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 (in radians) versus condition number. Beware of division by zeros!

   Repeat the same with the inverse trigonometric function $f(x) = \arctan(y)$. (Note, $f'(x) = 1/(1 + x^2)$).

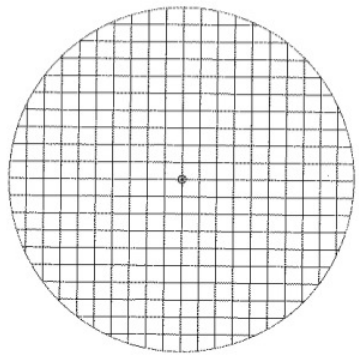
6. Write a script to find the overflow and underflow limits on the computer you are running.

7. 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.

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

9. To compute a planet's space coordinates, we have to solve the function: $f(x) = x - 1 - 0.5 \sin x$.
   Let the base point be $a = x_i = \pi/2$ on the interval $[0, \pi]$. Determine the highest order Taylor series expansion resulting in a maximum error of 0.015 on the specified interval. The error is equal to the absolute value of the difference between the given function and the specific Taylor series expansion.

10. Consider a circle of radius r of the form $x^2 + y^2 = r^2$. Divide them into squares of unit area (See figure below.) Clearly the circle's area is πr^2. If n is the number of "unclipped" squares within the circle, this number approximates the area of the circle: $n \approx \pi r^2$.



Write a script that accepts a radius r and displays the approximate computation of π as $\hat{\pi} \approx \frac{n}{r^2}$. Print also the absolute error $|\hat{\pi} - \pi|$

Modify the above script to accept a small number ($\Delta x < 1$) as an input and generates the approximation of the value of π based on the number of unclipped squares of area Δx^2 that fit inside a circle with unit radius. Print out the absolute and relative errors.

More questions if you have finished all of the above!

1. The derivative of $f(x) = 1/(1 - 3x^2)$ is given by $6x/(1 - 3x^2)^2$ Do you expect to have difficulties evaluating this function at $x = 0.577$ Try it using 3- and 4-digit arithmetic with chopping.

2. Use the Maclaurin series expansion of $\cos x$. (Try and work this out yourself.) Starting with the simplest zero-order approximation, add terms one at a time to estimate $\cos(\pi/4)$. After each new term is added compute the true and approximate percent relative errors. (Use built-in function in MATLAB to determine the true value.) Add terms until the absolute value of the approximate error estimate falls below an error criterion conforming to two significant figures.

3. Perform the same computation as above but for the Maclaurin series expansion for $\sin x$ to estimate $\sin(\pi/4)$.

4. If $|x| < 1$ it is known that $1/(1 - x) = 1 + x + x^2 + x^3 + \cdots$. Repeat above for $x = 0.1$.

5. Write a program to solve the quadratic equation $ax^2 + bx + c = 0$ using the standard quadratic formula:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

or the alternative formula

$$x = \frac{2c}{-b \mp \sqrt{b^2 - 4ac}}$$

Your program should accept the values for the coefficients a, b, c as input and produce the two roots of the equation as output. Your program should detect when the roots are not real and terminate. When should you use the two formulae?