

# SymPy une bibliothèque pure Python pour le calcul symbolique

Kamel Derouiche

PyConFr'14 - Lyon,  
25-28 Octobre 2014

Algerian IT Security Group &  
A<sup>2</sup>DEMTI



## Introduction

Contexte et motivation

SymPy c'est quoi?

Pourquoi SymPy?

Quelques logiciels qui utilisent SymPy

Différence entre SymPy et Sagemath

Numérique

Simplification

Fonction Numérique

Polynômes

SymPy et Latex

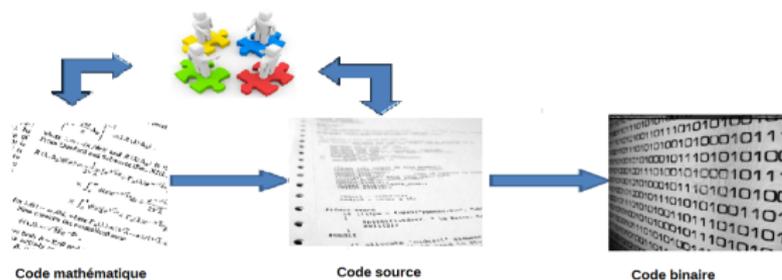
Graphe 3D

SymPy Gamma

Conclusion

Questions

## Contexte et motivation



- ▶ Les *softwares mathématiques*, sont des outils pour l'expérimentation, la modélisation et problème-solving
- ▶ Comment **utiliser** un logiciel de calcul formel pour **enseigner** dans les lycées, universités et ainsi **faire** de la recherche en mathématiques?

## SymPy c'est quoi ?



SymPy

- ▶ Lancé par Ondřej Čertík en 2005
- ▶ En production depuis 2007
- ▶ Objectif : devenir un CAS (Computer Algebra System) complet
- ▶ License BSD
- ▶ Installation simple : la seule dépendance requise est le langage Python.

## SymPy c'est quoi ?

- ▶ 100 % Python
- ▶ Multiplateforme
- ▶ Regroupe divers champs des mathématiques
- ▶ **Trois en un** :
  - ▶ Une bibliothèque : Il y a ceux qui préfère écrire des réponses sous forme de scripts!
  - ▶ Interactif : isympy via IPython en ligne de commande
  - ▶ Interface on-web : SymPy Gamma, SymPyLive

## Pourquoi choisir SymPy ?

ils existe beaucoup d'outils et bibliothèques en calcul formel : elle sont soit propriétaire ( Mathematica, Maple, Magma,...) où Libre/Open source (OpenAxiom, Ginac, Maxima, Pari, Singular, Yacas, Cantor, . . .). Pourquoi? :

- ▶ Éviter d'apprendre différent langage
- ▶ Licence restreinte
- ▶ Pas de graphiques pour certain (OpenAxiom, Pari,..)
- ▶ Communautés de développeurs restreinte

## Quelques logiciels qui utilisent SymPy

`odepy` Numerical ODE solvers in Python

`hamilton` Visualize and control mechanic systems through solving these systems

`grpy` Small GR-oriented package

`ncpol2sdpa` Sparse Semidefinite Programming Relaxations

`algebraic` Algebraic modeling system for Python

`aesthete` Integrated mathematics environment

`nipy` Analysis of neuroimaging data

`apply` Open source computational probability software

`sagemath`

## Différence entre SymPy et sagemath

- ▶  $\text{SymPy} \subseteq \text{Sagemath}^1$
- ▶ Sagemath dépend de différents composants, SymPy dépend d'un seul composant(Python)
- ▶ SymPy est léger, Sagemath est volumineux  $\simeq 1.x\text{Go}$

## Différence entre SymPy et Sagemath

```
In [1] : (x+1)^3
```

```
-----  
TypeError Traceback (most recent call last)
```

```
<ipython-input-78-0960a9242ed1> in <module>()  
----> 1 (x+1)^3
```

```
TypeError : unsupported operand type(s) for ^ : 'Add' and 'int'
```

## Quelques modules SymPy

- `concrete` symbolic products and summations
  - `core` Basic, Add, Mul, Pow, Function, . . .
- `matrices` Matrix class, orthogonalization etc.
- `ntheory` number theoretical functions
- `categories` category theory module
- `diffgeom` differential Geometry Module
- `parsing` Mathematica and Maxima parsers
- `physics` physical units, Pauli matrices
  - `series` compute limits and truncated series
- `printing` pretty-printing, code generation
- `simplify` rewrite expressions in other forms
- `utilities` test framework, compatibility stuff
- ...

## Numérique

- ▶ Déclarer ces variables

```
In[2] : t  
(...)  
NameError : name 't ' is not defined  
  
In[3] : t = symbols('t'); type(t)  
Out[4] : sympy.core.symbol.Symbol  
  
In[5] : x, y = Symbols('x y'); expr = x**2 + y**2
```

- ▶ Valeur numérique

```
In[7] : math.sqrt(20)  
Out[8] : 4.47213595499958  
  
In[9] : float(sqrt(20))  
Out[10] : 4.47213595499958
```

- ▶ Valeur symbolique

```
In[11] : import sympy  
In[12] : sympy.sqrt(20)  
Out[13] : 2√(5)
```

# Simplification

## Modèle pour QCM

► factor :

```
In[14] : expr = (y*x**4 + 6*y*2)*x*y**2  
In[15] : factor(expr)  
Out[16] :  
x.y3.(x4 + 12)
```

► expand :

```
In[17] : expr = (y*x**4 + 6*y*2)*x*y**2  
In[18] : factor(expr)  
Out[19] :  
x.y3.(x4 + 12)
```

► collect

```
In[20] : expr = (y*x**4 + 6*y*2)*x*y**2  
In[21] : collect(expr, x)  
Out[22]  
x.y3.(x4 + 12)
```

▶ Limite :

```
In[23] : limit(exp(1/x)/1-sin(x), x, 0)
```

```
Out[24] : ∞
```

▶ Dérivation :

```
In[25] : diff((x**y+y**2),x, y)
```

```
Out[26] :  $\frac{x^y(y \cdot \log(x)+1)}{x}$ 
```

▶ Intégration :

```
In[27] : integrate((exp(x)/1-x)
```

```
Out[28] :
```

```
 $-\frac{x^2}{2} + \exp(x)$ 
```

▶ Développement en série :

```
In[29] : (exp(1/x)/1-sin(x)).series(x, 0, 7)
```

```
In[30] :  $e^{\frac{1}{x}} - x + \frac{x^3}{6} - \frac{x^5}{120} + o(x^3)$ 
```

## Polynômes

```
In[31] : P = x**3+4*x**2+5*x+2
```

```
Out[32] :  $x^3 + 4x^2 + 5x + 2$ 
```

```
In[33] : solve(P, x)
```

```
Out[34] : [-2, -1]
```

```
In[35] : factor(P, domain='QQ')
```

```
(x + 1)2(x + 2)
```

```
In[36] : factor(P, domain='RR')
```

```
Out[37] : 1.0(1.0.x + 1.0)2.(1.0.x + 2.0)
```

```
In[38] : f = (x**2 + 2*x + 3)/(x**3 + 4*x**2 + 5*x + 2)
```

```
In[39] : apart(f)
```

```
Out[40] :  $\frac{2}{x+2} - \frac{2}{x+1} + \frac{2}{(x+1)^2}$ 
```

```
In[41] : cancel(f)
```

## Impression en mode Latex

```
In[42] : from sympy import integral, latex
In[43] : from sympy.abc import *

In[44] : latex(x**3-x*1)
Out[45] : 'x^3 - x'

In[46] : latex(x**3-x*1), mode='inline'
Out[47] : '$x^3 - x$'

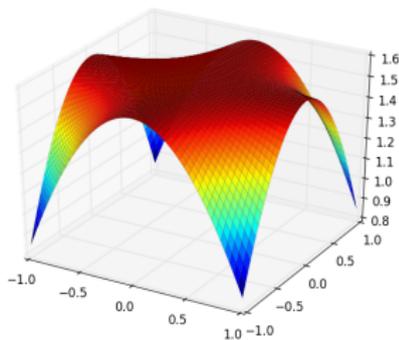
In[48] : latex(x**3-x*1), mode='equation')

In[50] : latex(Integral(x**3-x*1))
```

## Graphe 3D

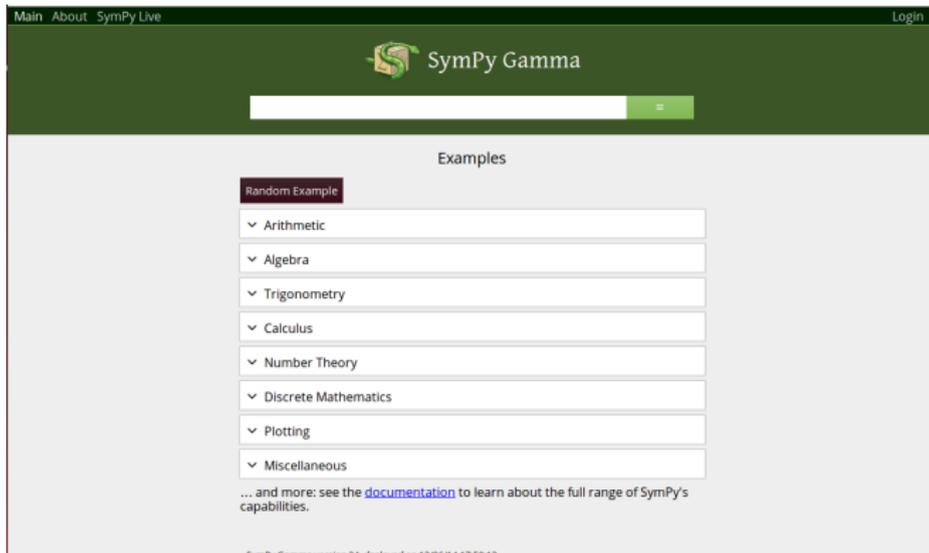
- ▶ Les plots avec SymPy dépend de deux bibliothèque Pyglet et matplotlib

```
In[10] : from sympy import exp, pi, I  
In[11] : from sympy.plotting import plot3d  
In[12] : plot3d((pi/2)*exp(I*x*y), (x, -1, 1), (y, -1, 1), nb_of_points_x=100,  
nb_of_points_y=100)
```



# L'interface SymPy Gamma

Console web interactif



The screenshot displays the SymPy Gamma web interface. At the top, there is a dark green header with navigation links for 'Main', 'About', and 'SymPy Live' on the left, and a 'Login' link on the right. The central part of the header features the SymPy Gamma logo and a search bar with a green submit button. Below the header, the main content area is titled 'Examples'. On the left side of this area, there is a vertical list of categories, each with a dropdown arrow: 'Arithmetic', 'Algebra', 'Trigonometry', 'Calculus', 'Number Theory', 'Discrete Mathematics', 'Plotting', and 'Miscellaneous'. A 'Random Example' button is positioned above this list. Below the categories, there is a text prompt: '... and more: see the [documentation](#) to learn about the full range of SymPy's capabilities.' At the bottom of the interface, a small version number is visible: 'SymPy Gamma version 2.6. Download on 13.06.14 13:40:13'.

## Communauté

Contribuer, question, documentation...

- ▶ Adresses utiles :
  - ▶ <https://github.com/sympy/sympy/wiki>
  - ▶ <http://docs.sympy.org/latest/index.html>
  - ▶ <https://twitter.com/SymPy>
- ▶ Contact, liste de diffusions us on our mailing list :
  - ▶ <https://groups.google.com/forum/#!forum/sympy>
- ▶ Récupérer les sources depuis le dépôt :
  - ▶ `git clone https://github.com/sympy/sympy.github.com.git`

## Conclusion

- ▶ Fonctionnalités de base issu de la structure de classe
- ▶ Projet évolue (GoSC, [planet.sympy.org](http://planet.sympy.org))
- ▶ Expérimenter les mathématiques avec Python

## Questions ?

**Contact** : [kamel.derouiche@gmail.com](mailto:kamel.derouiche@gmail.com), @kiaderouiche



**Merci pour votre attention**