Cet article est paru dans le Bulletin de l'AMQ, Volume 51, numéro 2, mai 2011.
http://newton.mat.ulaval.ca/amq/Bulletins.html

Sage, un logiciel libre de mathématiques

Sébastien Labbé
Laboratoire de Combinatoire et d'Informatique Mathématique
Université du Québec à Montréal
labbe.sebastien@courrier.uqam.ca
http://www.thales.math.uqam.ca/~labbes/

Résumé

Sage est un logiciel libre de mathématiques dont la mission est d'offrir une alternative aux logiciels propriétaires *Magma*, *Maple*, *Mathematica* et *Matlab*. Initié en 2006 aux États-Unis, le projet prend de l'ampleur aux quatre coins de la planète y compris au Québec.

Sage [1] est un logiciel libre de mathématiques disponible sous la licence publique générale de GNU. Il combine la puissance de plus de 90 logiciels libres spécialisés sous une interface commune basée sur le langage de programmation *Python*. Sa mission est d'offrir une alternative viable, libre et gratuite aux logiciels propriétaires *Magma*, *Maple*, *Mathematica* et *Matlab*.

Le projet, initié en 2006 par William Stein à l'Université de Seattle aux États-Unis, prend de l'ampleur. En cinq années, plus d'une centaine de mathématiciens à travers le monde ont contribué à l'amélioration du logiciel par l'ajout de fonctionnalités diverses.

Dans ce texte, nous présenterons un des principes important du logiciel Sage qui consiste à se baser sur des outils existants. Nous expliquerons en quoi les valeurs du logiciel libre respectent l'esprit des mathématiques. Puis, nous évoquerons les outils utilisés et développés par Sage qui lui permettent d'être compétitifs au niveau de la vitesse d'exécution. Nous décrirons ensuite quelques événements reliées au logiciel Sage s'étant produits au cours des dernières années au Québec plus particulièrement à l'Université du Québec à Montréal. Nous terminerons avec trois exemples d'utilisation.

Ne pas réinventer la roue

Lorsque William Stein a commencé ce projet, il a vite réalisé qu'il ne pouvait pas à lui seul construire un logiciel de mathématique, car cela représente beaucoup trop de travail [2]. C'est alors qu'il a décidé de rassembler les meilleurs logiciels libres existants et ceci est devenu l'un des principes de base du logiciel Sage : « ne pas réinventer la roue ». Ainsi, Sage rassemble les meilleurs logiciels libres existants, par exemple *Atlas* et *LinBox* pour l'algèbre linéaire, *Maxima* pour le calcul symbolique, *Pari/GP* pour les calculs en théorie des nombres ou *GAP* pour l'algèbre. À ceux-ci, s'ajoutent des bibliothèques pour la visualisation 2D et 3D tels que *Matplotlib* et *Jmol*.

De plus, cette philosophie d'utiliser au maximum ce qui existe s'étend au langage utilisateur. En effet, contrairement aux logiciels propriétaires de mathématiques bien connus, Sage a choisi d'utiliser le langage *Python*. Ceci permet de bénéficier du savoir des centaines de millions d'utilisateurs de *Python* à travers le monde et permet surtout aux concepteurs de Sage de se consacrer pleinement aux mathématiques et non à la conception d'un nouveau langage de programmation.

Finalement, le même principe s'applique à l'interface graphique, où Sage se sert d'un simple navigateur Web. Ceci permet d'utiliser Sage via Internet sur divers serveurs, le principal étant www.sagenb.org [10].

Respect de l'esprit des mathématiques

Les valeurs du logiciel libre correspondent bien à celles des mathématiques comme l'a écrit en 1993 Joachim Neubüser [3], qui a lancé le logiciel libre *GAP* en 1986 :

> ≪ *Vous pouvez lire le théorème de Sylow et sa preuve dans le livre Huppert à la bibliothèque, puis vous pouvez alors utiliser le théorème de Sylow pour le reste de votre vie gratuitement. Mais pour plusieurs systèmes de calcul formel des redevances de licence doivent être versés régulièrement. Dans cette situation deux des règles les plus élémentaires de conduite sont violés. En mathématiques, les résultats sont transmis gratuitement et les démonstrations sont publiques pour permettre leur vérification.* ≫

Vitesse d'exécution

Sage utilise *Cython* qui permet de traduire du code écrit en *Python* en langage *C*. C'est ce qui permet à Sage d'atteindre des niveaux inégalés pour la vitesse d'exécution de certains algorithmes. Par exemple, dans la documentation de Sage [6], on peut lire que ≪ le calcul du déterminant d'une matrice 500 x 500 sur des entiers aléatoire de 32 bits se fait en 4.12 secondes alors que le logiciel propriétaire *Magma* le fait en 62.87 secondes (testé sur un core2 duo 2.6Ghz sur OS X). ≫

Le logiciel Sage à l'UQAM

Plusieurs ateliers sur le logiciel Sage ont eu lieu à l'Université du Québec à Montréal (UQAM) au cours des dernières années où un petit bassin d'utilisateurs de Sage a commencé à se former. Certains étudiants gradués sont même devenus des développeurs actifs et ont ajouté à Sage des algorithmes nécessaires pour leurs propres recherches mathématiques.

En mai 2008, Franco Saliola, alors postdoctorant au Laboratoire de combinatoire informatique mathématique (LaCIM) à l'UQAM, a donné une formation de quelques jours à des étudiants et professeurs de l'UQAM. Puis, en juillet 2009, le LaCIM a organisé une formation de trois jours sur Sage [7] donnée par Sébastien Labbé. Cette fois, une vingtaine d'étudiants de l'Université de Montréal, de l'Université du Québec à Trois-Rivières, de l'Université McGill et de l'UQAM et quelques professeurs ont participé à l'activité. Puis, le 1^{er} septembre 2010, la journée Sage Day 25.5 a été organisé au LaCIM attirant une trentaine de participants dont une dizaine de professeurs de différents cégeps du Québec.

Au cours des prochaines années, le logiciel Sage sera certainement de plus en plus connu. Ainsi, les habituels coûts élevés des licences seront de moins en moins un obstacle à l'utilisation des logiciels de mathématiques dans les établissements scolaires tels que les universités, les cégeps et même les écoles secondaires.

Exemple 1 : Le calcul différentiel et intégral avec Sage

Sage peut dériver et intégrer plusieurs fonctions. Par exemple, pour dériver $\sin(u)$ par rapport à la variable u, on fait comme suit :

```
sage: u = var('u')
sage: diff(sin(u), u)
cos(u)
```

Le calcul de la quatrième dérivée de $\sin(x^2)$:

```
sage: diff(sin(x^2), x, 4)
16*x^4*sin(x^2) - 48*x^2*cos(x^2) - 12*sin(x^2)
```

Le calcul des dérivées partielles de $x^3 + 25y^2$ par rapport à x et à y respectivement :

```
sage: x,y = var('x,y')
sage: f = x^3 + 25*y^2
sage: f.diff(x)
3*x^2
sage: f.diff(y)
50*y
```

On calcule la limite $\lim_{x\to 0^+} \frac{1}{x}$ et de $\lim_{n\to\infty}(1+\frac{1}{n})^n$:

```
sage: lim(1/x, x=0, dir='+')
+Infinity
sage: n = var('n')
sage: lim((1+1/n)^n, n = oo)
e
```

Passons maintenant aux intégrales, tant indéfinies que définies. On calcule $\int x\sin(x^2)\,dx$ et $\int_0^1 \frac{x}{x^2+1}\,dx$:

```
sage: integral(x*sin(x^2), x)
-1/2*cos(x^2)
sage: integral(x/(x^2+1), x, 0, 1)
1/2*log(2)
```

On calcule la décomposition en fractions partielles de $\frac{1}{x^2-1}$:

```
sage: f = 1 / (x^2 - 1)
sage: f.partial_fraction(x)
1/2/(x - 1) - 1/2/(x + 1)
```

Exemple 2 : Sage est une combinaison de plusieurs logiciels

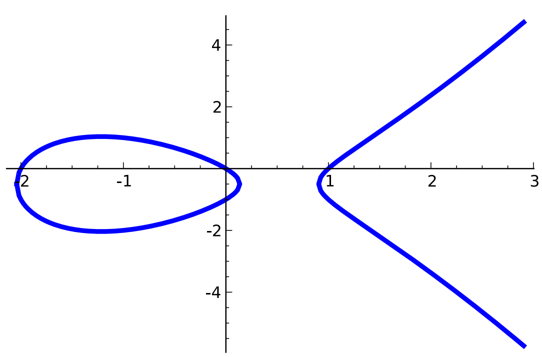
Les exemples suivants illustrent que Sage réussit à combiner plusieurs logiciels de mathématiques sans que ce soit visible aux yeux de l'utilisateur. Ils sont tirés d'une présentation [4] de William Stein, créateur de Sage et chercheur en théorie des nombres.

Construction d'une courbe elliptique utilisant la table de *John Cremona* :

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

Utilisation de *matplotlib* pour la dessiner :

```
sage: plot(E, thickness=3)
```



Utilisation de *mwrank* pour faire une 2-descente :

```
sage: print E.mwrank()
Curve [0,1,1,-2,0] :    Rank = 2
```

Calcul des coefficients de Fourier a_n avec le logiciel *PARI* :

```
sage: E.anlist(15)
[0, 1, -2, -2, 2, -3, 4, -5, 0, 1, 6, -4, -4, -3, 10, 6]
```

Calculer les zéros dans la bande critique de la série L avec *Lcalc* :

```
sage: E.lseries().zeros(5)
[0.000000000, 0.000000000, 2.87609907, 4.41689608, 5.79340263]
```

Sympow pour calculer le degré modulaire :

```
sage: E.modular_degree()
40
```

Calcul du rang du groupe 3-Selmer avec le logiciel (non libre) *Magma* :

```
sage: magma(E).ThreeSelmerGroup()
Abelian Group isomorphic to Z/3 + Z/3
Defined on 2 generators
Relations:
3*$.1 = 0
3*$.2 = 0
```

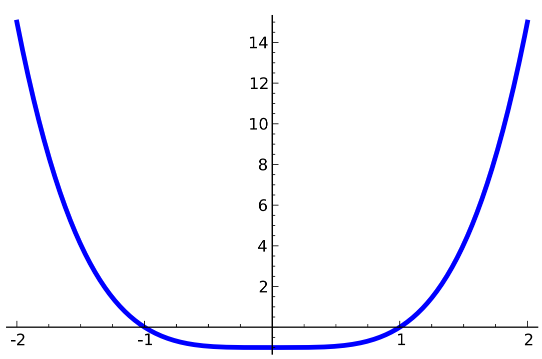
Exemple 3 : Fractale de Newton

Dans cet exemple, nous calculons les racines réelles et complexes d'une fonction polynomiale et nous dessinons une fractale de Newton [5].

Visualisation de la méthode de Newton sur les nombres réels

Définition de la fonction $f(x) = x^4 - 1$ que l'on dessine sur l'intervalle $[-2, 2]$:

```
sage: f(x) = x^4 - 1
sage: plot(f, -2, 2, thickness=3)
```



On définit la méthode de Newton qui permet de trouver numériquement les zéros d'une fonction f :

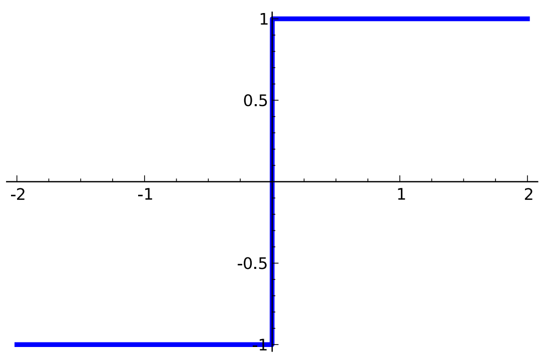
```
sage: def newton(f, z, precision=0.001) :
...       while abs(f(x=z)) >= precision:
...           z = z - f(x=z) / diff(f)(x=z)
...       return z
```

À l'aide de la méthode de Newton, on calcule le zéro dans le voisinage de 4 :

```
sage: newton(f, 4.0)
1.00000015203739
```

Sur l'intervalle $[-2, 2]$, les deux racines réelles distinctes de f sont trouvées par la méthode de Newton et correspondent aux plateaux de l'image ci-bas. En effet, la racine -1 est détectée à partir des nombres négatifs alors que la racine 1 est détectée à partir des nombres positifs :

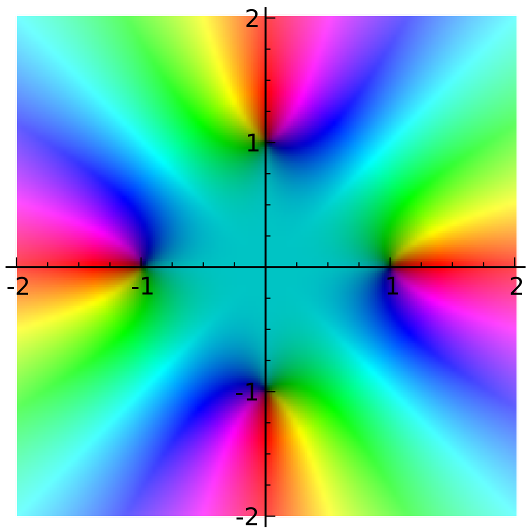
```
plot(lambda z : newton(f, z), (-2, 2), thickness=3)
```



Visualisation de la méthode de Newton sur les nombres complexes

Faisons maintenant le même exercice sur les nombres complexes. La commande `complex_plot` permet de visualiser la fonction f comme une fonction complexe $\mathbb{C} \to \mathbb{C}$. Le module du nombre complexe en sortie est indiqué par la luminosité (zéro étant noir et blanc étant l'infini), tandis que l'argument est représenté par la couleur (le rouge étant réel positif, et ainsi de suite avec les couleurs orange, jaune, vert, bleu et violet, à mesure que la valeur de l'argument augmente) :

```
sage: complex_plot(f, (-2,2), (-2,2))
```

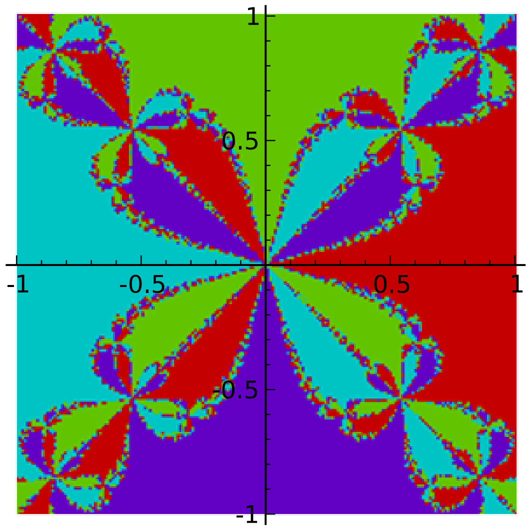


Toujours avec la méthode de Newton définie plus haut, on calcule numériquement les zéros dans le voisinage des nombres complexes $4i$ et $5 + i$ et on obtient les approximations de deux racines distinctes :

```
sage: newton(f, 4.0 * i)
1.00000015203739*I
sage: newton(f, 5.0 + i)
1.00000000948728 + 1.78556358514716e-7*I
```

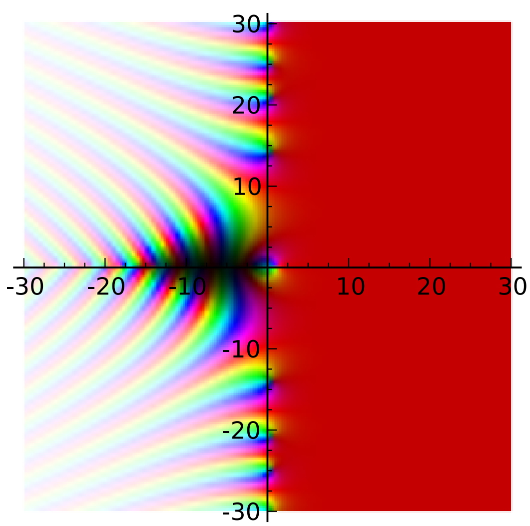
Grâce à la fonction `complex_plot`, on peut visualiser la convergence de la méthode de Newton sur les nombres complexes. On obtient non pas des plateaux comme pour les nombres réels, mais plutôt une fractale formée de quatre types de régions (couleurs) associées aux racines 1, -1, i et $-i$.

```
sage: complex_plot(lambda z : newton(f, z), (-1,1), (-1,1), plot_points=200)
```



Pour terminer, la fonction Zeta de Riemann :

```
sage: complex_plot(zeta, (-30,30), (-30,30))
```



Conclusion

Bien sûr, il y a beaucoup plus à mentionner sur le logiciel libre Sage. Cet article constitue un avant-goût. Plus d'informations sont disponibles sur le site web de Sage [1]. On peut aussi consulter la version française [8] du tutoriel sur Sage ou même le livre *Calcul mathématique avec Sage* [9] structuré selon le programme de l'épreuve de modélisation de l'agrégation de mathématiques en France. De plus, le Notebook publique de Sage [10] permet de tester et d'utiliser Sage via Internet. Finalement, on peut contacter le LaCIM ou l'auteur pour connaître les prochaines formations (souvent gratuites et ouverte au public) sur le logiciel Sage.

Références

[1] W. A. Stein et al., *Sage Mathematics Software (Version 4.6)*, The Sage Development Team, 2010, `http://www.sagemath.org`.

[2] W. A. Stein, *Mathematical Software and Me : A Very Personal Recollection*, 7 décembre 2009, `http://sagemath.blogspot.com/2009/12/mathematical-software-and-me-very.html`.

[3] J. Neubuser. *An invitation to computational group theory*. In C. M. Campbell, T. C. Hurley, E. F. Robertson, S. J. Tobin, and J. J. Ward (editors), Groups '93 Galway/St. Andrews, Volume 2, volume 212 of London Mathematical Society Lecture Note Series, pages 457–475. Cambridge University Press, 1995. `http://www.gap-system.org/Doc/Talks/cgt.ps`.

[4] W. A. Stein, *Sage : Unifying Mathematical Software*, Sage Days 16, Barcelone, 22 juin 2009, `http://modular.math.washington.edu/talks/20090622-sagedays16-thematic/`.

[5] Contributeurs de Wikipédia, Fractale de Newton, *Wikipédia, l'encyclopédie libre*, `http://fr.wikipedia.org/wiki/Fractale_de_Newton`, (Page consultée le janvier 24, 2011).

[6] Sage Reference Manual v4.6, *Dense matrices over the integer ring*, `http://www.sagemath.org/doc/reference/sage/matrix/matrix_integer_dense.html`.

[7] Jours Sage au LaCIM, juillet 2009, `http://wiki.sagemath.org/S%C3%A9bastienLabb%C3%A9/JoursSageUQAM`.

[8] Tutoriel sur Sage (en français), `http://sagemath.org/fr/html/tutorial/`.

[9] P. Zimmermann et al., *Calcul mathématique avec Sage*, Version 1.0, juillet 2010, 315 pages, `sagebook.gforge.inria.fr`.

[10] Le Notebook public de Sage, `http://www.sagenb.org/`.