Présentation du logiciel SageMath

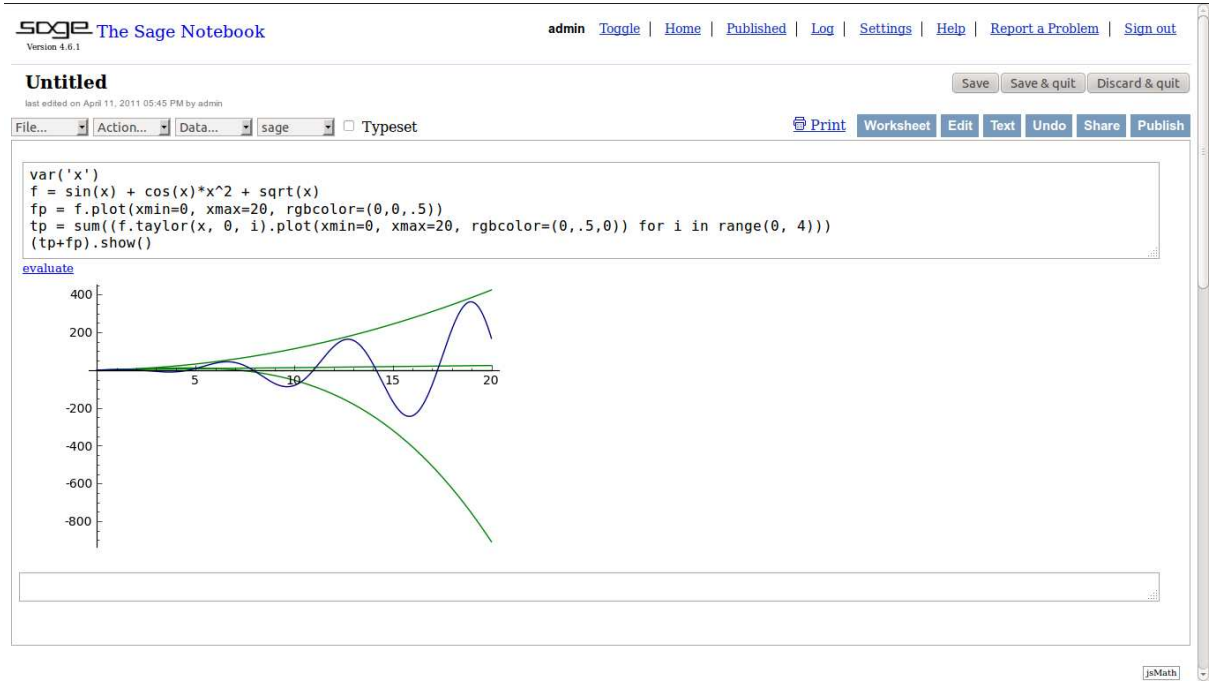
Cécile Armana

Laboratoire de Mathématiques de Besançon
Université de Franche-Comté, France

École CIMPA Kinshasa 2018

Qu'est SageMath ?

Logiciel généraliste de calcul mathématique



Téléchargeable pour être installé sur un ordinateur (Windows, Linux, MacOS) :

`https://www.sagemath.org/`

ou utilisable en ligne depuis un navigateur avec une connection à internet :

`https://cocalc.com/`

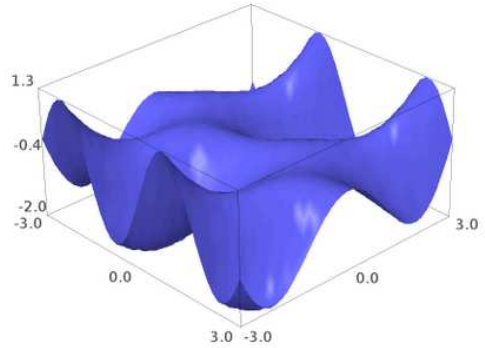
Un logiciel ouvert

Libre et gratuit :

- librement diffusable et utilisable (licence GNU GPL)
- code source librement consultable et modifiable (correction de bugs, ajout de nouvelles fonctionnalités)
- développé par une grande communauté d'utilisateurs

```
var('x,y'); plot3d(sin(x-y)*y*cos(x),(x,-3,3),(y,-3,3))
```

(x, y)



$\neq$ autres logiciels commerciaux de calcul mathématique :
Maple, Mathematica, Magma, Matlab

Plus de 200 développeurs à travers le monde

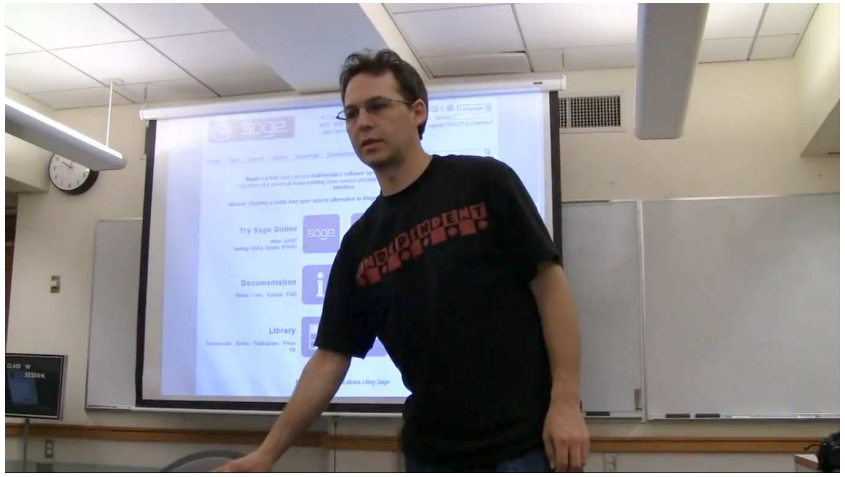
This map shows many contributors of the SageMath project from all around the world.
There are currently 270 in 190 different places.

Map Zoom: Earth - USA (UW, West , East) - Europe - Africa - Asia - S. America - Australia

http://www.sagemath.org/development-map.html

Derrière SageMath

Initié et développé depuis 2005 par William Stein, théoricien des nombres américain et professeur à l'Université de Washington



Software for Algebra and Geometry Experimentation

Que sait-il et que veut-il faire ?

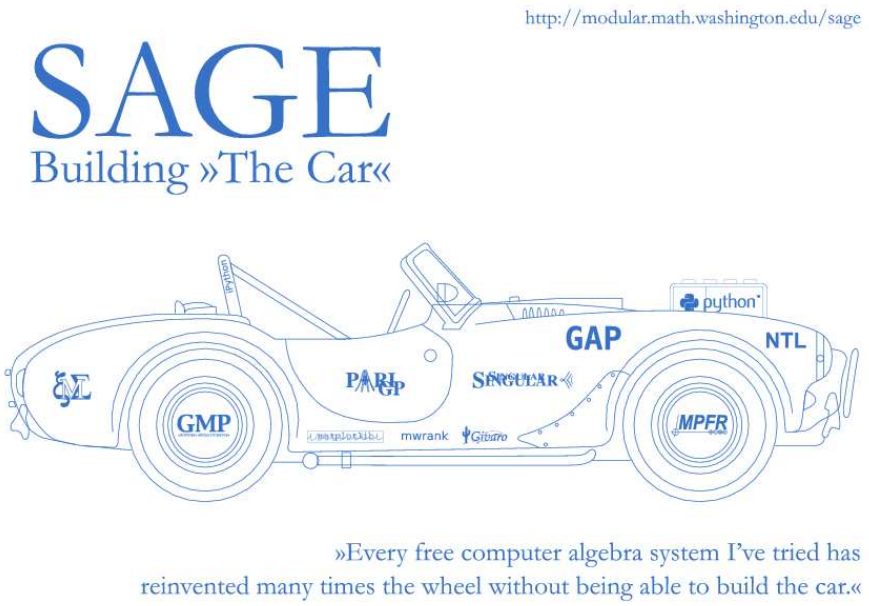
Proposer une alternative libre aux systèmes de calcul formel ou numérique propriétaires en couvrant l'essentiel des mathématiques et de l'informatique théorique :

- algèbre : algèbre linéaire, anneaux, corps, groupes, autres structures algébriques,...
- théorie des nombres
- géométrie algébrique, géométrie arithmétique
- mathématiques discrètes : combinatoire, théorie des graphes, théorie des codes, cryptographie,...
- analyse : fonctions usuelles, développements limités et asymptotiques, tracés,...
- géométrie et topologie
- probabilités et statistiques
- ...

Construire la voiture et non réinventer la roue

Écosystème s'appuyant sur :

- des logiciels libres développés auparavant et plus spécialisés : Axiom, Maxima, PARI/GP, GAP, Singular,...
- un langage de programmation généraliste, répandu et réputé, Python, disposant de bibliothèques de calcul scientifique : NumPy, Sympy,...



À quoi peut servir un logiciel de calcul formel ?

- vérifier des petits calculs
- représenter graphiquement des données
- faire de gros calculs, expérimenter
- produire ou tester des conjectures
- développer de nouveaux algorithmes

```
def verif_goldbach_jusqua(n):
    for m in IntegerRange(4,n+2,2):
        for i in IntegerRange(2,Integer(m/2)+1):
            if i.is_prime() and (m-i).is_prime():
                print i,"+",m-i,"=",m
```

```
verif_goldbach_jusqua(30)
```

```
2 + 2 = 4
3 + 3 = 6
3 + 5 = 8
3 + 7 = 10
5 + 5 = 10
5 + 7 = 12
3 + 11 = 14
7 + 7 = 14
3 + 13 = 16
5 + 11 = 16
5 + 13 = 18
7 + 11 = 18
3 + 17 = 20
7 + 13 = 20
3 + 19 = 22
5 + 17 = 22
11 + 11 = 22
5 + 19 = 24
7 + 17 = 24
11 + 13 = 24
3 + 23 = 26
7 + 19 = 26
13 + 13 = 26
5 + 23 = 28
11 + 17 = 28
7 + 23 = 30
11 + 19 = 30
13 + 17 = 30
```

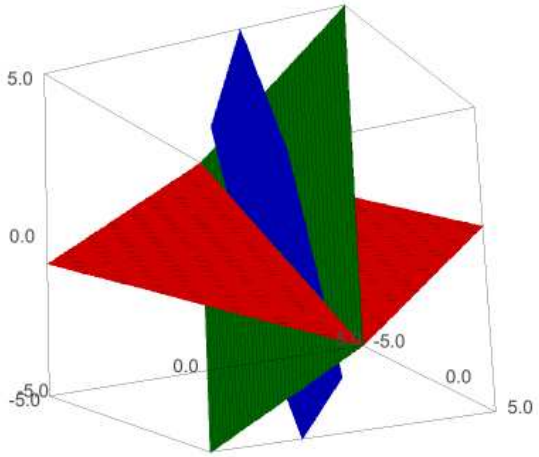
Exemple d'utilisation pour l'enseignement

Étude de l'intersection de trois plans de $\mathbb{R}^3$:

```
x,y,z=var('x,y,z')
f1 = 3*x+2*y+z
f2 = 2*x-3*y+5*z
f3 = x+y
```

```
P1 = implicit_plot3d(f1,(x,-3,3),(y,-5,5),(z,-5,5),color='blue')
P2 = implicit_plot3d(f2,(x,-5,5),(y,-5,5),(z,-5,5),color='red')
P3 = implicit_plot3d(f3,(x,-5,5),(y,-5,5),(z,-5,5),color='green')
```

```
show(P1+P2+P3)
```



Exemple d'utilisation pour la recherche (1/2)

V. Pasol et A. Popa, *Modular forms and period polynomials*,

Proc. Lond. Math. Soc. 107/4 (2013), 713-743.

4. Coboundary polynomials

In this section we show that the space of coboundary polynomials

$$C_w^\Gamma = \{P|(1-S) : P \in V_w^\Gamma \cap \ker(1-T)\}$$

is the radical of the bilinear form $\{\cdot,\cdot\}$ on W_w^Γ. The dimension of C_w^Γ equals the dimension of the Eisenstein subspace $\mathcal{E}_k(\Gamma) \subset M_k(\Gamma)$. For $\Gamma = \Gamma_0(N)$, we characterize those N for which $(C_w^\Gamma)^-$ is trivial, namely those N for which the map $\rho^- : S_k(\Gamma) \to (W_w^\Gamma)^-$ is an isomorphism, as in the full level case.

For $\Gamma_0(N)$ it turns out that $(C_w^\Gamma)^-$ is often trivial, in which case $\rho^- : S_k(\Gamma) \to (W_w^\Gamma)^-$ is an isomorphism just like for Γ_1. The following proposition was discovered using SAGE [SG].

Proposition 4.4. *Let $\Gamma = \Gamma_0(N)$. Then $(C_w^\Gamma)^- = \{0\}$ if and only if $N = 2^e N'$ with N' odd square free and $0 \leqslant e \leqslant 3$.*

Proof. From the proof of Lemma 4.2 we identify $(C_w^\Gamma)^-$ with the space $(\mathbb{C}^{e_\infty(\Gamma)})^-$ of vectors $(c_A)_{A\in\overline{\Gamma}\backslash\overline{\Gamma}_1}$ with $c_A = c_{AT}$ and $c_A = -c_{A'}$ (including for $k = 2$).

Assume N does not satisfy the conditions, so there exists $t \geqslant 3$ with $t^2|N$. We claim that $[A] \neq [A']$ for $A = \left(\begin{smallmatrix} x & y \\ t & z \end{smallmatrix}\right) \in \Gamma_1$, so $(C_w^\Gamma)^- \neq \{0\}$. Assuming by contradiction that $\gamma A T^s = A'$ for $\gamma = \left(\begin{smallmatrix} * & * \\ c & d \end{smallmatrix}\right) \in \Gamma$, it follows that $cx + dt = -t, c(y + sx) + d(z + st) = z$. The first equation implies that $t|d + 1$ while the second that $t|d - 1$, a contradiction with $t \geqslant 3$.

Assuming N satisfies the conditions, let $(c_A)_{A\in\overline{\Gamma}\backslash\overline{\Gamma}_1}$ with $c_A = c_{AT}$ and $c_A = -c_{A'}$. Identifying the coset space $\Gamma_0(N)\backslash\Gamma_1$ with $\mathbb{P}^1(\mathbb{Z}/N\mathbb{Z})$, it follows that $c_{(a:b)} = c_{(a:a+b)}$, $c_{(a:b)} = -c_{(-a:b)}$. The second relation implies $c_{(0:1)} = 0$. Let $N = dd'$ and $k \in \mathbb{Z}$, $(k, d) = 1$. We will show that $c_{(d:k)} = 0$. We have:

$$c_{(d:k)} = c_{(d:k+ad)} = -c_{(-d:k)} = -c_{((bd'-1)d:k)}$$

and it is enough to find $a, b \in \mathbb{Z}$ with $(bd' - 1, N) = 1$ and $k \equiv (bd' - 1)(k + ad) \pmod N$. The latter equation can be written $k(bd' - 2) \equiv ad \pmod{dd'}$ and the hypothesis on N ensures that $(d, d')|2$, so that we can find b, u such that $bd' - 2 = du$. Taking $a \equiv ku \pmod{d'}$, it follows that $(d : k + ad) = ((bd' - 1)d : k)$, which implies that $c_{(d:k)} = 0$. It follows that $c_A = 0$ for all $A \in \overline{\Gamma}\backslash\overline{\Gamma}_1$, finishing the proof. $\square$

Exemple d'utilisation pour la recherche (2/2)

Yu. Bilu, P. Parent, M. Rebolledo, *Rational points on* $X_0^+(p^r)$,
Ann. Inst. Fourier 63 (2013), 957-984.

Abstract

We show how the recent isogeny bounds due to Gaudron and Rémond allow to obtain the triviality of $X_0^+(p^r)(\mathbb{Q})$, for $r > 1$ and p a prime exceeding $2 \cdot 10^{11}$. This includes the case of the curves $X_{\text{split}}(p)$. We then prove, with the help of computer calculations, that the same holds true for p in the range $11 \leq p \leq 10^{14}$, $p \neq 13$. The combination of those results completes the qualitative study of such sets of rational points undertook in [4] and [5], with the exception of $p = 13$.

Exemple d'utilisation pour la recherche (2/2)

Yu. Bilu, P. Parent, M. Rebolledo, *Rational points on $X_0^+(p^r)$*,
Ann. Inst. Fourier 63 (2013), 957-984.

- **Algorithms** : we reproduce here the pseudo-codes of the algorithms described in Section 4.2. The original codes have been written with Sage [29]. We used the hilbert_class_polynomial function to compute H_{-D} and the crt function to apply Chinese Remainder Theorem.

Algorithm, Part I:

bad_discrim_and_primes$(\mathcal{D}, N)$

Require: A list $\mathcal{D}$ of imaginary quadratic discriminants and an integer $N > 1$ (as in Section 4.2).

1: set $L \leftarrow [3, 4, 7, 8, 11, 19, 43, 67, 163]$, Bad $\leftarrow []$, and Good $\leftarrow []$.
2: **for** d in $\mathcal{D}$ **do**
3: set $G \leftarrow H'_{-d}$
4: compute the prime factors $\mathcal{P}_D$ of $r_D := \gcd(\mathrm{Res}(G, H'_{-c^2 D}); c \in [\![2, 7]\!])$
5: **for** p in $\mathcal{P}_D$ **do**
6: **if** $p > 10$ and $p < N + 1$ and p not in Good and p not in Bad **then**
7: **if** $\chi_{-m}(p) = 1$ for all m in L **then**
8: add p to the list Bad
9: **else**
10: add p to the list Good.
11: add d to the list L (and go to step 3 with another d in $\mathcal{D}$).
12: **return** $[L, \text{Bad}]$.

Démo : manipuler des entiers

```
3+5
```

8

```
2^15
```

32768

```
factor(2018)
```

$2 \cdot 1009$

```
2018.factor()
```

$2 \cdot 1009$

```
1009.is_prime()
```

True

L'ensemble des entiers de la forme 7x avec x compris entre 0 et 49 et x congru à 1 modulo 3 s'obtient comme la liste suivante :

```
[7*x for x in range(50) if x%3 == 1]
```

$[7, 28, 49, 70, 91, 112, 133, 154, 175, 196, 217, 238, 259, 280, 301, 322, 343]$

Démo : manipuler équations et expressions symboliques

```
x,y = var('x,y')
factor(x^3-sin(y)^3)
```

$$\left(x^2 + x\sin(y) + \sin(y)^2\right)(x - \sin(y))$$

```
a,b,c = var('a,b,c')
solve(a*x^2+b*x+c == 0,x)
```

$$\left[x = -\frac{b + \sqrt{b^2 - 4\,ac}}{2\,a}, x = -\frac{b - \sqrt{b^2 - 4\,ac}}{2\,a}\right]$$

```
integrate(sin(x)*tan(x),x)
```

$$\frac{1}{2}\log(\sin(x) + 1) - \frac{1}{2}\log(\sin(x) - 1) - \sin(x)$$

```
x/(x^2+x)
```

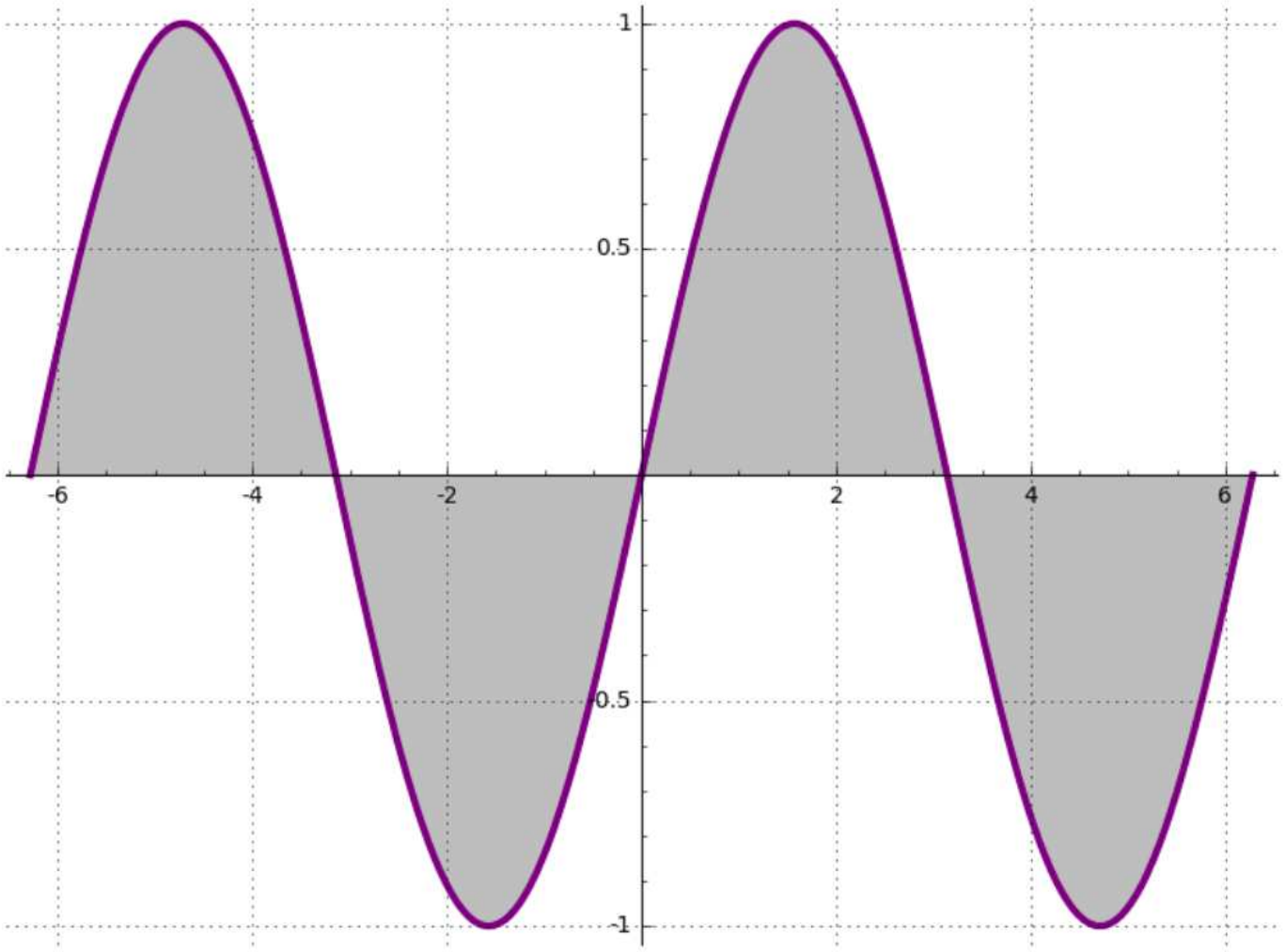
$$\frac{x}{x^2 + x}$$

```
(x/(x^2+x)).simplify_full()
```

$$\frac{1}{x + 1}$$

Démo : représentation graphique (2D)

```
plot(sin(x),(x,-2*pi,2*pi),thickness=3,color='purple',fill=True,gridlines=True)
```



Démo : manipulations dans l'anneau $\mathbb{Z}$

```
ZZ
```

$\mathbb{Z}$

```
1 in ZZ
```

True

```
ZZ.is_finite()
```

False

```
gcd(27,141)
```

3

```
xgcd(27,141)
```

$(3, 21, -4)$

```
27.quo_rem(3)
```

$(9, 0)$

Démo : manipulations dans un anneau $\mathbb{Z}/n\mathbb{Z}$

```
R = IntegerModRing(8)
R; R.list()
```

$\mathbf{Z}/8\mathbf{Z}$

$[0, 1, 2, 3, 4, 5, 6, 7]$

```
R.is_finite(), R.is_field()
```

$(\mathrm{True}, \mathrm{False})$

```
R(27)
```

3

```
R(27).parent()
```

$\mathbf{Z}/8\mathbf{Z}$

```
R(2).additive_order()
```

4

```
R(2).multiplicative_order()
```

```
Traceback (click to the left of this block for traceback)
...
ArithmeticError: multiplicative order of 2 not defined since it is
not a unit modulo 8
```

```
R(3).multiplicative_order()
```

2

Démo : manipulations dans un anneau de polynômes

```
R.<x> = PolynomialRing(QQ); R
```

$\mathbf{Q}[x]$

```
R.variable_name(); R.base_ring()
```

x

$\mathbf{Q}$

```
P = x^4-2*x^2+1; P
```

$x^4 - 2x^2 + 1$

```
P in R
```

True

```
P.is_irreducible(); P.factor()
```

False

$(x-1)^2 \cdot (x+1)^2$

```
P.degree(); P(2)
```

4

9

Démo : manipulations dans un anneau quotient

```
F = GF(3); F
```

$\mathbf{F}_3$

```
R.<x> = PolynomialRing(F)
(x^3-x^2+1).is_irreducible()
```

True

```
S.<y> = R.quotient(x^3-x^2+1); S
```

Univariate Quotient Polynomial Ring in y over Finite Field of size 3 with modulus x^3 + 2*x^2 + 1

```
y^3-y^2+1
```

0

```
S.is_field()
```

True

```
S.cardinality()
```

27

```
S.random_element()
```

$2y^2+1$

```
y^(27)
```

y

```
y^(10^5000)
```

y^2+y+1

Démo : de l'aide ! (1/2)

Complétion (touche TAB) et menu contextuel

```
R = ZZ['x']; R
```

$\mathbf{Z}[x]$

```
P = x^2+x+1
```

```
P.c
```

evaluate

```
P.canonicalize_radical
P.cartesian_product
P.category
P.coeff
P.coefficient
P.coefficients
P.coeffs

P.collect                P.cos
P.collect_common_factors P.cosh
P.combine                P.csgn
P.conjugate
P.content
P.contradicts
P.convert
```

Démo : de l'aide ! (2/2)

Aide à propos d'une commande : ?

```
P = x^2+x+1
```

```
P.coefficients?
```

File: /home/cecile/.sage/src/sage/symbolic/expression.pyx

Type: <type 'builtin_function_or_method'>

Definition: P.coefficients(x=None, sparse=True)

Docstring:

Return the coefficients of this symbolic expression as a polynomial in x.

INPUT:

- x – optional variable.

OUTPUT:

Depending on the value of `sparse`,

- A list of pairs `(expr, n)`, where `expr` is a symbolic expression and n is a power (`sparse=True`, default)
- A list of expressions where the n-th element is the coefficient of x^n when self is seen as polynomial in x (`sparse=False`).

EXAMPLES:

```
sage: var('x, y, a')
(x, y, a)
sage: p = x^3 - (x-3)*(x^2+x) + 1
sage: p.coefficients()
[[1, 0], [3, 1], [2, 2]]
sage: p.coefficients(sparse=False)
[1, 3, 2]
sage: p = x - x^3 + 5/7*x^5
sage: p.coefficients()
[[1, 1], [-1, 3], [5/7, 5]]
sage: p.coefficients(sparse=False)
[0, 1, 0, -1, 0, 5/7]
sage: p = expand((x-a*sqrt(2))^2 + x + 1); p
-2*sqrt(2)*a*x + 2*a^2 + x^2 + x + 1
sage: p.coefficients(a)
[[x^2 + x + 1, 0], [-2*sqrt(2)*x, 1], [2, 2]]
sage: p.coefficients(a, sparse=False)
[x^2 + x + 1, -2*sqrt(2)*x, 2]
sage: p.coefficients(x)
[[2*a^2 + 1, 0], [-2*sqrt(2)*a + 1, 1], [1, 2]]
sage: p.coefficients(x, sparse=False)
[2*a^2 + 1, -2*sqrt(2)*a + 1, 1]
```

Quelques portes d'entrée

- Le tutoriel Sage
  `http://doc.sagemath.org/html/fr/tutorial/`
- Calcul mathématique avec Sage
  `http://sagebook.gforge.inria.fr/`
- Sage quick reference cards
  `https://wiki.sagemath.org/quickref`
- Plein d'autres ressources :
  `http://www.sagemath.org/help.html`

Crédits illustrations et photos

- Logo SageMath : licence Creative Commons CC BY 3.0. Auteur : The Sage team. Source : `https://commons.wikimedia.org/wiki/File:Sage_logo_new.png`
- Capture d'écran de SageMath : licence Creative Commons CC BY-SA 2.0. Auteur : Sage. Source : `https://commons.wikimedia.org/wiki/File:Sage_-_plot.png`
- Carte des développeurs de SageMath : Google/SageMath. Source : `http://www.sagemath.org/development-map.html`
- Photo de William Stein : licence CC BY 3.0. Auteur : William A. Stein. Source : `https://en.wikipedia.org/wiki/File:William_A._Stein.png`
- Sage-car. Auteur : Martin Albrecht.
- Couverture de Calcul mathématique avec Sage : licence Creative Commons CC BY-SA 3.0. Auteurs : A. Casamayou, N. Cohen, G. Connan, T. Dumont, L. Fousse, F. Maltey, M. Meulien, M. Mezzarobba, C. Pernet, N. M. Thiéry, P. Zimmermann. Source : `http://sagebook.gforge.inria.fr/`