LAB # 05

02/11/25

1 Overview

This lab focuses on improving Newton's method as to make it more robust. To complete this lab, there is both a pre-lab section outlined below in addition to the in-lab exercises.

2 Pre-Lab

The pre-lab consists of filling out and completing the following chart. The blank chart was provided on the last page of the lab assignment.

Method:	Input:	Iteration:	Idea:
Bisection	Function $f(x)$ Interval $[a,b]$ Tolerance	$d = \frac{a+b}{2}$	Halves interval containing root
Fixed Point	Function $g(x)$ Initial Guess x_0 Interval $[a,b]$ Tolerance	$x_{n+1} = g(x_n)$	Iterates Equation
Newton	Function $f(x)$ Derivative $f'(x)$ Initial Guess x_0 Tolerance	$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$	Uses tangent to refine guess
Secant	Function $f(x)$ Initial Guesses x_0, x_1 Tolerance	$x_{n+1} = x_n - f(x_n)\frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})}$	Approximates derivative

Table 1: Part I of Table.

Method:	Requirements:	Pros:	Cons:
Bisection	Sign change over $[a,b]$	Always Converges Simple implementation	Very Slow Comparatively
Fixed Point	$g(x)$ is continuous	Faster	Convergence not guaranteed
Newton	$f'(x) \neq 0$ close x_0	Quadratic Convergence	Requires Derivative
Secant	Root must exist over $[a,b]$	No derivative	May not converge

Table 2: Part II of Table

3 Constructing a root finder

3.1 Exercises

As shown above in the pre-lab table **??**, each method has its own set of advantages and disadvantages. As a result, different methods perform better in certain situations compared to others. That being said, this lab focuses on creating a more robust version of Newton's method. The following exercises walks through the process of designing and building this new method.

3.1.1 Basin of convergence

In this part, we are tasked with creating a condition for our implementation of Newton's method which guarantees that the method will converge to a unique root given an initial guess in the neighborhood of the root we are tasked with finding. I used the following condition.

$$|f'(x)| < 1$$

Therefore, the basin of convergence consists of values of x such that $-1 < f'(x) < 1$.

3.1.2 Implementing a stopping bisection

For this part, the bisection sample code provided from class was used and modified. I added a function call to another function that checks if the current midpoint falls in the Newton's method basin of convergence within the bisection method while loop. If it does, it exits the bisection method and switches to the newton method. Otherwise, it continues as normal. The code can be found in the GitHub repository under **Lab 05.py**.

3.1.3 Different parameters

The original bisection sample code asks for a function $f(x)$, an interval $[a, b]$, a tolerance, and a max number of iterations. The new version, as shown in the code in the repository, requires the user to pass the derivative $f'(x)$ as well. The derivative is also passed as a parameter in order to check the basin of convergence condition as well as initiate the Newton's method function when the switch occurs.

3.1.4 Remaining Questions

Due to unforeseen issues while developing the aforementioned code as well as the time constraints of lab, I was not able to finish the exercises in this section.

3.2 Additional Exercise

Due to unforeseen issues while developing the aforementioned code as well as the time constraints of lab, I was not able to finish the exercises in this section.

4 Deliverables

The python code for lab, the code for this LaTeX rendering, as well as the rendering itself can be found in the GitHub repository under **Lab 05**.