0-montecarlo_methods

October 22, 2020

1 Métodos de Monte Carlo

Es el nombre que se le da a las técnicas que utilizan generación de (procesos) métodos aleatorios para resolver problemas en la computadora.

2 Números aleatorios

¿Qué es *random* ?¿Existe?

Las personas batallan mucho para generar aleatoriedad por si mismas.

Las computadoras son máquinas *deterministas*.

Caos –> Cambios infinitesimales pueden afectar drásticamente

Usaremos **números pseudo aleatorios**. Siempre llevan algo detrás.

[2]:
```
import random
random.seed(1)
#Determinamos una semilla y no cambia el aleatorio
random.gauss(0,1)
```

[2]:
```
1.2881847531554629
```

Ejercicio Programa el generador aleatorio de tipo *linear congruential*:

$$x_{n+1} = (ax_n + b) \mod c$$

Para que sea más fácil, utiliza el concepto de `generators` de **Python**

Ejercicio ¿Cuál es la secuencia?

Ejercicio ¿De qué longitud es la secuencia? ¿Está relacionada con el valor de c? ¿Cómo?

[3]:
```
%pylab inline
import numpy as np
import matplotlib.pyplot as plt
import matplotlib
```

```
Populating the interactive namespace from numpy and matplotlib
```

```
/opt/conda/lib/python3.8/site-packages/IPython/core/magics/pylab.py:159:
UserWarning: pylab import has clobbered these variables: ['random']
`%matplotlib` prevents importing * from pylab and numpy
  warn("pylab import has clobbered these variables: %s"  % clobbered +
```

[4]:
```python
def pseudo_random_number_generator(seed, a, b, c):
    yield 1
```

Ejercicio En los 70's fuéron muy populares los coeficientes **RANDU**, pero se demostró que fallaban miserablemente las pruebas de aleatoriedad, en particular los tripletes de números consecutivos caen en uno de 15 planos paralelos… Dibuja los primeros 100,000 tripletes y reproduce la imagen de la wikipedia.

Afortunadamente (por lo menos para los fines de este curso) `python` incluye un generador llamado **Mersenne Twister**, el cual podemos usar si importamos la biblioteca `random`

[5]:
```python
import random
#random.seed(10)
print(random.random())
print(random.random())
print(random.random())
print(random.random())
```

```
0.763774618976614
0.2550690257394217
0.49543508709194095
0.4494910647887381
```

3 Integración de Monte Carlo

Una técnica de **Monte Carlo** es la integración, el algoritmo es muy simple:

Recuerda que la integración devuelve el **área** *de una figura.*

1. Rodea el objeto al cual le quieres calcular el área con una figura de la cual conozcas el área (un rectángulo si es de dos dimensiones) y
2. Genera un número muy grande de puntos al azar dentro de la figura de la cual conoces el área.
3. El área del objeto es aproximadamente la fracción de puntos que cayeron dentro del objeto multiplicada por el área del objeto que la rodea.

3.0.1 Ejemplo: Calcular el volumen de una esfera de radio $r = 1$

Podemos usar la simetría del problema y concentrarnos en un cuadrante y luego multiplicar por 8 el resultado. Usemos como figura que conocemos un cubo de lado 1.

[14]:
```python
import time

tic = time.process_time()
```

```
#Encerramos la esfera en un cubo
volumen_cubo = 1 * 1 * 1

#Cantidad de puntos
N = 20000000
#Cuenta de cuántos cayeron dentro de la esfera
count = 0
for j in range(N):
    # Un punto al azar, random() genera 3 valores entre 0-1
    point = np.array([random.random(), random.random(), random.random()])
    # x^2 + y^2 + z^2 < 1 ¿Esta dentro? (por def de esfera)
    if sum(point**2) <= 1:
        count = count + 1

#Para saber qué fracción cayó dentro de la esfera
fraccion = float(count)/float(N)

#La fracción del volumen del cubo
volumen = fraccion * volumen_cubo

#Porque solo estabamos calculando en 0,1 y son 8 partes por ser esfera en R3
volumen_esfera = volumen * 8

toc = time.process_time()
```

El volumen de la esfera se puede calcular analiticamente y es igual a

$$V = \frac{4}{3}\pi r^3$$

[15]:
```
toc-tic
```

[15]: 233.93190486900002

[16]:
```
volumen_esfera
```

[16]: 4.1889968

[17]:
```
volumen_esfera_real = (4.0/3)*math.pi
volumen_esfera_real
```

[17]: 4.1887902047863905

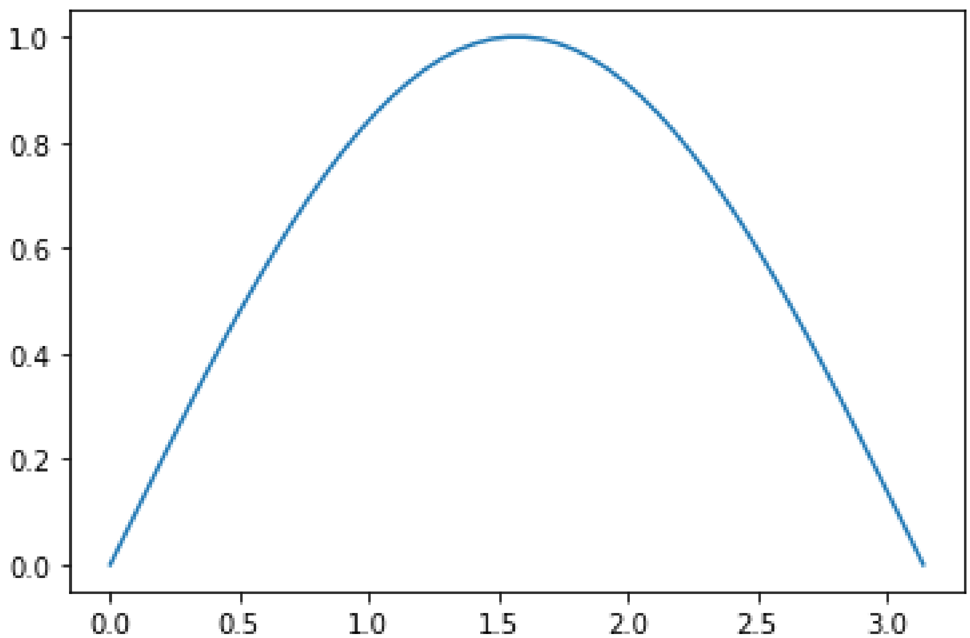
Ejercicio Calcula

$$I = \int_0^\pi \sin x dx$$

Usando técnicas de Monte Carlo. Compara con el resultado analítico.

[22]:
```
X = np.linspace(0,math.pi)
Y = np.sin(X)
plot(X,Y)
```

[22]:
```
[<matplotlib.lines.Line2D at 0x7f8031ff0be0>]
```



[23]:
```
area_rectangulo= math.pi * 1
N=10000000
count = 0
for i in range(N):
    point = np.array([random.random()*math.pi, random.random()])
    if point[1] < math.sin(point[0]):
        count+= 1
fraccion = float(count)/float(N)

area = fraccion * area_rectangulo
area
```

[23]:
```
1.9991242107779827
```

Ejercicio Encuentra el volumen de la intersección de una esfera y un cilindro. La esfera tiene $r = 1$ y está centrada en el origen. El cilindro tiene radio $r = 0.5$ y su eje es perpendicular al eje x y pasa por el punto $(0.5, 0, 0)$.