Quiz (II)
Finished by 18:30 on 5/4

1. Create a matlab script and change the filename to F7xxxxxx_quiz2.m. Link all the programs to solve following problems to this script. Make sure once type the filename' F7xxxxxx_quiz2', the results of the following problems will pop-up automatically in order. Remember not to type any 'clear all ', 'close all' command in any of the codes.

2. [F7xxxxxx_quiz2_prob2.m] Try to solve the roots of each quadratic equation using the original form and the numerator-rationalized form. Print out all the results for at least up to 16 decimal digits.
(1) $x^2 - (500 + k)x + 500k = 0 \ \ for\ k = 10^{-2}, 1\,0^{-4}, \ldots, 1\,0^{-10}$

(2) $0.0001x^2 + 30.0003x + 90 = 0$
(3) $x^2 + 300003\mathrm{x} + 900000 = 0$

(4) $x^2 - 302 \times 10^{152}x - 6 \times 10^{306} = 0$

After the comparison, how do you suggest to solve quadratic equation with good precision? [Write your comments on a txt or doc file]

3. [F7xxxxxx_quiz2_prob3.m] (1) Write two functions defined as follows to calculate e^{x} and e^{-x} for x > 0 based on Taylor's Expansion:

$$\mathrm{A(N,x)} = \Sigma_{\mathrm{k=0}}^{\mathrm{N}} \left(\frac{\mathrm{x}^{\mathrm{k}}}{\mathrm{k!}}\right) \cong \mathrm{e}^{\mathrm{x}}$$

$$B(\mathrm{N}, x) = \Sigma_{\mathrm{k=0}}^{\mathrm{N}} (-1)^k \left(\frac{x^k}{k!}\right) \cong e^{-x}$$

(2) Estimate $e^{-0.01}$ using the two functions with different N (N=0,1,2,…), the order of the expansion. e.g. $e^{-0.01} \cong 1/A(20,0.01) \cong B(20,0.01)$. Compare the result with Matlab's function exp(-0.01). How many terms does it require for the result to converge with accuracy up to 8 significant digits?

(3) How many terms does it require for e^{-1} to converge with accuracy up to 8 significant digits using 1/A and B respectively?

(4) Repeat (2) to calculate e^{-20} using $1/A(N, 20)$ and B(N,20) with N = 16,17,18,19,20,21, and 22.

i. Explain if there is any strange result showing up.

ii. When the computation converges, the two results are different. If you don't compare the two results with Matlab's embedded function, can you tell which one is more accurate and why?

iii. After the comparisons, could you propose to an exponential function with good precision?

4. [F7xxxxxxx_quiz2_prob4.m] Write a program to find the approximated machine epsilon of the double precision floating point number instead of just using eps from Matlab's embedded function. [BONUS]