Calculs Scientifique - CM 2 - Sumpy

Saturday, February 3, 2024 9:45 PM

- **CM 2:**

expr.func	# shows type of the expression
expr.args	# shows the sub-expression of the original expression
srepr(expr)	# stands for string representation
apart(polynome)	# will break down the P to a sum of rational fractions
evalf()	# numerical evaluation
expr.subs({a:-2, b:5, c:3, d:0})	# replace multiple values once
solve([expr, expr2], [x,y])	
Eq(expr, value_to_equate)	
Rational(1, 3)	# A class to generate a rational number
latex(expr)	# generate a LaTeX representation of a given expression
Lambda([x], expr)	# A class to generate a function
lambdify(x, expr)	# transform an expression to a function
integrate(expr, (x, 0, 1))	# calculate the integrate between 0 and 1
diff(expr , x , 3)	# third derivative of expr
limit(expr, x, 0, dir='+')	# x-->0, the positive side
expr.series(x, at a, deg n)	# Taylor series
sympify(expr)	# transform a Python expression to a Sympy expression
plot(function) **plot**(function, ylim=(a, b)) **plot**(function, (x, -2, 2), title='') **title** attribute **show** attribute **legend** attribute **line_color** attribute **.line_color** = 'b' **.show**() **plot**(fct1, fct2) **plot1.append**(plot2[0]) **plot1.extend(**plot2**)**	# draw a plot from functions # a plot between -2 and 2 ordonee # a plot between -2 and 2 absis # give a title to the plot # a method to make the color of the plot blue # multiple plots in the same graph
plotting.plot3d(fct, (x, -2, 2), (y, -1, 1))	# generate 3d plot
Equivalent(A, B)	# returns whether two logic expressions are equivalents ro not

Basics

Sympy help:	`help(function)`
Declare symbol:	`x = Symbol('x')`
Substitution:	`expr.subs(old, new)`
Numerical evaluation:	`expr.evalf()`
Expanding:	`expr.expand()`
Common denominator:	`ratsimp(expr)`
Simplify expression:	`simplify(expr)`

Constants

π:	`pi`
e:	`E`
∞:	`oo`
i:	`I`

Numbers types

Integers ($\mathbb{Z}$):	`Integer(x)`
Rationals ($\mathbb{Q}$):	`Rational(p, q)`
Reals ($\mathbb{R}$):	`Float(x)`

Basic funtions

Trigonometric:	`sin cos tan cot`
Cyclometric:	`asin acos atan acot`
Hyperbolic:	`sinh cosh tanh coth`
Area hyperbolic:	`asinh acosh atanh acoth`
Exponential:	`exp(x)`
Square root:	`sqrt(x)`
Logarithm ($\log_b a$):	`log(a, b)`
Natural logarithm:	`log(a)`
Gamma ($\Gamma(x)$):	`gamma(x)`
Absolute value:	`abs(x)`

Calculus

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

Equations

Equation $f(x) = 0$:	`solve(f, x)`
System of equations:	`solve([f, g], [x, y])`
Differential equation:	`dsolve(equation, f(x))`

Geometry

Points:	`a = Point(xcoord, ycoord)`
Lines:	`l = Line(pointA, pointB)`
Circles:	`c = Circle(center, radius)`
Triangles:	`t = Triangle(a, b, c)`
Area:	`object.area`
Intersection:	`intersection(a, b)`
Checking tangency:	`c.is_tangent(l)`

Plotting

Plot:	`Plot(f, [a, b])`
Zoom: $+/-$:	`R/F or PgUp/PgDn or Numpad +/-`
Rotate X,Y axis:	`Arrow Keys or WASD`
Rotate Z axis:	`Q and E or Numpad 7 and 9`
View XY:	`F1`
View XZ:	`F2`
View YZ:	`F3`
View Perspective:	`F4`
Axes Visibility:	`F5`
Axes Colors:	`F6`
Screenshot:	`F8`
Exit plot:	`ESC`

Discrete math

Factorial ($n!$):	`factorial(n)`
Binomial coefficient $\binom{n}{k}$:	`binomial(n, k)`
Sum ($\sum_{n=a}^{b} expr$):	`summation(expr, (n, a, b))`
Product ($\prod_{n=a}^{b} expr$):	`product(expr, (n, a, b))`

Linear algebra

Matrix definition:	`m = Matrix([[a, b], [c, d]])`
Determinant:	`m.det()`
Inverse:	`m.inv()`
Identity matrix $n \times n$:	`eye(n)`
Zero matrix $n \times n$:	`zeros(n)`
Ones matrix $n \times n$:	`ones(n)`

Printing

LaTeX print:	`print latex()`
Python print:	`print python()`
Pretty print:	`pprint()`

Examples

Find 100 digits of π^e:

```
(pi**E).n(100)
```

Expand $(x+y)^2(x-y)(x^2+y)$:

```
((x + y)**2 * (x - y) * (x**2 + y)).expand()
```

Simplify $\frac{1}{x} + \frac{x \sin x - 1}{x^2 - 1}$:

```
simplify((1/x) + (x * sin(x) - 1)/(x**2 - 1))
```

Check if line passing through points $(0,1)$ and $(1,1)$ is tangent to circle with center at $(5,5)$ and radius 3:

```
Circle(Point(5,5), 3).is_tangent(
Line(Point(0,1), Point(1,1)))
```

Find roots of $x^4 - 4x^3 + 2x^2 - x = 0$:

```
solve(x**4 - 4*x**3 + 2*x**2 - x, x)
```

Solve the equations system: $x + y = 4$, $xy = 3$:

```
solve([x + y - 4, x*y - 3], [x, y])
```

Calculate limit of the sequence $\sqrt[n]{n}$:

```
limit(n**(1/n), n, oo)
```

Calculate left-sided limit of the function $\frac{|x|}{x}$ in 0:

```
limit(abs(x)/x, x, 0, dir='-')
```

Calculate the sum $\sum_{n=0}^{100} n^2$:

```
summation(n**2, (n, 0, 100))
```

Calculate the sum $\sum_{n=0}^{\infty} \frac{1}{n^2}$:

```
summation(1/n**2, (n, 0, oo))
```

Calculate the integral $\int \cos^3 x\,dx$:

```
integrate(cos(x)**3, x)
```

Calculate the integral $\int_1^{\infty} \frac{dx}{x^2}$:

```
integrate(1/x**2, (x, 1, oo))
```

Find 10 terms of series expansion of $\frac{1}{1-2x}$ at 0:

```
(1/(1 - 2*x)).series(x, 0, 10)
```

Solve the differential equation $f''(x) + 9f(x) = 1$:

```
dsolve(f(x).diff(x, x) + 9*f(x) - 1, f(x))
```