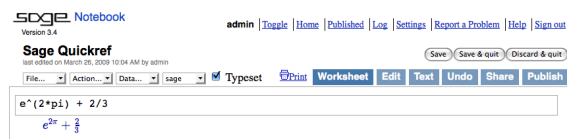


Sage Guía de Referencia Rápida

William Stein (baseda en el trabajo de P. Jipsen)
GNU Free Document License, extend for your own use
Adaptación al español : Javier Honrubia González

Notebook



Evaluar celda: <shift-enter>

Evaluar celda creando una nueva: <alt-enter>

Dividir celda: <control-; >

Unir celdas: <control-backspace>

Insertar celda matemática: <click> en la línea azul entre celdas

Insertar celda de texto/HTML: <shift-click> en la línea azul entre celdas

Borrar celda: borrar el contenido y después <backspace>

Línea de comandos

`com<tab>` completa *comando*

`*bar*?` lista de comandos que contienen “bar”

`command?<tab>` muestra la documentación del comando

`command??<tab>` muestra el código fuente del comando

`a.<tab>` muestra los métodos del objeto **a** (más: `dir(a)`)

`a._<tab>` muestra los métodos ocultos del objeto **a**

`search_doc("cadena o regexp")` búsqueda en el texto de la documentación

`search_src("cadena o regexp")` búsqueda en el código fuente

_ último resultado

Números

Enteros: $\mathbf{Z} = \mathbb{Z}$ p.ej. -2 -1 0 1 10^{100}

Racionales: $\mathbf{Q} = \mathbb{Q}$ p.ej. 1/2 1/1000 314/100 -2/1

Reales: $\mathbf{R} \approx \mathbb{R}$ p.ej. .5 0.001 3.14 1.23e10000

Complejos: $\mathbf{C} \approx \mathbb{C}$ p.ej. $\mathbb{C}(1,1)$ $\mathbb{C}(2.5,-3)$

Doble precisión: RDF and CDF p.ej. $\text{CDF}(2.1,3)$

Módulo n : $\mathbf{Z}/n\mathbf{Z} = \mathbb{Z}_{\text{mod}}$ p.ej. $\text{Mod}(2,3)$ $\mathbb{Z}_{\text{mod}}(3)(2)$

Cuerpos finitos: $\mathbf{F}_q = \mathbb{F}_q$ p.ej. $\mathbb{F}_3(3)(2)$ $\mathbb{F}_9("a").0$

Polinomios: $R[x,y]$ p.ej. $\mathbb{S}.\langle x,y \rangle = \mathbb{Q}\langle x^2+y^3 \rangle$

Series: $R[[t]]$ p.ej. $\mathbb{S}.\langle t \rangle = \mathbb{Q}[[t]]$ $1/2+2*t+0*(t^2)$

Números p -ádicos: $\mathbf{Z}_p \approx \mathbb{Z}_p$, $\mathbf{Q}_p \approx \mathbb{Q}_p$ p.ej. $2+3*5+0*(5^2)$

Cierre algebraico: $\overline{\mathbf{Q}} = \mathbb{Q}\overline{}$ p.ej. $\mathbb{Q}\overline{(2^{1/5})}$

Interválo aritmético: RIF p.ej. `sage: RIF((1,1.00001))`

Campo numérico: $\mathbf{R}.\langle x \rangle = \mathbb{Q}\langle x \rangle$; $\mathbf{K}.\langle a \rangle = \text{NumberField}(x^3+x+1)$

Aritmética

$ab = \mathbf{a*b}$ $\frac{a}{b} = \mathbf{a/b}$ $a^b = \mathbf{a^b}$ $\sqrt{x} = \mathbf{sqrt(x)}$
 $\sqrt[n]{x} = \mathbf{x^(1/n)}$ $|x| = \mathbf{abs(x)}$ $\log_b(x) = \mathbf{log(x,b)}$

Sumas: $\sum_{i=k}^n f(i) = \mathbf{sum(f(i) for i in (k..n))}$

Productos: $\prod_{i=k}^n f(i) = \mathbf{prod(f(i) for i in (k..n))}$

Constantes y funciones

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

$\phi = \mathbf{golden_ratio}$ $\gamma = \mathbf{euler_gamma}$

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

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

Función en Python `def f(x): return x^2`

Funciones interactivas

Escribe `@interact` antes de la función (las variables determinan los controles)

`@interact`

```
def f(n=[0..4], s=(1..5), c=Color("red")):  
    var("x"); show(plot(sin(n+x^s), -pi, pi, color=c))
```

Expresiones simbólicas

Define nuevas variables simbólicas: `var("t u v y z")`

Función simbólica: p.ej. $f(x) = x^2$ `f(x)=x^2`

Relaciones: `f==g f<=g f>=g f<g f>g`

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

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

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

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

Cálculo

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

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

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

`diff = differentiate = derivative`

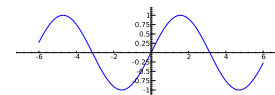
$\int f(x)dx = \mathbf{integral(f(x), x)}$

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

$\int_a^b f(x)dx \approx \mathbf{numerical_integral(f(x), a, b)}$

Polinomio de Taylor, grado n en a : `taylor(f(x), x, a, n)`

Gráficos 2D



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

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

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

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

opciones están en `plot.options`, p.ej. `thickness=pixel`,

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

`show(gráfico, opciones)`

usa `figsize=[w,h]` para ajustar tamaño

usa `aspect_ratio=número` para ajustar la relación de aspecto

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

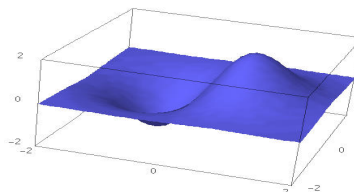
`parametric_plot((f(t),g(t)), (t, tmin, tmax), opciones)`

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

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

`animate(lista de gráficos, opciones).show(delay=20)`

Gráficos 3D



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

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

`text3d("txt", (x,y,z), opciones)`

`tetrahedron((x,y,z), tamaño, opciones)`

`cube((x,y,z), tamaño, opciones)`

`octahedron((x,y,z), tamaño, opciones)`

`dodecahedron((x,y,z), tamaño, opciones)`

`icosahedron((x,y,z), tamaño, opciones)`

`plot3d(f(x,y), (x,xb,xe), (y,yb,ye), opciones)`

`parametric_plot3d((f,g,h), (t,tb,te), opciones)`

`parametric_plot3d((f(u,v),g(u,v),h(u,v)),
(u,ub,ue), (v,vb,ve), opciones)`

opciones: `aspect_ratio=[1,1,1]`, `color="red"`

`opacity=0.5`, `figsize=6`, `viewer="tachyon"`

Matemáticas discretas

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

Resto de n dividido por $k = n\%k$ $k|n$ iff $n\%k==0$

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

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

Strings: p.ej. $s = \text{"Hola"} = \text{"Ho"} + \text{'la'}$

$s[0]=\text{"H"}$ $s[-1]=\text{"a"}$ $s[1:3]=\text{"ol"}$ $s[3:]=\text{"a"}$

Listas: p.ej. $[1, \text{"Hola"}, x] = [] + [1, \text{"Hola"}] + [x]$

Tuplas: p.ej. $(1, \text{"Hola"}, x)$ (inmutable)

Conjuntos: p.ej. $\{1, 2, 1, a\} = \text{Set}([1, 2, 1, \text{"a"}]) (= \{1, 2, a\})$

Comprensión de listas \approx notación constructiva, p.ej.

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

Teoría de grafos



Grafo: $G = \text{Graph}(\{0: [1, 2, 3], 2: [4]\})$

Grafo dirigido: $\text{DiGraph}(\text{diccionario})$

Familias de grafos: $\text{graphs.}(\text{tab})$

Invariantes: $G.\text{chromatic_polynomial}()$, $G.\text{is_planar}()$

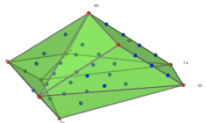
Caminos: $G.\text{shortest_path}()$

Visualizar: $G.\text{plot}()$, $G.\text{plot3d}()$

Automorfismos: $G.\text{automorphism_group}()$,

$G1.\text{is_isomorphic}(G2)$, $G1.\text{is_subgraph}(G2)$

Combinatoria



Secuencias enteras: $\text{sloane_find}(\text{lista})$, $\text{sloane.}(\text{tab})$

Particiones: $P = \text{Partitions}(n)$ $P.\text{count}()$

Combinaciones: $C = \text{Combinations}(\text{lista})$ $C.\text{list}()$

Producto cartesiano: $\text{CartesianProduct}(P, C)$

Tabla: $\text{Tableau}([[1, 2, 3], [4, 5]])$

Palabras: $W = \text{Words}(\text{"abc"})$; $W(\text{"aabca"})$

Posets: $\text{Poset}([[1, 2], [4], [3], [4], []])$

Sistema de raíces: $\text{RootSystem}(\text{"A"}, 3)$

Cristales: $\text{CrystalOfTableaux}(\text{"A"}, 3, \text{shape}=[3, 2])$

Politopos de redes: $A = \text{random_matrix}(ZZ, 3, 6, x=7)$

$L = \text{LatticePolytope}(A)$ $L.\text{npoints}()$ $L.\text{plot3d}()$

Álgebra matricial

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

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

$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix} = \text{matrix}(QQ, 2, 3, [1, 2, 3, 4, 5, 6])$

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

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

Resolver $Ax = v$: $A \setminus v$ or $A.\text{solve_right}(v)$

Resolver $xA = v$: $A.\text{solve_left}(v)$

Forma reducida escalonada: $A.\text{echelon_form}()$

Rango y dimensión del núcleo: $A.\text{rank}()$ $A.\text{nullity}()$

Forma de Hessenberg: $A.\text{hessenberg_form}()$

Polinomio característico: $A.\text{charpoly}()$

Autovalores: $A.\text{eigenvalues}()$

Autovectores: $A.\text{eigenvectors_right}()$ (also left)

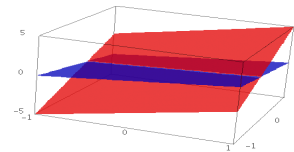
Gram-Schmidt: $A.\text{gram_schmidt}()$

Visualizar: $A.\text{plot}()$

Reducción LLL: $\text{matrix}(ZZ, \dots).\text{LLL}()$

Forma Hermite: $\text{matrix}(ZZ, \dots).\text{hermite_form}()$

Álgebra lineal



Espacio vectorial $K^n = K^n$ p.ej. QQ^3 RR^2 CC^4

Subespacio: $\text{span}(\text{vectors}, \text{campo})$

p.ej., $\text{span}([[1, 2, 3], [2, 3, 5]], QQ)$

Núcleo: $A.\text{right_kernel}()$ (también left)

Suma e intersección: $V + W$ and $V.\text{intersection}(W)$

Base: $V.\text{basis}()$

Matriz de la base: $V.\text{basis_matrix}()$

Restricción de la matriz al subespacio: $A.\text{restrict}(V)$

Coordenadas respecto a la base: $V.\text{coordinates}(\text{vector})$

Métodos numéricos

Paquetes: $\text{import numpy, scipy, cvxopt}$

Minimización: $\text{var}(\text{"x y z"})$

$\text{minimize}(x^2 + x*y^3 + (1-z)^2 - 1, [1, 1, 1])$

Teoría de números

Primos: $\text{prime_range}(n, m)$, is_prime , next_prime

Factorización: $\text{factor}(n)$, $\text{qsieve}(n)$, $\text{ecm.factor}(n)$

Símbolo de Kronecker: $\left(\frac{a}{b}\right) = \text{kronecker_symbol}(a, b)$

Fracciones continuadas: $\text{continued_fraction}(x)$

Números de Bernoulli: $\text{bernoulli}(n)$, $\text{bernoulli_mod_p}(p)$

Curvas elípticas: $\text{EllipticCurve}([a_1, a_2, a_3, a_4, a_6])$

Caracteres de Dirichlet: $\text{DirichletGroup}(N)$

Formas modulares: $\text{ModularForms}(\text{nivel}, \text{peso})$

Símbolos modulares: $\text{ModularSymbols}(\text{nivel}, \text{peso}, \text{signo})$

Módulos Brandt: $\text{BrandtModule}(\text{nivel}, \text{peso})$

Varietades modulares abelianas: $J_0(N)$, $J_1(N)$

Teoría de grupos

$G = \text{PermutationGroup}([[(1, 2, 3), (4, 5)], [(3, 4)]])$

$\text{SymmetricGroup}(n)$, $\text{AlternatingGroup}(n)$

Grupos abelianos: $\text{AbelianGroup}([3, 15])$

Grupos matriciales: GL , SL , Sp , SU , GU , SO , GO

Funciones: $G.\text{syllow_subgroup}(p)$, $G.\text{character_table}()$,

$G.\text{normal_subgroups}()$, $G.\text{cayley_graph}()$

Anillos no conmutativos

Cuaterniones: $Q.<i, j, k> = \text{QuaternionAlgebra}(a, b)$

Álgebra libre: $R.<a, b, c> = \text{FreeAlgebra}(QQ, 3)$

Módulos en Python

$\text{import nombre_del_módulo}$

$\text{nombre_del_módulo.}(\text{tab})$ y $\text{help}(\text{nombre_del_módulo})$

Perfilado y depuración

time comando : muestra información temporal

$\text{timeit}(\text{"comando"})$: cronometra el tiempo de ejecución

$t = \text{cputime}()$; $\text{cputime}(t)$: tiempo de CPU transcurrido

$t = \text{walltime}()$; $\text{walltime}(t)$: tiempo de muro transcurrido

$\%pdb$: activa el depurador interactivo (línea de comando)

$\%prun \text{comando}$: perfila comando (línea de comando)
