HOMEWORK #1

Due: October 1 (Thursday) by midnight

Instructions:

- The assignment consists of *two* questions, worth 3 and 6 points.
- Submit your assignment *electronically* to the Email address `math3q03@math.mcmaster.ca`; hardcopy submissions will not be accepted.
- It is obligatory to use the MATLAB template file available at `http://www.math.mcmaster.ca/bprotas/MATH3Q03/template.m` (see also the link in the "Computer Programs" section of the course website); submissions non compliant with this template will not be accepted.
- All plots should have suitable axis labels and legends.
- Make sure to enter your name and student I.D. number in the appropriate section of the template.
- Late submissions and submissions which do not comply with these guidelines will not be accepted.

---

1. For each of the three numbers $a_1 = 10^{-3}$, $a_2 = 1$ and $a_3 = 10^3$ determine the relative round-off error of the addition operation defined as follows

$$E = \frac{|\epsilon - [(a_i + \epsilon) - a_i]|}{\epsilon}, \qquad i = 1, 2, 3, \tag{1}$$

where $\epsilon \in [10^{-20}, \dots, 10^{-10}]$. Plot the error E as a function of ϵ in log-linear coordinates using the function `logspace` to represent ϵ with 100 samples. The plots of errors in the three cases ($i = 1, 2, 3$) should all appear on figure 1 and be distinguished using different symbols and colors. For each a_i, $i = 1, 2, 3$, estimate the value of ϵ for which the entire arithmetic precision is lost and save it in the variables `Answer1`, `Answer2` and `Answer3`. Your MATLAB code must not use the function `eps`.
(3 points)

2. Consider the function $f(x) = e^{-x} - \sin(x)$ and write a MATLAB code that will perform the following tasks:

   (a) find the root x^* of the equation $f(x) = 0$ in the interval $\Omega = [0, 1]$ with the bisection, secant and fixed-point methods using the codes posted on the course website (i.e., `bisect.m`, `secant.m` and `fixpoint.m`); use suitable starting points to initialize the iterations and print the approximate solutions obtained with each of the methods,

   (b) add your own MATLAB code implementing Newton's method to solve this problem; print the approximate solution obtained,

   (c) plot the function $f(x)$ for $x \in \Omega$ with a solid line and mark the data points $\{x_n, f(x_n)\}$ obtained at the different iterations $n = 1, 2, \dots$ with the four methods in points (a) and (b); use distinct symbols and colors for each of the methods; this plot should appear as figure 2,

(d) plot of the quantity $|x_n - x^*|$ as a function of n (the iteration index) for the results obtained using the four methods mentioned above; use different symbols to mark data points corresponding to the different methods and use linear-log coordinates; as the "exact" solution use $x^* = 0.58853274398186107743$; this plot should appear as figure 3.

Note that you may need to modify the MATLAB codes provided on the course website to ensure that they return suitable data. In problems (a) and (b) limit the number of iterations to 100. When using the fixed-point iterations, transform the problem to the form $g(x) = x$, where $g(x) = f(x) + x$.
(6 points)