FIT3139: Lab questions for week 2

1. Write a program that accepts a real number as an input and then outputs a binary representation of it.

2. Write a script to compute the absolute and relative errors in Stirling's approximation of a factorial: $n! \approx \sqrt{(}2\pi n)(n/e)^n$ for $1 \leq n \leq 15$. What happens to the absolute error, does it grow or shrink? What about relative error?

3. Write a program to compute the exponential function e^x using the infinite series

   $$e^x = 1 + x + x^2/2! + x^3/3! + ...$$

   In real-world applications, we will not know that the true answer for a given value of a function, *a priori*. For these situations, an alternative is to normalize the error using the best available estimate of the true value. For iterative numerical evaluations of a function, the approximate relative error ϵ_a is computed as follows:

   $$\epsilon_a = \frac{\text{current approximation-previous approximation}}{\text{current approximation}} \times 100\%.$$

   For varying values of x in the range 0 to 1, print out the approximate relative errors.

   Since the series expansion is over infinite terms, you will have to think about when you want to stop the iterations. Explore various possibilities.

4. (This is not a programming question) Consider the function $f(x) = x^2$. Find its condition number (use the calculus based approximation introduced in the lecture). Is $f(x)$ well-conditioned? What does it say about its sensitivity.

   Repeat the exercise for the function $f(x) = \sqrt{x}$

5. Write a script to evaluate the sensitivity and conditioning of the function $f(x) = \tan(x)$ (Note, in this case $f'(x) = 1 + \tan^2(x)$). Plot a graph of x versus condition number.

6. Write a script which computes the smallest number ϵ such that $1 + \epsilon > 1$. This number is the machine epsilon. Compare this with the number generated by using the built-in function `eps` in MATLAB or `sys.float_info.epsilon` in Python.

7. As above, develop your own script to determine the smallest positive real number used in MATLAB. Base your algorithm on the

FIT3139: Lab questions for week 2

notion that your computer will be unable to reliably distinguish between zero and a quantity that is smaller than this number. Compare your answer with the built-in `realmin` in MATLAB, or `sys.float_info.min*sys.float_info.epsilon` in Python.

8. (Not a programming exercise) Compute the forward and backward error for the function $y = f(x) = \cos(x)$ for $x = 1$. Consider the function approximation of $\cos(x)$, $\tilde{y} = \tilde{f}(x) = 1 - \frac{x^2}{2}$

9. (Not a programming exercise) A floating point system is given by the numbers b, p, L and U.

   - What is the number of normalized floating-point numbers in a given floating-point system?
   - What is the smallest positive normalized floating-point number?
   - What is the largest positive normalized floating-point number?