APPM 4600 Lab 5
Creating more robust root finding methods

1 Overview

In this lab, you will improve Newton's method so that it will be more robust; i.e. open to less sensitive initial conditions. The pre-lab has you filling out a chart about the different root finding methods you learned about in class. Part of this chart is determining the pros and cons of each method. If time allows, you will build your own hybrid root finding method out of the methods you have learned about in this class.

2 Before lab

Complete the table on the last page of this lab report document.

3 Constructing a robust and rapidly convergent root finder

As you saw in the Before Lab section all of the methods have advantages and limitations. In this lab, we are going to build a more robust version of Newton's method avoiding the daunting task of knowing if your initial guess is "close enough" to the root. This is asking a lot of a user. The exercises in this section will walk you through the process of designing and building this new method.

3.1 Exercises

1. Viewing Newton's method as a fixed point method, derive a condition which guarantees that Newton's method will converge to a unique root for all initial guess in a neighborhood of the root.
   Hint: This will involve derivatives of the function f which you are trying to find the root of.
   The condition you find is what is called the *basin of convergence* for Newton's method.

2. Build your own Bisection code which terminates when the midpoint lies in the basin of convergence for Newton's method. *(You can start with the code from class.)*

3. Do you need to change the input of the original bisection method? If so, how did it change?

4. Appending to your modified bisection code put the Newton's method code but set it up so that takes the midpoint found by the bisection method as input. *(These should be put in one subroutine.)*

5. What are the advantages of your new fancy method? What are the limitations?

6. Consider the function $f(x) = e^{x^2+7x-30} - 1$. $x = 3$ is a root of this function. Apply the methods below with the specified initial data.
   (a) Bisection with $[a, b] = [2, 4.5]$.
   (b) Newton's method with $x_0 = 4.5$.
   (c) Your new hybrid method with $[a, b] = [2, 4.5]$.

How many iterations did each method take? Which method converged the fastest? Which method is the most cost effective? (Count function evaluations needed until convergence)

3.2 Additional exercise

If you finish early, pick one of the cons of the root finding methods we studied in class and try to build a new method that avoids that con without introducing *too* many new requirements.

4 Deliverable

Push your hybrid root finding method to Git and report your solutions to the questions in the exercise section on Canvas.

Chart for root finding

Method	Input	Iteration	Idea behind method	Required for convergence	Pros	Cons
Bisection						
Fixed point						
Newton						
Secant						