CSE 224
Lab 04

Prepared By: Ridwan Mahbub, Lecturer, CSE, BUBT

Description:

This lab introduces students to the fundamental concepts of numerical analysis, focusing on measuring errors in numerical computations and implementing the bisection method for solving equations. Students will gain hands-on experience in quantifying errors, understanding the sources of errors, and applying the bisection method to approximate solutions for nonlinear equations.

Objectives:

- Understand the significance of numerical analysis in solving real-world problems.
- Learn to measure and analyze different types of errors in numerical computations.
- Gain proficiency in the bisection method as a technique for finding approximate solutions to equations.
- Develop programming skills to implement error measurement and the bisection method in practical scenarios.
- Apply the concepts learned to solve simple nonlinear equations.

Lab Content:

Definition of real error and absolute error

- Calculation of real error and absolute error with examples.
- Understanding Pros and Cons associated with each error.

- Concept of significant digits

The Bisection Method: Theory and Implementation

- Explanation of the bisection method as an iterative root-finding technique.
- Algorithm and steps involved in the bisection method.
- Coding guidelines and best practices for implementing the bisection method
- Advantages and disadvantages of the Bisection Method.

Implementing Error Measurement and the Bisection Method ● Writing a program to measure errors in numerical calculations. ● Implementing the bisection method to approximate solutions of nonlinear equations.

- Analyzing the convergence behavior of the bisection method.

Tasks:

1. Define a function for the following equation as a function file named **functionDemo.m**:

   $$f(x) = x^3 - 23x^2 + 142x - 120$$

2. Complete the following code for bisection method:

```
function [root, iterations, errors] = bisection_method(guess1, guess2,
max_iterations, tolerance)
    if functionDemo(guess1) * functionDemo(guess2) >= 0
        error("The guess does not satisfy the required conditions");
    end

    iterations = 0;
    errors = [];
    prevGuess = 0;

    while iterations < max_iterations
    %
    % Insert code for bisection method
    %
    end
endfunction



%Example of calling the function in the command window
[root, iterations, errors] = bisection_method(1
1, 20, 100, 1e-6);
```

3. Plot the relative errors in a graph with the number of iterations.