

# SIMULACIÓN DE LAS ECUACIONES DE LANGEVIN EN DOS Y TRES DIMENSIONES

Julio Pérez Vásquez

*(18 de julio de 2025)*

## ¿Por qué decidí estudiar este problema?

Siempre me ha llamado la atención cómo las partículas pueden moverse de forma tan impredecible en sistemas reales, como en líquidos o gases. Justamente, el movimiento browniano es un buen ejemplo de esto. Las ecuaciones de Langevin son una herramienta que nos permite meter en el modelo tanto la fricción (que frena la partícula) como el ruido aleatorio (lo impredecible del entorno). Por eso, me interesó simular estos sistemas de manera numérica, para ver cómo se comportan en la práctica y explorar casos donde no existe una solución analítica sencilla.

## ¿Qué hace mi código?

Lo que hago en este proyecto es simular el movimiento de una partícula usando las ecuaciones de Langevin, tanto en 2D como en 3D. Además, me enfoco en comparar el efecto de dos tipos de ruido: uniforme y gaussiano. Así puedo analizar cómo cambia la trayectoria de la partícula dependiendo de la aleatoriedad que le agrego.

## Árbol de directorios de mi proyecto

```
proyecto_final
|- langevin2D
|  |- include/
|  |  |-- langevin2D.h
|  |  |-- rk_4.h
|  |- scripts/
|  |  |-- plot2D.gp
|  |- src/
|  |  |-- langevin2D.cpp
|  |  |-- main.cpp
|  |  |-- rk_4.cpp
|  |- makefile
```

```
proyecto_final
|- langevin3D
|  |- include/
|  |  |-- langevin3D.h
|  |  |-- rk_4.h
|  |- scripts/
|  |  |-- plot3D.gp
|  |- src/
|  |  |-- langevin3D.cpp
|  |  |-- main.cpp
|  |  |-- rk_4.cpp
|  |- makefile
```

## Archivos principales:

- `main.cpp`: Aquí configuro la simulación y la ejecuto.
- `langevin2D.cpp/h`: Defino toda la dinámica del sistema Langevin en 2D.
- `rk_4.cpp/h`: Implemento el método de Runge-Kutta 4 para resolver las ecuaciones diferenciales.
- `plot2D.gp`: Script para graficar los resultados y visualizar lo que pasa.

## ¿Cómo funciona cada parte?

**Método RK4:** Uso este método porque es preciso y bastante confiable para resolver ecuaciones diferenciales. El código está hecho de tal forma de que sea re-utilizable.

```
1 class rk4
2 {
3 public:
4     using ODEFunction = std::function<std::vector<double>(double, const
        ↪ std::vector<double>&)>;
5 private:
6     ODEFunction f;
7     ...
8 };
```

**Sistema Langevin 2D:** En esta parte defino cómo la fricción y el ruido afectan a la partícula. El ruido puede ser uniforme o gaussiano, para comparar ambos casos.

```
1 std::vector<double> LangevinSystem2D::operator()(double t, const std::vector<double>&
    ↪ y)
2 {
3     double eta_x, eta_y;
4     if (noise_type_ == UNIFORM)
5     {
6         eta_x = uniform_dist_(gen_);
7         eta_y = uniform_dist_(gen_);
8     }
9     else
10    {
11        eta_x = gaussian_dist_(gen_);
12        eta_y = gaussian_dist_(gen_);
13    }
14    double vx = y[1];
15    double vy = y[3];
16    double dvxdt = -gamma_ * vx + eta_x;
17    double dvydt = -gamma_ * vy + eta_y;
18    return {vx, dvxdt, vy, dvydt};
19 }
```

**Programa principal:** Acá inicializo los parámetros y corro la simulación para cada tipo de ruido, guardando los datos y luego graficando.

```
1 int main()  
2 {  
3     LangevinSystem2D uniform_system(LangevinSystem2D::UNIFORM, GAMMA);  
4     LangevinSystem2D gaussian_system(LangevinSystem2D::GAUSSIAN, GAMMA);  
5  
6     run_simulation("uniforme", uniform_system);  
7     run_simulation("gaussiano", gaussian_system);  
8     ...  
9 }
```

Una vez entendido lo anterior procedemos a compilar para luego ejecutar nuestro código, así tenemos

```
$> ./main  
Iniciando simulación 2D con ruido uniforme...  
-> Datos guardados en 'data/langevin2d_uniforme.dat'  
Iniciando simulación 2D con ruido gaussiano...  
-> Datos guardados en 'data/langevin2d_gaussiano.dat'  
  
Simulaciones 2D completadas.  
$>
```

## Resultados

Al final lo que obtengo son archivos de datos y gráficos. Estos muestran cómo se mueve la partícula bajo diferentes condiciones de ruido, y nos permiten ver claramente la diferencia entre usar ruido uniforme y gaussiano.

**\*Aclaración:** En este informe presento únicamente las gráficas correspondientes al caso de Langevin 2D, con el objetivo de mantener el documento conciso y claro. No obstante, las gráficas para Langevin 3D también se generan en su propio workspace y están completamente disponibles para su consulta.

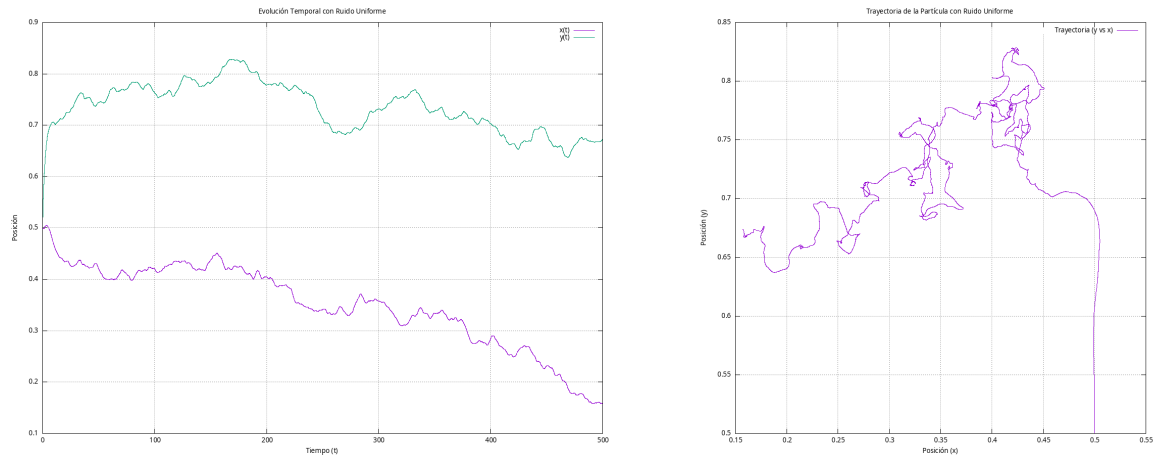


Figure 1: Evolución temporal de las coordenadas  $x(t)$  y  $y(t)$  (izquierda) y trayectoria en el plano  $(x, y)$  (derecha) para una partícula bajo las ecuaciones de Langevin en 2D con ruido uniforme.

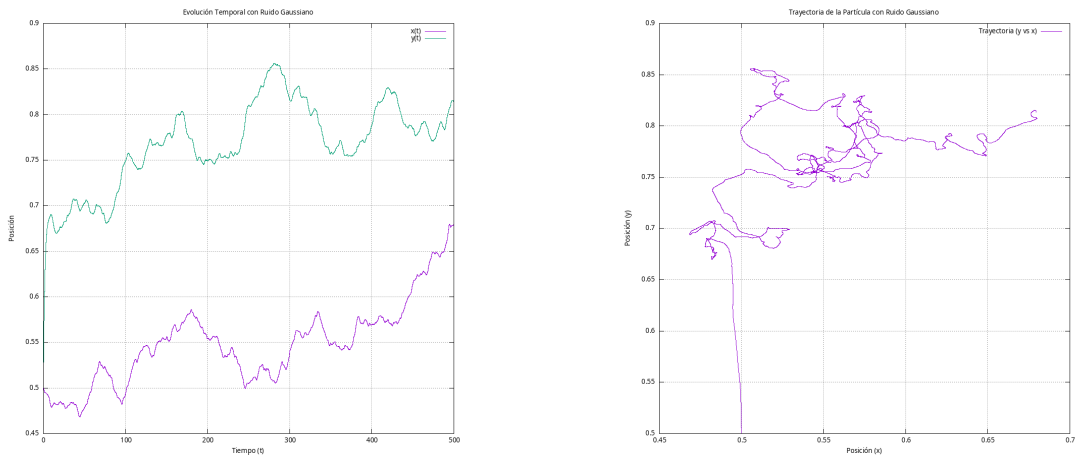


Figure 2: Evolución temporal de las coordenadas  $x(t)$  y  $y(t)$  (izquierda) y trayectoria en el plano  $(x, y)$  (derecha) para una partícula bajo las ecuaciones de Langevin en 2D con ruido gaussiano.

### Diferencias entre ruido uniforme y gaussiano:

Al observar los resultados, se nota que el tipo de ruido afecta de manera importante la trayectoria y el comportamiento de la partícula. El *ruido uniforme* genera fluctuaciones que siempre tienen valores dentro de un rango fijo, lo que hace que el movimiento de la partícula sea más regular y la dispersión más uniforme. En cambio, con *ruido gaussiano*, la mayoría de las fluctuaciones son pequeñas, pero pueden ocurrir saltos grandes ocasionales. Esto provoca trayectorias menos predecibles y una dispersión más amplia. En resumen, el ruido uniforme mantiene el movimiento más controlado, mientras que el ruido gaussiano permite variaciones mayores y más aleatorias en la trayectoria.

## Conclusión

En este trabajo pude estudiar el movimiento estocástico en dos y tres dimensiones usando simulaciones numéricas basadas en las ecuaciones de Langevin. Comparar el ruido uniforme y el gaussiano

mostró claramente cómo el tipo de aleatoriedad afecta la trayectoria y la dispersión de las partículas. Esto demuestra la importancia de elegir bien el modelo de ruido al investigar sistemas físicos u otros fenómenos donde la aleatoriedad juega un papel central. Además, desarrollar y aplicar estas simulaciones me permitió adquirir herramientas útiles para analizar problemas donde no es posible encontrar una solución analítica.