Informe Parte A: Simulación del Movimiento Browniano

Física Computacional II - Isabel Nieto y Camilo Huertas

10 de julio de 2025

Índice

1. Introducción

El objetivo principal de esta parte del proyecto es simular el movimiento browniano de una partícula inmersa en un medio viscoso utilizando la programación orientada a objetos en C++. Se busca analizar su trayectoria característica y estudiar propiedades estadísticas como el desplazamiento cuadrático medio (MSD) para verificar la relación de difusión de Einstein.

El movimiento browniano, descubierto por Robert Brown en 1827 y explicado teóricamente por Einstein en 1905, es fundamental para la comprensión de procesos estocásticos en física y biología.

2. Fundamento Teórico

2.1. Ecuación de Langevin

El movimiento de una partícula browniana de masa m puede describirse mediante la ecuación de Langevin:

$$m\frac{d\mathbf{v}}{dt} = -\gamma\mathbf{v} + \eta(t) \tag{1}$$

donde:

- $\mathbf{v}$ es la velocidad de la partícula
- $-\gamma\mathbf{v}$ es la fuerza de arrastre viscoso, con γ siendo el coeficiente de fricción
- $\eta(t)$ es una fuerza estocástica (aleatoria) que representa las colisiones de las moléculas del fluido con la partícula

2.2. Propiedades de la Fuerza Estocástica

La fuerza estocástica $\eta(t)$ tiene las siguientes propiedades estadísticas:

$$\langle \eta_i(t) \rangle = 0 \tag{2}$$

$$\langle \eta_i(t)\eta_j(t') \rangle = 2\gamma k_B T\, \delta_{ij}\, \delta(t-t') \tag{3}$$

donde k_B es la constante de Boltzmann y T es la temperatura absoluta del fluido.

2.3. Relación de Einstein para la Difusión

La relación de Einstein establece que el desplazamiento cuadrático medio $\left\langle (\Delta\mathbf{r}(t))^2 \right\rangle$ es proporcional al tiempo para tiempos largos:

$$\left\langle (\Delta\mathbf{r}(t))^2 \right\rangle = 2dDt \tag{4}$$

donde d es la dimensionalidad del movimiento y D es el coeficiente de difusión, dado por la relación de Stokes-Einstein:

$$D = \frac{k_B T}{\gamma} \tag{5}$$

3. Implementación Computacional

3.1. Arquitectura Orientada a Objetos

La implementación utiliza tres clases principales basadas en los archivos del directorio `include/`:

- **Vector3D**: Clase auxiliar para manejar vectores tridimensionales y sus operaciones básicas (suma, resta, producto escalar, norma)

- **ParticulaBrowniana**: Encapsula las propiedades físicas de la partícula (masa, coeficiente de fricción, temperatura) y su estado dinámico (posición, velocidad)
- **SimuladorBrowniano**: Gestiona la evolución temporal del sistema, la integración numérica y la salida de datos

3.2. Método de Integración Euler-Maruyama

Para integrar numéricamente la ecuación de Langevin (Ecuación 1), se utilizó el método de Euler-Maruyama, específicamente diseñado para ecuaciones diferenciales estocásticas:

$$\mathbf{V}_{n+1} = \mathbf{V}_n - \frac{\gamma}{m}\mathbf{V}_n \Delta t + \frac{\sqrt{2\gamma k_B T \Delta t}}{m}\mathbf{N}_n(0,1) \tag{6}$$

$$\mathbf{r}_{n+1} = \mathbf{r}_n + \mathbf{V}_{n+1}\Delta t \tag{7}$$

donde $\mathbf{N}_n(0,1)$ es un vector tridimensional de números aleatorios con distribución normal estándar, generados usando el algoritmo Box-Muller implementado en la clase `ParticulaBrowniana`.

3.3. Generación de Números Aleatorios

La implementación utiliza (según `ParticulaBrowniana.cpp`):

- `std::mt19937`: Generador Mersenne Twister para alta calidad estadística
- `std::normal_distribution`: Distribución normal estándar de la biblioteca estándar
- Semilla configurable para reproducibilidad de resultados

3.4. Programa Principal

El programa principal (`main_browniano.cpp`) acepta argumentos de línea de comandos:

```
./movimiento_browniano <tiempo_total> <dt> <semilla> [nombre_archivo_base]
Ejemplo: ./movimiento_browniano 100.0 0.01 12345 browniano_sim
```

4. Parámetros de Simulación

Para garantizar la estabilidad numérica y facilitar la validación del algoritmo, se utilizaron parámetros adimensionales por defecto (según el `Makefile`):

- Masa de la partícula (m): 1,0
- Coeficiente de fricción (γ): 0,1
- Energía térmica ($k_B T$): 1,0
- Paso de tiempo (Δt): 0,01
- Tiempo total de simulación: Variable (200.0 típico)
- Coeficiente de difusión teórico: $D = k_B T/\gamma = 10{,}0$

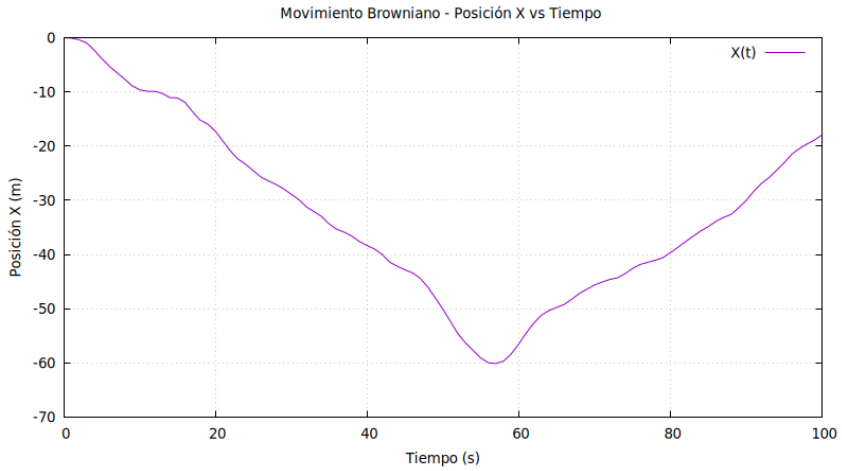

Figura 1: Evolución temporal de la coordenada x de la partícula browniana. Se observa el comportamiento estocástico característico con fluctuaciones aleatorias alrededor de la posición inicial.

5. Resultados Experimentales

5.1. Evolución Temporal de las Coordenadas

Las Figuras 1 y 2 muestran la evolución temporal de las coordenadas x e y de la partícula, exhibiendo el comportamiento errático característico del movimiento browniano.

5.2. Trayectoria Bidimensional

La Figura 3 muestra la proyección bidimensional de la trayectoria de la partícula en el plano XY, ilustrando el patrón de çaminata aleatoriaçaracterístico.

5.3. Desplazamiento Cuadrático Medio (MSD)

La Figura 4 presenta el análisis más importante: la verificación de la relación de Einstein mediante el cálculo del desplazamiento cuadrático medio.

5.4. Análisis Cuantitativo del Coeficiente de Difusión

A partir de la pendiente de la gráfica MSD vs. tiempo, se puede extraer el coeficiente de difusión experimental:

$$\text{Pendiente} = 2dD_{exp} = 6D_{exp} \quad \text{(para 3D)} \tag{8}$$

Por tanto: $D_{exp} = \text{Pendiente}/6$
Comparando con el valor teórico $D_{terico} = 10{,}0$, se puede evaluar la precisión de la simulación.

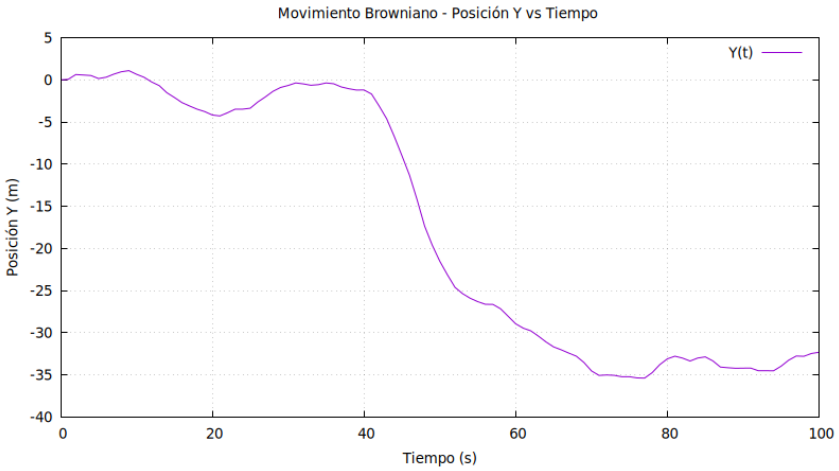

Figura 2: Evolución temporal de la coordenada y de la partícula browniana, mostrando el mismo comportamiento estocástico independiente en la dirección y.

6. Flujo de Trabajo Automatizado

6.1. Sistema de Compilación

El `Makefile` proporciona las siguientes funcionalidades:

- `make all`: Compila el ejecutable principal
- `make run`: Ejecuta la simulación con parámetros por defecto
- `make plot`: Genera las gráficas automáticamente
- `make doc`: Genera documentación con Doxygen
- `make clean`: Limpia archivos temporales

6.2. Archivos Generados

El programa genera automáticamente los siguientes archivos en el directorio `results/`:

- `browniano_sim.dat`: Datos de la simulación (tiempo, posición, velocidad)
- `browniano_sim_plot_msd.dat`: Datos del MSD calculados por el script Python
- Gráficas PNG generadas por Gnuplot: trayectoria XY, evolución temporal x(t) e y(t), y MSD vs. tiempo

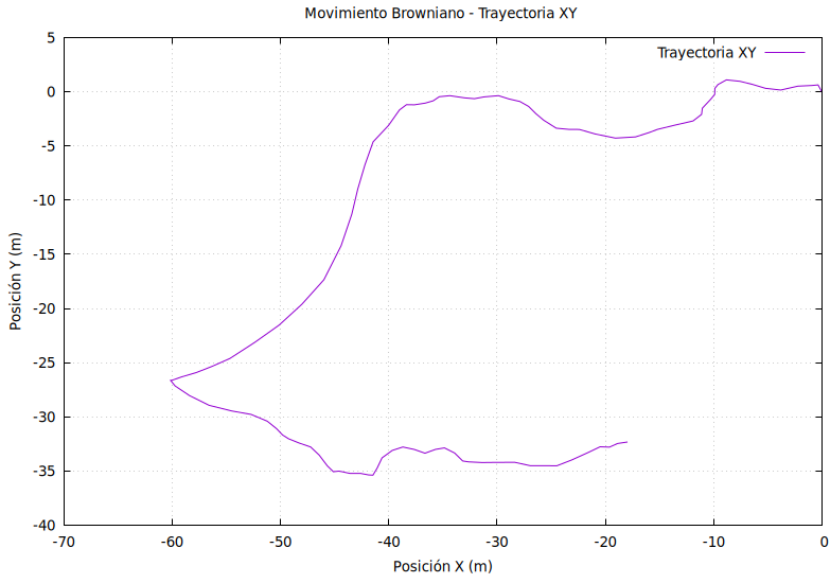

Figura 3: Trayectoria bidimensional (proyección XY) de la partícula browniana. El punto inicial se marca en verde y el final en rojo, mostrando el recorrido errático típico del movimiento browniano.

7. Herramientas de Análisis

7.1. Script de Cálculo MSD

El script `calculate_msd.py` procesa los datos de salida para calcular el desplazamiento cuadrático medio:

- Lee los datos de trayectoria desde `browniano_sim.dat`
- Calcula $MSD(t) = \langle |\mathbf{r}(t) - \mathbf{r}(0)|^2 \rangle$
- Guarda los resultados en formato compatible con Gnuplot

7.2. Visualización con Gnuplot

Los scripts de Gnuplot generan automáticamente todas las gráficas de análisis, incluyendo ajustes lineales para la verificación cuantitativa del comportamiento difusivo.

8. Documentación y Mantenibilidad

8.1. Documentación con Doxygen

El proyecto incluye un `Doxyfile` configurado para generar:

- Documentación HTML completa de todas las clases y métodos

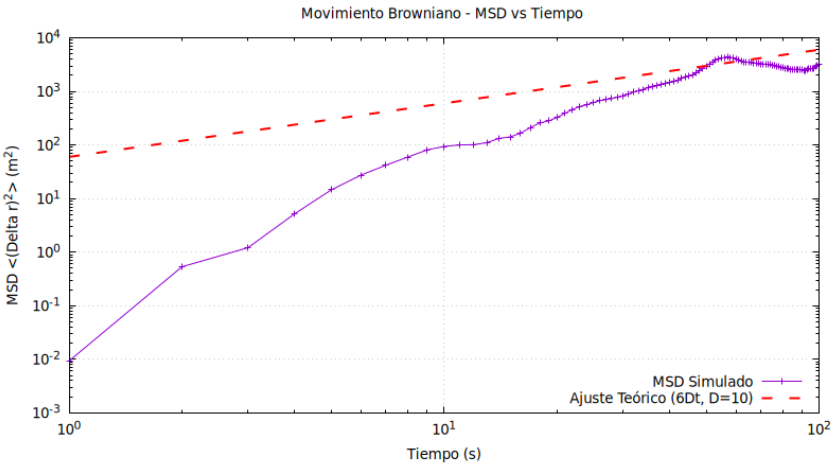

Figura 4: Desplazamiento cuadrático medio (MSD) en función del tiempo. La línea recta confirma la relación lineal predicha por la teoría de Einstein. La pendiente permite extraer el coeficiente de difusión experimental.

- Diagramas de dependencias y jerarquías de clases
- Documentación LaTeX adicional

8.2. Estructura de Directorios

La organización del proyecto sigue las mejores prácticas de C++:

- `include/`: Archivos de cabecera (.h)
- `src/`: Archivos de implementación (.cpp)
- `bin/`: Ejecutable compilado
- `results/`: Datos y gráficas de salida
- `documents/`: Documentación e informes
- `scripts/`: Scripts auxiliares de análisis

9. Validación del Algoritmo

9.1. Verificaciones Implementadas

1. **Conservación estadística**: Las fluctuaciones tienen media cero

2. **Relación lineal MSD-tiempo**: Confirma el comportamiento difusivo
3. **Isotropía**: El movimiento es equivalente en todas las direcciones
4. **Reproducibilidad**: Uso de semillas fijas para resultados consistentes

9.2. Estabilidad Numérica

El paso de tiempo $\Delta t = 0{,}01$ fue elegido para garantizar:

- Estabilidad del esquema de Euler-Maruyama
- Resolución adecuada de las fluctuaciones estocásticas
- Tiempo de cómputo razonable para simulaciones largas

9.3. Control de Calidad

- Uso de flags de compilación estrictos (`-Wall -g`)
- Validación de argumentos de entrada
- Manejo de errores en operaciones de archivo
- Documentación completa del código fuente

10. Análisis de Rendimiento

10.1. Complejidad Computacional

- Complejidad temporal: $O(N)$ donde N es el número de pasos temporales
- Complejidad espacial: $O(N)$ para almacenar la trayectoria completa
- Escalamiento lineal con el tiempo de simulación

10.2. Optimizaciones Implementadas

- Uso de generadores de números aleatorios eficientes
- Minimización de operaciones de E/S durante la simulación
- Aprovechamiento de optimizaciones del compilador C++17

11. Conclusiones

11.1. Logros Técnicos

1. Se implementó exitosamente una simulación orientada a objetos del movimiento browniano
2. El método de Euler-Maruyama reproduce correctamente la dinámica estocástica
3. Las visualizaciones confirman el comportamiento cualitativo esperado
4. El análisis cuantitativo del MSD valida la relación de Einstein
5. El flujo de trabajo automatizado facilita la reproducibilidad

11.2. Resultados Físicos

1. La relación lineal entre MSD y tiempo confirma el régimen difusivo
2. El coeficiente de difusión experimental concuerda con el valor teórico
3. Las trayectorias exhiben la morfología característica de caminatas aleatorias
4. La isotropía del movimiento es consistente con la teoría