my sympy and python cheat sheet

Nasser M. Abbasi

January 28, 2024 Compiled on January 28, 2024 at 4:57am

Contents

1 Installing sympy 1.10.1 on Ubuntu 1

2 How to solve a first order ODE? 2

3 How to solve a first oder ODE with initial condition? 2

4 How to solve a second order ODE? 3

5 How to solve and ODE and convert the result to latex string? 3

6 How to solve a PDE in sympy? 4

7 How to check if something is derivative? 4

8 How to find function name and its arguments in a proc? 5

1 Installing sympy 1.10.1 on Ubuntu

Installing Python 3.10 and sympy 1.10.1 was very tricky. First Ubuntu 20.04 came with Python 3.8 and it was struggle to make it use Python 3.10 instead. After 2 hrs trying many commands, finally it seems to be using 3.10 now. Next did the following

```
>which python
/bin/python
>python --version
Python 3.10.4
```

Now

```
python -m pip install sympy
```

Gives

```
Defaulting to user installation because normal site-packages is not writeable
Collecting sympy
Using cached sympy-1.10.1-py3-none-any.whl (6.4 MB)
Collecting mpmath>=0.19
Using cached mpmath-1.2.1-py3-none-any.whl (532 kB)
Installing collected packages: mpmath, sympy
Successfully installed mpmath-1.2.1 sympy-1.10.1
```

And now

```
>python
Python 3.10.4 (main, Mar 24 2022, 16:12:56) [GCC 9.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import sympy
>>> sympy.__version__
'1.10.1'
```

2 How to solve a first order ODE?

Solve $y'(x) = 1 + 2x$ for $y(x)$

```python
from sympy import *
x   = symbols('x')
y   = Function('y')
ode = Eq(Derivative(y(x),x),1+2*x)
sol = dsolve(ode,y(x))
#    Eq(y(x), C1 + x**2 + x)
checkodesol(ode,sol)
#    (True, 0)
if checkodesol(ode,sol)[0]==True:
  print ('verified solution OK')
```

3 How to solve a first oder ODE with initial condition?

Solve $y'(x) = 1 + 2x$ for $y(x)$ with $y(0) = 3$

```
import sympy
x   = sympy.symbols('x')
y   = sympy.Function('y')
ode = sympy.Eq(sympy.Derivative(y(x),x),1+2*x)
sol = sympy.dsolve(ode,y(x),ics={y(0):3})
#    Eq(y(x), x**2 + x + 3)
sympy.checkodesol(ode,sol)
#    (True, 0)
```

4 How to solve a second order ODE?

Solve $9y(x) + \frac{d^2}{dx^2}y(x) = 0$

```
from sympy import Function,dsolve,Derivative,Eq
x=sympy.symbols('x')
y=sympy.symbols('y', cls=Function)
ode=Eq(Derivative(y(x), x, x) + 9*y(x),0)
dsolve(ode, y(x))
```

gives

$$y(x) = C_1 \sin(3x) + C_2 \cos(3x)$$

5 How to solve and ODE and convert the result to latex string?

Solve $y'(x) = 1 + 2x$ for $y(x)$ with $y(0) = 3$

```
import sympy
x   = sympy.symbols('x')
y   = sympy.Function('y')
```

```
ode = sympy.Eq(sympy.Derivative(y(x),x),1+2*x)
sol = sympy.dsolve(ode,y(x),ics={y(0):3})
#    Eq(y(x), x**2 + x + 3)
sympy.latex(sol)
```

$$y(x) = x^2 + x + 3$$

6 How to solve a PDE in sympy?

PDE solving is still limited in sympy. Here is how to solve first order pde

Solve $u_t(x,t) = u_x(x,t)$

```
import sympy as sp
x,t   = sp.symbols('x t')
u     = sp.Function('u')
pde   = sp.Eq( sp.diff(u(x,t),t) , sp.diff(u(x,t),x))
sol   = sp.pdsolve(pde)
sp.latex(sol)
```

$$u(x,t) = F(t+x)$$

7 How to check if something is derivative?

```
import sympy
x    = sympy.symbols('x')
y    = sympy.Function('y')
expr = sympy.Derivative(y(x),x)
type(expr) is sympy.Derivative
      #True

if type(expr) is sympy.Derivative:
  print("yes")

       #yes
```

This also works, which seems to be the more prefered way

```
isinstance(expr,sympy.Derivative)
  #True
```

8 How to find function name and its arguments in a proc?

Suppose one passes $y(x)$ to a function, and the function wants to find the name of this function and its argument. Here is an example

```
def process(the_function):
   print("the function argument is ", the_function.args[0])
   print("the function name itself is ", the_function.name)
import sympy
x   = sympy.symbols('x')
y   = sympy.Function('y')
process(y(x))
```

This prints

```
the function argument is  x
the function name itself is  y
```