



# Discrete Structures: CMPSC 102

Oliver BONHAM-CARTER

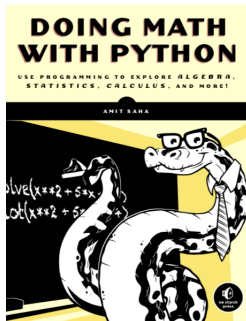
Fall 2019

Week 13

SymPy

Installing  
SymPy

Practical 12



## Saha, Chapter 4: SymPy

- Doing symbolic math with Python
- Note: we may have to use Python2 to study the libraries.

# What is Symbolic Math?

SymPy

Installing  
SymPy

Practical 12

- Can you see the difference between 2.8284271247461903 and  $2*\text{sqrt}(2)$ ?
  - These are both the results of  $\sqrt{8}$
  - One is easier to follow than another
- Using a large floating point value may have eventual rounding errors and is difficult to track when following the steps in working out mathematical work
- SymPy makes working with math easier as the notation is conserved during the work.

# Complication without Complexity

Such as ...

SymPy

Installing  
SymPy

Practical 12

Solve for  $x$

$$\begin{aligned} a * x^2 + b * x + c &= 0 \\ a = 3, b = 2, c = 3, x &=? \end{aligned} \tag{1}$$

Solve for  $x, y, z$

$$\begin{aligned} x + 2 * y + 0 * z &= 1 \\ 3 * x + 2 * y + 2 * z &= 1 \\ 2 * x + 0 * y + 0 * z &= 1 \\ x = ?, y = ?, z &=? \end{aligned} \tag{2}$$

# What is SymPy?

## SymPy

### Installing SymPy

#### Practical 12



<https://www.sympy.org>

- SymPy is a symbolic manipulation library for python.
- Solve mathematical problems using non-numeric data types
- You can work with math where the computed values remain in the contexts of fractions, equations, etc (otherwise these values would be immediately converted to floating points, for example)

# What is SymPy?

## SymPy

### Installing SymPy

#### Practical 12

## Points to consider

- Enter expressions in symbolic form with symbolic data types.
- Expand or simplify symbolic expressions.
- Find symbolic roots, limits, minima, maxima, etc.
- Differentiate and integrate symbolic functions.
- Generate Taylor series of functions (among other tools).
- Solve algebraic and differential equations symbolically.
- Solve simultaneous equations (even some nonlinear).
- Do variable precision arithmetic.
- Create graphical representations of symbolic functions.

## Installing Software



75%

### Websites

- <https://docs.sympy.org/latest/tutorial/intro.html>
- <https://www.sympy.org/en/index.html>

### Installation Commands from Bash or Command Prompt

```
python3 -m pip install -U pip # install PIP
# python3 -m pip install --user --upgrade pip # alternative

python3 -m pip install -U sympy #install Sympy core
# python3 -m pip install -U --user sympy # alternative
```



## Checking the Version

```
import sympy  
print(sympy.__version__) # '1.4'
```



## Using Python

```
import sympy
sympy.sqrt(8) # 2*sqrt(2)
sympy.sqrt(20) # 2*sqrt(5)
```

Input:

$$\sqrt{8}$$

Exact result:

$$2\sqrt{2}$$

Decimal approximation:

2.8284271247461900976033774

Input:

$$\sqrt{20}$$

Exact result:

$$2\sqrt{5}$$

Decimal approximation:

4.4721359549995793928183

Graphics from: <https://www.wolframalpha.com>

# Study the Steps of Simplification

SymPy

Installing  
SymPy

Practical 12

## The Step-By-Step Way

- When you work each step of the math by hand, you can see how the results are generated.

## Using Python

- $\sqrt{8} = \sqrt{4 * 2} = \sqrt{4} * \sqrt{2} = 2 * \sqrt{2} = 2.83$
- $\sqrt{20} = \sqrt{4 * 5} = \sqrt{4} * \sqrt{5} = 2 * \sqrt{5} = 4.47$

## Endless Possibilities

What else can this system do?!  
(you tell me!)

# Study the Steps of Simplification

## Simple examples

SymPy

Installing  
SymPy

Practical 12

```
from sympy import Symbol
x = Symbol('x')
y = Symbol('y')
s = x*y + x*y
print(s)
```

```
from sympy import Symbol
x = Symbol('x')
p = (x + 2)*(x + 3)
print(p)
```

```
from sympy import pprint, Symbol
x = Symbol('x')
y = Symbol('y')
expr = x*x + 2*x*y + y*y
pprint(expr)
```

## Practical 12, GitHub Repository

Since we will not be meeting on Friday,  
we will have our practical today.

Due without exceptions at 2pm, 25<sup>th</sup> November 2019

<https://classroom.github.com/a/AwN13f3p>

**THINK**

# Practical 12

Explore on your own

SymPy

Installing  
SymPy

Practical 12

What to do: Follow a portion of the SymPy tutorial!

- **Follow the intro tutorial to get the overall**
  - <https://docs.sympy.org/latest/tutorial/intro.html>
- **Now, pick an area to follow another tutorial**
  - <https://docs.sympy.org/latest/tutorial/>
- Use the interactive interpreter to play with the code or use your local python installation.
- Respond to the questions in `writing/miniReflection.md`

**THINK**