MATH-411 Numerical Analysis—Homework 1
Rochester Institute of Technology, Fall 2022

Due: Friday September 02, 2022 at 11.59pm EST.
Instructor: Blessing Emerenini
E-mail: boesma@rit.edu
Remark:

- All assignments are uploaded on MyCourses as pdf.
- For this assignment you can type your solution in Microsoft word and then convert to pdf **or** you can use Latex and automatically generate your pdf.
- Figure out how to upload your files on MyCourses before the due dates. Late homeworks are **not** accepted.
- You can discuss ideas on how to tackle the problems on **Piazza** but do not post solutions. Thanks.

Please show all your work clearly. If the assignment involves MATLAB, please turn in your code and figures as well.

1. Convert the following base 10 numbers to binary. Use overbar notation for nonterminating binary numbers.
   (a) 10.5, (b) 1/3, (c) 12.8

2. Find the IEEE double precision representation fl(x), and find the exact difference fl(x)-x for the given real numbers. Check that the relative rounding error is no more than $\frac{1}{2}\varepsilon_{mach}$. (a) $x = 2.75$, (b) $x = 2.7$, (c) $x = \dfrac{10}{3}$

3. A machine stores floating point numbers in 7-bit words. The first bit is stored for the sign of the number, the next three for the biased exponent and the next three for the magnitude of the mantissa. You are asked to represent 33.35 in the above word. The error you will get in this case would be
   (A) underflow
   (B) overflow
   (C) NaN
   (D) No error will be registered.
   Explain why this is the case.

4. Consider a binary floating-point number system containing numbers of the form

$$\pm 0.1d_1d_2 \times 2^e, \qquad -4 \le e \le 6,$$

   where $d_1, d_2 \in \{0,1\}$ are binary bits. Suppose that the system uses a conventional rounding to the nearest policy to convert a real number to its binary floating-point number and to do floating point arithmatic.

(a) What are the smallest and largest positive numbers (in decimal) in this floating point system?

(b) What is ε_{mach} in this system?

(c) What is the floating-point representation (in binary and decimal) of the number 9 in this system?

(d) Give an example that shows $fl(fl(a+b)+c) \neq fl(a+fl(b+c))$, where a,b,c are floating-point numbers contained in this system.

5. As you have likely seen in calculus and analysis, the Maclaurin series for $f(x) = e^{2x}$ converges for $-\infty < x < \infty$ and is given by

$$e^{2x} = \sum_{n=0}^{\infty} \frac{(2x)^n}{n!}$$

Let $A_n(x)$ be the nth sum of this series for a given x.
(a) Write a MATLAB program that calculates the terms of the series until the relative error is less than 10^{-10}. To write this program most efficiently, you can take advantage of the fact that the $n+1$st term is equal to the nth term times $\frac{2x}{n+1}$. Also, use the MATLAB exp command to calculate the 'true' values of e^{2x}. Finally, make sure you a maximum iteration threshold of 120 so that it does not run forever is the estimate is not converging. How many terms are needed to converge to this bound for $x = 3$, $x = -3$, and $x = -9$?
(b)Now change your stopping criteria in the previous program so that the program stops when $A_n(x) = A_{n+1}(x)$ (when doing this, please comment out the original stopping criteria and put the new on in). Obviously this never happens using infinite precision, but it will on a computer. Why?
(c) For $x = 15, 6, -6, -15$ find the following pieces of information: what is the estimate of e^{2x} generated by the series, what is the 'true' value, how many iterations does it take, and what are the absolute and relative errors?
(d) You should notice that the estimate for $x = -15$ is almost laughably bad. Why do you think the relative error is so much higher for negative values?