Updated on August 31, 2021

Math 3607: Homework 2

Due: 10:00PM, Tuesday, September 7, 2021

TOTAL: 30 points

- Problems marked with are to be done by hand; those marked with are to be solved using a computer.
- **Important note.** Do not use *Symbolic Math Toolbox*. Any work done using `sym` or `syms` will receive NO credit.

1. (Leap year: Exercise 2, Lecture 3) A year is a *leap year* if it is a multiple of 4, except for years divisible by 100 but not by 400. In simpler terms, a non-century year is a leap year if it is divisible by 4; a century year is a leap year if it is divisible by 400.

   For example,

   - Last year (2020) was a leap year. (non-century year; divisible by 4)
   - 1900 was not a leap year. (century year; not divisible by 400)
   - 2000 was a leap year. (century year; divisible by 400)

   (a) Write a script which determines whether a given year is a leap year or not.

   (b) Run your script with the years in the example above.

   (c) (Optional) Use (or modify if you like) the script to find all leap years between 1900 and 2021.

2. (3-D coordinate conversion) Recall from calculus that Cartesian coordinates (x, y, z) in $\mathbb{R}^3$ are related to spherical coordinates (ρ, ϕ, θ) by

$$x = \rho \sin\phi \cos\theta, \quad y = \rho \sin\phi \sin\theta, \quad z = \rho\cos\phi,$$

   where $\phi \in [0, \pi]$ and $\theta \in [0, 2\pi)$.

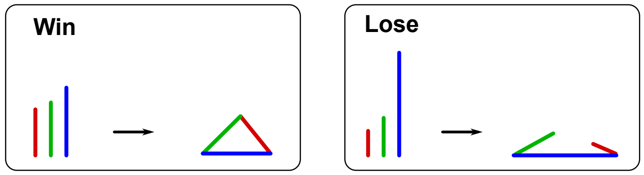
   (a) Write down the expressions for ρ, ϕ, θ in terms of x, y, z. Show your work.

   (b) Write a script which takes Cartesian coordinates as inputs and converts them to their spherical counterparts. (Convention: for points on the z-axis, set $\theta = 0$.)

   (c) Then run your script with the following inputs:

   - $(x, y, z) = (1, 2, -2)$
   - $(x, y, z) = (0, -3, 4)$
   - $(x, y, z) = (-5, -12, 1)$
   - $(x, y, z) = (0, 0, -7)$

3. (Game of 3-stick: Exercise 2, Lecture 4) In the game of 3-Stick, you pick three sticks each having a random length between 0 and 1. You win if you can form a triangle using three sticks; otherwise, you lose.



Simulate one million games. Use the results to estimate the probability of winning a game.

Note. For this problem, you do not need to write a separate script. Simply write your program in a single code block.

Note. The only output must be the summary of the million simulations. Use `disp` or `fprintf`.

4. (Gap of 10: modified from Exercise, Lecture 5) Simulate the tossing of a *biased* coin whose tails is 3 times more likely to be showing than its heads, until the gap between the number of heads and that of tails reaches 10. Print out the number of tosses needed. Then use a for-loop to repeat the simulation ten times. Your output should look like

```
Simulation  1 done in 17 tosses.
Simulation  2 done in 14 tosses.
Simulation  3 done in 18 tosses.
Simulation  4 done in 22 tosses.
Simulation  5 done in 48 tosses.
Simulation  6 done in 18 tosses.
Simulation  7 done in 12 tosses.
Simulation  8 done in 20 tosses.
Simulation  9 done in 16 tosses.
Simulation 10 done in 26 tosses.
```

Note. As in the previous problem, write your code in a single code block.

5. (Sequences Converging to π) Each of the following sequences converges to π:

$$a_n = \frac{6}{\sqrt{3}} \sum_{k=0}^{n} \frac{(-1)^k}{3^k(2k+1)},$$

$$b_n = 16 \sum_{k=0}^{n} \frac{(-1)^k}{5^{2k+1}(2k+1)} - 4 \sum_{k=0}^{n} \frac{(-1)^k}{239^{2k+1}(2k+1)}.$$

Write a program that prints $a_0, \ldots, a_{n_a}$, where n_a is the smallest integer so that $|a_{n_a} - \pi| \leq 10^{-6}$ and prints $b_0, \ldots, b_{n_b}$, where n_b is the smallest integer so that $|b_{n_b} - \pi| \leq 10^{-6}$.

Note. Write your program in a single code block.