CSCI3656: Numerical Computation
Homework 1: Due 5pm on Friday, Sep. 11

Turn in your own writeup that includes your own code. List the names of the people you collaborated with. Submit an electronic copy (scanned, photo, PDF, ...) via Canvas.

1. Execute the following lines in an interpreter (Matlab or Python).

   ```
   >> format long
   >> x = 9.4
   >> y = x - 9
   >> z = y - 0.4
   ```

   What did you get for `z`? What should it be in exact arithmetic? Why is it not what it should be? (Hint: For a detailed description, see Chapter 0.3.3 in Sauer's *Numerical Analysis*.)

2. Consider the following two polynomials:

   $$\begin{aligned} p_1(x) &= (x-2)^9 \\ p_2(x) &= x^9 - 18x^8 + 144x^7 - 672x^6 + 2016x^5 - 4032x^4 + 5376x^3 - 4608x^2 + 2304x - 512 \end{aligned} \tag{1}$$

   Convince yourself that $p_1(x) = p_2(x)$ in exact arithmetic. (No need to show your work on this in the write-up).

   Given a polynomial expressed as

   $$p(x) \;=\; \sum_{i=0}^{n} a_i\, x^i, \tag{2}$$

   Horner's algorithm for evaluating the polynomial at some given x is: (i) initialize $p = a_n$, (ii) for $i = n-1$ down to 0, do $p = p * x + a_i$. Implement Horner's algorithm. (I'm using Matlab.)

   Note that $x = 2$ is a root of $p_1(x)$ and $p_2(x)$. Generate 8000 equally spaced points in the interval $[1.92, 2.08]$. (In Matlab, you can do this with `linspace`.) Evaluate and plot $p_1(x)$ at each point in the interval. In a separate figure, evaluate and plot $p_2(x)$ using Horner's algorithm. In exact arithmetic, these should be the same. What's going on in these plots? (Hint: For a detailed description, see Chapter 0.1 in Sauer's *Numerical Analysis* or Chapter 1.4 and surrounding text in Demmel's *Applied Numerical Linear Algebra*.)

3. Consider the functions

   $$f_1(x) = \frac{1-\cos(x)}{\sin^2(x)}, \qquad f_2(x) = \frac{1}{1+\cos(x)}. \tag{3}$$

   Using trig identities, show that $f_1(x) = f_2(x)$. (Please show your work on this one.) Implement f_1 and f_2. Make a table of your implementations evaluated at the points $x_k = 10^{-k}$ for $k = 0, 1, \dots, 12$. You should see that f_1 loses all accuracy as k increases (that is, as x_k approaches zero), while f_2 retains its accuracy. Explain why. (Hint: For a detailed description, see Chapter 0.4 in Sauer's *Numerical Analysis*.)