Project 3: Secant and Bisection Methods

Correspondent TA: Ramazan Yılmaz

Please refer your questions to *ramazan.yilmaz1@boun.edu.tr*.

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 (fully commented and documented) to Moodle. Late submission penalty is 20% for up to one week after the deadline. No credits will be given for late submissions beyond one week.

• You will write in C/C++. You can use Dev-C++ as a compiler or any other compiler you wish. You can download Dev-C++ from: http://dev-c.en.malavida.com/

• Your project will not only be graded on whether it works or not, but also on whether it has good programming style.

• You must turn in:

– *Source codes(.cpp):* Fully commented. You must be explicit in your comments. An educated person reading your code should clearly understand the purpose of each line. Executable (.exe) or object (.obj) files are not accepted.

– *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 must be indicated in the beginning of the Readme file.

All submitted files must be in a single .zip file (not .rar), whose name contains your complete name. (PLEASE DO NOT SUBMIT ANY OTHER FILES THAN SPECIFIED!)

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.

You should implement both methods separately first. Then you should use a hybrid method where you start with bisection method for the first two iterations and then continue with the 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).

Programming Details:

• You have n+1 command line inputs for the coefficients of

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0.$$

The coefficients will be inputted in the order of $a_n \rightarrow a_0$. The value of *n* will not be given. Use dynamically allocated memory to store the coefficients.

• You have 3 more command line arguments for the initial guesses x_0, x_1 for $x_1 > x_0$ and the tolerance value tol.

Example:

The command line arguments are:

2 2 -7 1 -7 1.5 1.8 0.001

Your program must solve $2x^4 + 2x^3 - 7x^2 + x - 7 = 0$ with $x_0 = 1.5, x_1 = 1.8$, $tol = 0.001$, and find a root close to 1.67.