Universidad Autónoma De Sinaloa

Facultad de Ciencias Física-Matemáticas

Simulación de movimiento browniano en python

Fisica computacional

Autor:
Lopez Rodriguez José Julián

Enero-Junio 2022

Resumen

Al analizar a través de un microscopio los movimientos de las partículas de polen suspendidas en el agua, estos movimientos parecen ser aleatorios, siguen un movimiento errático y zigzagueante. En de 1827 Robert Brown, observo a través de un microscopio las partículas contenidas en el polen se movía de una forma bastante peculiar. Parecía ser un sistema caótico. Inicialmente creyó que este movimiento era causado por que algunas células dentro de estas partículas continuaban vivas. Sin embargo, nuevamente hizo su experimento pero ahora con partículas contenidas en polen muerto por mas de un siglo. Nuevamente observo los mismo resultados. Ahora sabemos que el causante de este movimiento son las coaliciones de las moléculas del fluido. De hecho el movimiento browniano fue una evidencia de la existencia del átomo y las moléculas. En el presente estudio se busca hacer una simulación de partículas de polen sobre agua, en el lenguaje de programación python.

Índice

1. Introducción

El ahora conocido movimiento browniano fue investigado por primera vez por el mismo Robert Brow (investigación que fue publicada en 1827 [1]) y otros científiques de la época. Las observaciones mostraron que un aumento en la temperatura del fluido hace que el movimiento sea con mayor rapidez. En cambio a mayor viscosidad del fluido el movimiento se ralentiza. La teoría cinética de la materia dio una explicación cualitativa. Las moléculas que componen el fluido están sometidas a un movimiento térmico, cuyas velocidades tienen una distribución de Maxwell-Boltzmann. Cualquier partícula en el fluido sufrirá constante colisiones por parte de las moléculas del fluido, cada colisión modificara el vector velocidad en el fluido. El resultado sera un movimiento zigzagueado y errático a través del fluido. Actualmente, el movimiento browniano ha adquirido gran importancia en áreas científicas e ingenierías. Como la electrónica, la computación, la biología y la economía. Que en esta ultima fue propuesta por Louis Bachelier[2] que la propuso para la aplicación en la fluctuación en los mercados.

2. Primera simulación del movimiento browniano

2.1. Primer modelado de el movimiento de las partículas

Para el primer modelado usaremos el modelo de movimiento browniano propuesto por M. Smoluchowski [3]. En este modelo se simplifica el movimiento de la partícula y se supone que la partícula se mueve dando pequeños pasos de longitud d sobre el eje x, y otro pequeño salto de la misma longitud sobre el eje y

2.1.1. Distribución de probabilidad para la distancia de las partículas

Consideremos los siguientes eventos. En que cada evento una partícula da un paso, este puede ser a la derecha o hacia la izquierda, al final la partícula da N pasos, es decir transcurrieron N eventos. Queremos calcular cuantos de estos N pasos son hacia la derecha. En cada evento tenemos solo dos posibles resultados (un paso hacia la derecha o uno a la izquierda). Sea k el numero de pasos que son hacia las derecha y sea p la posibilidad de que la partícula avance hacia la derecha. Podemos reescribir la pregunta por ¿Cual es la posibilidad de que la partícula avance k pasos hacia la derecha con un total de N pasos, si la posibilidad de que un solo paso sea hacia la derecha es p?. Por hipótesis sabemos que un paso hacia la derecha es independiente del siguiente paso (Es decir son eventos independientes). Sea $f(k, N)$ la función que denote esta probabilidad. Supongamos que hay tres eventos. Es decir $N = 3$ y queremos saber cual es la posibilidad de que la partícula de 2 pasos hacia la derecha. Es decir $k = 2$. Esto es

p	p	p
1	1	0
1	0	1
0	1	1

Cuadro 1: $f(2, 3)$

El numero de columna representa los pasos que dio (en este caso 3). El numero 1 representa que la partícula dio un paso hacia la derecha. y un 0 representa que la partícula no dio un paso hacia la derecha, k representa el numero de pasos a la derecha que nos interesa saber su probabilidad. El numero de renglones representa el numero de conmutaciones tal que cumplen que $k = 2$. La probabilidad de que la partícula de 2

pasos a la derecha es igual a p^2 y la probabilidad de que la partícula no de un paso a la derecha es $(1-p)$. Ahora como hay tres formas de cumplir con esta condición. Es decir:

$$(p)(p)(1-p) = p^2(1-p)$$
$$(p)(1-p)(p) = p^2(1-p)$$
$$(1-p)(p)(p) = p^2(1-p)$$

Se tiene que

$$f(2,3) = 3p^2(1-p)$$

Entonces la probabilidad de que k pasos sean hacia la derecha es de p^k y la probabilidad de que $N-k$ no sean hacia la derecha es de $(1-p)^{N-k}$. Ahora el numero de combinaciones se denotara por $\binom{N}{k}$ que sabemos que esto es

$$\binom{N}{k} = \frac{n!}{k!(n-k)!}$$

Entonces $f(k,N) = \binom{N}{k}p^k(1-p)^{N-k}$ es la función de distribución buscada. A esta función de distribución se le conoce como función de distribución binomial. Que para un valor N fijo:

$$P(X=k) = \binom{N}{k}p^k(1-p)^{N-k}$$

Supongamos ahora que una partícula se mueve N pasos, y además que de estos N pasos, N_r son los pasos hacia la derecha y N_l son los pasos a la izquierda. Entonces:

$$N = N_l + N_r$$

Entonces queremos saber cual es la probabilidad de que $N_r = k$ para un cierto k dado. Por las observaciones hechas por Robert Brow [1] la probabilidad de que dé un paso hacia la derecha es igual a que si da un paso a la izquierda, es decir $p = \frac{1}{2}$ sustituyendo esto en f

$$\begin{aligned} P(X=k) &= \binom{N}{k}\left(\frac{1}{2}\right)^k\left(1-\frac{1}{2}\right)^{N-k} \\ &= \binom{N}{k}\left(\frac{1}{2}\right)^k\left(\frac{1}{2}\right)^{N-k} \\ &= \binom{N}{k}\left(\frac{1}{2}\right)^N \end{aligned} \tag{1}$$

Si en cada paso recorre una distancia h se tendrá entonces que la posición de una una partícula después de N pasos (suponiendo que se encontraba en el origen) es:

$$\begin{aligned} x &= (N_r - N_l)\,h \\ &= (2N_r - N)\,h \end{aligned} \tag{2}$$

Donde se sustituyo $N_l = N - N_r$

Ahora graficando distancia contra probabilidad de que se encuentre en una distancia dada con $N = 0$ (Es decir 0 pasos) tenemos la **figura 1**.

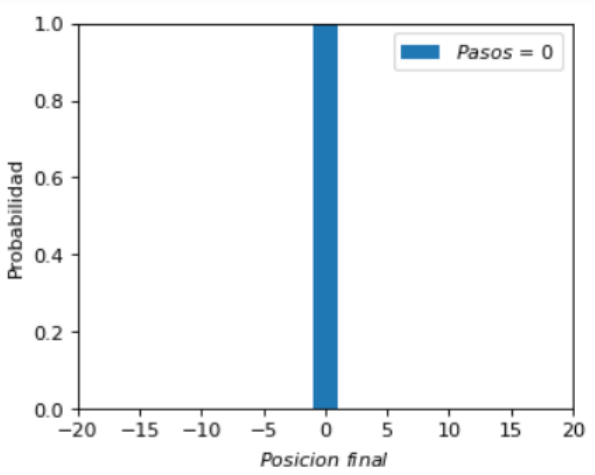


Figura 1: Distribución de probabilidad para $N = 0$

Lo cual tiene mucho sentido ya que si la partícula no se mueve, entonces la probabilidad de que la partícula se quede en el origen es de 100 %. Volviendo a graficar distancia contra probabilidad con $N = 10$ (10 pasos) obtenemos la **figura 2**.

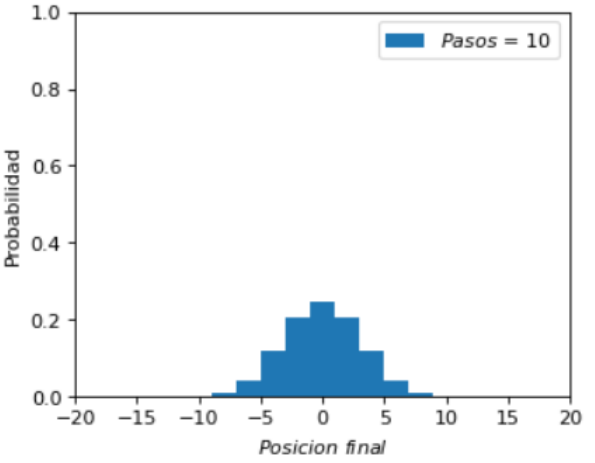


Figura 2: Distribución de probabilidad para $N = 10$

Vemos que para $N = 10$ sigue siendo muy probable que la partícula al dar el décimo paso llegue de nuevo al origen. Ahora para $N = 20$ (20 pasos) obtenemos la **figura 3**.

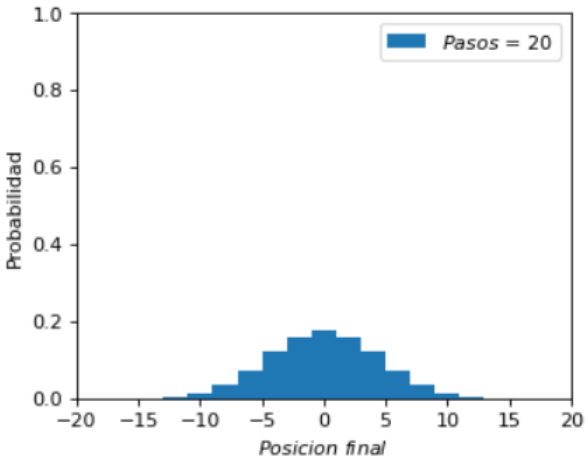


Figura 3: Función de distribución con $N = 20$

Que vemos que cada vez mas la función se distribuye mas y mas. Es decir, cada vez se concentra menos en el centro. Por ultimo se tiene que si graficamos de nuevo con $N = 30$ obtenemos la **figura 4**.

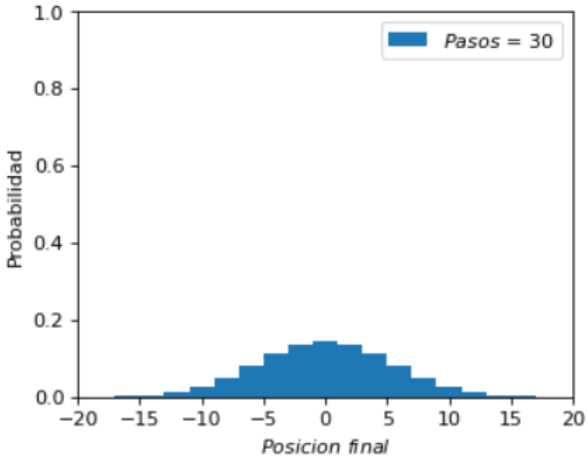


Figura 4: Distribución de probabilidad para $N = 30$

Analizando la **figura 1**, **figura 2**, **figura 3** y **figura 4** se observa que mientras N crece la probabilidad de que la partícula quede en el origen después de N pasos es cada vez menor. También se observa que la probabilidad de que quede en una posición x después de los N pasos cada vez se extiende mas y mas, lo cual es algo de esperarse, ya que es un hecho experimental que las partículas se disipan, es decir, pasan de estar en una región donde hay mucha concentración de partículas a una región con menor concentración.

2.1.2. Aproximación de la función de distribución binomial a la función de distribución normal cuando N tiende a infinito y h tiende a 0 de tal forma que Nh es constante

Pero este modelo señalado descrito anteriormente tiene un grave problema y es que las partículas no dan saltos, x (la distancia) debe de ser una función continua. Tenemos que tomar el limite en donde pasamos de considerar una variable discreta y comenzamos a utilizar una variable continua. Es decir debemos tomar el limite cuando $N \to \infty$ y $h \to 0$ pero por conveniencia esto se hará de tal forma que $hN = constante$. El tiempo promedio que existe entre una colisión entre una molécula del fluido y la partícula que esta bajo el movimiento browniano es:

$$\tau = \frac{N}{\tau}$$

Para encontrar esta aproximación hace falta una propiedad y un teorema [4].

Teorema 1. *Supongamos que X y Y son variables aleatorias tal que $\mathrm{M}_X(k) = \mathrm{M}_Y(k)$ para algún intervalo que contenga el cero. Entonces la distribución de probabilidad de X y Y son iguales*

Propiedad 1. *Ahora sea $aX + b$ una transformación lineal donde a y b son constante se tiene entonces que*

$$M_{b+aX}(k) = E\left[e^{k(b+aX)}\right] = E\left[e^{kb+kaX}\right]$$

$$= E\left[e^{kb} \cdot e^{kaX}\right] = e^{bk} \cdot E\left[e^{(ak)x}\right] = e^{bk} M_x(ak)$$

Una vez aclarado esto analizamos la función de distribución binomial:

$$\mathrm{P}(X = n) = \binom{N}{n} p^n (1-p)^{N-n}$$

Calculando la función generadora de momentos

$$M_X(k) = E[e^{kX}] = \sum_{x=0}^{n} e^{kx} \frac{n!}{(n-x)!x!} p^n (1-p)^{n-x}$$

$$= \sum_{x=0}^{n} \frac{n!}{(n-x)!x!} \left(e^k p\right)^x (1-p)^{n-x}$$

Desarrollando la sumatoria

$$= (1-p)^n + n\left(e^k p\right)(1-p)^{n-1} + \frac{n(n-1)}{2!}\left(e^k p\right)^2 (1-p)^{n-2} + \ldots + \left(e^k p\right)^n$$

Utilizando el Terorema del binomio que afirma que

$$(x+y)^n = \sum_{k=0}^{n} \binom{n}{k} x^{n-k} y^k$$

Entonces

$$= \left[(1-p) + e^k p\right]^n$$

O bien, si $n = N$

$$\mathrm{M}_n(k) = \left(pe^k + 1 - p\right)^N .$$

Ahora para la función de distribución normal

$$\mathrm{P}(x = k) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}\right)$$

De [4] se tiene que su función generadora de momentos es:

$$\mathrm{M}_x(k) = e^{\mu k + \sigma^2 k^2/2}.$$

Ahora recordando que la posición de una partícula browniana es $x = (2N_r - N)\,h$ y además con $p = \frac{1}{2}$ se tiene que:

$$\mathrm{M}_x(k) = E\left[e^{kx}\right]$$

$$\mathrm{M}_x(k) = E\left[e^{k(2N_r - N)h}\right]$$

$$\mathrm{M}_x(k) = E\left[e^{(2khN_r - khN)}\right]$$

$$\mathrm{M}_x(k) = e^{-hNk}\mathrm{M}_{2hN_r}(k)$$

Y utilizando ahora la función generadora de momentos de la función de distribución binomial

$$\begin{aligned}\mathrm{M}_x(k) &= e^{-hNk}\left(\frac{1}{2}e^{2hk} + \frac{1}{2}\right)^N \\ &= e^{-hNk}\left(\frac{1}{2}e^{2hNk/N} + \frac{1}{2}\right)^N \\ &= e^{-yk}\left(\frac{1}{2}e^{2yk/N} + \frac{1}{2}\right)^N\end{aligned}$$

Donde se hizo $y = hN = cons.$ Ahora el termino $e^{2yk/N}$ se puede hacer la siguiente aproximación

$$e^{2yk/N} \approx 1 + \frac{2yk}{N} + \left(\frac{yk}{N}\right)^2$$

Esto debido a la expansión de e^x de orden 2

$$e^x \approx 1 + x + \frac{x^2}{2!}$$

Sustituyendo de vuelta en $M_x(k)$

$$\mathrm{M}_x(k) \approx e^{-yk}\left(1 + \frac{yk + y^2k^2/2N}{N}\right)^N$$

Ahora utilizando el siguiente limite

$$\lim_{n\to\infty}\left(1 + \frac{x}{n}\right)^n = e^x$$

se tiene que

$$\begin{aligned}\mathrm{M}_x(k) &\approx e^{-yk}e^{yk + y^2k^2/2N} \\ &= e^{(y^2/N)k^2/2}\end{aligned}$$

Pero esto es la función generadora de momentos de la función de distribución normal con $\mu = 0$ y $\sigma^2 = \frac{y^2}{N}$. Por tanto se tiene que

$$\begin{aligned}\mathrm{p}(x) &\approx \frac{1}{\sqrt{2\pi y^2/N}} \exp\left(-\frac{1}{2}\frac{x^2}{y^2/N}\right) \\ &= \frac{1}{\sqrt{2\pi h^2 t/\tau}} \exp\left(-\frac{1}{2}\frac{x^2}{h^2 t/\tau}\right)\end{aligned}$$

Comparando con la función de densidad normalizada

$$\mathrm{p}(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}\right)$$

se tiene que $\mu = 0$ (como ya se sabia) donde μ es la media, $\sigma^2 = \frac{h^2 t}{\tau}$, donde sabemos que $\frac{h^2}{\tau}$ es constante.

2.1.3. Ecuación de concentración de partículas

Si hacemos $D = \frac{h^2}{2\tau}$

$$\mathrm{p}(x) = \frac{1}{\sqrt{2\pi(2)Dt}} \exp\left(-\frac{1}{2}\frac{x^2}{2Dt}\right)$$

$$\mathrm{p}(x) = \frac{1}{\sqrt{4\pi Dt}} \exp\left(-\frac{x^2}{4Dt}\right)$$

Entonces la probabilidad de que una partícula de un grupo de $\mathcal{N}$ partículas se encuentre a una distancia x en un tiempo t es de:

$$P(x,t) = \frac{\mathcal{N}}{\sqrt{4\pi Dt}} \exp\left(-\frac{x^2}{4Dt}\right)$$

Entonces la concentración de un grupo de $\mathcal{N}$ partículas, puede ser descrita por la ecuación

$$n(x,t) = \frac{\mathcal{N}}{\sqrt{4\pi Dt}} \exp\left(-\frac{x^2}{4Dt}\right)$$

2.1.4. Media y varianza

Pero para precisar mas la relación entre el movimiento browniano y su difusión supongamos entonces que tenemos $\mathcal{N}$ partículas brownianas que pueden moverse en el eje x y en el eje y que están descritas por la función $n(x,t)$. Supongamos también que cada movimiento de cada partícula es independiente (justo como lo describió Robert Brown). Las partículas pueden ser modeladas como se describió en la sección anterior.

Se vio anteriormente que

$$\bar{x} = \frac{1}{\mathcal{N}}\sum_{k=1}^{\mathcal{N}} x_k = 0$$

y para la desviación estándar

$$\sigma_x^2 = \frac{1}{\mathcal{N}} \sum_{k=1}^{\mathcal{N}} \left[(x_k - \bar{x})^2 \right]$$
$$= \frac{y^2}{N} = h^2 N.$$

2.1.5. Resultados

Se crean 500 partículas de alrededor del origen con una distribución uniforme. El radio es igual a la unidad la longitud de paso es de 0.1 de unidad. Se tiene entonces que las partículas generadas fueron

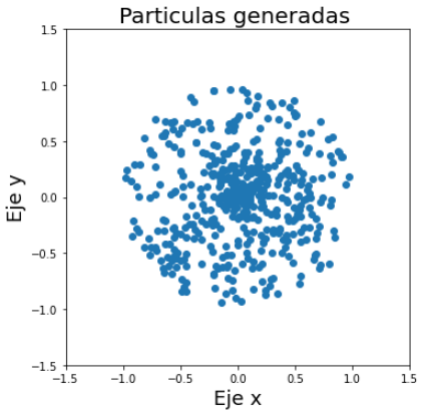


Figura 5: Partículas generadas, x prom 0.0, y prom 0.02, Var(x) 0.17, Var(y) 0.15

Después de una unidad de tiempo se tiene que el sistema a evolucionado, es decir cada partícula se movió un paso hacia algún lado y otro hacia arriba o hacia abajo, se tiene

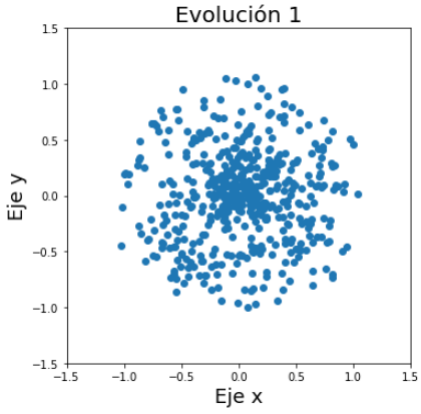


Figura 6: Primera evolución del sistma, x prom -0.0, y prom 0.02, Var(x) 0.18, Var(y) 0.17

se deja de nuevo que el sistema evoluciones nuevamente, pero ahora se deja que pase 5 unidades de tiempo, se tiene entonces la siguiente figura

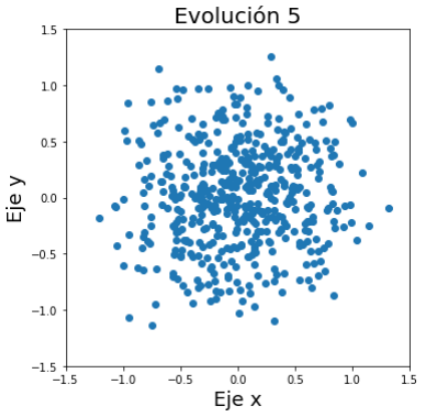


Figura 7: 5ta evolución de las partículas, x prom -0.01, y prom 0.03 Var(x) 0.21 Var(y) 0.21

Y nuevamente seguimos así sucesivamente hasta la evolución numero 10

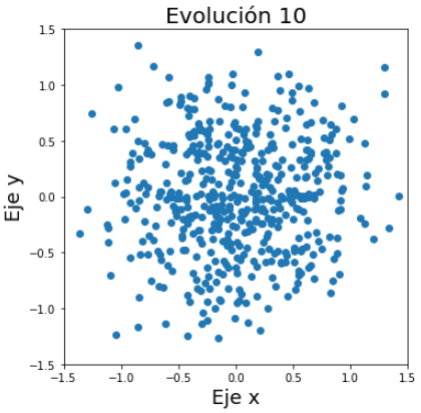


Figura 8: x prom -0.03, y prom 0.04, Var(x) 0.27, Var(y) 0.27

2.2. Primer modelado movimiento browniano con corriente

Anteriormente se analizo un sistema de partículas brownianas que están en constante borbandeando con las moléculas de agua. Sin embargo en algunas ocasiones necesitaremos agregar una corriente. Por ejemplo si tomamos en cuanta las colisiones entre las propias partículas brownianas habrá un gradiente de concentración uniforme en el sistema y se tendrá que existirá una velocidad resultante y constante. ¿O bien que pasaría si hay un flujo uniforme en el agua en donde se encuentran las moléculas brownianas?.

Sea $\rho(\boldsymbol{r}, t)$ la densidad de población de las partículas. Supongamos que el numero total de un sistema se conserva. Entonces sea V un volumen arbitrario. Donde las partículas que están dentro de este volumen están disminuyendo ya que estas partículas están saliendo del volumen debido a una corriente. Es decir, matemáticamente:

$$\int_V \frac{\partial}{\partial t}\rho(\mathbf{r},t)\mathrm{d}V = -\oint_S \mathbf{j}\cdot\hat{n}\ \mathrm{d}A$$

Donde $\boldsymbol{j}$ denota el flujo de la partículas. Usando el teorema de divergencia se tiene que

$$\int_V \frac{\partial}{\partial t}\rho(\mathbf{r},t)\mathrm{d}V = -\int_V \nabla\cdot\mathbf{j}\mathrm{d}V$$

Esto se debe cumplir para todos los volúmenes ya que elegimos uno arbitrariamente, se tiene entonces que

$$\frac{\partial}{\partial t}\rho(\mathbf{r},t) + \nabla\cdot\mathbf{j} = 0$$

Entonces consideremos dos corrientes

- La difusión debido a la corriente del agua
- El choque entre las mismas partículas

Para la primera usaremos la primera ley de Fick, donde $\mathbf{j}_1 = -D\Delta\rho$ donde D es el coeficiente de difusión, esta ley físicamente nos señala que una partícula tendrá un movimiento en dirección contraria a la posición donde hay mayor concentración de partículas. Para el segundo usaremos $\mathbf{j}_2 = p\mathbf{v}$ donde $\mathbf{v}$ es la velocidad preferencial de las partículas que se encuentran en el fluido. Sumando $\mathbf{j}_1$ y $\mathbf{j}_2$ se tiene que:

$$\mathbf{j} = \mathbf{j}_1 + \mathbf{j}_2 = -D\nabla\rho + \rho\mathbf{v}.$$

Sustituyendo esto en la expresion para $\frac{\partial}{\partial t}\rho(\mathbf{r},t)$ se tiene que

$$\begin{aligned}\frac{\partial}{\partial t}\rho(\mathbf{r},t) &= -\nabla\cdot\mathbf{j}\\ &= -\nabla\cdot(-D\nabla\rho+\rho\mathbf{v})\\ &= \nabla\cdot(D\nabla\rho) - \mathbf{v}\cdot\nabla\rho - \rho\nabla\cdot\mathbf{v}\\ &= D\nabla^2\rho - \mathbf{v}\cdot\nabla\rho\end{aligned}$$

Y por ultimo asumimos que D no varia y además que ($\nabla\cdot\mathbf{v} = 0$ es decir, el volumen tampoco varia. Entonces la corriente de difusión en una dimensión es

$$\frac{\partial}{\partial t}\rho(x,t) = D\frac{\partial^2}{\partial x^2}\rho(x,t) - v_x\frac{\partial}{\partial x}\rho(x,t)$$

La posición ahora esta dada por $\tilde{x} = (2N_r - N)\,h + v_x t$ donde v_x es la velocidad de la corriente y además $t = N\tau$. El termino $v_x t$ es una constante. Entonces la función generadora de momento de $\tilde{x}$ es

$$\mathrm{M}_{\bar{x}}(k) = e^{-v_x tk}\mathrm{M}_{(2N_r-N)h}(k)$$

Usando el resultado que obtuvimos en la sección anterior para $\mathrm{M}_{(2N_r-N)h}(k)$ con N grandes

$$\mathrm{M}_{\bar{x}}(k) \approx e^{-v_x tk + (y^2/N)k^2/2}$$

Y esto es la función generadora de momentos para una distribución normal, donde $\mu = v_x t$ y $\sigma^2 = y^2/N$.

Entonces la función de densidad de probabilidad de la posición de las partículas brownianas con corriente es

$$\mathrm{p}(x) \approx \frac{1}{\sqrt{4\pi Dt}}\exp\left(-\frac{1}{2}\frac{(x-v_x t)^2}{2Dt}\right)$$

2.2.1. Resultados para el primer modelado con corriente.

Se crean 500 partículas de alrededor del origen con una distribución uniforme. El radio es igual a la unidad. La longitud de paso es de 0.1 de unidad. Se tiene entonces que las partículas generadas fueron

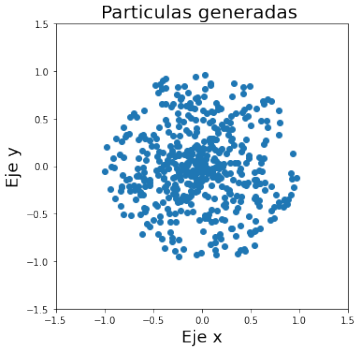


Figura 9: x prom -0.03, y prom -0.01, Var(x) 0.17, Var(y) 0.17

Después de una unidad de tiempo se tiene que el sistema a evolucionado, es decir cada partícula se movió un paso hacia algún lado y otro hacia arriba o hacia abajo, se tiene

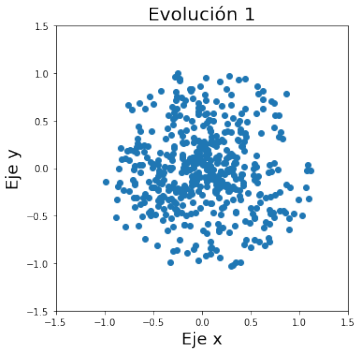


Figura 10: x prom 0.02, y prom -0.02, Var(x), 0.18 Var(y) 0.18

se deja de nuevo que el sistema evoluciones nuevamente, pero ahora se deja que pase 5 unidades de tiempo, se tiene entonces la siguiente figura

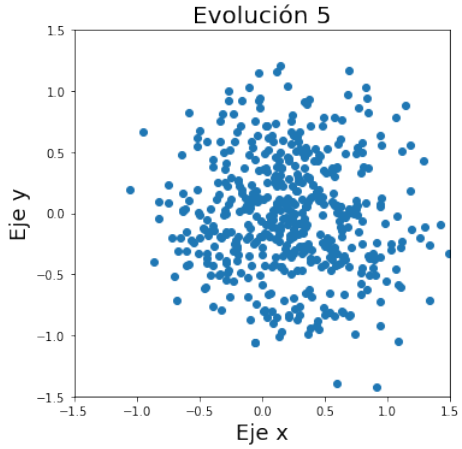


Figura 11: x prom 0.21, y prom -0.02, Var(x) 0.21, Var(y) 0.22

nuevamente se deja que el sistema evolucione nuevamente otros 5 unidades de tiempo. Se tiene

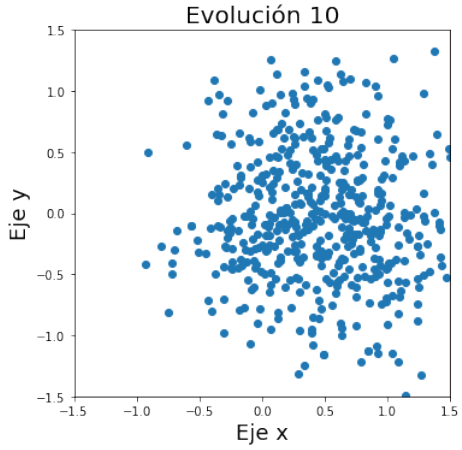


Figura 12: x prom 0.46, y prom -0.03, Var(x) 0.27, Var(y) 0.28

2.3. Código

```
def creador_de_particulas(N,R):
  '''
    Parametros
      N (int) Numero de particulas que el quiere generar

      R (float) Radio de la 'Nube de particulas'
    Arroja
      -(array) un array que contiene N cordenadas en el eje x

      -(array) un array que contiene N cordenadas en el eje y
  '''
  angulos=2*np.pi*np.random.rand(N) #Creamos un array que contenga N angulos aleatorios entre 0 y pi
  radios= R*np.random.rand(N)#Creamos un array que contenga N radios aleatorios menores a R
  cordenadasx=radios*np.cos(angulos)#Se caucula la distancia en x. Como el angulo y el radio son
aleatorios entonces la cordenada en x tambien es aleatoria
  cordenadasy=radios*np.sin(angulos)#Se calcula la distancia en y. Como el angulo y el radio son
aleatorios entonces la cordenada en y tambien es aleatoria
  return(cordenadasx,cordenadasy)
```

```
def evolucionador_de_particulas(cordenadasx,cordenadasy,d,pr,pu,corrientex,corrientey):
  '''
    Parametros
      -cordenadasx (array) posiciones en x
      -cordenadasy (array) posiciones en y
      -d           (float) la longitud de cada paso
      -pr          (float) probabilidad que de un paso a la derecha (positivo y menor a 1)
      -pu          (float) probabilidad que de un paso hacia arriva (positivo y menor a 1)
      -corrientex  (float) (aun no programado)
      -corrientey  (float) (aun no programado)
    Arroja
      - (array) un array con las cordenadas x evolucionadas
      - (array) un array con las cordenadas y evolucionadas
  '''
  if cordenadasx.size != cordenadasy.size:    #Verificamos que existan las mismas cordenadas en x y en
y
    if cordenadasx.size<cordenadasy.size:
      print('Verificar la entrada, hay mas cordenadas en el eje y')
    else:
      print('Verificar la entrada, hay mas cordenadas en el eje x')

  arrayaleatorio=np.random.rand(cordenadasx.size) #Se crea un array aleatorio de dimencion las mismas
que el numero de cordenada en x

  for i, pasoDerecha in enumerate(arrayaleatorio < pr):#Se decide si el paso sera a la derecha o hacia
la izquierda dependiendo del array aleatorio
    if pasoDerecha:
      cordenadasx[i]=d+cordenadasx[i]
    else:
      cordenadasx[i]=-d+cordenadasx[i]

  arrayaleatorio=np.random.rand(cordenadasy.size) #Se renueva el array aleatorio de dimencion las
mismas que el numero de cordenada en y

  for i, pasoArriba in enumerate(arrayaleatorio < pr): #Se decide si el paso sera a la derecha o hacia
la izquierda dependiendo del array aleatorio
    if pasoArriba:
      cordenadasy[i]=d+cordenadasy[i]
    else:
      cordenadasy[i]=-d+cordenadasy[i]

  cordenadasx += corrientex*d # se le suma la corriente en x
  cordenadasy += corrientey*d # se le suma la corriente en y

  return (cordenadasx,cordenadasy)
```

```
def graficadora_de_evolucion(N,R,d,pr,pu,corrientex,corrientey,evoluciones):
  '''
    Parametros
      -N           (int)  Numero de particulas
      -R           (float) Radio de la nube de particulas
      -d           (float) la longitud de cada paso
      -pr          (float) probabilidad que de un paso a la derecha (positivo y menor a 1)
      -pu          (float) probabilidad que de un paso hacia arriva (positivo y menor a 1)
      -corrientex  (float) magnitud de la corriente en el eje x
      -corrientey  (float) magnitud de la corriente en el eje y
      -Evoluciones  (int)  Numero de evoluciones que el usuario quiere analizar
    Arroja
      - solo imprime (prints)
  '''
  ab=creador_de_particulas(N,R)

  a=ab[0]
  b=ab[1]

  for i in range  (0,evoluciones+1): #se coloca un for para poder colocar diferentes titulos
dependiendo la imagen
    if i==0:
      print('Partículas generadas '+str(list(a)) + ','+ str(list(b))) #se imprimen las cordenadas
iniciales

      x_avg, y_avg = np.average(a), np.average(b) #Se calcula el promeido de el eje x y el promedio en
el eje y
      x_var = np.sum((a - x_avg)**2) / N #se calcula la varianaza en el eje x
      y_var = np.sum((b - y_avg)**2) / N #se calcula la varianza en el eje y

      print( ) #esto es solo para dejar un espacio entre cada impresion
      print('x prom '+str(round(x_avg,2))+', y prom '+str(round(y_avg,2))+ ' Var(x) ' +
str(round(x_var,2)) + ' Var(y) '+  str(round(y_var,2)))#Se imprime el resultado


      fig = plt.figure(figsize=(5.6, 5.6)) #Declaramos una nueva imagen
      plt.title("Particulas generadas ",fontsize=20) #Colocamos un titulo
      plt.xlabel("Eje x",fontsize=18) #colocamos los ejes
      plt.ylabel("Eje y",fontsize=18)
      plt.plot(a,b, "o") #dibeñamos cada puntito
      plt.xlim(-1.5,1.5) #le damos el tamaño a los ejes
      plt.ylim(-1.5,1.5)
      plt.show()  #mostramos
    else:
      c=evolucionador_de_particulas(a,b,d,pr,pu,corrientex,corrientey) #se le manda a la funcion
evolucionador de particulas
      print('Posición de las parículas despues de '+str(i)+' evoluciones '+str(list(c[0])) + ','+
str(list(c[1]))) #se imprimen las nuevas coordenadas
      a=c[0] #se guardan las nuevas cordenadas
      b=c[1]

      x_avg, y_avg = np.average(a), np.average(b) #Se calcula en nuevo promedio de posiciones
      x_var = np.sum((a - x_avg)**2) / N #se calcula la nueva varianza en el eje x
      y_var = np.sum((b - y_avg)**2) / N #se calcula la nueva varianza en el eje y

      print( )
      print('x prom '+str(round(x_avg,2))+', y prom '+str(round(y_avg,2))+ ' Var(x) ' +
str(round(x_var,2)) + ' Var(y) '+  str(round(y_var,2)))  #Se imprime el resultado

      fig = plt.figure(figsize=(5.6, 5.6))
      plt.title("Evolución "+ str(i),fontsize=20)
      plt.xlabel("Eje x",fontsize=18)
      plt.ylabel("Eje y",fontsize=18)
      plt.plot(a,b, "o")
      plt.xlim(-1.5,1.5)
      plt.ylim(-1.5,1.5)
      plt.show()
```

3. Segunda simulación del movimiento browniano

Anteriormente se vio un modelado que se basaba en estadística para modelar el movimiento observado por Robert Brown en 1827. En esta sección crearemos otra simulación que comparada al modelo anterior el análisis sera mucho mas corto pero aun así la simulación sera mas sofisticada. En esta simulación en lugar de asignar un salto aleatorio en la partícula browniana se le asignara una masa y un radio. Modelaremos las moléculas de agua como partículas esféricas de tamaño finito con una distribución de velocidad determinada donde la partícula browniana estará inicialmente en reposo.

3.1. Segundo modelado de las partículas

En esta simulación supondremos que tenemos N moléculas de agua con una velocidad inicial $\mathbf{v}$, la partícula browniana tendrá una velocidad inicial nula. También supondremos que la masa de la partícula browniana tiene distinta a la masa de una molécula. Supondremos que todos los choques de las moléculas entre ellas mismas y entre la partícula browniana es son totalmente elásticos. Para llevar a cabo esto usaremos álgebra vectorial, ya que de esta forma los cálculos se simplificaran mucho.
Concentremos en una colisión cualquiera del sistema, donde dos esferas (esfera 2 y esfera 1) con vectores posición $\mathbf{r_2}$ y $\mathbf{r_1}$, Además tienen velocidades arbitrarias $\mathbf{v_2}$ y $\mathbf{v_1}$. Primero creemos un vector unitario que apunte del centro de la esfera 2 y a la esfera 1. Este vector esta dado

$$\boldsymbol{n} = \frac{\boldsymbol{r}_2 - \boldsymbol{r}_1}{\|\boldsymbol{r}_2 - \boldsymbol{r}_1\|}$$

Donde definimos el producto punto como

$$\|\boldsymbol{r}_2 - \boldsymbol{r}_1\| = \sqrt{(\boldsymbol{r}_2 - \boldsymbol{r}_1)\cdot(\boldsymbol{r}_2 - \boldsymbol{r}_1)}$$

La velocidad relativa esta dada por

$$v_{rel} = \mathbf{n}\cdot(\mathbf{v_2} - \mathbf{v_1})$$

De manera similar la velocidad relativa después del impacto es

$$v_{\text{choq}} = \boldsymbol{n}\cdot((\boldsymbol{v}_2 + \Delta\boldsymbol{v}_2) - (\boldsymbol{v}_1 + \Delta\boldsymbol{v}_1))$$

Sabemos que en los choques se cumple que:

$$v_{\text{choq}} = -\epsilon v_{rel}$$

Donde ϵ es el coeficiente de elasticidad. Entonces se tiene que:

$$\begin{aligned}
\boldsymbol{n}\cdot((\boldsymbol{v}_2 + \Delta\boldsymbol{v}_2) - (\boldsymbol{v}_1 + \Delta\boldsymbol{v}_1)) &= -\epsilon\boldsymbol{n}\cdot(\boldsymbol{v}_2 - \boldsymbol{v}_1)\\
\boldsymbol{n}\cdot\left(\left(\boldsymbol{v}_2 + \frac{J}{m_2}\boldsymbol{n}\right) - \left(\boldsymbol{v}_1 - \frac{J}{m_1}\boldsymbol{n}\right)\right) &= -\epsilon\boldsymbol{n}\cdot(\boldsymbol{v}_2 - \boldsymbol{v}_1)\\
\boldsymbol{n}\cdot\left(\frac{J}{m_2}\boldsymbol{n} + \frac{J}{m_1}\boldsymbol{n}\right) &= -\epsilon\boldsymbol{n}\cdot(\boldsymbol{v}_2 - \boldsymbol{v}_1) - \boldsymbol{n}\cdot(\boldsymbol{v}_2 - \boldsymbol{v}_1)\\
\boldsymbol{n}\cdot\left(\frac{1}{m_2} + \frac{1}{m_1}\right)\boldsymbol{n}J &= -(1+\epsilon)\boldsymbol{n}\cdot(\boldsymbol{v}_2 - \boldsymbol{v}_1)\\
J &= -(1+\epsilon)\frac{\boldsymbol{n}\cdot(\boldsymbol{v}_2 - \boldsymbol{v}_1)}{\frac{1}{m_2} + \frac{1}{m_1}}
\end{aligned}$$

Donde $\boldsymbol{J} = \Delta\boldsymbol{p} = m\Delta\boldsymbol{v}$ Para un choque elástico $\epsilon = 1$, Entonces

$$J = -2\frac{\boldsymbol{n}\cdot(\boldsymbol{v}_2 - \boldsymbol{v}_1)}{\frac{1}{m_2} + \frac{1}{m_1}}$$

Pero también sabemos que

$$\mathbf{J} = \Delta\mathbf{P} = m\Delta\mathbf{v}$$

Despejando $\Delta\mathbf{v}$

$$\Delta\boldsymbol{v}_1 = -\frac{J}{m_1}\boldsymbol{n} \quad \Delta\boldsymbol{v}_2 = +\frac{J}{m_2}\boldsymbol{n}$$

Sustituyendo J y simplificando se tiene que para una colisión entre dos partículas de agua (masas iguales) se tiene

$$\vec{v}_1^{\,\text{nueva}} = \vec{v}_1 - \frac{(\vec{v}_1 - \vec{v}_2)\cdot(\vec{r}_1 - \vec{r}_2)}{|\vec{r}_1 - \vec{r}_2|^2}(\vec{r}_1 - \vec{r}_2)$$

$$\vec{v}_2^{\,\text{nueva}} = \vec{v}_2 - \frac{(\vec{v}_2 - \vec{v}_1)\cdot(\vec{r}_2 - \vec{r}_1)}{|\vec{r}_1 - \vec{r}_2|^2}(\vec{r}_2 - \vec{r}_1)$$

Para una colisión entre la partícula browniana y una partícula del agua se tiene que (masas distintas)

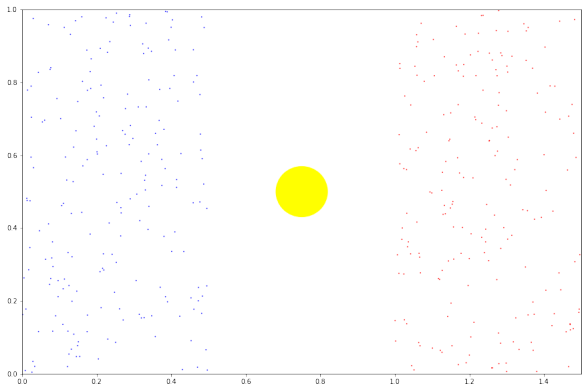
$$\vec{v}_1^{\,\text{nueva}} = \vec{v}_1 - \left(\frac{2m_2}{m_1 + m_2}\right)\frac{(\vec{v}_1 - \vec{v}_2)\cdot(\vec{r}_1 - \vec{r}_2)}{|\vec{r}_1 - \vec{r}_2|^2}(\vec{r}_1 - \vec{r}_2)$$

$$\vec{v}_2^{\,\text{nueva}} = \vec{v}_2 - \left(\frac{2m_1}{m_1 + m_2}\right)\frac{(\vec{v}_2 - \vec{v}_1)\cdot(\vec{r}_2 - \vec{r}_1)}{|\vec{r}_1 - \vec{r}_2|^2}(\vec{r}_2 - \vec{r}_1)$$

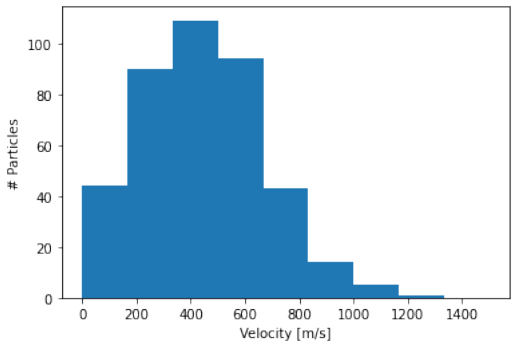
Estas ecuaciones las que usaremos para calcular la velocidad final después de una colisión

3.2. Resultados del segundo modelado

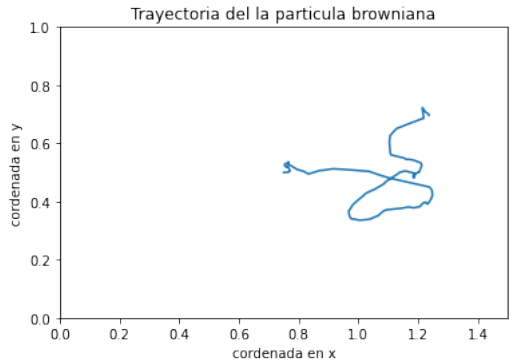
Primera simulación
Se generaron 400 partículas con un una velocidad inicial de $500m/s$. Las moléculas del agua tenían un radio de $0.0015m$ y una masa de $0.0015kg$. La partícula browniana tenia una masa de $0.01kg$ y un radio de 0.07



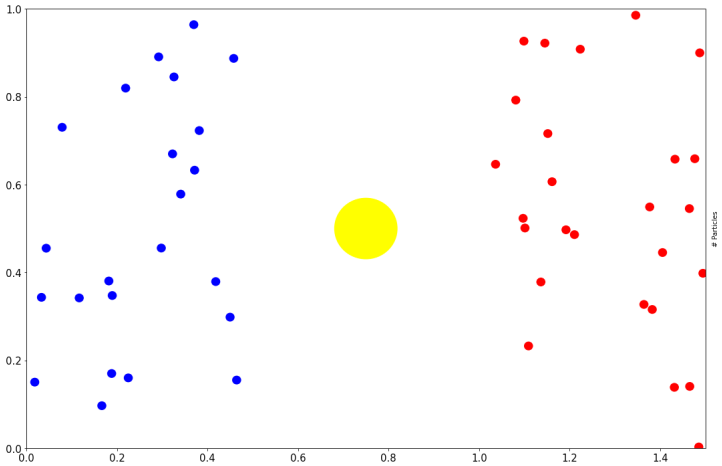
El histograma de las velocidades fue



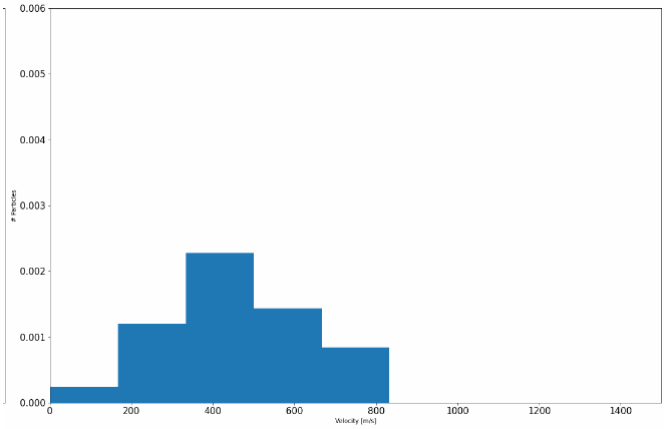
El movimiento de la partícula browniana fue



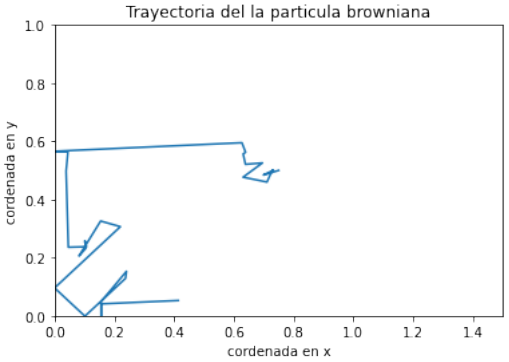
Segunda simulación
Se generaron 50 partículas con un una velocidad inicial de $500m/s$. Las moléculas del agua tenían un radio de $0.01m$ y una masa de $0.01kg$. La partícula browniana tenia una masa de $0.01kg$ y un radio de 0.07



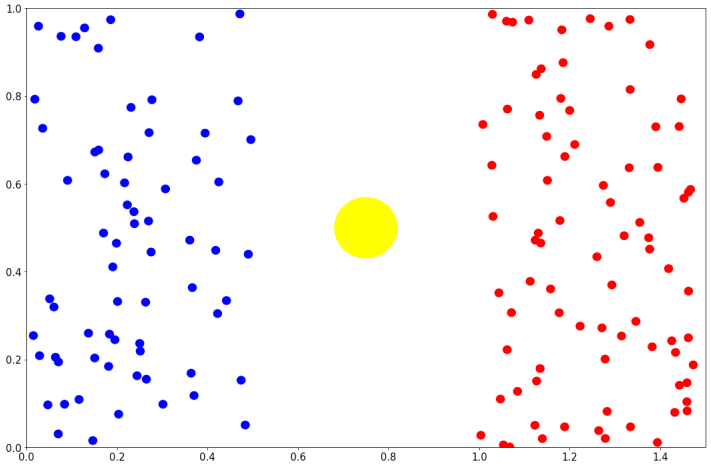
El histograma de las velocidades fue

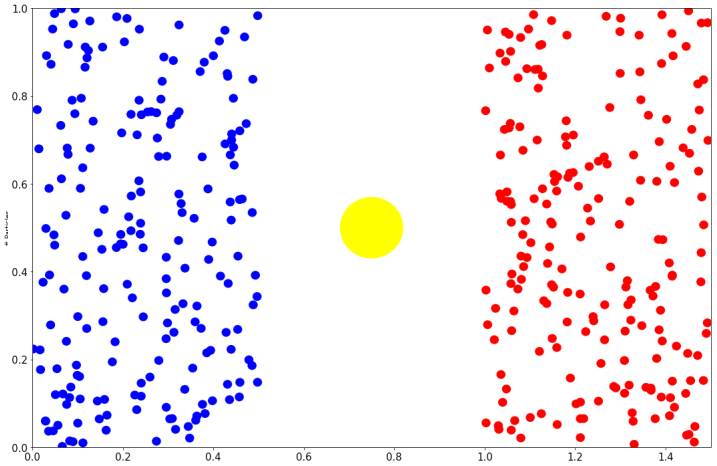


El movimiento de la partícula browniana fue

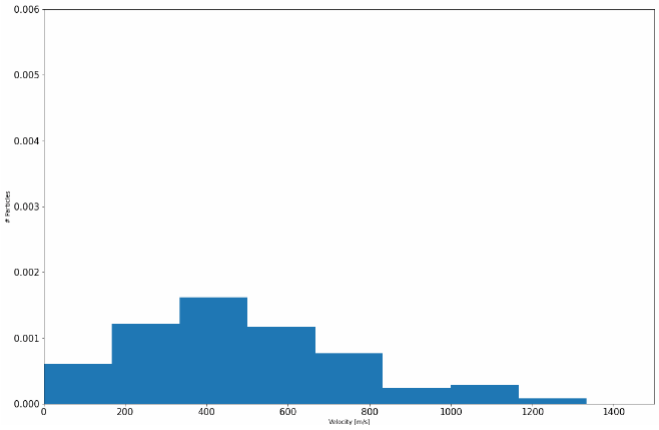


Tercera simulación
Se generaron 150 partículas con un una velocidad inicial de $500m/s$. Las moléculas del agua tenían un radio de $0.01m$ y una masa de $0.01kg$. La partícula browniana tenia una masa de $0.01kg$ y un radio de 0.07

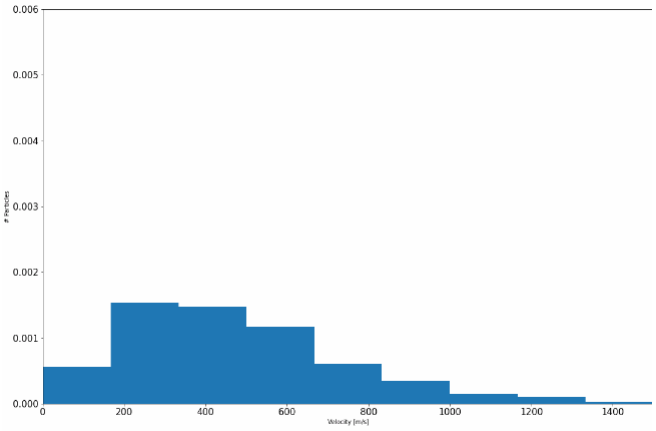




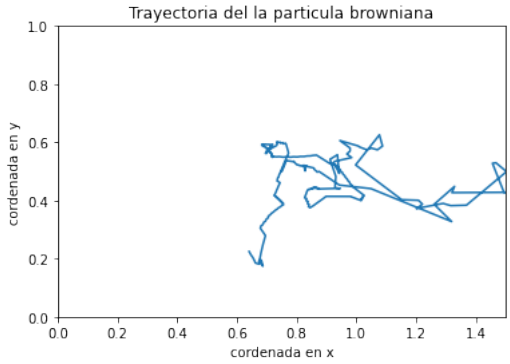
El histograma de las velocidades fue



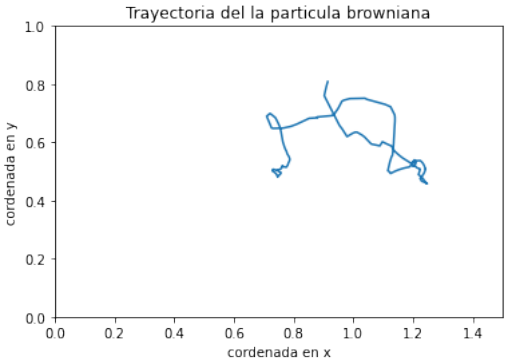
El histograma de las velocidades fue



El movimiento de la partícula browniana fue

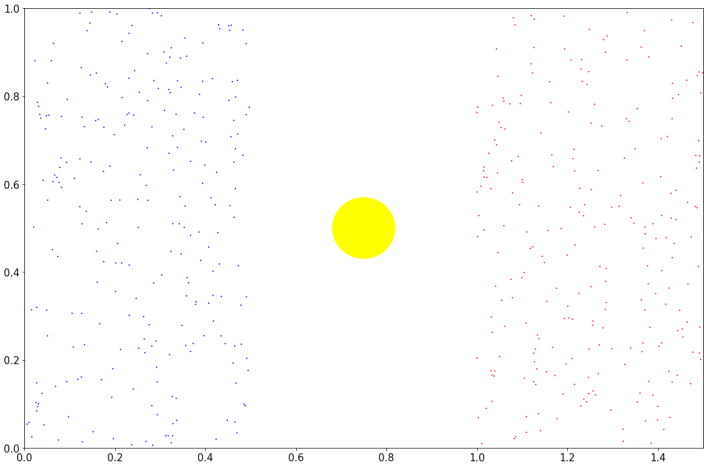


El movimiento de la partícula browniana fue



Cuarta simulación
Se generaron 400 partículas con un una velocidad inicial de $500m/s$. Las moléculas del agua tenían un radio de $0.01m$ y una masa de $0.01kg$. La partícula browniana tenia una masa de $0.07kg$ y un radio de 0.07

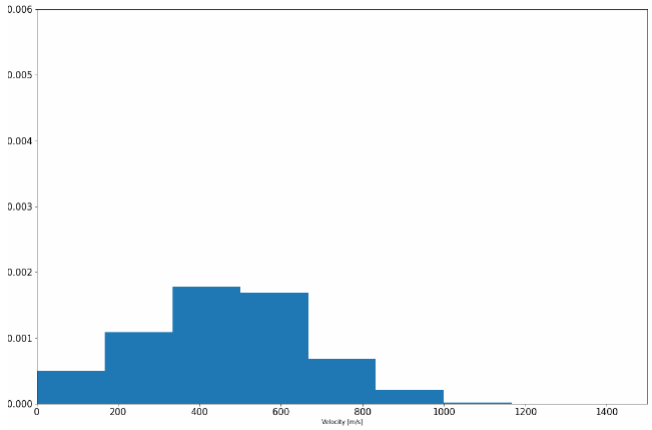
Quinta simulación
Se generaron 400 partículas con un una velocidad inicial de $500m/s$. Las moléculas del agua tenían un radio de $0.0015m$ y una masa de $0.0015kg$. La partícula browniana tenia una masa de $0.07kg$ y un radio de 0.07



3.3. Código

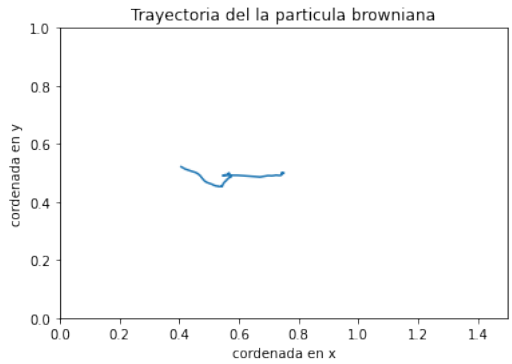
```
n_particles = 400
r = np.random.random((2,n_particles))
ixr = r[0]>0.5
ixl = r[0]<=0.5
ifirst= np.full(n_particles,False);
ifirst[0]=True
for i in range (r[0].size):
  if .5 < r[0][i]:
    r[0][i]=r[0][i]+0.5

#Le cambiamos las cordenadas a la paricula browniana para que comience el centro
r[0][0]=.75
r[1][0]=.5
ids = np.arange(n_particles)
ids_pairs = np.asarray(list(combinations(ids,2)))
v = np.zeros((2,n_particles))
v[0][ixr] = -500
v[0][ixl] = 500
v[0][ifirst] = 0
radius = 0.0015
mc=0.0015
radiusG = .07
mg=.05
ts=1000
dt=0.000008
d_cutoff =2*radius
d_grande=radius+radiusG

rs, vs = motion(r, v, ids_pairs, ts, dt, 2*radius, radius+radiusG, mc, mg)
```

El histograma de las velocidades fue




```
def get_delta_pairs(x):
    return np.diff(np.asarray(list(combinations(x,2))), axis=1).ravel() #calcula la distancia entre dos particulas en una dimencion

def get_deltad_pairs(r):
    return np.sqrt(get_delta_pairs(r[0])**2 + get_delta_pairs(r[1])**2) #calcula la distancia entre dos particulas total (2 dimenciones)

def compute_new_v(v1c, v2c, r1c, r2c, v1g, v2g, r1g, r2g, mc, mg): #Esta funcion me crea los nuevos valores de velocidad
    v1cnew = v1c - np.diag((v1c-v2c).T@(r1c-r2c))/np.sum((r1c-r2c)**2, axis=0) * (r1c-r2c)
    v2cnew = v2c - np.diag((v2c-v1c).T@(r2c-r1c))/np.sum((r2c-r1c)**2, axis=0) * (r2c-r1c)

    v1gnew = v1g -(2*mc/(mg+mc))*np.diag((v1g-v2g).T@(r1g-r2g))/np.sum((r1g-r2g)**2, axis=0) * (r1g-r2g)
#CUIDADO CON EL FACTOR SI ESTA MAL AQUI PUEDE SER EL PORQUE
    v2gnew = v2g -(2*mg/(mg+mc))*np.diag((v2g-v1g).T@(r2g-r1g))/np.sum((r2g-r1g)**2, axis=0) * (r2g-r1g)
#CUIDADO CON EL FACTOR SI ESTA MAL AQUI PUEDE SER EL PORQUE
    return v1cnew, v2cnew, v1gnew, v2gnew

def Nataly_q(a): #esta funcion te devuelve los mismos pares pero sin los ceros iniciales
    contador = 0
    for i in range (len(a)):
        if a[i][0]==0:
          contador = contador + 1
    return a[contador:len(a)]

def Nataly_d(a): #esta funcion te devuelve los mismos pares pero solos los que incluyen los ceros iniciales
    contador = 0
    for i in range (len(a)):
        if a[i][0]==0:
          contador = contador + 1
    return a[0:contador]
```

El movimiento de la partícula browniana fue



```python
def motion(r, v, id_pairs, ts, dt, d_cutoff, d_grande, mc, mg):
    rs = np.zeros((ts, r.shape[0], r.shape[1])) #creamos ts matrices r (esto se hace para guardar todas las posiciones por calcular)
    vs = np.zeros((ts, v.shape[0], v.shape[1])) #creamos ts matrices v (esto se hace para guardar todas las velocidades por calcular)
    # Valores iniciales
    rs[0] = r.copy() #hacemos una copia de la primera matriz (valores inicales)
    vs[0] = v.copy() #hacemos una copia de la primera matriz (valores inicales)

    for i in range(1,ts):

        ic = id_pairs[get_deltad_pairs(r) < d_cutoff] #esta es una lista de las particulas que colicionan para esta condicon (2r)
        ig = id_pairs[get_deltad_pairs(r) < d_grande] #esta es una lista de las particulas que colicionan para esta condicon (d_grande)

        icc=Nataly_q(ic) #aqui quite todas las colisiones que incluyen a la particula grande
        igg=Nataly_d(ig) #aqui solo deje las colisiones que tienen que ver con la particula grande

        v[:,icc[:,0]], v[:,icc[:,1]],  v[:,igg[:,0]], v[:,igg[:,1]]= compute_new_v(v[:,icc[:,0]], v[:,icc[:,1]],  r[:,icc[:,0]],  r[:,icc[:,1]], v[:,igg[:,0]],  v[:,igg[:,1]],  r[:,igg[:,0]], r[:,igg[:,1]], mc, mg) #Estoy uzando la funcion compute new

        v[0,r[0]>1.5] = -np.abs(v[0,r[0]>1.5]) #Las siguientes lineas evitan que las particulas se salgan del cuadrado 1X1 cambiando en cada segundo un 1 por un 2
        v[0,r[0]<0] = np.abs(v[0,r[0]<0])
        v[1,r[1]>1] = -np.abs(v[1,r[1]>1])
        v[1,r[1]<0] = np.abs(v[1,r[1]<0])

        r = r + v*dt
        rs[i] = r.copy()
        vs[i] = v.copy()
    return rs, vs
```

4. Conclusión y comparación

El modelo que establece una probabilidad de movimiento en cada eje es bastante bueno, vemos que la nube de partícula si se expande tal y como pasa cuando abrimos un perfume y aun sin poner la nariz cerca del frasco nos llega su olor. Además es mucho mas rápido que el modelo de colisiones. El modelo de colisiones es bastante mas interesante, pero mucho mas costoso computacional-mente. Por otro lado se vio que la partícula browniana sigue una trayectoria en algunos casos tal y como Robert Brown, es decir zigzagueado.

Referencias

[1] Robert Brown , *A brief account of microscopical observations made in the months of June, July and August, 1827, on the particles contained in the pollen of plants; and on the general existence of active molecules in organic and inorganic bodies*, The Ray Society, London, 1866, págs. 463.

[2] Louis Bachelier, *The Moment Generating Function of the Normal Distribution* 1996

[3] M. Smoluchowski, *Zur kinetischen Theorie der Brownschen Molekularbewegung und der Suspensionen*, 1906 págs. 756.

[4] Maria Garcia, Aurora Hermoso, Juan Antonio, Patricia Roman, y Francisco Torres, *P_T05_FGMNormal, Grupo de invocación docente de la universidad de Granada.*