National Higher School of Mathematics
Department of Preparatory Cycle
Computational Mathematical Tools

Lab-work N°01

Submitted By:
Full name :
....
ID :
....
Group / Section :
....

Submitted To:
Mr. A. Ameraoui
Assoc. Professor
Dept. of PC

February, 2023

NATIONAL HIGHER SCHOOL
OF MATHEMATICS
المدرسة الوطنية
العليا في الرياضيات

1 Non-linear equations solving problem

Let's define the function

$$f(x) = \cos(x) - xe^x, \text{ for } x \in \left[0, \frac{\pi}{2}\right]$$

1. Using **Python** package `matplotlib.pyplot`, plot the graph of the function $f(x)$, while checking the uniqueness of the root of the equation $f(x) = 0$ (let denoted this root x_r).

```
In [1]: import numpy as np
In [2]: import matplotlib.pyplot as plt
In [3]: def f(x: float):    # Or use f= lambda x : np.cos(x) - x*np.exp(x)


...
```

2. Write **Python** function `NRroot()`, that allows the approximation of x_r using the bisection method, for a given interval $[a, b]$ and a precision $\varepsilon > 0$.

```
def BSroot(a: float, b: float, eps: float):

...
```

Report what the function returns for $a = 0, b = \frac{\pi}{2}$ and $\varepsilon = 10^{-8}$

```
In [4]:


...
```

3. Write **Python** function `NRroot()`, to find the approximate value of the root x_r, using the Newton-Raphson method, for a given $x_0 \in [a, b]$. (the method requires the derivative of the function f).

```
def fprime(x: float):


def NRroot(x0: float, eps: float):


...
```

Report what the function returns for $x_0 = \frac{\pi}{4}$ and $\varepsilon = 10^{-8}$

```
In [5]:



...
```

4. Write `Python` function `SCroot()`, to approximate the value of x_r to within $\varepsilon > 0$, using the secant sequence $(x_n)_{n\geqslant 0}$, defined for a given $x_0, x_1 \in [a,b]$:

$$\begin{cases} x_0, x_1 \in [a,b] \\ x_{n+1} = x_n - f(x_n)\dfrac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})}, \text{ for all } n \geqslant 1 \end{cases}$$

```
def SCroot(x0: float, x1: float, eps: float):


...
```

Report what the function returns for $x_0 = 0, x_1 = \frac{\pi}{4}$ and $\varepsilon = 10^{-8}$

```
In [6]:


...
```

5. We rewrite the function $f(x)$, in the form $x = \cos(x)e^{-x} = g(x)$. Write `Python` function `FProot()`, to approximate the value of x_r to within $\varepsilon > 0$, using the fixed-point method $x_{n+1} = g(x_n)$ and for a given $x_0 \in [a,b]$.

```
from typing import Callable
Func = Callable[[float], float]
def FProot(g: Func, x0: float, eps: float):


...
```

Report what the function returns for $x_0 = \frac{\pi}{4}$ and $\varepsilon = 10^{-8}$

```
In [7]:


...
```

6. Consider now, the Steffensen convergence acceleration procedure, given by :

$$\begin{cases} x_0 \in [a,b] \\ y_n = g(x_n) \\ x_{n+1} = x_n - \dfrac{(y_n - x_n)^2}{g(y_n) - 2y_n + x_n}, \text{ for all } n \geqslant 0 \end{cases}$$

Write `Python` function `STroot()`, to approximate the value of x_r to within $\varepsilon > 0$, using the Steffensen method and for a given $x_0 \in [a,b]$.

```python
def STroot(x0: float, eps: float):



...
```

Report what the function returns for $x_0 = \frac{\pi}{4}$ and $\varepsilon = 10^{-8}$

```python
In [8]:


...
```

7. Complete and comment on the results obtained in the following table :

ε	Nbr iterations BS	Nbr iterations NR	Nbr iterations FP	Nbr iterations ST
10^{-1}				
10^{-2}				
10^{-3}				
10^{-4}				
10^{-5}				
10^{-6}				
10^{-7}				
10^{-8}				
10^{-9}				
10^{-10}				

Comments :