Exercises: (Chapter 1 and Chapter 2 §1, §2, and §3)

1. Download https://math.msu.edu/~iwenmark/Teaching/MTH451/HWandCode/TaylorExample.zip and then modify the Matlab script "TaylorExample.m" to approximate $f(x) = \cos(\exp(2x))$.[1] You should graph the first, second, and third degree Taylor polynomial approximations of f (based at 0) over the interval $(-.5, .5)$, together with f itself. If you modify the MATLAB script "TaylorExample.m" to do this, make sure to either disable the portion of the code that reads in and plots "TaylorExamp.txt", or, better yet, modify the C code that produces "TaylorExamp.txt" too so that it produces a new "TaylorExamp.txt" that contains the correct third degree Taylor polynomial for this f. **In any case, turn in a printout of both your graph, and of the (modified) code you used to produce it**.

2. Download and run "Round.m" in MATLAB (available at https://math.msu.edu/~iwenmark/Teaching/MTH451/HWandCode/HW2scripts.zip).[2] The script approximates the value $2^k\pi$ to single precision accuracy for each of $k = 1, 2, 3, \ldots, 20$ in two different ways:

   (a) "Approx1" is calculated by computing $2^k\pi$ directly (note that this can be implemented by simply shifting the decimal point in the binary representation of π back k times)

   (b) "Approx2" is calculated by adding π to itself $2^k - 1$ times

   The plot produced by the script contains two relative error curves: the relative error $E1(k) = \frac{|\text{Approx1}-2^k\pi|}{2^k\pi}$ as a function of k, and the relative error $E2(k) = \frac{|\text{Approx2}-2^k\pi|}{2^k\pi}$ as a function of k. Which error curve increases fastest as k increases (i.e., which approximation is least accurate), one or two? **Explain why the worse approximation is so much worse.**

3. Compute the dot product of the following two vectors (available already typed in into MATLAB at https://math.msu.edu/~iwenmark/Teaching/MTH451/HWandCode/HW2scripts.zip):

   $$x = [2.718281828, -3.141592654, 1.414213562, 0.5772156649, 0.3010299957], \text{ and}$$
   $$y = [1486.2497, 8.78366.9879, -22.37492, 4773714.647, 0.000185049].$$

   Compute the summation in four ways:

   (a) Forward order $\sum_{i=1}^{n} x_i y_i$

   (b) Reverse order $\sum_{i=n}^{1} x_i y_i$

   (c) Largest-to-smallest order (add all positive numbers in order from largest to smallest, then add negative numbers in order from smallest to largest, and then add the two partial sums)

   (d) Smallest-to-largest order (reverse the order of the adding in the previous method)

   Use both single ("single()" in MATLAB) and double ("double()" in MATLAB) precision for a total of eight answers. Compare the results with the *correct* value to seven decimal places, $1.006571 * 10^{-9}$. Which method of computing the dot product is most accurate? Which method of computing the dot product is least accurate? **Explain your answers**.

4. Using the definition $\sinh(x) := \frac{\exp(x)-\exp(-x)}{2}$, discuss the problem of computing $\sinh(x)$ without loss of significance. For what values of $x \in \mathbb{R}$ will loss of significance be an issue? How can you modify the computation of $\sinh(x)$ for those values of x to avoid loss of significance (e.g., write out a truncated series you could use to compute $\sinh(x)$ to help you keep significance for those values of x)?

5. The condition number of the function $f(x) = x^\alpha$ is independent of x. What is the condition number?

6. What is the condition number of $\ln(x)$? For what values of $x \in \mathbb{R}$ is it largest?

[1] Alternatively, as always, you may instead recode this script in a different coding language of your choice.

[2] Alternatively, as always, you may instead recode this script in a different coding language of your choice.