

# Simulación

## Práctica 1: Método Monte-Carlo

Profesor: Ángel Isabel Moreno Saucedo  
Semestre Febrero - Junio 2021

### 1. Introducción

En esta práctica se calculará el valor de una integral definida. En el libro de Simulation de Ross, S.M. [2], muestra que una de las aplicaciones principales de los números aleatorios es el cálculo de integrales. Para esto se utilizará el método de aproximación de integrales. (método *Monte-Carlo*)

### 2. Método Monte-Carlo

Para el cálculo de la integral utilizaremos lo siguiente

$$\int_a^b g(x)dx \approx \sum_{i=1}^k \frac{(b-a)g(a+(b-a)u_i)}{k} \quad (1)$$

donde  $u_i \sim \mathcal{U}(0,1)$  y  $k$  es lo suficientemente grande. La integral

$$\int_3^7 \frac{1}{e^x + e^{-x}} dx \quad (2)$$

es la que aproximaremos a su solución. Los Cuadros (1) y (2) muestran el código en R y python, respectivamente, del método Monte-Carlo para aproximar el valor de la integral (2).

```
1 f <- function(x) { return(1 / (exp(x) + exp(-x))) }
2
3 simulacion_MC <- function(F, a, b, cantidad) {
4   acumulado <- 0
5   for (i in 1:cantidad) {
6     x <- runif(1)
7     acumulado <- acumulado + F(a + (b - a) * x)
8   }
9   return(((b - a) * acumulado) / cantidad)
10 }
11
12 desde <- 3
13 hasta <- 7
14 cantidad <- 50000
15 replicas <- 30
16
17 for (i in 1:replicas) {
18   integral <- simulacion_MC(f, desde, hasta, cantidad)
19   print(paste("Replica ", i, ": ", integral, sep = ""))
20 }
```

Cuadro 1: Código en R del método Monte-Carlo.

```

1 # -*- coding: utf-8 -*-
2 """
3 Created on Mon Feb 22 07:32:59 2021
4
5 @author: Angel Moreno
6 """
7
8 from math import exp
9 def f(x):
10     return 1 / (exp(x) + exp(-x))
11
12 import random
13 def simulacion_MC(F, a, b, cantidad):
14     acumulado = 0
15     for j in range(cantidad):
16         x = random.random()
17         acumulado += F(a + (b - a) * x)
18     return ((b - a) * acumulado)/cantidad
19
20 desde = 3
21 hasta = 7
22 cantidad = 50000
23 replicas = 30
24
25 for i in range(replicas):
26     integral = simulacion_MC(f, desde, hasta, cantidad)
27     print(f'Replica {i + 1}: {integral}')

```

Cuadro 2: Código en python del método Monte-Carlo.

El valor que tomases de referencia es el que nos da Wolfram Alfa[1] de 0.048834.

### 3. Tarea

Determina el tamaño de muestra requerido por cada lugar decimal de precisión del estimado obtenido para el integral, comparando con Wolfram Alpha para por lo menos desde uno hasta seis decimales. Elige tamaños de muestra de los valores aleatorios y realice réplicas para validar que cada vez que se ejecute el código nos garantice la precisión deseada. Realice un reporte donde explique la simulación y escriba las conclusiones, añada visualizaciones.

#### 3.1. Puntos Extra

Aproxime alguna integral definida en el intervalo  $[0, \infty]$ . Luego compare el resultado con el que arroja Wolfram Alfa [1]. Determine que muestra es la adecuada para aproximar a cinco dígitos con el resultados de Wolfram Alfa.

## Referencias

- [1] Wolfram Research, Inc. Mathematica, Version 12.2, 2020. URL <https://www.wolframalpha.com/calculators/integral-calculator>. Champaign, IL.
- [2] Sheldon M. Ross. *Simulation (2. ed.)*. PRENTICE HALL, 1999. ISBN 970-I7-0259-x.