage.combinat`

`sage.crypto`

`sage.databases`

`sage.finance`

`sage.functions`

`sage.geometry`

`sage.graphs`

`sage.groups`

`sage.homology`

`sage.interfaces`

`sage.lfunctions`

`sage.logic`

`sage.matrix`

`sage.modular`

`sage.modules`

`sage.monoids`

`sage.numerical`

`sage.parallel`

`sage.plot`

`sage.rings`

`sage.sat`

`sage.schemes`

`sage.sets`

`sage.stats`

`sage.symbolic`

...

- S'appuie sur les logiciels tiers embarqués
- \simeq 460 000 lignes de code spécifique (+ 900 k de doc, tests, ...)



Une bibliothèque Python

<code>sage.algebras</code>	<code>sage.logic</code>
<code>sage.calculus</code>	<code>sage.matrix</code>
<code>sage.categories</code>	<code>sage.modular</code>
<code>sage.coding</code>	<code>sage.modules</code>
<code>sage.combinat</code>	<code>sage.monoids</code>
<code>sage.crypto</code>	<code>sage.numerical</code>
<code>sage.databases</code>	<code>sage.parallel</code>
<code>sage.finance</code>	<code>sage.plot</code>
<code>sage.functions</code>	<code>sage.rings</code>
<code>sage.geometry</code>	<code>sage.sat</code>
<code>sage.graphs</code>	<code>sage.schemes</code>
<code>sage.groups</code>	<code>sage.sets</code>
<code>sage.homology</code>	<code>sage.stats</code>
<code>sage.interfaces</code>	<code>sage.symbolic</code>
<code>sage.lfunctions</code>	<code>...</code>

- S'appuie sur les logiciels tiers embarqués
- \simeq 460 000 lignes de code spécifique (+ 900 k de doc, tests, ...)



Une bibliothèque Python

<code>sage.algebras</code>	<code>sage.logic</code>
<code>sage.calculus</code>	<code>sage.matrix</code>
<code>sage.categories</code>	<code>sage.modular</code>
<code>sage.coding</code>	<code>sage.modules</code>
<code>sage.combinat</code>	<code>sage.monoids</code>
<code>sage.crypto</code>	<code>sage.numerical</code>
<code>sage.databases</code>	<code>sage.parallel</code>
<code>sage.finance</code>	<code>sage.plot</code>
<code>sage.functions</code>	<code>sage.rings</code>
<code>sage.geometry</code>	<code>sage.sat</code>
<code>sage.graphs</code>	<code>sage.schemes</code>
<code>sage.groups</code>	<code>sage.sets</code>
<code>sage.homology</code>	<code>sage.stats</code>
<code>sage.interfaces</code>	<code>sage.symbolic</code>
<code>sage.lfunctions</code>	<code>...</code>

- S'appuie sur les logiciels tiers embarqués
- \simeq 460 000 lignes de code spécifique (+ 900 k de doc, tests, ...)



Un système interactif

```
sage: %display ascii_art
sage: version()
SageMath Version 6.8.beta0, Release Date: 2015-
sage: taylor(exp(x), x, 0, 5)
      5      4      3      2
      x      x      x      x
--- + -- + -- + -- + x + 1
120  24  6  2
sage: GF(17)['t'].random_element(degree=7)
9*t^7 + t^6 + 3*t^5 + 15*t^4 + 11*t^3 + 11*t^2
sage: DyckWords(10)[456]
```

```
  /\
 /\  /\
/\  /\  /\
sage: pi.n(digits=30)
3.14159265358979323846264338328
sage: MatrixSpace(GF(17), 3).random_element().L
([ [1 0 0] [ 1 0 0] [12 16 1] )
 [ [0 1 0] [ 2 1 0] [ 0 3 4] )
 [ [0 0 1], [10 6 1], [ 0 0 10] )
sage: factor(234567890987654323467890)
2 * 5 * 1117 * 20999811189584093417
sage: import urllib2
sage: f = urllib2.urlopen("http://sagemath.org/
sage: f.read(121)
'<!DOCTYPE html>\n<html xmlns="http://www.w3.org
tml" lang="en" xml:lang="en">\n<head profile="h
com/-/spec/o'
```

The screenshot shows the SageMath web interface in a browser window titled "Untitled - Sage - Konqueror". The address bar shows "http://localhost:8080/home/admin/35/".

The main content area is divided into three sections:

- Graph:** A graph with 6 nodes (labeled 1, 2, 3, 4, 5, 6) and several edges connecting them.
- Code Editor:** Contains the following code:

```
E = EllipticCurve('37a1')
E.matrix_of_frobenius(11, prec=3)
```

Below the code is an "evaluate" button.
- Output:** Shows the result of the evaluation:
$$\begin{pmatrix} 9 \cdot 11 + 9 \cdot 11^3 + O(11^4) & 10 + 11 + O(11^3) \\ 2 \cdot 11 + 11^2 + O(11^4) & 6 + 11 + 10 \cdot 11^2 + O(11^3) \end{pmatrix}$$

Below the code editor, there is another code block:

```
f = (lambda u,v: cos(u), lambda u,v: sin(u)+cos(v),
lambda u,v: sin(v))
parametric_plot3d(f, (0, 2*pi), (-pi, pi))
```

At the bottom of the interface, a 3D plot is displayed, showing a blue surface (a portion of a sphere or similar shape) within a 3D coordinate system.

Un système interactif

```
sage: %display ascii_art
sage: version()
SageMath Version 6.8.beta0, Release Date: 2015-
sage: taylor(exp(x), x, 0, 5)
  5    4    3    2
  x    x    x    x
--- + -- + -- + -- + x + 1
120  24  6   2
sage: GF(17)['t'].random_element(degree=7)
9*t^7 + t^6 + 3*t^5 + 15*t^4 + 11*t^3 + 11*t^2
sage: DyckWords(10)[456]
```



```
sage: pi.n(digits=30)
3.14159265358979323846264338328
sage: MatrixSpace(GF(17), 3).random_element().L
([ [1 0 0] [ 1 0 0] [12 16 1] )
 [ [0 1 0] [ 2 1 0] [ 0 3 4] )
 [ [0 0 1], [10 6 1], [ 0 0 10] )
sage: factor(234567890987654323467890)
2 * 5 * 1117 * 20999811189584093417
sage: import urllib2
sage: f = urllib2.urlopen("http://sagemath.org/
sage: f.read(121)
'<!DOCTYPE html>\n<html xmlns="http://www.w3.or
tml" lang="en" xml:lang="en">\n<head profile="h
com/-/spec/o'
```

Démo

```
E = EllipticCurve('37a1')
E.matrix_of_frobenius(11,prec=3)

lambda^3 + O(11^4)      10 + 11 + O(11^3)
lambda^2 + O(11^4)    6 + 11 + 10 * 11^2 + O(11^3)

f = (lambda u,v: cos(u), lambda u,v: sin(u)+cos(v),
     lambda u,v: sin(v))
parametric_plot3d(f, (0, 2*pi), (-pi, pi))
```

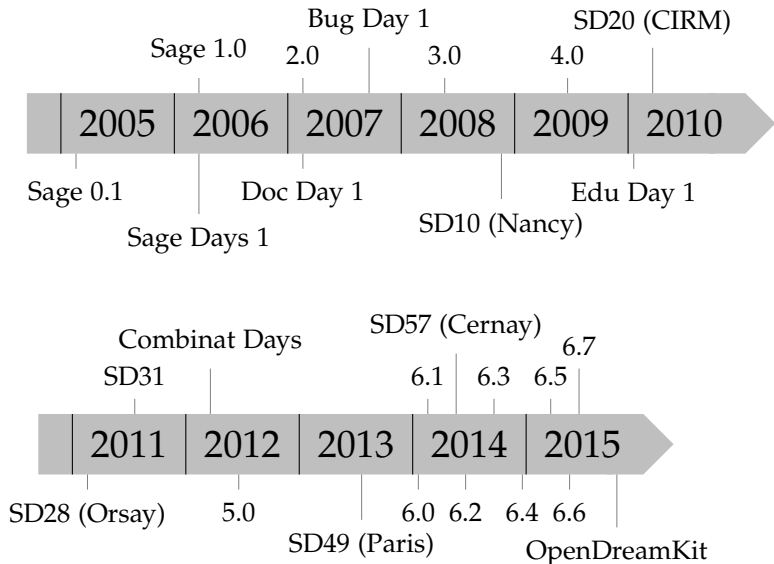


Qu'est-ce que Sage ?

- 1 Une distribution
- 2 Une bibliothèque Python
- 3 Un système interactif
- 4 Une communauté



Histoire



Développement et communauté

Langages principaux : Python + Cython

Développeurs faciles à trouver, bibliothèques riches,
glu facile, pas trop inefficace

Outils : git + trac, listes de diffusion, wiki...

Sage Days

Bugs nombreux mais code facile d'accès

