Tarea 3

Ana Lucio, Ana Carbajal, Eduardo Rivera, César Luna, René Zamora

9 de septiembre de 2022

Resumen

En esta actividad se realizó una gráfica la cual describe el comportamiento acerca de cómo se comporta la aproximación al número pi, utilizando el programa python, también se vio lo que es el método Monte Carlo y cómo se puede utilizar.

1. Introducción

l término Monte Carlo se aplica a un conjunto de métodos matemáticos que se empezaron a usar en los 40s para el desarrollo de armas nucleares en Los Álamos. Consisten en resolver un problema mediante la invención de juegos de azar cuyo comportamiento simula algún fenómeno real gobernado por una distribución de probabilidad o sirve para realizar un cálculo. Más técnicamente, un Monte Carlo es un proceso estocástico numérico, es decir, una secuencia de estados cuya evolución viene determinada por sucesos aleatorios. Se utilizará este método junto con un lenguaje de alto nivel de programación; 'Phyton', este se emplea para desarrollar aplicaciones de todo tipo y en este caso lo usaremos para conocer el comportamiento de la aproximación al número pi.

2. Desarrollo

2.1. Método Monte Carlo

Los Métodos de Monte-Carlo son técnicas para analizar fenómenos por medio de algoritmos computacionales, que utilizan y dependen fundamentalmente de la generación de números aleatorios.

El uso de los métodos de Montecarlo como herramienta de investigación proviene del trabajo realizado en el desarrollo de la bomba atómica durante la Segunda Guerra Mundial en el Laboratorio Nacional de Los Álamos en EE. UU[3]. Este trabajo conllevaba la simulación de problemas probabilísticos de hidrodinámica concernientes a la difusión de neutrones en el material de fisión. Esta difusión posee un comportamiento eminentemente aleatorio. En la actualidad es parte fundamental de los algoritmos de raytracing para la generación de imágenes 3D.

El uso de este método es para aproximar el valor de Pi; consiste en dibujar un cuadrado y dentro de ese cuadrado dibujar un círculo con diámetro de igual medición que uno de los lados del cuadrado, después se dibujan puntos de manera aleatoria sobre la superficie dibujada.

1. Dibuja un círculo unitario, y al cuadrado de lado 2 que lo inscribe.
2. Lanza un número $nndepuntosaleatoriosuniformesdentrodelcuadrado.Cuentaelnúmerodepuntosdentrodelcírculo, i.e.punt$
3. El cociente de los puntos dentro del círculo dividido entre n es un estimado de, pi/4. Multiplica el resultado por 4 para estimar pi.

En este cálculo se tienen que hacer dos consideraciones importantes:

- Si los puntos no están uniformemente distribuidos, el método es inválido.
- La aproximación será pobre si solo se lanzan unos pocos puntos. En promedio, la aproximación mejora conforme se aumenta el número de puntos.

***Aplicaciones del método Monte Carlo**[1].

- Transporte de la radiación en la materia.
- Calculo integral.
- Teoría de transporte
- Problemas de optimización

2.2. Código y Gráfica

A continuación los códigos y gráficas que se realizaron aplicando el Método Monte Carl[2].

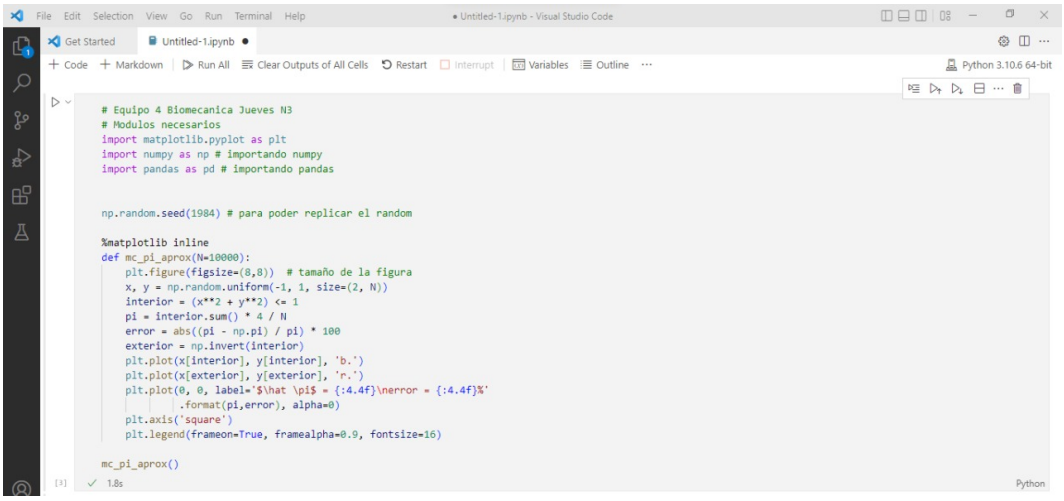


Figura 1: Código con aproximamiento a n=10000

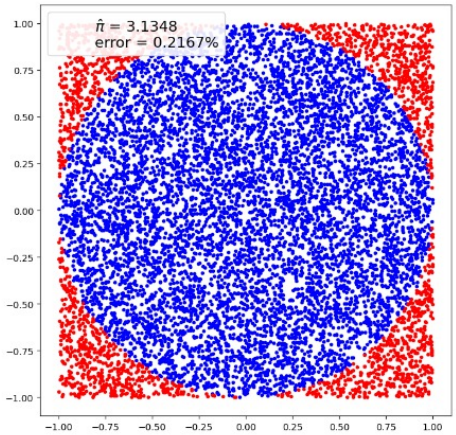
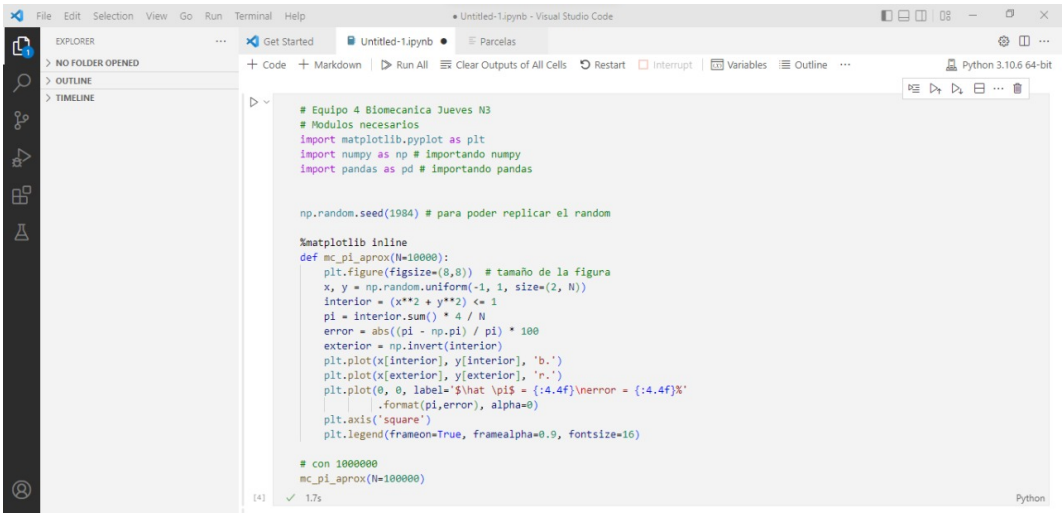


Figura 2: Gráfica con aproximamiento a n=10000



```
# Equipo 4 Biomecanica Jueves N3
# Modulos necesarios
import matplotlib.pyplot as plt
import numpy as np # importando numpy
import pandas as pd # importando pandas


np.random.seed(1984) # para poder replicar el random

%matplotlib inline
def mc_pi_aprox(N=10000):
    plt.figure(figsize=(8,8))  # tamaño de la figura
    x, y = np.random.uniform(-1, 1, size=(2, N))
    interior = (x**2 + y**2) <= 1
    pi = interior.sum() * 4 / N
    error = abs((pi - np.pi) / pi) * 100
    exterior = np.invert(interior)
    plt.plot(x[interior], y[interior], 'b.')
    plt.plot(x[exterior], y[exterior], 'r.')
    plt.plot(0, 0, label='$\hat \pi$ = {:4.4f}\nerror = {:4.4f}%'
             .format(pi,error), alpha=0)
    plt.axis('square')
    plt.legend(frameon=True, framealpha=0.9, fontsize=16)

# con 1000000
mc_pi_aprox(N=100000)
```

Figura 3: Código con aproximamiento a n=1000000

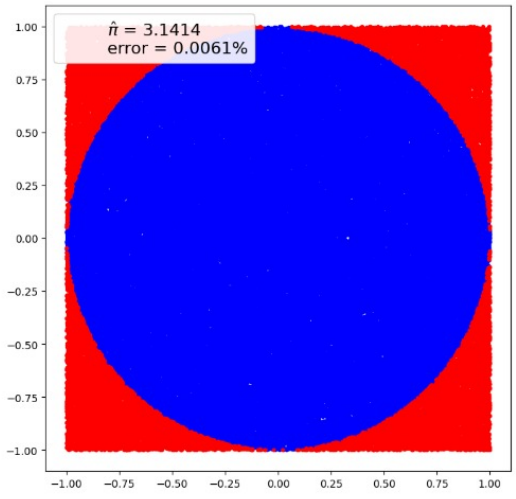


Figura 4: Gráfica con aproximamiento a n=1000000

```python
#Equipo 4 biomecanica
from multiprocessing import Pool
import matplotlib.pyplot as plt
from random import randint
import statistics
width = 10000
heigth = width
radio = width

npuntos = 0
ndentro = 0
radio2 = radio * radio
replicas = 80
promediopi = []
listareplicas = []
listapromedios = []


if __name__ == '__main__':
        with Pool(4) as p:
            for j in range(replicas):
                for i in range(1,10000):
                    x = randint(0,width)
                    y = randint(0,width)
                    npuntos += 1
                    if x * x + y * y <= radio2:
                        ndentro += 1
                    pi = ndentro * 4 /npuntos
                    promediopi.append(pi)
                print(statistics.mean(promediopi))
                listareplicas.append(j)
                listapromedios.append(statistics.mean(promediopi))
```

Figura 5: Código de convergimiento del número pi

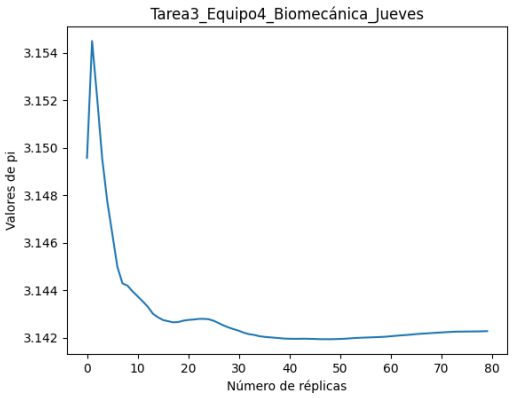


Figura 6: Gráfica de convergimiento del número pi

3. Conclusiones

Gracias a esta actividad nuestro equipo pudo entender mejor el origen del número "pi", también pudimos aprender el método Montecarlo el cual es una secuencia de estados cuya evolución viene determinada por sucesos meramente aleatorios. Todo esto lo desarrollamos mediante el lenguaje de programación Python, una vez sacando las aproximaciones al número pi se tabuló y conforme a eso se grafico.

Referencias

[1] R.M. Landrove. MÉtodo de monte carlo y sus aplicaciones, Octubre 1994. URL http://fisica.ciens.ucv.ve/~rmartin/lspcp/mmca.html.

[2] R.E. López Briega. Introducción a los métodos de monte-carlo con python, Marzo 2017. URL https://relopezbriega.github.io/blog/2017/01/10/introduccion-a-los-metodos-de-monte-carlo-con-python/.

[3] P. Pérez M. Valente. Aplicación de la técnica de simulación monte carlo, 2018. URL https://www.famaf.unc.edu.ar/~pperez1/manuales/cim/cap6.html#:~:text=La%20simulaci%C3%B3n%20Monte%20Carlo%20es,m%C3%A9dicas%20que%20utilizan%20radiaciones%20ionizantes.