Homework 03

Out of problem 1,2, and 3 chose only one to implement. Problems 4, 5, and 6 are to be done unconditionally.

With your email submission attach listings of each function which you implement except may be for problem 6.

General requirements:

1. all root finding functions must have optional outputs with the function value at solution point, and number of iterations. So the general root finding function definition should look like
   `function [x_sol, f_at_x_sol, N_iterations] = find_root_method(f_handle,...)`

2. all relevant input parameters should be validated against possible user errors.

3. Test your implementation with $f(x) = \exp(x) - 5$ at initial bracket [0,3]

   - Where ever the initial bracket is not applied (for example Newton-Raphson algorithm) use the right limit of the test bracket as a starting point of the algorithm.

4. All methods should be tested for the following parameters eps_f=1e-8 and eps_x=1e-10.

Problem 1 - optional (5 points)

Write proper implementation of the bisection algorithm. Define your function as
`function [x_sol, f_at_x_sol, N_iterations] =bisection(f, xn, xp, eps_f, eps_x)`

Problem 2 - optional (5 points)

Write proper implementation of the false position algorithm. Define your function as
`function [x_sol, f_at_x_sol, N_iterations] = regula_falsi(f, xn, xp, eps_f, eps_x)`

Problem 3 - optional (5 points)

Write proper implementation of the secant algorithm. Define your function as
`function [x_sol, f_at_x_sol, N_iterations] = secant(f, x1, x2, eps_f, eps_x)`

Problem 4 (5 points)

Write proper implementation of Newton-Raphson algorithm. Define your function as
`function [x_sol, f_at_x_sol, N_iterations] = NewtonRaphson(f, xstart, eps_f, eps_x, df_handle)` Note that `df_handle` is a handle to calculate derivative of the function `f` it could be either analytical representation of $f'(x)$ or its numerical estimate via the central difference formula.

Problem 5 (5 points)

Write proper implementation of Ridders' algorithm. Define your function as
`function [x_sol, f_at_x_sol, N_iterations] = Ridders(f, x1, x2, eps_f, eps_x)`

Problem 6 (5 points)

For each method find the root of the following two functions

1. $f1(x) = \cos(x) - x$ with the 'x' initial bracket [0,1]
2. $f2(x) = \tanh(x - \pi)$ with the 'x' initial bracket [-10,10]

Make a comparison table for the above algorithms with following rows

1. Method name
2. root of $f1(x)$
3. initial bracket or starting value used for $f1$
4. Number of iterations to solve $f1$
5. root of $f2(x)$
6. initial bracket or starting value used for $f2$
7. Number of iterations to solve $f2$

make columns corresponding to 3 algorithms which you chose to implement.

If algorithm diverges with suggested initial bracket, indicate so and appropriately modify the bracket, indicate modified bracket in the above table as well. Make your conclusions about speed and robustness of the methods

Bonus problem 7 (5 points)

Plot the $\log_{10}$ of the absolute error of $\sin(x)$ derivative at $x = \pi/4$ calculated with forward and central difference methods vs the $\log_{10}$ the h value. See `loglog` help for plotting with logarithmic axes. The values of h should cover the range $10^{-16} \cdots 10^{-1}$ (see Matlab `logspace` function designed for such cases).

Why error decreases as h goes down and then start to increase?

The minimum of the absolute error indicates optimal values of h.