SymPy Cheat Sheet

J. Juang, version 1.1

Notebook (and commandline)

Sympy Packages: `from sympy import`
Declare Symbols: `x = symbol('x')`
Declare Variables: `x = var('x')`
Substituting Symbols: *expression*.subs(*old,new*)
e.g. x+x.subs(x,1)=x+1
Sympy Help: `help()`

Numerical types

Integers: $\mathbb{Z}$ = `integer(x)` e.g. `-2 0 1 10^100`
Decimals: $\mathbb{R} \approx$ `float(x)` e.g. `2.58`
Rationals: $\mathbb{Q}$ = `rational (x,y)= x/y  e.g. 1/2`
Complex: $\mathbb{C} \approx$ `complex`(*expr*) e.g. `1+i  2.5-3*i`

Arithmetic

ab = `a*b` $\frac{a}{b}$ = `a/b` a^b = `a**b` $\sqrt{x}$ = `sqrt(x)`
$\sqrt[n]{x}$ = `x^(1/n)` or `mpmath.root(x,n)` $|x|$ = `abs(x)`
$\log_b(x)$ = `log(x,b)` $\log_1 0(x)$ = `log10(x)`
x!=`factorial(x)` $\gamma(x)$ = `gamma(x)`

Constants: π = `mpmath.pi` e = `mpmath.e`
i = `j` ∞ = `inf` ϕ = `mpmath.phi`
Approximate to n digits: `mpmath.dps=n`

Basic Functions

Rationalizing the denominator: radsimp(*expr*)
Common Denominator: ratsimp (*expr*)
Trigonometry: `sin cos tan sec csc cot`
Inverse Trigonometric:`asin acos atan asec acsc acot`
Radians to Degrees: degrees(x)
Degrees to Radians: radians(x) Exponential Function: e^x = `mpmath.exp(x)`

Algebraic Expressions

Expanding: (*expression*).expand()
Simplifying: simplify(*expression*)
Solving Polynomials: solve(f(x),x)
Solving Systems: solve([f(x,y),g(x,y)],[x,y])
e.g. solve([x + 5*y - 2, -3*x + 6*y], [x, y])
Sum:$a + b$ = `a._add_(b)=Add(a,b)`

$\sum_{x=k}^{n} f(x)$ = `nsum(lambda x: f(x), [k, n])`

Calculus

$\lim_{x \to a} f(x)$ = `limit(f(x), x, a)`
$\lim_{x \to a^-} f(x)$ = `limit(f(x), x=a, dir='-')`
$\lim_{x \to a^+} f(x)$ = `limit(f(x), x=a, dir='+')`
$\frac{d}{dx}(f(x))$ = `diff(f(x),x)`
$\frac{\partial}{\partial x}(f(x,y))$ = `diff(f(x,y),x)`
$\int f(x)dx$ = `integrate(f(x),x)`
$\int_a^b f(x)dx$ = `integrate(f(x),(x,a,b))`
Taylor polynomial, deg n about a: `f(x).series(x,n,a)`

Geometry

from sympy.geometry import *
Point: a = Point(0, 0) `Point.is_collinear(a, b, c)`
l=`line([Point(`x_1`,`y_1`), Point(`x_2`,`y_2`)]`
t=`triangle(a,b,c)`
c=`circle((`x`,`y`),`r`)`
`t.area, t.medians[a]`
`c.is_tangent(l),intersection(c,l)`

Graphing

2D Plot from a to b: `mpmath.plot([f(x),g(x)],[a,b])`
3D Plot (pylab only):`mpmath.splot(f(x),[a,b],[a,b])`
Zoom: `R and F, Page Up and Down, Numpad + and -`
Rotate View X,Y axis: `Arrow Keys, A,S,D,W`
Rotate View Z axis: `Q and E, Numpad 7 and 9`
Rotate Ordinate Z axis: `Z and C, Numpad 1 and 3`
View XY: `F1` View XZ: `F2` View YZ: `F3`
View Perspective: `F4` Reset: `X, Numpad 5`
Toggle Axes Visible: `F5` Axes Colors: `F6`
Close Window: `ESCAPE` Screenshot: `F8`

Discrete math

$n!$ = `factorial(n)` $\binom{x}{m}$ = `binomial(x,m)`
Strings: e.g. `s = 'Hello'` = `"Hello"` = `""+"He"+'llo'`
`s[0]='H'  s[-1]='o'  s[1:3]='el'  s[3:]='lo'`
Lists: e.g. `[1,'Hello',x] = []+[1,'Hello']+[x]`
Tuples: e.g. `(1,'Hello',x)` (immutable)
Sets: e.g. $\{1,2,1,a\}$ = `Set([1,2,1,'a'])` $(= \{1,2,a\})$

Linear algebra

from sympy.matrices import *

$\begin{pmatrix}1\\2\end{pmatrix}$ = `Matrix ([[1],[2]])`
$\begin{pmatrix}1&2\\3&4\end{pmatrix}$ = `Matrix([[1,2],[3,4]])`
$\begin{vmatrix}1&2\\3&4\end{vmatrix}$ = `M.det()` $\begin{vmatrix}1&2\\3&4\end{vmatrix}^{-1}$ =`M.inv()`

Dot Product: `v1.dot(v2)` Cross Product:`v1.cross(v2)`

Combining Matrices: `M1.row_join(M2)`,`M1.col_join(M2)`

For an nxn matrix, Identity: `eye(n)`
Zeroes: `zeroes(n)` Ones: `ones(n)`

Statistics

from sympy.statistics import *
Normal distribution: N=Normal(mu,sigma)
N.mean, N.median, N.variance, N.stdev, N.pdf(), N.cdf()
Random number generator:N.random()
Probability on an interval:N.probability(a,b)
Confidence interval for x confidence level: N.confidence(x)

Printing

LaTeX: `latex()`
MathML: `mathml()`
Python: `print python ()`
Unicode: `pprint()`