Exercise #4

11. September 2023

Exercises marked with a (J) should be handed together with a Jupyter notebook.

Problems marked with 4N or 4D must be submitted only by the respective course, while unmarked problems must be submitted by both courses.

A serious attempt in solving $70 - 75\%$ of all the tasks must be done in **both** theory and coding (if present) in order to pass the exercise.

Optional exercises will not be corrected.

Problem 1. (Bisection Method - **J**)
Consider the function

$$f(x) = (1 - 3^x)x^2 + 4(x-1)3^x + 4(1-x).$$

We want to find roots of $f(x)$ in the interval $[-2, 3]$.

a) Plot the function on the interval $[-2, 3]$.

b) Do 4 iterations of the bisection method by hand. After 4 iterations you have a narrowed down interval $[a_n, b_n]$ in which the solution lies. Taking $\tilde{x} = (a_n + b_n)/2$ as your solution, which solution did you find? What is the error bound, i.e. the maximum distance between $\tilde{x}$ and the exact solution?

c) How many iterations are at most required to guarantee an error smaller than 10^{-3}? Compute it analytically.

d) (J) Implement the bisection algorithm in python to find roots of $f(x)$ with an error smaller than 10^{-3} (again in the interval $[-2,3]$).

Problem 2. (Fixed point method)

We consider the solution of the equation

$$\cos(e^{-x}) = 2\sqrt{x}.$$

a) Show that the following fixed point method

$$x = g(x) \quad \text{with} \quad g(x) = \frac{\cos^2(e^{-x})}{4}$$

has a unique solution $\hat{x} \geq 0$.

b) If you run the code below, you will obtain a value x that is a numerical approximation of $\hat{x}$. Provide an upper bound for the error $e := |\hat{x} - x|$.

```
import numpy as np

def g(x):
    return np.cos(np.exp(-x))**2/4

x = 0
x_old = 1

while np.abs(x_old-x) > 1e-6:
    x_old = x
    x = g(x)
    print(x)
```

Problem 3. (Newton's method - **J**)
Consider the function

$$f(x) = \cos(x) - \sqrt{x}.$$

We want to find the root of $f(x)$.

a) Compute two steps with Newton's method by hand, starting at $x_0 = 1$.

b) (J) Implement Newton's method for the problem at hand and use it to find the root of $f(x)$ with error less than 10^{-3}. Use $x_0 = 1$ as starting point.

Problem 4.

Consider the following implementation of an iterative method:

```
from numpy import cos, sin, log

x = 0.5
err = abs(cos(x)+log(x))

while err > 1e-6:
    dx = -(cos(x)+log(x))/(-sin(x)+1/x)
    x += - dx
    err = abs(dx)

print(x)
```

a) What method is supposed to be implemented above?

b) Write down the specific equation being (iteratively) solved by the algorithm.

c) An error is present in the code, spot it.