

Introduction à Sage

Marc MEZZAROBBA, Équipe Algorithms, INRIA Paris-Rocquencourt

Cet exposé est un tutoriel destiné surtout à ceux qui n'ont jamais utilisé Sage [1] et voudraient découvrir rapidement les possibilités de ce logiciel.

Sage est un système de calcul formel et numérique, en un mot de *calcul mathématique*, qui s'est donné pour mission de fournir une alternative libre (*free/open source*) à Magma, Maple, Mathematica et Matlab. Il est disponible, ainsi qu'une abondante documentation, à l'adresse <http://sagemath.org/>. Les développeurs de Sage aiment à dire qu'ils veulent construire la voiture au lieu de réinventer la roue. C'est pourquoi Sage est tout à la fois :

- une *distribution* de près d'une centaine de logiciels libres tiers — ATLAS, FLINT, gsl, LAPACK, LinBox, Maxima, PARI, R, etc. —, avec des dépendances externes réduites au minimum ;
- une *interface* aux logiciels mathématiques majeurs inclus ou non dans la distribution, dont Axiom, Maple, Mathematica, Matlab, Maxima, Octave, R ou encore Scilab ;
- et lui-même un *système de calcul mathématique* à part entière, avec ses propres types d'objets et sa propre interface cohérente, qui s'appuie sur les programmes de la distribution Sage complétés de plusieurs centaines de milliers de lignes de code spécifique.

Ses fonctionnalités couvrent (à des degrés divers) des domaines comme le calcul symbolique élémentaire, le calcul numérique à précision machine ou arbitraire, l'algèbre linéaire symbolique et numérique, l'algèbre commutative, la théorie des nombres, la combinatoire, la théorie des graphes ou la cryptographie. Il n'y a pas de langage Sage : Sage se présente essentiellement comme une bibliothèque Python, et son utilisation interactive se fait à travers un interpréteur de commandes Python.

Les quelques commandes suivantes donnent un avant-goût de l'utilisation de Sage.

```
sage: 1 + 1
2
sage: ( 50.factorial() - 1 ).factor()
3282689 * 9264993790673858548163596419296731686851127709314075137791
sage: integrate(1/(1+x^2), x, -infinity, infinity)
pi
sage: _.numerical_approx(250)
3.1415926535897932384626433832795028841971693993751058209749445923078164063
sage: u, v = SR.var('u,v'); f = function('f', u, v)
sage: diff(f(u=cos(v), v=u^2*v), v)
u^2*D[1](f)(cos(v), u^2*v) - sin(v)*D[0](f)(cos(v), u^2*v)
sage: R.<z> = QQ[] # affecte a R l'anneau Q[z] et a z son indeterminee
sage: p = R.random_element(degree=3); p
-2*z^3 - 1/2*z^2 + 2*z - 1
sage: ((2*z+1)*p^100).roots(ComplexIntervalField())
[(-1.307397616924093?, 100), (-0.5000000000000000?, 1), (0.528698808462047?
- 0.320806319199897?*I, 100), (0.528698808462047? + 0.320806319199897?*I, 100)]
sage: S.<w> = GF(17)[] # polynomes sur F_17
sage: sum(i*w^i for i in (1..10))
10*w^10 + 9*w^9 + 8*w^8 + 7*w^7 + 6*w^6 + 5*w^5 + 4*w^4 + 3*w^3 + 2*w^2 + w
sage: _.factor()
(10) * w * (w + 10) * (w + 14) * (w^3 + 16*w^2 + 16*w + 3) * (w^4 + 15*w^2 + 2*w + 1)
```

Références

- [1] WILLIAM A. STEIN *et al.*, *Sage Mathematics Software (Version 4.6.2)*, The Sage Development Team, 2011, <http://www.sagemath.org/>.