

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
FACULTAD DE CIENCIAS FÍSICO MATEMÁTICAS

Aplicaciones de Nanotecnología Computacional
Proyecto 5: *Caminatas aleatorias y radio de giro*

José Abraham Morales Vidales, 1941505
René Rogelio Corrales Millán, 1941487
Diego Armando Alvarado Silos, 1941598

Profesor: Dr. Omar González Amezcua

16 de abril de 2021

Resumen

Mediante la utilización de un algoritmo de cómputo se analizaron miles de caminatas con el fin de realizar estadística y determinar el valor al que convergen con diferentes tamaños.

Palabras clave: Caminatas aleatorias, radio de convergencia

Introducción

Las caminatas aleatorias (RW del inglés *random walk*) son útiles para describir un amplio espectro de situaciones que van desde una simplificación del movimiento browniano hasta interesantes aplicaciones en economía sobre los precios de acciones. Aún con todas sus fuertes repercusiones, su programación es increíblemente sencilla.

Conocer las características que distinguen a las RW puede ser de gran ayuda para los modelados que se deseen realizar con ellas. Una de estas características o propiedad definida por el *tamaño* de la RW es el *radio de giro*.

Marco teórico

Una RW puede ser generada utilizando la siguiente fórmula iterativa:

$$x_n = x_{n-1} + \xi_n$$

Donde ξ_n es una variable aleatoria que puede tomar los valores de -1 y 1 (en el caso de una RW unidimensional). Esta fórmula iterativa será válida también para calcular x_{n-1} , x_{n-2} , x_{n-3} , ..., x_1 . Entonces podemos determinar la fórmula para la caminata aleatoria de la siguiente forma:

$$x_n = x_0 + \sum_{i=1}^n \xi_i$$

En el caso de una RW en 2 dimensiones las variables x_n no serían números reales sino vectores, y la variable aleatoria ξ_n puede tomar los valores (1, 0), (-1, 0), (0, 1), (0, -1).

El parámetro que se desea analizar en este proyecto es el *Radio de Giro cuadrado*, el cual nos

da información acerca del tamaño de la caminata, y se calcula de la siguiente manera:

$$R_g^2 = \frac{1}{2N_p^2} \sum_{n=1}^{N_p} \sum_{m=1}^{N_p} [(x_n - x_m)^2 + (y_n - y_m)^2]$$

Donde N_p es el número de pasos de la RW, x y y definen las coordenadas de cada paso. Este valor por sí solo no dice mucho acerca de la naturaleza de una RW. Sin embargo, es una herramienta útil al momento de analizar una gran cantidad de RW. Entonces se debe calcular el promedio de los radios de giro cuadrados de distintas caminatas, con diferente número de pasos; esto se hace con la fórmula común para el promedio:

$$\langle R_g^2 \rangle = \frac{1}{N_c} \sum_{i=0}^{N_c} (R_g^2)_i$$

Entonces, la idea general del código es generar una cantidad de RW significativa, 10^6 aproximadamente, para un número de pasos definido; y encontrar el radio de giro promedio para ese número de pasos.

Objetivo general

Determinar el radio de giro de diferentes RW.

Objetivo específico.

1. Medir y comparar los radios de giro en espacios 2-D y 3-D.
2. Determinar una relación entre el radio de giro y el tamaño de las RW.

Resultados

Se generó un código en Python que generara las caminatas aleatorias, mediante POO (programación orientada a objetos) donde cada caminata forma parte de un objeto de clase "trayectoria" y la función relacionada a ese objeto es realizar una caminata aleatoria que se define como "caminar". Se le especifica un número de pasos y la dimensión de la caminata.

```

class trayectoria():
    def __init__(self,num_steps,dim):
        self.num_steps=num_steps+1
        self.Co=[]
        self.dimension=dim
    def caminar(self):
        for h in range(self.dimension):
            self.Co.append([0])
        for s in range(1,self.num_steps):
            for h in range(self.dimension):
                self.Co[h].append(self.Co[h][s-1]*(-1)**rd.randint(1, 2))
            for h in range(self.dimension):
                self.Co[h].pop(0)
        return np.array(self.Co)

```

Figura 1.- Código que define la clase “trayectoria”

El ciclo principal y el que consume mas recursos computacionales es el encargado de calcular el radio de giro, debido a que en Python los cálculos sobre arreglos grandes son muy tardados optamos por utilizar F2py para integrar código de Fortran. Se realizo un código en Fortran con la subrutina que calcula el radio de giro. Después con f2py se convirtió ese documento con extensión “.f” en un documento que puede ser leído como un documento de Python pero que funciona con un compilador de Fortran.

```

subroutine radio(dim,num_steps,tra,suma)
integer(kind=4),intent(in):: num_steps,dim
real(kind=8),dimension(dim,num_steps),intent(in)::tra
real(kind=8),intent(out):: suma
integer:: i,s,j
suma=0.0
do i=1,num_steps
    do s=1,num_steps
        do j=1,dim
            suma=suma+(tra(j,i)-tra(j,s))**2
        end do
    end do
end do
suma=(1/(2*((num_steps)**2.0)))*suma
end subroutine radio

```

Figura 2.- Subrutina de Fortran para calcular radio de giro a partir de una trayectoria dada en forma de un arreglo bidimensional.

De esa manera podemos importar un nuevo archivo como un paquete de Python con la función “radio” que esta en la subrutina del archivo de Fortran.

Utilizamos 1000 caminatas para cada número de pasos ya que ese número es el más optimo (en cuestión de tiempo) y es un muestreo suficientemente exacto para los propósitos de este proyecto.

Para utilizar el código correctamente se debe establecer un numero de pasos máximo definido por la variable “max_steps” y un intervalo en el

número de pasos que sea divisor de “max_steps” asignado en la variable “inc”.

```

dim=3 #Dimensiones de la caminata
trayectorias_num=1000
max_steps=10000 #numero de pasos maximo
inc=1000 #intervalo de pasos; tiene que ser un divisor de max_steps
tiempo=[]
radio_final=[]
for num_steps in range(inc,max_steps+1,inc):
    radio_final.append(radio(dim,num_steps,tra,suma))

```

Figura 3.

1. RW en 2-D

Se computaron los radios de giro para distintos números de pasos y todas las gráficas obtenidas arrojaron una relación directamente proporcional entre el número de pasos y el radio de giro promedio como se puede apreciar en la figura 4.

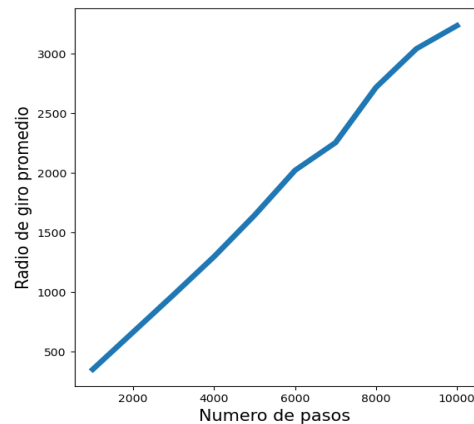


Figura 4.- Grafica del radio de giro para caminatas de 2 dimensiones.

2. RW en 3-D

Análogo al caso bidimensional se computaron los radios de giro para distintos números de pasos y todas las gráficas obtenidas arrojaron una relación directamente proporcional entre el número de pasos y el radio de giro promedio como se puede apreciar en la figura 5.

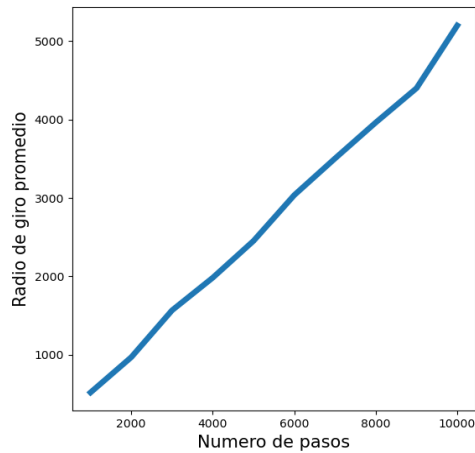


Figura 5.-Grafica del radio de giro para caminatas de 3 dimensiones.

Tanto para el caso 2-D como para el caso 3-D se trataron los datos de salida con mínimos cuadrados para determinar la dependencia lineal que existe entre el radio de giro y el número de pasos de las RW; para el caso 2-D se obtuvo una pendiente 0.3305 unidades de radio de giro al cuadrado por número de pasos, por otro lado, para el caso 3-D, se obtuvo una pendiente de 0.5047.

Conclusiones

Para ambos casos existe una relación lineal entre el número de pasos de las caminatas y los radios de giro promedio. Esta cantidad en principio no tiene validez estadística si se aplica sobre una sola RW, sin embargo, cuando se trata de un gran número de ellas, este número se convierte en una característica especial del conjunto. Usar Fortran en Python redujo ampliamente el tiempo de cálculo, pues de las horas de cálculo que se requerían con Python, con Fortran, esto se redujo a minutos.