Project – Phase 2

1- Objective

The aim of this project is to compare and analyze the behavior of the different numerical methods used for calculating the roots of equations:

1. Bisection
2. False-Position
3. Fixed point
4. Original Newton-Raphson
5. Modified Newton-Raphson
6. Secant Method.

2- Description

You are required to implement a root finder program which takes as an input the equation, the technique to use and its required parameters (e.g., interval for the bisection method).

3- Specification

You're required to build upon the GUI application of phase 1, to include the following:

1. Accepts a free-text input for a non-linear equation:
   a. The equations containing different function: {poly, exp, cos, sin}.
   b. The variable used is "x"

2. You have to plot the function first to aid the user to get the initial guess(es). Plot g(x) with y = x in case of fixed point, f(x) in the remaining cases.

3. Choose any of the previously mentioned methods to solve the given equation via a drop-down list.

4. Parameters, if it applicable for the chosen solving method.

5. A way to enter the precision (number of significant figures), EPS and the max number of iterations otherwise default values are used; Default #SFs = System Default, Default Epsilon = 0.00001, and Default Max Iterations = 50.

6. A Solve button to display the approximate root (if exists), number of iterations , approximate relative error , number of correct significant figures, and the execution time.

7. The error messages (if applicable) must be descriptive.

4- Bonus

You can implement *Single step mode* simulation showing each step of the algorithm.

5- Deliverables

You must deliver the following:

1. A fully commented and clean code. You may use any programming language.
2. A well-formatted report that contains the following:
   a. Flowchart or pseudo-code for each method.
   b. Sample runs for each method; they must include different cases (normal and tricky cases).
   c. **Analyzing the performance** of all implemented numerical methods by finding the root of **at least two different equations**. The analysis should compare the methods based on three key metrics: the **number of iterations** required, the final **approximate relative error**, and the **execution time**.
   d. Data structure used and how helpful was your choice.

6- Notes

Please note the following:

1- You should continue working in the team formed in phase 1.
2- Code plagiarism detection will be applied and if a violation is detected, the cheating policy will be applied.