The goal of this assignment is to explore a few methods of root-finding, that is, methods for solving
\[ f(x) = 0. \]  
(1)
We will investigate their convergence rate/order, sensitivity and pitfalls. We assume that there exists a solution (root) \( x \) of (1). In fact, we seek a solution of the problem
\[ x = \exp(-x) \]  
(2)
which is set up in the form (1) using either \( f_1(x) = x - \exp(-x) \), or \( f_2(x) = x\exp(x) - 1 \), or \( f_3(x) = 1/x - \exp(x) \).

**Notes and additional info:** Let \( x \) be the true solution to (1). We say that a method has convergence rate \( C/\text{order} \alpha \) if \( \varepsilon_{n+1} \leq C\varepsilon_n^\alpha \) for iterates \( n = 1, 2, 3, \ldots \). A good estimate for \( \alpha \) can be provided by calculating
\[ \alpha \approx \frac{\log(\varepsilon_{n+1}) - \log(\varepsilon_n)}{\log(\varepsilon_n) - \log(\varepsilon_{n-1})} \]
for a few subsequent iterations. Note that in order to do that, one must know the exact solution \( x \). If it is not known, then \( \varepsilon_n \) cannot be calculated exactly. In Newton (and secant) method a good estimate for \( \varepsilon_n \) is \( \varepsilon_n \approx v_n = x_{n+1} - x_n \).

1. Plot \( y = x, y = e^{-x} \) and estimate where these curves intersect (in class we guessed it happens somewhere in the interval (0, 1)). Plot the functions \( f_i(x) \) and estimate the solution to the equation from the graph. Are your answers consistent? Are there any other solutions to (2) (consider a sufficiently large interval)? Discuss smoothness of each function \( f_i \), critical points etc., and anything else that may be relevant to solving (2) numerically. BE AS CONCISE AS YOU CAN IN YOUR ANSWERS and in # OF PLOTS.

2. Use bisection method to solve the problem (1). Note: you can use Atkinson/Han’s MATLAB code. I DO NOT NEED TO SEE YOUR CODE.
   a) Use interval [0,1] with eps=1.e-5 and max # iterations 25. Try it for \( f_1, f_2, f_3 \). Is there a problem for any of \( f_i \)? Explain why. Modify the setup to eliminate this problem.
   b) Predict how many iterations it will take to solve a) for \( f_1 \) on the same interval with precision 1e-6, b) on the interval [0.2,0.7] with precision 1e-5. Why? Verify.
   c) Find the solution using the smallest reasonable precision. Show all decimal and hex digits of the solution: call it \( \text{xbis} \). Is \( f_i(\text{xbis}) = 0? \) Comment.

3. Use Newton method to solve the problem. Note: you can use Atkinson/Han’s MATLAB code but you may have to modify it. DO NOT SHOW THE CODE HERE.
   Explore sensitivity to the starting point: try using \( x_0 = 1.0, 0.0, -1.0, 10.0 \) for each \( f_i \). Use tolerance eps=1.e-10. Problems? Discuss.
Pick one “good” starting point and one “good” $f_i$ and find $x_{newt}$ by iterating to “machine epsilon”. Is $f_i(x_{newt}) = 0$? Compare $x_{newt}$ and $xbis$ (what is the “right” method of comparison). Discuss.

4. Investigate convergence rate of the Newton method: assume $x$ is equal $x_{newt}$. Is $\alpha$ really equal 2 as predicted by theory? What if we do not know $x$ and only use estimates $v_n$ of $\epsilon_n$?

**Note:** here you have to write your own code or modify Atkinson/Han’s M-file `newton.m` and make it deliver, for each iteration $n$, the error $\epsilon_n$ and its estimate $v_n$. SHOW THE CODE.

5. Write your own loop or routine to find the solution of a problem $x = \exp(-\beta x)$ using the fixed-point method with the initial guess $x_0 = 0$. SHOW THE CODE. Try $\beta = 10, 2, 1, 0.1$. For what $\beta$ are we guaranteed that the fixed-point method converges? Does it actually converge? Estimate the convergence rate (if applicable) for $\beta = 0.1$.

**Extra:** do #4 with secant method.