Homework 1. Due on Jan. 25

In this homework, we study the convergence of interval bisection method, Newton's method, and globally convergent Newton's method for solving a nonlinear equation of one variable.

1. First use the MATLAB built-in function *fzero* to find the solution of the nonlinear equation

$$e^{2\sin(x)} - x = 0. \tag{1}$$

Plot the function $y = e^{2\sin(x)} - x$ on the interval $x \in [-10,\ 10]$.

Write a MATLAB function which returns the function value $f(x) = e^{2\sin(x)} - x$ and its derivative (if needed; see *nargout* in MATLAB) for a given input x.

2. Write a MATLAB function which implements the Interval Bisection method to solve (1).

In the implementation, you can choose the initial interval as, for example, $[-8,\ 8]$. The iteration can be stopped when $|f(x_k)| \leq 10^{-12}$, at iterate x_k. To evaluate the function value $f(x)$, call the function written in Question **1**.

Does the interval bisection method find the same root as obtained from *fzero*?

3. Write a MATLAB function to implement the Newton's method to solve (1). The stopping criterion is also when $|f(x_k)| \leq 10^{-12}$. In the algorithm, both the function value and its derivative are needed, which can be obtained by using the function written in Question **1**.

Experiment different initial guesses in the Newton's method, e.g., $x_0 = -6, -4, -2, 0, 2, 4, 6$, and discuss how the convergence depends on the initial guess.

4. Write a MATLAB function to implement the globally convergent Newton's method to solve (1). In this algorithm, the Newton's method need to be combined with the interval bisection method, as discussed in the lecture.

In the implementation, an initial interval, e.g., $[-8,\ 8]$, need be provided. The initial guess x_0 can be taken either as the left or the right end point of the initial interval.

Does this algorithm improve the convergence significantly compared with the original Newton's method in Question **3**, for certain initial guesses used there?

5. Adjust the above functions such that for each of the three methods, save the errors $|e_k|$ for all the iterates into a vector. Draw the three error vectors for the three algorithms, respectively, in the same graph.

Generate such graphs corresponding to different initial guesses used in the Newton's method, e.g., $x_0 = -6, -4, -2, 0, 2, 4, 6$, as done in Question **3**, respectively.

From those graphs, what can you say about the performances of the three algorithms? Which algorithm is both robust and fast?