Deadlines:

For all Groups: April 25 2023

Grading system:

For every problem you solve you get a score. Your score is your mark for this laboratory.

- 1 problem – 1p

- 2 problem – 2p

- 3 problem – 2p

- 4 problem – 2p

- 5 problem – 2p

- 6 problem – 0.5p

- 7 problem – 0.5p

Introduction:

Welcome to the first numerical analysis lab. Here you will learn some interesting things. The exercises will be solved using Python, Julia, MATLAB or Octave. I have listened to your wishes on how the lab should be so it is built from two parts. The first part is from problem 1-5 and it's pretty easy, maybe only problem 5 is more difficult but it's easy overall it's kind of like homework for these exercises if you do it you get a 9. Problem 6 and 7 are harder and this is for those who know programming well or want to know more and so if you solve these problems too you can get a grade 10. The bonus problem is an interesting one but it's complex and it's up to you if you want to solve this problem.

1. Bisection method

Use the bisection method to find a root of the equation $e^x - x^2 = 0$ accurate to 8 decimal places on the interval [-2, 0].

Restrictions:

You don't have to use any libraries for this exercise. Exceptions are Numpy and Math libraries

2. Interesting function

Let g(x) = $2x - cx^2$ where c is a positive constant. Prove that if the fixed-point iteration p_n = g(p_n−1) converges to a non-zero limit, then the limit is 1/c.

Tip:
Use Newton's method for finding solution

Restrictions:
You don't have to use any libraries for this exercise. Exceptions are Numpy and Math libraries

3. Muller's method

Use Muller's method to find a root of the equation $x^3 + 2x^2 + 10x - 20 = 0$ accurate to 8 decimal places, starting with the initial guesses x_0 = 0, x_1 = 1, and x_2 = 2.

Restrictions:
You don't have to use any libraries for this exercise. Exceptions are Numpy and Math libraries

4. Nonlinear equations

A research team is investigating the behaviour of a nonlinear system. They have formulated a system of nonlinear equations that describe the behaviour of the system. They want to find the roots of the system of equations to better understand the behaviour of the system. Write a program that implements the Newton-Raphson method to find the roots of the system of nonlinear equations. The program should take as input the initial guess, the system of equations, and the desired tolerance. The output should be the roots of the system of equations with an accuracy of at least the specified tolerance value.

Restrictions:
You don't have to use any libraries for this exercise. Exceptions are Numpy and Math libraries

5. Improving algorithm

You are working for a social media company that is trying to improve its recommendation algorithm. The algorithm involves solving a nonlinear equation of the form f(x) = 0, where f(x) is a function that depends on the user's preferences and other factors. You decide to use the secant method to find the root of the equation because it is faster than the bisection method. However, you realise that the secant method may not always converge to the correct root, especially for certain types of functions. Your task is to implement a modified version of the secant method that is more robust and can handle different types of input data. You decide to use a hybrid approach that combines the secant method with the bisection method. Write a program that implements the hybrid secant-bisection method for finding the root of the nonlinear equation. Your program should be able to handle different types of input data.

Restrictions:
You don't have to use any libraries for this exercise. Exceptions are Numpy and Math libraries

6. Editing images

UTM creates its own application for image editing. You are working on this graphics project, your part in this project involves rotating and scaling images. Develop a program that can perform affine transformations on images using matrices. Use the NumPy library to implement matrix operations. Write test cases for your program to ensure its accuracy.

Restrictions:
You don't have to use any libraries for this exercise. Exceptions are Numpy and Math libraries

7. Calculator

You are a software engineer working on a project that requires precise numerical calculations. However, you have noticed that your program is producing incorrect results for some calculations involving very large or very small numbers. Upon further investigation, you discover that this is due to the limitations of computer representation of numbers. Your task is to write a program that can accurately perform calculations involving very large or very small numbers, accounting for the limitations of computer representation. Specifically, you must implement algorithms for:

-Converting decimal numbers to binary and vice versa.

-Addition, subtraction, multiplication, and division of binary numbers.

-Computing the absolute and relative error of a given approximation.

-Implementing data structures such as arrays and linked lists to efficiently store and manipulate large numbers.

Your program must be able to accurately perform calculations on numbers of up to 100 digits in length, and should include appropriate error-handling mechanisms. Your program must also include a user interface that allows the user to input numbers and choose the desired operation. Your program will be evaluated based on the accuracy of its calculations.

Bonus: Interesting project

Mr. Bostan's friend is a mathematician working on a research project that involves solving a system of nonlinear equations. The system has a very large number of equations and variables, and it is difficult to solve using traditional methods. Mr. Bostan told you to help him with this project. You decide to use the Newton-Krylov method, a powerful and efficient method for solving large-scale systems of nonlinear equations. However, you realise that the method requires a lot of computational resources and may not be suitable for all types of systems. Your task is to implement the Newton-Krylov method for solving the system of nonlinear equations. You also need to optimise your implementation to make it as efficient as possible, using techniques such as sparse matrix storage and parallel processing. Write a program that implements the Newton-Krylov method for solving the system of nonlinear equations. Your program should be able to handle large-scale systems.