

Sage: méthodes numériques

Thierry DUMONT, Institut Camille Jordan, Lyon.

Mots-clés : Sage, calcul scientifique, Python, Cython.

Sur le site web de Sage (<http://www.sagemath.org/>), on peut lire:

Mission: Creating a viable free open source alternative to Magma, Maple, Mathematica and Matlab.

Sage est un logiciel jeune: il n'est pas encore raisonnable de comparer la richesse des bibliothèques numériques de Sage et celle de Matlab. Mais c'est un logiciel en développement très rapide. S'il n'est pas encore *actuellement* une véritable alternative à Matlab, il devrait être à terme *plus* qu'une alternative en intégrant les méthodes numériques dans un système beaucoup plus général.

Python, le langage de Sage, est déjà beaucoup utilisé en calcul scientifique (couplage de codes, ajout d'un langage de script, développements autonomes). C'est la clé du développement rapide de Sage et de l'intégration logicielle qu'il offre:

- Sage utilise la plupart des bibliothèques Python développées par ailleurs en calcul numérique: (Numpy, Scipy, Matplotlib etc.), elles mêmes basées sur des bibliothèques classiques (lapack, gsl, blitz, atlas...).
- Toute bibliothèque écrite dans les langages habituels (C, C++, Fortran...) est candidate à être incorporée à Sage. Il y aura à terme des éléments finis, par exemple.
- Tout logiciel intégré dans Sage acquiert une interface utilisateur Python qui en rend l'utilisation transparente, et il entre dans une communauté de logiciels, avec lesquels il est susceptible d'interagir. On peut donc envisager de faire coopérer facilement des méthodes numériques, du calcul symbolique, des calculs exacts, etc.
- Les performances: Python est interprété; mais dans la plupart des applications, seules les boucles les plus extérieures seront écrites dans ce langage: les calculs consommateurs de ressources sont délégués à des bibliothèques compilées. De plus, Sage utilise *Cython* qui permet d'obtenir à partir de programmes écrits en Python (éventuellement légèrement modifiés) des performances comparables à celles des langages compilés. Il est donc possible de faire des développements en Python, tout en obtenant de bonnes performances.

Dès à présent, certaines possibilités sont nouvelles en calcul numérique. Citons:

- la possibilité de faire des calculs avec des nombres flottants de précision quelconque,
- l'algèbre linéaire exacte (dans les rationnels).

Sage a aussi des avantages pédagogiques: il ne faut apprendre qu'un seul langage de programmation là où l'approche classique en logiciels séparés nécessite la connaissance d'un langage pour le calcul numérique et d'un autre pour le calcul symbolique. Et Python est un langage énormément utilisé en dehors des mathématiques.

Dans mon exposé, je montrerai quelques exemples simples d'utilisation, principalement en algèbre linéaire.