



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

Thierry Dumont.

Institut Camille Jordan, Lyon.

25 Octobre 2014.

tdumont@math.univ-lyon1.fr

Un logiciel libre de mathématiques (licence GPL) :

- construit au-dessus de nombreux logiciels libres existants,
- intégrant des développements propres.

Un logiciel libre de mathématiques (licence GPL) :

- construit au-dessus de nombreux logiciels libres existants,
- intégrant des développements propres.

Mission :

Créer un alternative libre viable à Magma, Maple, Mathematica et Matlab.

Sage est originaire de l'Université de l'état de Washington.

William Stein, Professeur.

Si vous ne savez pas ce que c'est que Matlab, Magma, Maple, Mathematica :

- Matlab : méthodes numériques (flottants) « Matrix Laboratory ».
- Maple, Mathematica, Magma : calculs exacts (« Computer Algebra » : dériver intégrer des fonctions, mais aussi plein d'autres choses).

Si vous ne savez pas ce que c'est que Matlab, Magma, Maple, Mathematica :

- Matlab : méthodes numériques (flottants) « Matrix Laboratory ».
- Maple, Mathematica, Magma : calculs exacts (« Computer Algebra » : dériver intégrer des fonctions, mais aussi plein d'autres choses).

$$f(x) = \exp(\cos(\log(\sin(x) + 1))).$$

```
sage: f=exp(cos(log(sin(x)+1)))
```

```
sage: ft=taylor(f,x,0,6)
```

```
sage:
```

$$\frac{53}{240} x^6 e - \frac{1}{6} x^5 e - \frac{1}{8} x^4 e + \frac{1}{2} x^3 e - \frac{1}{2} x^2 e + e.$$

Si vous ne savez pas ce que c'est que Matlab, Magma, Maple, Mathematica :

- Matlab : méthodes numériques (flottants) « Matrix Laboratory ».
- Maple, Mathematica, Magma : calculs exacts (« Computer Algebra » : dériver intégrer des fonctions, mais aussi plein d'autres choses).

$$f(x) = \exp(\cos(\log(\sin(x) + 1))).$$

```
sage: f=exp(cos(log(sin(x)+1)))
```

```
sage: ft=taylor(f,x,0,6)
```

```
sage:
```

$$\frac{53}{240} x^6 e - \frac{1}{6} x^5 e - \frac{1}{8} x^4 e + \frac{1}{2} x^3 e - \frac{1}{2} x^2 e + e.$$

Pourquoi créer un nouveau logiciel ?

Pourquoi créer un nouveau logiciel ?

- Peut on raisonnablement faire des sciences avec des logiciels boîte noire ? Le logiciel fermé est contradictoire avec la démarche scientifique.

Pourquoi créer un nouveau logiciel ?

- Peut on raisonnablement faire des sciences avec des logiciels boîte noire ? Le logiciel fermé est contradictoire avec la démarche scientifique.
- D'autre part, on dispose de nombreux systèmes de calcul, libres, mais cantonnés à un usage limité, et nécessitant un apprentissage non négligeable.

Pourquoi créer un nouveau logiciel ?

- Peut on raisonnablement faire des sciences avec des logiciels boîte noire ? Le logiciel fermé est contradictoire avec la démarche scientifique.
- D'autre part, on dispose de nombreux systèmes de calcul, libres, mais cantonnés à un usage limité, et nécessitant un apprentissage non négligeable.

Exemples :

Gap	Groups and programming	
LinBox	Algèbre linéaire exacte	Grenoble
Maxima	Calcul symbolique	
MPFR	flottants toutes précision	Inria Nancy
PARI	théorie des nombres	Bordeaux
Singular	Polynomes	Kaiserslautern

...

Pourquoi créer un nouveau logiciel ?

- Peut on raisonnablement faire des sciences avec des logiciels boîte noire ? Le logiciel fermé est contradictoire avec la démarche scientifique.
- D'autre part, on dispose de nombreux systèmes de calcul, libres, mais cantonnés à un usage limité, et nécessitant un apprentissage non négligeable.

Exemples :

Gap	Groups and programming	
LinBox	Algèbre linéaire exacte	Grenoble
Maxima	Calcul symbolique	
MPFR	flottants toutes précision	Inria Nancy
PARI	théorie des nombres	Bordeaux
Singular	Polynomes	Kaiserslautern

...

Comment fédérer tout ces logiciels de manière transparente ?

Python !

Comment fédérer tout ces logiciels de manière transparente ?

Python !

Défi

Modéliser les structures mathématiques.

Comment fédérer tout ces logiciels de manière transparente ?

Python !

Défi

Modéliser les structures mathématiques.

En mathématique, on considère des *ensembles*, à partir desquels on définit des *ensembles structurés* par raffinements successifs, en ajoutant des propriétés.

Comment fédérer tout ces logiciels de manière transparente ?

Python !

Défi

Modéliser les structures mathématiques.

En mathématique, on considère des *ensembles*, à partir desquels on définit des *ensembles structurés* par raffinements successifs, en ajoutant des propriétés.

- par exemple les *groupes*, sont des ensembles avec une addition $+$ qui vérifie certains axiomes,

Comment fédérer tout ces logiciels de manière transparente ?

Python !

Défi

Modéliser les structures mathématiques.

En mathématique, on considère des *ensembles*, à partir desquels on définit des *ensembles structurés* par raffinements successifs, en ajoutant des propriétés.

- par exemple les *groupes*, sont des ensembles avec une addition $+$ qui vérifie certains axiomes,
- les *anneaux* sont des *groupes* avec en plus une multiplication \times qui vérifie certains axiomes,

Comment fédérer tout ces logiciels de manière transparente ?

Python !

Défi

Modéliser les structures mathématiques.

En mathématique, on considère des *ensembles*, à partir desquels on définit des *ensembles structurés* par raffinements successifs, en ajoutant des propriétés.

- par exemple les *groupes*, sont des ensembles avec une addition $+$ qui vérifie certains axiomes,
- les *anneaux* sont des *groupes* avec en plus une multiplication \times qui vérifie certains axiomes,
- les *corps* sont des *anneaux* avec plus d'axiomes.

Comment fédérer tout ces logiciels de manière transparente ?

Python !

Défi

Modéliser les structures mathématiques.

En mathématique, on considère des *ensembles*, à partir desquels on définit des *ensembles structurés* par raffinements successifs, en ajoutant des propriétés.

- par exemple les *groupes*, sont des ensembles avec une addition $+$ qui vérifie certains axiomes,
- les *anneaux* sont des *groupes* avec en plus une multiplication \times qui vérifie certains axiomes,
- les *corps* sont des *anneaux* avec plus d'axiomes.

Hiérarchie d'objets, généricité etc...

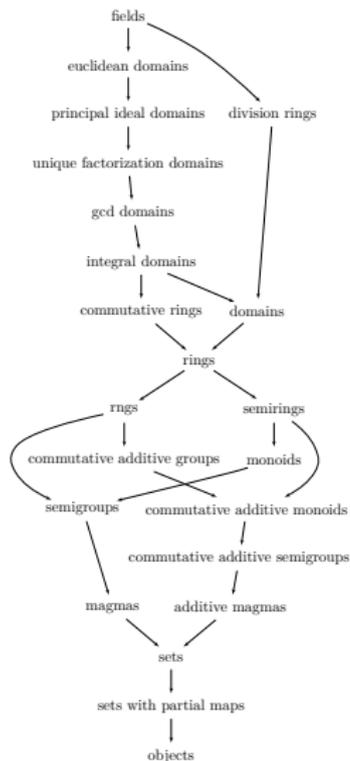


FIGURE 5.1 – Un petit morceau du graphe des catégories dans Sage.

- Il est très important que tout soit objet.
- Classes abstraites, etc...
- Conversion et coercion : une des parties les moins faciles de l'affaire.

Bien entendu, tout doit être le plus possible transparent pour l'utilisateur standard.

- Il est très important que tout soit objet.
- Classes abstraites, etc...
- Conversion et coercion : une des parties les moins faciles de l'affaire.

Bien entendu, tout doit être le plus possible transparent pour l'utilisateur standard.

Python : certainement un des meilleurs langages existants pour cela.

Avantages :

- pas de définitions d'un nouveau langage.
- quantité énorme d'interfaces et d'outils existants.

et puis... apprendre Sage, c'est apprendre Python !

Sage contient 90 packages standard (+ optionnels, + bases de données).

Cython est utilisé intensivement dans Sage :

- Pour réaliser des interfaces avec les bibliothèques existantes.
- Pour pallier la lenteur de Python.

Pas de Cython, pas de Sage !

Un lien fort avec \LaTeX .

- Chaque objet fournit une représentation \LaTeX .
- SageTeX : appeler Sage depuis \LaTeX ! (et incorporer résultats, graphiques, etc...).

Le notebook

Apparu bien avant le notebook de Ipython.

Le notebook

Apparu bien avant le notebook de Ipython.

Nombreux ingrédients : Mathjax, Jinja2, etc..

Représentation $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$: jolies sorties dans le navigateur grâce à Mathjax !

Le notebook

Apparu bien avant le notebook de Ipython.

Nombreux ingrédients : Mathjax, Jinja2, etc..

Représentation \LaTeX : jolies sorties dans le navigateur grâce à Mathjax !

Permet de réaliser des serveurs Sage (sécurisés) ; exemples :

- Université Lyon1 (Dpt. de Mathématiques et Dpt. de Mécanique).
- Serveurs publics :
 - <https://cloud.sagemath.com/>
(accès Android possible -limité-)

Le notebook

Apparu bien avant le notebook de Ipython.

Nombreux ingrédients : Mathjax, Jinja2, etc..

Représentation \LaTeX : jolies sorties dans le navigateur grâce à Mathjax !

Permet de réaliser des serveurs Sage (sécurisés) ; exemples :

- Université Lyon1 (Dpt. de Mathématiques et Dpt. de Mécanique).
- Serveurs publics :
 - <https://cloud.sagemath.com/>
(accès Android possible -limité-)

Bientôt (dès la prochaine release) : `ipython notebook=sage`

Installation, Packaging

- Systèmes Unix (distributions Linux, MacOS). Windows avec une machine virtuelle.
- Binaires disponibles, PPA Ubuntu.
- Compilation : *presque tout* est compilé depuis les sources : Python, Lisp, et les bibliothèques. On a juste besoin du minimum : autotools, C, C++ et fortran.

Un petit air de Debian : notion de packages .spkg

Nouvelles versions fréquentes (\simeq 2 mois).

```
cd sage
du -sk .
4912644 .
Bigre !
```


La réception de Sage

- Enthousiasme plus ou moins grand en enseignement...

La réception de Sage

- Enthousiasme plus ou moins grand en enseignement...
- Mais plutôt bien reçu chez les étudiants.

La réception de Sage

- Enthousiasme plus ou moins grand en enseignement...
- Mais plutôt bien reçu chez les étudiants.
- Nette progression dans la recherche Universitaire (en maths).

La réception de Sage

- Enthousiasme plus ou moins grand en enseignement...
- Mais plutôt bien reçu chez les étudiants.
- Nette progression dans la recherche Universitaire (en maths).
- Sage est disponible à l'agrégation de mathématiques.

La réception de Sage

- Enthousiasme plus ou moins grand en enseignement...
- Mais plutôt bien reçu chez les étudiants.
- Nette progression dans la recherche Universitaire (en maths).
- Sage est disponible à l'agrégation de mathématiques.

Performances :

- Meilleur que les concurrents non libres dans de nombreux domaines (voir les benchmarks sur le site du projet).

La réception de Sage

- Enthousiasme plus ou moins grand en enseignement...
- Mais plutôt bien reçu chez les étudiants.
- Nette progression dans la recherche Universitaire (en maths).
- Sage est disponible à l'agrégation de mathématiques.

Performances :

- Meilleur que les concurrents non libres dans de nombreux domaines (voir les benchmarks sur le site du projet).
- L'alternative à Matlab est subordonnée au développement de Scipy... et la culture Matlab est tellement partagée...

La réception de Sage

- Enthousiasme plus ou moins grand en enseignement...
- Mais plutôt bien reçu chez les étudiants.
- Nette progression dans la recherche Universitaire (en maths).
- Sage est disponible à l'agrégation de mathématiques.

Performances :

- Meilleur que les concurrents non libres dans de nombreux domaines (voir les benchmarks sur le site du projet).
- L'alternative à Matlab est subordonnée au développement de Scipy... et la culture Matlab est tellement partagée...

et pour finir, un peu de pub :



A. Casamayou, N. Cohen, G. Connan, T. Dumont, L. Fousse, F. Maltey,
M. Meulien, M. Mezzarobba, C. Pernet, N. M. Thiéry, P. Zimmermann.
CC-BY-SA. mai 2013, ISBN : 9781481191043.

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