

Anexo Trabajo Práctico 3

Grupo 9: Kramer Esteban, Scmazzon Martina

Abril 2019

1 Introducción

A raíz de lo hablado en clase se generaron una serie de gráficos con el fin de describir y encontrar la función que más se adecúe al Desplazamiento Cuadrático Medio de la partícula grande.

2 Implementación

2.1 Descripción

Para lograr el objetivo se obtuvo desde el código creado en Java distintos MSDs para intervalos de tiempo incrementales múltiplos de 5 segundos.

Se realizaron dos experiencias con 100 y 200 partículas en 195 segundos. Dado que al comenzar el algoritmo algunos de los valores arrojados no describen una tendencia lineal, los primeros 7 valores fueron desestimados, comenzando así a partir del intervalo de 30 segundos.

Los valores fueron almacenados en un archivo de texto para luego ser abiertos desde Octave.

2.2 Código Octave

Al comenzar la experiencia generamos pendientes de 0 a 100 con intervalos de 1, graficamos dichas rectas junto con los valores de MSDs obtenidos, y notamos que la diferencia era muy grande, por lo tanto decidimos decrementar los intervalos con el fin de encontrar la pendiente mas adecuada.

```
# Valor de pendiente. Determina la cantidad de ecuaciones 'y' que voy a tener  
m = (0:0.00001:0.0009);
```

```
# Valor de t, donde voy a evaluar la funcion 'y'  
t = (30:5:195);
```

Por otro lado, generamos una matriz donde las filas van a representar las distintas rectas y sus respectivas pendientes y las columnas $y(t)$, donde t expresa el tiempo. Graficamos en una misma figura, las rectas con distintas pendientes

y los valores de MSDs obtenidos con el fin de obtener un aproximado a nivel visual.

```

for i=1:length(m)
    for j = 1:length(t)
        # Matriz de valor de 'ym'. Columnas = y(t) Filas = Cada Ym
        y(i,j) = m(i)*t(j);
    endfor
endfor

figure(1)
for i = 1:length(m)
    plot(t,y(i,:), 'b-')
    hold on
end
xlabel( 'Tiempo [s] ')
ylabel( 'y(t) ')
title( 'Grafico de y(t) para distintas pendientes ')
grid on
grid minor

# Abrimos archivo generado en Java
d = dlmread( 'valuesForGraph.txt ')
figure(1)
plot(t,d, 'r-')

```

Ahora, vamos en busca de la segunda figura, el error. Siguiendo la misma línea de pensamiento generamos una nueva matriz de error, en la misma se almacenara por cada fila el error correspondiente a la pendiente indicada. Si tenemos 11 pendientes, tendremos 11 filas y una única columna en la matriz. Graficaremos dicho error.

Nos quedaremos con el error mas pequeño y generaremos una última figura, una en la cual tengamos la pendiente que haya dado el menor error posible y la función de los MSDs obtenidos.

```

error = 0;
# for externo, me va a agarrar la fila del 'm' correspondiente en 'y'
for i = 1:length(m)
    c = y(i,:);
    # ahora voy a recorrer y tomar los valores de 'd'
    for j = 1:length(d)
        error += (d(j) - c(j))^2;
    endfor
    e(i,1) = error;
    error = 0;
endfor

```

```

figure(2)
for i = 1:length(m)
    plot(m,e(:,1), 'r. ')
    xlabel( 'Pendientes ');
    ylabel( 'Error ');
    title( 'Error entre DCM y las distintas pendientes planteadas ')
    hold on
endfor

```

```

figure(3)
for i = 1:length(m)
    index = find(e == min(e));
    plot(t,y(index,:), 'b-')
    xlabel( 'Tiempo [s] ')
    ylabel( 'y(t) ')
    title( 'Grafico de la funcion con menor error vs. DCM ')
    hold on
endfor

```

```

figure(3)
plot(t,d, 'r-')

```

3 Resultados

3.1 Experiencia 1

Con una configuración de 100 partículas en 195 segundos.

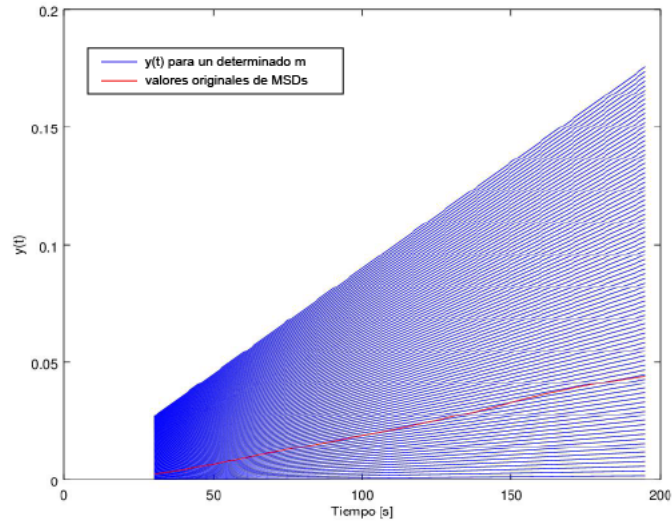


Figura 1: Rectas de pendientes aleatorias vs MSDs obtenidos

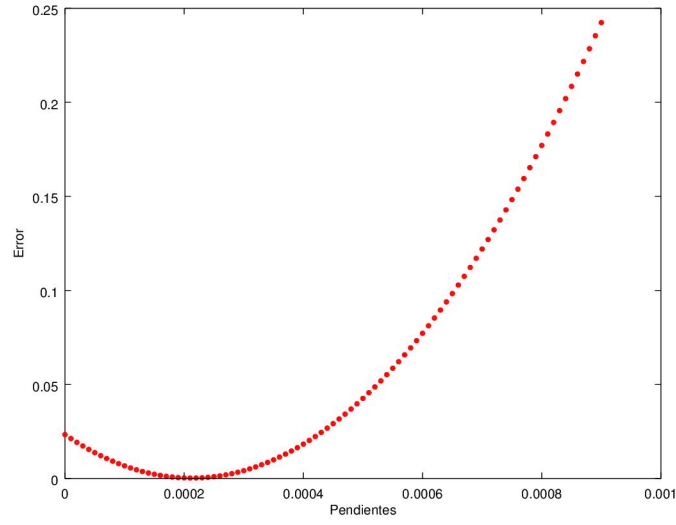


Figura 2: Error en función de las distintas pendientes

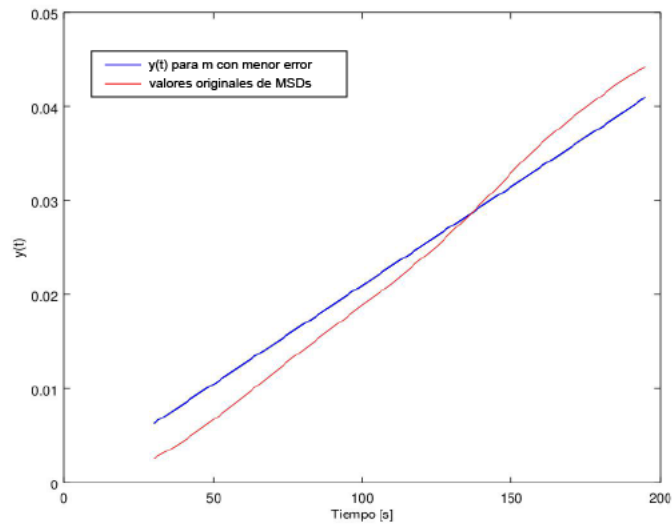


Figura 3: Recta con menor error vs. MSDs obtenidos

Generando pendientes entre 0 y 0.0009 con un intervalo de 0.00001 el menor error obtenido fue $2.5611\text{e-}04$. La pendiente correspondiente es $m = 2.1000\text{e-}04$

3.2 Experiencia 2

Con una configuración de 200 partículas en 195 segundos.

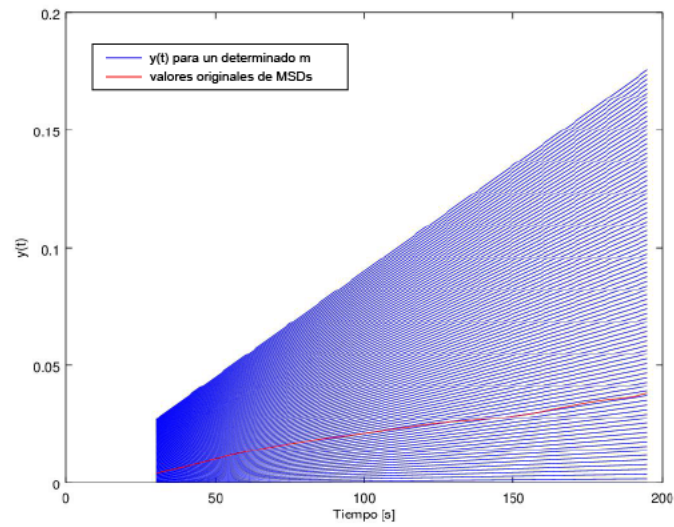


Figura 4: Rectas de pendientes aleatorias vs MSDs obtenidos

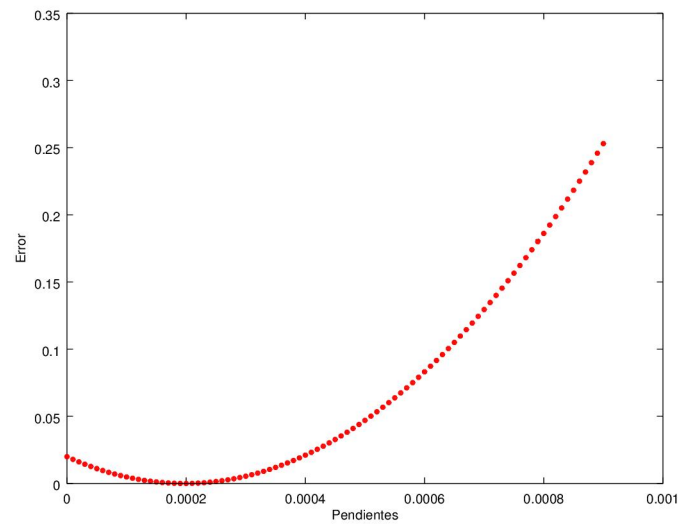


Figura 5: Error en función de las distintas pendientes

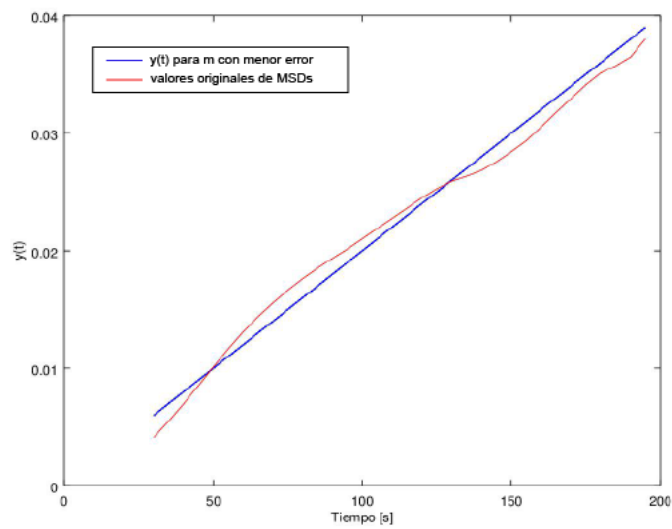


Figura 6: Recta con menor error vs. MSDs obtenidos

Generando pendientes entre 0 y 0.0009 con un intervalo de 0.00001 el menor error obtenido fue 4.7442e-05. La pendiente correspondiente es $m = 2.0000\text{e-}04$