

Sympy

This guide is heavily inspired from the Sympy tutorial Introduction available here :
<https://docs.sympy.org/latest/tutorial/intro.html>

Symbolic computation deals with the computation of mathematical objects symbolically. This means that the mathematical objects are represented exactly, not approximately, and mathematical expressions with unevaluated variables are left in symbolic form.

Python is great for maths, and for getting numerical answers for numerical computations :

```
>>> import math
>>> math.sqrt(8)
```

But, when providing results, python provides a numerical answer which is not perfectly accurate, but suppose we want to simplify an expression instead of getting the numerical value:

```
>>> import sympy
>>> sympy.sqrt(8)
```

Symbolic Expressions

The real power of SymPy is the ability to use symbols within expression that allows to solve complex equations`:

```
>>> from sympy import symbols
>>> x, y = symbols('x y')
>>> expr = x + 2*y
>>> expr
```

We now have defined a Sympy expression containing two symbols, we can play around with that:

```
>>> expr + 1
>>> expr - x
>>> x*expr
```

Did you notice how `x*expr` became `x*(x+2*y)` and not `x^2+2xy` ? We can actually get the expanded form of the expression:

```
>>> from sympy import expand, factor
>>> expanded_expr = expand(x*expr)
```

```
>>> expanded_expr
```

You can also easily factor an expanded expression:

```
>>> factor(expanded_expr)
```

Powerful Symbolic Computations

Let's now use Sympy for some complex computations:

```
>>> from sympy import *
>>> x, t, z, nu = symbols('x t z nu')
>>> init_printing(use_unicode=True)
```

The second line prints mathematical expressions in a nicer way in the terminal.

Let's get the derivative

```
>>> diff(sin(x)*exp(x), x)
```

Let's compute $\int (e^x \sin(x) + e^x \cos(x)) dx$:

```
>>> integrate(exp(x)*sin(x) + exp(x)*cos(x), x)
```

Let's compute $\int_{-\infty}^{\infty} (\sin(x^2)) dx$:

```
>>> integrate(sin(x**2), (x, -oo, oo))
```

Sympy can also be used to solve equations

Solve $x^2 - 2 = 0$:

```
solve(x**2 - 2, x)
```

Solve the differential equation $y'' - y = e^t$ (might take a second or two):

```
>>> y = Function('y')
>>> dsolve(Eq(y(t).diff(t, t) - y(t), exp(t)), y(t))
```

Find the eigenvalues of the matrix $\begin{bmatrix} 1 & 2 \\ 2 & 2 \end{bmatrix}$:

```
>>> Matrix([[1, 2], [2, 2]]).eigenvals()
```

Finally, sympy can also provide answers already formatted in a Latex format so you can just copy and paste the result in your .tex document:

```
>>> latex(Integral(cos(x)**2, (x, 0, pi)))
```