### 311 – Numerical Computations Lab 5: Installing Scipy and Numpy, The bisection Method

## A-Matrices in Python/ Matrices in Numpy:

In Python, they are basically: List of lists

```
a = [[7, 1, 3, 2], [7, 3], [6, 1, 2]]

for i in range(len(a)):

    for j in range(len(a[i])):

        print(a[i][j], end=' ')

    print()
```

#### Output:

7 1 3 2

73

612

### Same Program:

```
a = [[7, 1, 3, 2], [7, 3], [6, 1,2]]

for i in a:

for j in i:

print(j, end=' ')

print()
```

However, we will later shift to "Matrices" in the special library: <u>Numpy</u>.

# **B-** Default arguments in Python:

```
def f (a=1, b=2, c=3):  # 3 optional arguments
    return a+2*b-c
print(f())
print(f(5))
print(f(2,3))
print(f(c=8))
```

```
def f (a, b=2, c=3): #2 optional arguments
return a+2*b-c

print(f(5))
print(f(2,3))
print(f(4,c=8))
print(f(c=8))
print(f())
```

# However, the following function is Not valid (Syntax Error):

```
def f (a=4, b, c=3):
return a+2*b-c
```

# Because non-default argument follows default argument

# C- Installing NumPy and SciPy:

#### 1-Install PIP:

**PIP** is a <u>package management system</u> used to install and manage software packages in Python.

Start On the command prompt by:

```
C:\Users\Galal>python -V

Python 3.9.1

C:\Users\Galal>pip -V (or pip --version)

pip 21.0.1 .....
```

If pip is not installed, you have to install it first. Now you can proceed:

```
pip install numpy
```

```
pip install scipy
```

• Now the following program should work:

```
from scipy import optimize

def f(x):

return (x**2 - 1)

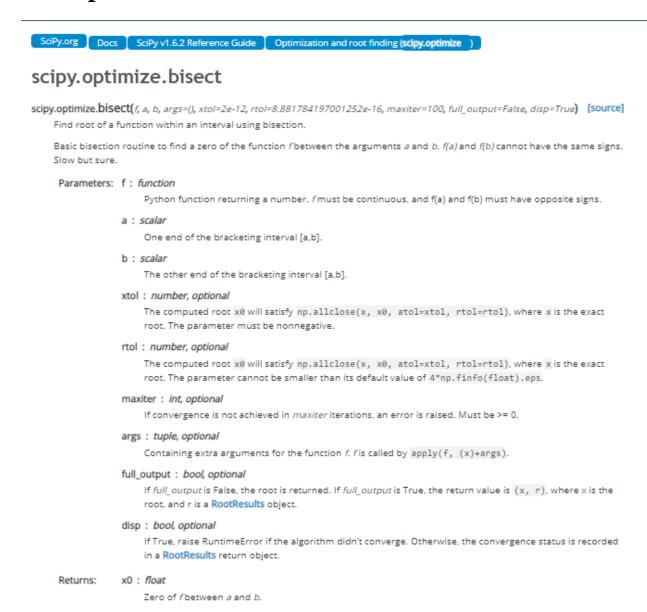
print(optimize.bisect(f, 0, 2))

print(optimize.bisect(f, -2, 0))
```

So How to set the desired tolerance, for example????
The secret is in the **Default Arguments (See Section B)**,,
(See also Next Page)!!

# **D-** Numpy and Scipy Documentation:

#### **Example:** bisect function documentation:



#### The documentation homepage:

https://docs.scipy.org/doc/

# **E-** Example: Solving the Cubic Equation

a-The algebraic method!!! (Don't try it !!!)

$$x = \sqrt[3]{\left(\frac{-b^3}{27a^3} + \frac{bc}{6a^2} - \frac{d}{2a}\right) + \sqrt{\left(\frac{-b^3}{27a^3} + \frac{bc}{6a^2} - \frac{d}{2a}\right)^2 + \left(\frac{c}{3a} - \frac{b^2}{9a^2}\right)^3}}$$

$$+ \sqrt[3]{\left(\frac{-b^3}{27a^3} + \frac{bc}{6a^2} - \frac{d}{2a}\right)} - \sqrt{\left(\frac{-b^3}{27a^3} + \frac{bc}{6a^2} - \frac{d}{2a}\right)^2 + \left(\frac{c}{3a} - \frac{b^2}{9a^2}\right)^3} - \frac{b}{3a}.$$

**Abel–Ruffini theorem:** There is no algebraic solution to general polynomial equations of degree five or higher with arbitrary coefficients.

b- Write your own (non-recursive) version of the bisection method:

def bisection(f,a,b,e):

- For simplicity, assume initially that: f(a)f(b) < 0.
- Stop and return the value x, where  $f(x) \le e$ .
- Print the sequence of solutions (helpful if debugging is needed).

c-Use your function to solve the cubic equation:

$$0.4 x^3 - 9 x^2 - 12 = 0$$

(Use the two ends: 20 and 25 and e=0.005)