Cairo University
SBE 2250 Numerical Methods in Biomedical Engineering - Spring 2021

Task 2 Roots and Optimization

Read Chapters 5, 6, and 7 in the course textbook: Applied Numerical Methods with MATLAB® for Engineers and Scientists, Fourth Edition, 2017. Include all Python code snippets as well as any computed results and figures in your submission.

Part I. Roots - Bracketing Methods

1. Solve **Problem 5.14 page 161** in the textbook. Plot the graphical solution (as a solid line) and the computed iterations (as asterisks '*') in Python.

2. Solve **Problem 5.18 page 162** in the textbook. Plot the graphical solution as well as the computed iterations in Python.

Part II. Roots - Open Methods

3. Solve **Problem 6.19 page 192** in the textbook. Instead of using the *fzero* function, develop the solutions in Python using a graphical approach, the Newton-Raphson method, the secant method, and the modified secant method. Plot the computed iterations of each method with initial guesses of 1 and 1000. Create a suitable figure legend.

Part III. Optimization

4. Implement in Python the golden-section search algorithm for finding the optimum value of a one-dimensional unimodal **polynomial** function, with stopping criteria of user-specified relative error and maximum number of iterations. Set appropriate default values for these stopping criteria.
   The inputs to your Python function should be:
   a) An array of the polynomial coefficients:
      Ex 1: For f (x) =a + bx + cx^2 + dx^3 + ex^4, input the list [a , b ,c ,d ,e].
      Ex 2: For f (x) = 4x − $1.8x^2$ + $1.2x^3$ − $0.3x^4$, input the list [0 , 4 ,-1.8 ,1.2 ,-0.3].
   b) The lower bracket value, x_l.

c) The upper bracket value, x_u.
d) The relative error, $\mathcal{E}_s$.
e) The maximum number of iterations, N_{max}.

Use your implementation to solve **Problem 7.7 page 215**. Compare your manual calculations with the Python-based results. Comment on the convergence of your results. Also, create a figure in Python. Plot the function *f* (*x*) in red, all iterations of the golden-section search method in green, and all iterations of the parabolic interpolation method in blue.

This is a team task (with 6-7 students per team). Each team should submit one PDF file on Blackboard. The submission should contain the following: The Python code (include the code in the PDF and also as well-documented working Python scripts)

1. The output from the code

2. All hand calculations

3. All hand calculations

Submission Deadline: 11:59 PM on June 1, 2021