

Sage Quick Reference (Basic Math)

Peter Jipsen, version 1.1

latest version at wiki.sagemath.org/quickref

GNU Free Document License, extend for your own use

But : relier les notations standard aux commandes Sage

Interface web (et interface texte)

Pour évaluer une cellule : <shift-enter>

`com<tab>` essaye de compléter la *commande*

`commande?<tab>` montre la documentation

`commande??<tab>` montre la source

`a.<tab>` montre toutes les méthodes pour l'objet `a`

`search_doc('chaîne ou regexp')` cherche dans la doc.

`search_src('chaîne ou regexp')` cherche dans les sources

`lprint()` bascule le mode sortie LaTeX

`version()` donne la version de Sage

Insérer une cellule : cliquer sur la ligne bleue

Supprimer une cellule : supprimer le contenu puis `backspace`

Types numériques

Entiers : $\mathbb{Z} = \mathbb{ZZ}$ par ex. `-2 -1 0 1 10^100`

Rationnels : $\mathbb{Q} = \mathbb{QQ}$ par ex. `1/2 1/1000 314/100 -42`

Décimaux : $\mathbb{R} \approx \mathbb{RR}$ par ex. `.5 0.001 3.14 -42.`

Complexes : $\mathbb{C} \approx \mathbb{CC}$ par ex. `1+i 2.5-3*i`

Constantes et fonctions de base

Constantes : $\pi = \text{pi}$ $e = \text{e}$ $i = \text{i}$ $\infty = \text{oo}$

Approximation : `pi.n(digits=18) = 3.14159265358979324`

Fonctions : `sin cos tan sec csc cot sinh cosh tanh sech csch coth log ln exp`

$ab = \text{a*b}$ $\frac{a}{b} = \text{a/b}$ $a^b = \text{a^b}$ $\sqrt{x} = \text{sqrt}(x)$

$\sqrt[n]{x} = \text{x}^{1/n}$ $|x| = \text{abs}(x)$ $\log_b(x) = \text{log}(x,b)$

Variables symboliques : `t,u,v,y,z = var('t u v y z')`

Définir une fonction : par ex. $f(x) = x^2$ `f(x)=x^2`

ou `f=lambda x: x^2` ou `def f(x): return x^2`

Opérations sur les expressions

`factor(...)` `expand(...)` `(...).simplify(...)`

Équations symboliques : `f(x)==g(x)`

`_` est le résultat précédent

Résoudre $f(x) = g(x)$: `solve(f(x)==g(x),x)`

`solve([f(x,y)==0, g(x,y)==0], x,y)`

`find_root(f(x), a, b)` trouve $x \in [a, b]$ t.q. $f(x) \approx 0$

$\sum_{i=k}^n f(i) = \text{sum}([f(i) \text{ for } i \text{ in } [k..n]])$

$\prod_{i=k}^n f(i) = \text{prod}([f(i) \text{ for } i \text{ in } [k..n]])$

Calcul différentiel et intégral

$\lim_{x \rightarrow a} f(x) = \text{limit}(f(x), x=a)$

$\lim_{x \rightarrow a^-} f(x) = \text{limit}(f(x), x=a, \text{dir}='minus')$

$\lim_{x \rightarrow a^+} f(x) = \text{limit}(f(x), x=a, \text{dir}='plus')$

$\frac{d}{dx}(f(x)) = \text{diff}(f(x), x)$

$\frac{\partial}{\partial x}(f(x, y)) = \text{diff}(f(x, y), x)$

Dériver : `diff = differentiate = derivative`

$\int f(x)dx = \text{integral}(f(x), x)$

Intégrer : `integral = integrate`

$\int_a^b f(x)dx = \text{integral}(f(x), x, a, b)$

Dev. de Taylor, ordre n en a : `taylor(f(x), x, a, n)`

Graphiques dans le plan

`line([(x1, y1), ..., (xn, yn)], options)`

`polygon([(x1, y1), ..., (xn, yn)], options)`

`circle((x, y), r, options)`

`text("txt", (x, y), options)`

options comme dans `plot.options`, par ex. `thickness=pixel`,

`rgbcolor=(r, g, b)`, `hue=h` avec $0 \leq r, b, g, h \leq 1$

utiliser l'option `figsize=[w, h]` pour ajuster le rapport largeur/hauteur

`plot(f(x), xmin, xmax, options)`

`parametric_plot((f(t), g(t)), tmin, tmax, options)`

`polar_plot(f(t), tmin, tmax, options)`

pour combiner : `circle((1, 1), 1)+line([(0, 0), (2, 2)])`

`animate(liste d'objets graphiques, options).show(delay=20)`

Graphiques dans l'espace

`line3d([(x1, y1, z1), ..., (xn, yn, zn)], options)`

`sphere((x, y, z), r, options)`

`tetrahedron((x, y, z), size, options)`

`cube((x, y, z), size, options)`

options par ex. `aspect_ratio=[1, 1, 1]` `color='red'` `opacity`

`plot3d(f(x, y), [xb, xe], [yb, ye], options)`

ajouter l'option `plot_points=[m, n]` ou utiliser `plot3d_adaptive`

`parametric_plot3d((f(t), g(t), h(t)), [tb, te], options)`

`parametric_plot3d((f(u, v), g(u, v), h(u, v)), [ub, ue], [vb, ve], options)`

utiliser + pour combiner des objets graphiques

Math. discrètes

Partie entière $\lfloor x \rfloor = \text{floor}(x)$ $\lceil x \rceil = \text{ceil}(x)$

Reste de la division de n par $k = n\%k$ $k|n$ ssi $n\%k==0$

$n! = \text{factorial}(n)$ $\binom{x}{m} = \text{binomial}(x, m)$

$\phi = \text{golden_ratio}$ $\phi(n) = \text{euler_phi}(n)$

Chaînes : `s = 'Salut' = "Salut" = ""+"Sa"+"lut"`

`s[0]='S' s[-1]='t' s[1:3]='al' s[3:]='ut'`

Listes : par ex. `[1, 'Salut', x] = []+[1, 'Salut']+[x]`

Tuples : par ex. `(1, 'Salut', x)` (non mutable)

Ensembles : $\{1, 2, 1, a\} = \text{Set}([1, 2, 1, 'a'])$ ($= \{1, 2, a\}$)

Création de liste \approx notation ensembliste, par ex.

$\{f(x) : x \in X, x > 0\} = \text{Set}([f(x) \text{ for } x \text{ in } X \text{ if } x > 0])$

Algèbre linéaire

$\begin{pmatrix} 1 \\ 2 \end{pmatrix} = \text{vector}([1, 2])$, $\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} = \text{matrix}([[1, 2], [3, 4]])$

$\begin{vmatrix} 1 & 2 \\ 3 & 4 \end{vmatrix} = \text{det}(\text{matrix}([[1, 2], [3, 4]]))$

$Av = A*v$ $A^{-1} = A^{-1}$ $A^t = A.\text{transpose}()$

méthodes: `nrows()` `ncols()` `nullity()` `rank()` `trace()`...

nbr de colonnes, nbr de lignes, dim. noyau, rang, trace

Modules et paquetages

`from nom_module import *` (beaucoup sont préchargés)

`calculus coding combinat crypto functions games`

`geometry graphs groups logic matrix numerical plot`

`probability rings sets stats`

`sage.nom_module.all.<tab>` montre les commandes

Paquetages standards : Maxima GP/PARI GAP Singular R ...

Paquetages opt. : Biopython Fricas(Axiom) Gnuplot ...

`%nom_paquetage` pour charger

`time commande` pour montrer la durée du calcul