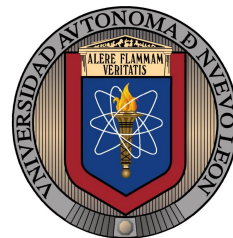




Facultad de Ingeniería Mecánica y Eléctrica

UANL

MECATRÓNICA



Biomecánica

Grupo:002

Tarea Actividad 3

FECHA DE ENTREGA:08/09/2022

Profesor:

Dra. Yadira Moreno Vera

Alumnos:

Jorge Luis Salinas Garza, Bernardo Gil Villarreal, Mauricio Julian Salazar Salazar, Cristian Arturo Garza Cavazos, Victor Emmanuel Cantu Corpus

Índice

1. Introducción	3
2. Método Monte Carlo	3
3. Orígenes del método	4
4. Ejemplo	4
5. Cálculo-estimación del número por medio de técnicas Monte Carlo	6
6. Experimentación	7
7. Resultados	8
8. Conclusión	8

1. Introducción

En esta actividad aprenderemos sobre el método de monte carlo en el lenguaje de programación en python, el compilador a utilizar es el mismo integrado, en el cual simularemos el método monte carlo y graficamos los resultados haciendo variar algunas constantes para su misma apreciación. La simulación de Monte Carlo es una técnica que combina conceptos estadísticos (muestreo aleatorio) con la capacidad que tienen los ordenadores para generar números pseudo aleatorios y automatizar cálculos.

2. Método Monte Carlo

El método de Montecarlo es un método no determinista o estadístico numérico, usado para aproximar expresiones matemáticas complejas y costosas de evaluar con exactitud. El método se llamó así en referencia al Casino de Montecarlo (Mónaco) por ser “la capital del juego de azar”, al ser la ruleta un generador simple de números aleatorios. El nombre y el desarrollo sistemático de los métodos de Montecarlo datan aproximadamente de 1944 y se mejoraron enormemente con el desarrollo de la computadora u ordenador. 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. 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. En la primera etapa de estas investigaciones, John von Neumann y Stanislaw Ulam refinaron esta ruleta y los métodos “de división” de tareas. Sin embargo, el desarrollo sistemático de estas ideas tuvo que esperar al trabajo de Harris y Herman Kahn en 1948. Aproximadamente en el mismo año, Enrico Fermi, Nicholas Metropolis y Ulam obtuvieron estimadores para los valores característicos de la ecuación de Schrödinger para la captura de neutrones a nivel nuclear usando este método. El método de Montecarlo proporciona soluciones aproximadas a una gran variedad de problemas matemáticos posibilitando la realización de experimentos con muestreos de números pseudoaleatorios en una computadora. El método es aplicable a cualquier tipo de problema, ya sea estocástico o determinista. A diferencia de los métodos numéricos que se basan en evaluaciones en N puntos en un espacio M -dimensional para producir una solución aproximada, el método de Montecarlo tiene un error absoluto de la estimación que decrece como

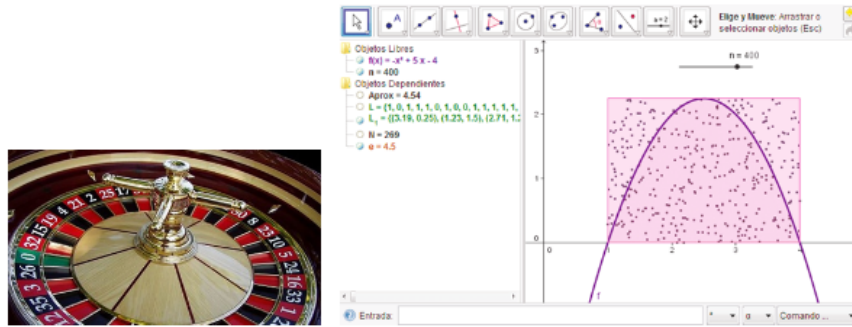
$$\frac{1}{\sqrt{N}}$$

3. Orígenes del método

La invención del método de Montecarlo se asigna a Stanislaw Ulam y a John von Neumann. Ulam ha explicado cómo se le ocurrió la idea mientras jugaba un solitario durante una enfermedad en 1946. Advirtió que resulta mucho más simple tener una idea del resultado general del solitario haciendo pruebas múltiples con las cartas y contando las proporciones de los resultados que computar todas las posibilidades de combinación formalmente. Se le ocurrió que esta misma observación debía aplicarse a su trabajo de Los Álamos sobre difusión de neutrones, para la cual resulta prácticamente imposible solucionar las ecuaciones íntegro-diferenciales que gobiernan la dispersión, la absorción y la fisión. “La idea consistía en probar con experimentos mentales los miles de posibilidades, y en cada etapa, determinar por casualidad, por un número aleatorio distribuido según las probabilidades, qué sucedería y totalizar todas las posibilidades y tener una idea de la conducta del proceso físico”. Podían utilizarse máquinas de computación, que comenzaban a estar disponibles, para efectuar las pruebas numéricas y en efecto reemplazar el aparato experimental del físico. Durante una de las visitas de von Neumann a Los Álamos en 1946, Ulam le mencionó el método. Después de cierto escepticismo inicial, von Neumann se entusiasmó con la idea y pronto comenzó a desarrollar sus posibilidades en un procedimiento sistemático. Ulam expresó que Montecarlo “comenzó a tener forma concreta y empezó a desarrollarse con todas sus fallas de teoría rudimentaria después de que se lo propuse a Johnny”.

4. Ejemplo

Los programas de diseño asistido por ordenador (CAD) pueden determinar rápidamente el volumen de modelos muy complejos. Estos modelos, en general, no tienen una expresión analítica para determinar su volumen (por ejemplo, para un prisma, área de la base multiplicada por la altura), y la única solución es dividir el modelo en un conjunto de pequeños submodelos (teselación) cuyo volumen pueda determinarse (por ejemplo, dividir el modelo en miles de tetraedros). Sin embargo, esto consume muchos recursos, tanto para la teselación como para el cálculo del volumen de cada uno de los elementos. Por ello utilizan métodos de Montecarlo, más robustos y eficientes. Como el software sí que conoce la expresión analítica de la geometría del modelo (posición de los nodos, aristas y superficies) puede determinar si un punto está dentro del modelo o está fuera con un coste mucho menor que el de determinar un volumen.



Se define la eficiencia del método Monte Carlo (ϵ) como

$$\epsilon \equiv \sigma^2 T$$

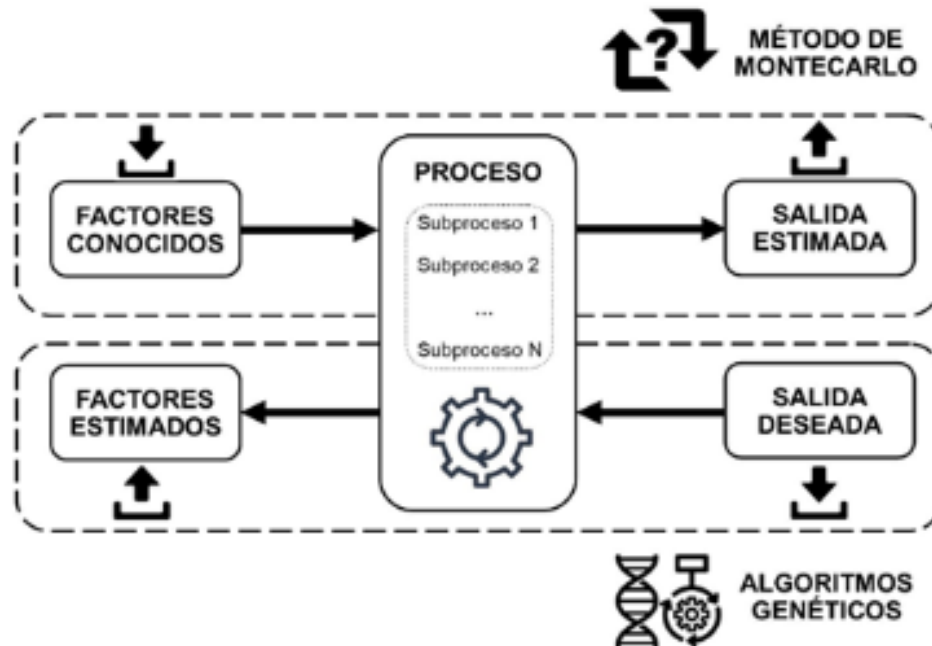
donde T es el tiempo de cálculo. Como el valor de T está fuertemente relacionado con el número de puntos usados en la computación, se suele dar también esta otra definición para la eficiencia:

$$\epsilon \equiv N \sigma^2$$

Y, a partir de esta, la eficiencia relativa (rel):

$$\epsilon_{rel} \equiv \frac{\epsilon[N]}{\epsilon[N']} = \frac{N}{N'} \frac{\sigma^2}{(\sigma')^2}$$

Si $\epsilon_{rel} > 1$, entonces el método que corresponde a N_2 es “mejor” que el método con N_1 . Si el número de puntos utilizados es el mismo, la eficiencia relativa queda reducida al cociente de las

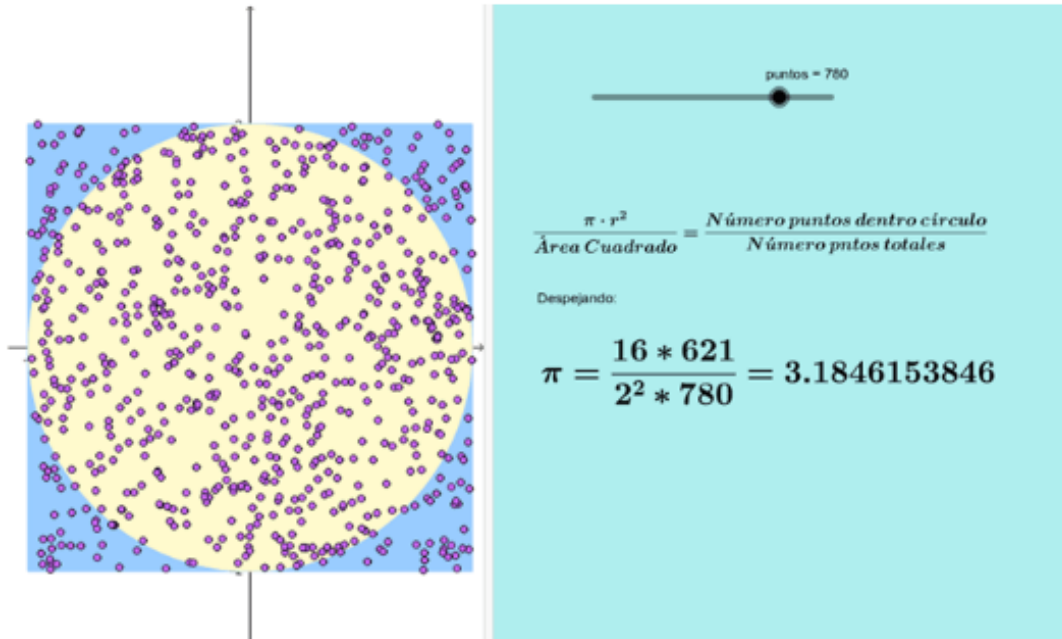


varianzas

5. Cálculo-estimación del número por medio de técnicas Monte Carlo

Uno de los métodos más antiguos utilizados para estimar el valor de π es el método de Buffon, que emplea una serie de líneas paralelas y una vara, cuya longitud guarda correlación con la separación entre líneas, para ser arrojada y determinar el ángulo que forma éstas con las líneas, así como la línea que atraviesa. El método propuesto a continuación, representa una analogía al método de Buffon. Se considera un círculo de radio unidad centrado en el origen. El área del círculo en el primer cuadrante será $\pi/4$. Un modo de resolver este problema usando el método Monte Carlo con técnica éxito-fracaso, también denominado método de rechazo, es el siguiente:

1. Generar un par de números aleatorios 1 y 2 uniformemente distribuidos en $[0,1]$.
2. Determinar un punto en el primer cuadrante, de coordenadas (x,y) a partir de 1 y 2.
3. Determinar la distancia D del punto (x,y) al origen, $D = \sqrt{x^2 + y^2}$.
4. Examinar si la distancia D es mayor o menor al radio R ($R=1$).
5. Considerar con “éxito” los procesos que den lugar a puntos en el plano dentro de círculo y como “fracaso” los que estén fuera.
6. Calcular las proporciones de éxito y de fracaso.



6. Experimentación

Código a utilizar

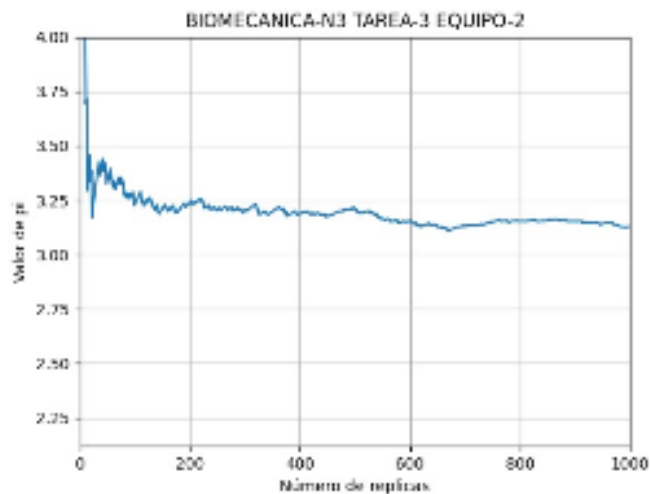
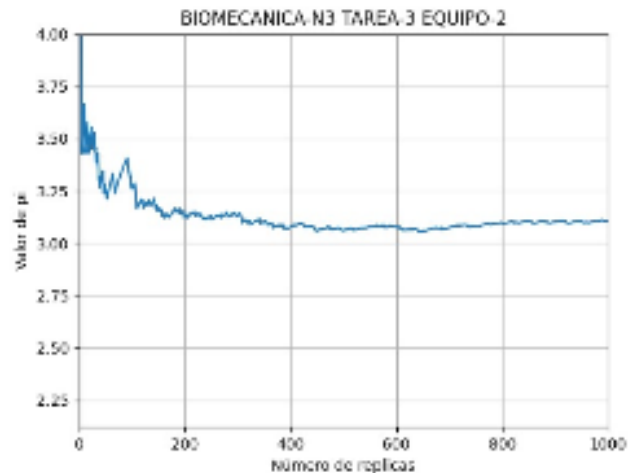
```

1 #EQUIPO 2 BIOMECANICA N3 L-M-1
2 #MAURICIO JULIAN SALAZAR SALAZAR 1906944
3 #VICTOR EMMANUEL CANTU CORPUS 1909659
4 #CHRISTIAN ARTURO GARZA CAVAZOS 1909877
5 #JORGE LUIS SALINAS GARZA 1916367
6 #BERNARDO GIL VILLARREAL 1901876
7 from random import randint
8 from multiprocessing import Pool
9 import statistics
10 import matplotlib.pyplot as plt
11 width = 10000
12 height = width
13 radio = width
14 npuntos = 0
15 ndentro = 0
16 radio2 = radio * radio
17 replicas = 10
18 promediopi = []
19
20 for j in range(replicas):
21     for i in range(1, 100000):
22         x = randint(0, radio)
23         y = randint(0, radio)
24         npuntos += 1
25         if x*x + y*y <= radio2:
26             ndentro += 1
27             pi = ndentro * 4 / npuntos
28             promediopi.append(pi)
29     print(statistics.mean(promediopi))
30 print(statistics.mean(promediopi))
31 plt.xlim(0, 10000)
32 plt.ylim(2.12, 4)
33 plt.plot(promediopi)
34 plt.xlabel('Número de replicas')
35 plt.ylabel('Valor de pi')
36 plt.title('BIOMECANICA N3 AREA-3 EQUIPO-2')
37 plt.grid()
38 plt.show()

```

7. Resultados

Código a utilizar



8. Conclusión

Gracias a la realización de esta práctica podemos darnos cuenta de cómo un simple método como lo es el Monte Carlo puede generar muchas funciones. Este método permite obtener soluciones probabilísticas para ecuaciones donde no es satisfactorio utilizar un método determinado por falta de precisión en los datos de entrada. El objetivo de este método no es el de brindar decisiones sino apoyar a la toma de estas y en base a esto podemos decir que existen diversos programas como python o como matlab que ayudan al soporte de problemas complejos, ya que será cuestión de

analizar si elegir aquel que se adecue mejor a sus necesidades. Podemos observar que la primera gráfica inició con un valor de π muy alto y a mayor número de réplicas mayor precisión del número constante de π .

Referencias

- [1] López, M. V. and Mariño, S. I. (2002). Aplicación del método de montecarlo para el cálculo de integrales definidas. In *IV Workshop de Investigadores en Ciencias de la Computación*.
- [2] Rodríguez-Aragón, L. J. (2011). Simulación, método de montecarlo. *Recuperado de: <https://previa.uclm.es/profesorado/licesio/docencia/mcoi/tema4-guion.pdf>*.
- [3] Sobol, I. M. (1976). *Método de Montecarlo*. Mir Moscow.

Los créditos de las fotografías pertenecen a sus respectivos autores.