EE 242 Numerical Methods for Electrical Engineering

Project 3: Secant and Bisection Methods

(-10% if late by 1-2 days, -20% if late by 3-7 days, -100% if late by >8 days)

Project Instructions:

• For this project, you will work alone. No collaborations of any sort will be allowed with others. Any violation, regardless of the scope, will be directly referred to the department's Ethical Commission.

• You will submit your program source code (fully commented and documented) to CIMS. You should turn in the program source code, not the project file; i.e. upload the ***.cpp** file)

• Your project will not only be graded on whether it works or not, but also on whether it has good programming style or not. Object oriented programming (OOP) is a must, and your code will be graded according to that.

• You should turn in:

– *Source code:* Fully commented. You should be explicit in your comments. An educated person reading your code should clearly understand the purpose of each line. Executable or object files are not accepted. The file name should be **source.cpp**

– *A Readme file:* A short file named **readme.txt** containing information regarding how to compile and run your program including the necessary arguments. If your program is incomplete, this should be indicated in the beginning of the Readme file.

Important: You should upload a **.zip file with only your name and surname as the file name**, containing your source.cpp and readme.txt.

DO NOT upload .zip files with arbitrary names, or you will be penalized.

Project Goals:

In this project you will be implementing secant and bisection algorithms in order to solve $f(x) = 0$ for any given polynomial f.

Your program should take the coefficients of the function, initial guesses and the tolerance value as command line arguments and return the resulting values of x as well as the numbers of iterations for each method. For the bisection method, your program should give the warning that your algorithm doesn't work if the signs of initial guesses are same. You should implement both methods separately first. Then you should use a hybrid method where you start with bisection methods for the first two iterations and then continue with secant method for the rest of the iterations. Your program should print out the number of iterations required for each of the 3 methods (i.e., bisection, secant, and hybrid). If the number of iterations for each method exceeds 15, your program should be terminated and it should give notes as “ the number of iterations exceeded the threshold. ”

Project Details:

- You have n+1 command line inputs for the coefficients of $f(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0$ in the order from a_n to a_0. Use dynamically allocated memory to store these.
- You have 3 more command line arguments for the initial guesses x_0, x_1 ($x_1 > x_0$) and the tolerance value tol.

Example:
The command line arguments are:
1.5 1.8 0.001 2 2 -7 1 -7
(lower initial value, higher initial value, tolerance, and the coefficients)
Your program should solve $2x^4 + 2x^3 - 7x^2 + x - 7 = 0$ and find a root close to 1.67. As an output, for each of the three methods, your program should print i) the root found by it ii) the number of iterations required for the method to converge.

Note: Please do not use functions or expressions for setting the output decimal precision. They truncate or round the result. This may cause false positive or false negatives (i.e. Do not use setprecision function). Show the result as it is.