SageMath
Guia de referencia rápida: Cálculo
Adaptado de William Stein
Versão do Sage: 9.2
`http://wiki.sagemath.org/quickref`

Adaptação: Rogério T.C.

Constantes e funções pré-definidas

Constantes: $\pi =$ `pi` $e =$ `e` $=$ `exp(1)` $i =$ `I` $=$ `i`

$\infty =$ `oo` $=$ `infinity` $\ln(2) =$ `log(2)` $=$ `log2`

Aproximação: `pi.n(digits=18)` $= 3.14159265358979324$

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

Funções e variáveis simbólicas

Criar variáveis simbolicas:
`var("t u theta")` ou `var("t,u,theta")`

Use `*` para multiplocação e `^` ou `**` para exponenciação:
$2x^5 + \sqrt{2}\cos(x)$ = `2*x^5 + sqrt(2)*cos(x)`

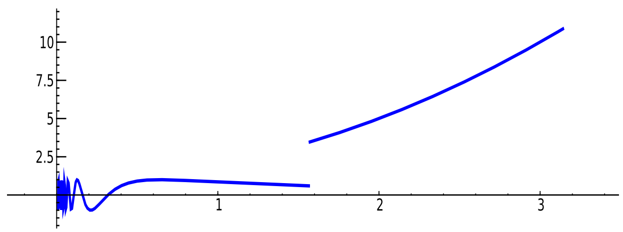
Definindo funções simbólicas:
`f(a,b,theta) = a + b*theta^2`

Função simbólica não determinada:
`f = function("f")(theta)`

Função definida por partes:

$$f(x) = \begin{cases} \sin(1/x), & 0 < x < \pi/2 \\ x^2+1, & \pi/2 < x < \pi \end{cases}$$

`piecewise([[(0,pi/2),sin(1/x)],[(pi/2,pi),x^2+1]])`



Limitando domínio de variável simbólica:
`assume(x>2)`
`assume(x, 'real')`

Verificando o que já foi assumido:
`assumptions()`

Eliminando o que já foi assumido:
`forget(x>2)`
`forget()`

Manipulação simbólica

Sendo f uma função ou expressão simbólica:

Simplificar: `f.simplify_exp()`, `f.simplify_full()`, `f.simplify_log()`, `f.simplify_radical()`, `f.simplify_rational()`, `f.simplify_trig()`

Expandir: `f.expand()`, `f.expand_rational()`

Fatorar: `f.factor()`

`(x^3-y^3).factor()` $\Rightarrow$ $(x^2+xy+y^2)(x-y)$

Limites

$\lim_{x\to a} f(x) =$ `limit(f(x), x=a)`
`limit(sin(x)/x, x=0)`

$\lim_{x\to a^+} f(x) =$ `limit(f(x), x=a, dir='plus')`
`limit(1/x, x=0, dir='plus')`

$\lim_{x\to a^-} f(x) =$ `limit(f(x), x=a, dir='minus')`
`limit(1/x, x=0, dir='minus')`

Derivadas

$\frac{d}{dx}(f(x)) =$ `diff(f(x),x)` ou `f.diff(x)`

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

Exemplo: `diff(x*y + sin(x^2) + e^(-x), x)`

Integrais

$\int f(x)dx =$ `integral(f,x)` ou `f.integrate(x)`
`integral(x*cos(x^2), x)`

$\int_a^b f(x)dx =$ `integral(f,x,a,b)`
`integral(x*cos(x^2), x, 0, sqrt(pi))`

$\int_a^b f(x)dx \approx$ `numerical_integral(f(x),a,b)[0]`

Exemplo:
`assume(x>0)`
`numerical_integral(x*cos(x^2),0,1)[0]`

Expansão em série de Taylor

Polinômio de Taylor de grau n em torno de a:

$\sum_{k=0}^{n} \frac{f^{(k)}(a)}{k!}(x-a)^k =$ `taylor(f,x,a,n)`

Exemplo: `taylor(1/sqrt(1+x), x, 0, 5)`

Séries (finitas ou infinitas)

$\sum_{k=0}^{n} f(k) =$ `sum(f(k),k,1,n)`

$\sum_{k=0}^{\infty} f(k) =$ `sum(f(k),k,1,oo)`

Exemplos:

`sum(k^2,k,1,n)` $\Rightarrow$ $\sum_{k=1}^{n} k^2 = \frac{(2n+1)(n+1)n}{6}$

`sum(1/k^2,k,1,oo)` $\Rightarrow$ $\sum_{k=1}^{\infty} \frac{1}{k^2} = \frac{\pi^2}{6}$

Cálculo de várias variáveis

Gradiente: `f.gradient()` or `f.gradient(`*vars*`)`
`(x^2+y^2).gradient([x,y])`

Hessiana: `f.hessian()`
`(x^2+y^2).hessian()`

Matrix jacobiana: `jacobian(f,` *vars*`)`
`jacobian(x^2 - 2*x*y, (x,y))`

Equações e relações

Relações:

$f = g$: `f == g`, $f \neq g$: `f != g`,
$f \leq g$: `f <= g`, $f \geq g$: `f >= g`,
$f < g$: `f < g`, $f > g$: `f > g`

Resolvendo equações, inequações e sistemas

$f = g$: `solve(f == g, x)`,
$f \leq g$: `solve(f <= g, x)`,
$\begin{cases} f = 0 \\ g = 0 \end{cases}$ `solve([f == 0, g == 0], x,y)`

Exemplo: `solve([x^2+y^2==1, (x-1)^2+y^2==1],x,y)`

Gráficos básicos

2D: (para opções e exemplos consulte `plot?`)
`plot(f(x),(x,xmin,xmax),opções)`
`parametric_plot((f(t),g(t)),(t,tmin,tmax),opções)`
`polarplot(f(t),(t,tmin,tmax),opções)`
`line([(x1,y1),(x2,y2)...,(xn,yn)],opções)`
`text("texto",(x,y),options)`

3D:
`plot3d(f(x,y),(x,xmin,xmax),(y,ymin,ymax),opções)`
`parametric_plot3d((f,g,h),(u,umin,umax),(v,vmin,vmax),opções)`
`line3d([(x1,y1,z1),...,(xn,yn,zn)],opções)`
`text3d("texto",(x,y,z),opções)`