Ton Duc Thang University
Faculty of Information Technology

# Applied Calculus for IT - 501031
# Lab 04

## 1. Sympy

SymPy is a Python library for symbolic mathematics. It aims to become a full-featured computer algebra system (CAS) while keeping the code as simple as possible in order to be comprehensible and easily extensible.

### 1.1 Setup:

If you have Python and PIP already installed on a system, then install it using this command:

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

### 1.2 Introduction

Symbolic computation systems such as SymPy are capable of computing symbolic expressions with variables. Let us define a symbolic expression, representing the mathematical expression *x+2y*.

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

Note that we wrote $x + 2 * y$ just as we would if x and y were ordinary Python variables. But in this case, instead of evaluating to something, the expression remains as just $x + 2 * y$. Now let us play around with it:

```
>>> expr + 1
x + 2*y + 1
>>> expr - x
2*y
```

https://docs.sympy.org/latest/tutorials/intro-tutorial/intro.html

### 1.3 Practice examples:

Solve $x^2 + 4x + 4 = 0$.

```
import sympy as sp
```

```
x = sp.symbols('x')
root = sp.solve(x**2 + 4*x + 4)
print(root)
```

Plot functions:

```
import sympy as sp
import math

x = sp.symbols('x')

f5a = abs(x)**(1/2)
sp.plot(f5a, (x, -10, 10), line_color='red')  #tuple

f5d = math.e**x
sp.plot(f5d, (x, -10, 10), line_color='blue')

f5e = sp.log(x)
sp.plot(f5e, (x, -10, 10))
```

Evaluate a certain expression:

```
import sympy as sp

x = sp.symbols('x')
f = x*x - x + 1

print( f.subs(x, 2) ) # f(2)
```

**Exercise 0**

Write a Python program to plot the following functions on a graph, and mark the intersection point of f1 and f2:

$$f_1(x) = -x + 5$$
$$f_2(x) = \frac{x}{2} + 2$$

Hint:

```
import sympy as sp
import matplotlib.pyplot as plt
import numpy as np

# ve f1 va f2
```

```
x = np.arange(...)
f1 = lambda x: ...
f2 = lambda x: ...

y1 = ...
y2 = ...

plt.plot(x, y1)
plt.plot(x, y2)

# ve diem giao nhau
x = sp.symbols('x')
f1 = -x + 5
f2 = ...

x_root = sp.solve(...)
y_root = f1.subs(...)  #f1(x_root[0]) = f1(2)

plt.plot(..., y_root, 'ro')
plt.title('Find intersection point of f1(x) = -x + 5 and f2(x) = x/2 + 2')
plt.grid(linestyle='--')
plt.show()
```

**Exercises 1, 3, 4, 5, 6, 7, 10 in the PDF file "Lab04.ex.pdf"**

**Hint:**

**Limit of a function:**

```
import sympy as sp
import math
x = sp.symbols('x')
#1c
f1c = math.e ** (1/x)
lm = sp.limit(f1c, x, 1)
print("1c - The limit of f(x) at x = 1: " + str(lm) )
#1i
f = (2*x*x) / ( 3 - 3*sp.cos(x) )
lm = sp.limit(f, x, 0)
print("1i - The limit of f(x) at x = 0: " + str(lm) )

#infinity: sp.oo

#factorial: sp.factorial(...)
```

**Right/Left limit:**

**Theorem.**

- $\lim_{x\to a} f(x) = L \iff \lim_{x\to a^+} f(x) = \lim_{x\to a^-} f(x) = L.$

$$f(x) \to L \Leftarrow x \to a \iff \begin{cases} x \to a^+ & \Rightarrow f(x) \to L \\ x \to a^- & \Rightarrow f(x) \to L \end{cases}$$

```
import sympy as sp
import math

#3.1
x = sp.symbols('x')
f3_1 = 1/( 1 + 2**(1/x) )
lmRight = sp.limit(f3_1, x, 0, '+')
print ("Right limit = " , lmRight )
lmLeft = sp.limit(f3_1, x, 0, '-')
print ("Left  limit = " , lmLeft )
```

**Continuity**

## Definition of Continuity

- A function $f$ is **continuous at a number** $a$ if

$$\lim_{x\to a} f(x) = f(a).$$

If $f$ is not continuous at $a$, we say $f$ is **discontinuous** at $a$.

- **Remark.** The definition consists of the 3 properties:

i) $f$ is defined at $a$ (i.e., $a$ is in the domain of $f$), and

ii) $\lim_{x\to a} f(x)$ exists, and

iii) $\lim_{x\to a} f(x) = f(a)$.

**Ex6:**

```
import numpy as np
import sympy as sp

x = sp.symbols('x')
f6a = ...

#At point x = 0
lm_x_0 = sp.limit(f6a, x, 0)
#Compare lm_x_0 and f(0)
...

#Other points x != 0
for c in np.arange(-100, 100, 1):
  if c != 0:
    lm_x_c = sp.limit(f6a, x, c)
    #Compare lm_x_c and f(c)

#f(x) is continuous for all x # 0, 3
```

**Homework:**
**Exercises 8 in the PDF file "Lab04.ex.pdf"**

## 2. References

- Python Tutorial on the W3schools website: https://www.w3schools.com/python/default.asp
- Python Tutorial on the Tutorials Point website: https://www.tutorialspoint.com/python/index.htm

**-- THE END --**