

# Introduction à



Marc MEZZAROBBA

Projet ALGORITHMS, INRIA Paris Rocquencourt

# 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://sagenb.org/>

















# Qu'est-ce que Sage ?

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



# Une distribution

```
$ wget http://mirror/sagemath/src/sage-4.6.2.tar
  && tar xf sage-4.6.2.tar && cd sage-4.6.2 && make
$ ./sage
```

ATLAS • BLAS • *boehm\_gc* • *boost* • cddlib • cephes  
cliquer • cvxopt •  Python • ECL • eclip • *f2c* • fpLLL  
FLINT • GAP • gfan •  Gambit • GLPK • GSL  
IP[y]: •  • LAPACK • lcalc •  LinBox • matplotlib  
 Maxima • M4RI •  mercurial • MPFI •  MPFR  
mpmath • networkx • NTL •  NumPy • PALP  
 PARI-GP • PIL •  PolyBoKi • Pynac •  python<sup>™</sup>  
 • rubiks •  SciPy •  SINGULAR • symmetrica  
sympow • *tachyon* • zn\_poly • + 47 autres  
+ 49 paquets optionnels + 62 paquets expérimentaux



# Une bibliothèque Python

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

- S'appuie sur les logiciels tiers embarqués
- $\simeq$  200 000 lignes de code spécifique (hors doc + tests)



# Une bibliothèque Python

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

- S'appuie sur les logiciels tiers embarqués
- $\simeq$  200 000 lignes de code spécifique (hors doc + tests)



# Une bibliothèque Python

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

- S'appuie sur les logiciels tiers embarqués
- $\simeq$  200 000 lignes de code spécifique (hors doc + tests)





# Une bibliothèque Python

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

- S'appuie sur les logiciels tiers embarqués
- $\simeq$  200 000 lignes de code spécifique (hors doc + tests)



# Une bibliothèque Python

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

- S'appuie sur les logiciels tiers embarqués
- $\simeq$  200 000 lignes de code spécifique (hors doc + tests)



# Une bibliothèque Python

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

- S'appuie sur les logiciels tiers embarqués
- $\simeq$  200 000 lignes de code spécifique (hors doc + tests)



# Un système interactif

```
-$ sage
```

```
-----  
| Sage Version 4.6.2, Release Date: 2011-02-25 |  
| Type notebook() for the GUI, and license() for information. |  
-----
```

```
sage: taylor(exp(x), x, 0, 5)  
1/120*x^5 + 1/24*x^4 + 1/6*x^3 + 1/2*x^2 + x + 1  
  
sage:  
sage: MatrixSpace(RR,5,3).random_element()  
[-0.570390764900653  0.521446993576251 -0.950894560265950]  
[-0.942431942330060  0.254122819002693  0.916721924359961]  
[-0.195702504102615 -0.350489870318781 -0.214359534055980]  
[ 0.487076746020482  0.461116221981387 -0.665179594662514]  
[ 0.180194930460366  0.616390883848273 -0.389309976296204]  
  
sage:  
sage: import urllib2  
sage: f = urllib2.urlopen("http://sagemath.org/")  
sage: f.read(121)  
'<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"\nhttp  
://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">'  
  
sage:
```

The screenshot shows the Sage Notebook interface in a browser window titled "Enveloppes (2) (Sage) - Iceweasel". The interface includes a navigation bar with "Sage The Sage" logo, user information "admin", and links for "Toggle", "Home", "Published", "Log", "Settings", "Help", "Report a Problem", and "Sign out". The notebook title is "Enveloppes (2)" and it shows it was last edited on May 15, 2011. The interface has a menu bar with "File...", "Action...", "Data...", "sage", and "Typeset", and a toolbar with "Print", "Worksheet", "Edit", "Text", "Undo", "Share", and "Publish".

The main content area shows the following Sage session:

```
R.<x,y,t> = QQ[]; eq = x^2+(y-t)^2-1/2*(t^2 + 1)  
eq  
  
x^2 + y^2 - 2yt + 1/2 t^2 - 1/2  
  
fig_circles = add((eq(t=k/5)*QQ[x,y]).plot()  
for k in (-15..15))  
options = {'aspect_ratio': 1, 'xmin': -2,  
'xmax': 2, 'ymin': -3, 'ymax': 3, 'frame':  
True, 'axes': False, 'fontsize': 8}  
fig_circles.show(**options)
```

Below the code, there is an "evaluate" button and a plot showing a series of blue circles. The plot has a y-axis ranging from 1 to 3 and an x-axis ranging from -2 to 2. The circles are centered at different y-values and have varying radii, creating a pattern that resembles a series of overlapping parabolas or a "circular envelope".



# Démo

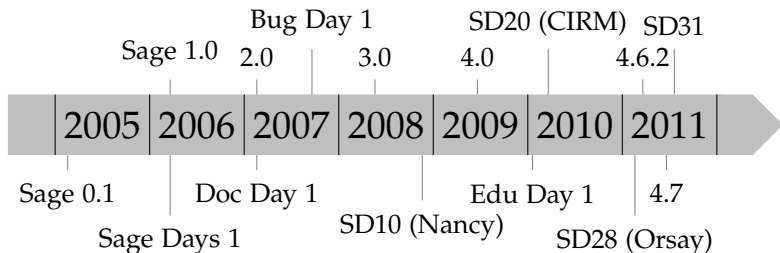
© 2008 Duk

# Qu'est-ce que Sage ?

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



# Développement et communauté



## ■ Outils

- ▶ Listes de diffusion
- ▶ Mercurial + trac
- ▶ Wiki

## ■ Sage Days

## ■ Python

- ▶ Répandu – ouverture
- ▶ Interfaçage aisé

## ■ Sage sera ce que vous en ferez !

