

# Introducción a la Simulación Computacional

## Guía 1: Monte Carlo

Segundo Cuatrimestre de 2021

### Problema 1: Monte Carlo con muestreo directo

- a) Realice un programa para estimar el número  $\pi$  en base a la obtención de una serie pares  $(x, y)$ , donde  $x$  e  $y$  son obtenidos de una distribución uniforme de números pseudo-aleatorios entre 0 y 1.

Si se piensa en ejes cartesianos y que el primer cuadrante, en una área de  $A = 1 \times 1 = 1$  contiene un cuarto de círculo de radio  $r = 1$ . La superficie de éste es  $S = \frac{1}{4}\pi r^2 = \frac{1}{4}\pi$ . La fracción entre áreas será proporcional al cociente de pares  $(x, y)$  que caen dentro del círculo, respecto de todos los pares obtenidos.

$$\frac{N_{circ}}{N_{tot}} = \frac{\int_0^1 dx \sqrt{1-x^2}}{A} = \frac{S_{1/4\text{circ}}}{A} = \frac{\pi/4}{1}$$

**Nota:** Este problema puede pensarse como un ejemplo en el cuál se utiliza Monte Carlo para realizar una integral definida.

- b) Discutir por qué funciona el muestreo y en qué condiciones el muestreo directo no sería una buena opción para calcular el valor medio de una cantidad.

**Problema 2:** Se utilizará el método de Monte Carlo para crear un generador de números aleatorios con una distribución arbitraria  $f(x)$ .

Escriba un programa que genere  $N = 1000$  pares de numeros aleatorios  $(X, Y)$  a partir de la función *uni()* de *ziggurat.f90*, que cumplan  $0 < X < 2$  y  $0 < Y < 1$ .

- a) Utilizando los comandos *open*, *read* y *close*, logre que el programa lea de un archivo externo ('input.dat') la cantidad de puntos aleatorios  $N$  que se van a generar. No se olvide de crear el archivo 'input.dat' y escribir un numero entero en el.
- b) Defina la función de densidad de probabilidad  $f(x) = (x - 1)^2$ .
- c) Escriba una rutina que acepte (escriba a archivo) el número aleatorio  $X$ , si  $Y \leq f(x)$ . ¿Puede estimar que fracción de números será aceptado?
- d) Haga un histograma de los números  $X$  aceptados, normalícelo y compárelo con la función  $f(x)$ .
- e) Utilizando inteligentemente las funciones de *ziggurat* (*uni()*, *rnor()*, *rexp()*), realice un muestreo con *importance sampling*, para obtener valores de  $X$  con más probabilidad de ser aceptados.
- f) Compare la eficiencia ( $X$  aceptados / pares  $(X, Y)$  generados) y el tiempo de simulación, entre los métodos de fuerza bruta e *importance sampling*.

**Nota:** para que la comparación del tiempo de simulación tenga sentido, debe hacerse a igual número de aceptaciones.