Computational Problems on the Langevin Equation

PK Mohanty (PH4108), 2025

Q7. Anusha— **Q5.** Anunay— , **Q4.** Gautam— , **Q9.** Aishi— , **Q3.** Debanjan— , **Q1.** Krishnabh— , **Q6.** Tanish— , **Q11.** Durgesh— , **Q10.** Pratik— , **Q8.** Vibhor— , **Q2.** Himanshu—

1. **Overdamped Free Diffusion:**The Langevin equationof an overdamped particle is $dx = \sqrt{2D}dW$ or equivalently, $\frac{dx}{dt} = \sqrt{2D}\eta(t)$. Write a program to simulate 1000 independent 1D trajectories of an overdamped particle with $D = 0.5\ \mu m^2/s$ for 10 seconds ($dt = 0.01$ s). From these trajectories, compute and plot the ensemble-averaged Mean-Squared Displacement (MSD) as a function of time. Compare your result with the theoretical expectation $\langle x^2(t)\rangle = 2Dt$ by plotting both on the same graph.

2. **Velocity Autocorrelation Function:** Simulate the underdamped Langevin equation in 1D:

$$m\frac{dv}{dt} = -\gamma v + \sqrt{2\gamma k_B T}\ \eta(t)$$

for a particle with $m = 10^{-12}$ kg, $\gamma = 10^{-8}$ kg/s, at $T = 300$ K. Use $dt = 1 \times 10^{-6}$ s to generate a long trajectory. Calculate the velocity autocorrelation function $\langle v(t)v(0)\rangle$ from your simulation and verify that it decays exponentially as $\frac{k_B T}{m}e^{-\gamma t/m}$.

3. **Diffusion in a Harmonic Potential:** Simulate the motion of an overdamped particle in a 1D harmonic potential $U(x) = \frac{1}{2}kx^2$. The equation is:

$$dx = -\frac{k}{\gamma}x\ dt + \sqrt{2D}\ dW$$

with $k = 1 \times 10^{-6}$ N/m, $\gamma = 1 \times 10^{-8}$ kg/s, $T = 300$ K, for 10^5 steps ($dt = 0.1$ s). Plot the stationary probability distribution $P(x)$ from the histogram of the trajectory and compare it with the theoretical Boltzmann distribution $P(x) \propto \exp(-U(x)/k_B T)$.

4. **Escape Over a Potential Barrier:** Simulate the overdamped dynamics $dx = -\frac{1}{\gamma}U'(x)dt + \sqrt{2D}dW$ in a double-well potential $U(x) = x^4 - 2x^2$. Use parameters $\gamma = 1$, $D = 0.1$, $dt = 0.01$, and initial condition $x_0 = -1$. Run 100 simulations and calculate the average time for the particle to first cross $x = 0$ (the escape time from the left well). Plot a few sample trajectories.

5. **2D Brownian Motion:** Simulate a 2D overdamped Brownian particle with anisotropic diffusion: $D_x = 1.0, D_y = 0.2$ ($\mu m^2/s$). Generate a single long trajectory ($T = 100$ s, $dt = 0.1$ s) starting from the origin. Plot the trajectory and the time-averaged MSD for both x and y directions on the same plot, showing that they follow $2D_x t$ and $2D_y t$ respectively.

6. **Stochastic Resonance:** Simulate the overdamped Langevin equation in a bi-stable potential $U(x) = -\frac{1}{2}x^2 + \frac{1}{4}x^4$ with an added weak periodic force:

$$dx = [-U'(x) + A\sin(\omega t)]dt + \sqrt{2D}dW$$

Use $A = 0.1$, $\omega = 0.1$, and explore different noise intensities $D = [0.01, 0.1, 0.5]$. For each D, run a simulation and compute the power spectral density of $x(t)$. Determine which noise intensity maximizes the signal-to-noise ratio at the driving frequency ω.

7. **Mean First-Passage Time:** For an overdamped particle in a 1D box $x \in [0, L]$ with absorbing boundaries at $x = 0$ and $x = L = 5$, simulate the free diffusion equation ($D = 1$) starting from $x_0 = 2.5$. Run 1000 simulations and record the time when the particle first hits either boundary. Plot the histogram of these first-passage times and compute the average. Compare with the theoretical mean first-passage time for this setup.

8. **Noise-Induced Oscillations:** Simulate the "Brusselator" model, a chemical reaction system with noise:

$$dx = [a - (b + 1)x + x^2 y]dt + \sqrt{2D}dW_1$$
$$dy = [bx - x^2 y]dt + \sqrt{2D}dW_2$$

with $a = 1$, $b = 2.5$, $D = 0.01$, $dt = 0.01$, for 1000 seconds. Plot the phase portrait (x vs y) and the time series of $x(t)$. Demonstrate how noise sustains oscillations that would otherwise die out in the deterministic system.

9. **Non-Markovian Process (Ornstein-Uhlenbeck Noise):** Simulate a particle driven by colored noise. First, generate an Ornstein-Uhlenbeck noise process $z(t)$ with $dz = -\frac{z}{\tau}dt + \sqrt{\frac{2}{\tau}}dW$, with $\tau = 1.0$. Then, use this noise to drive the equation: $dx = z(t)dt$. Use $dt = 0.01$ and simulate for 1000 seconds. Compute and plot the MSD of $x(t)$ and show that it exhibits ballistic motion ($\langle x^2\rangle \propto t^2$) at short times and diffusive motion ($\langle x^2\rangle \propto t$) at long times.

10. **Particle in a Tilted Periodic Potential:** Simulate an overdamped particle in a tilted washboard potential: $U(x) = -U_0\cos(x) - Fx$, with $U_0 = 2$, $F = 0.5$. The dynamics are $dx = -U'(x)dt + \sqrt{2D}dW$. Use $D = 0.5$, $dt = 0.1$, and simulate for 5000 steps. Track the net displacement Δx over time. Run 100 simulations and plot the average velocity $\langle v\rangle = \langle \Delta x\rangle/T$ as a function of the force F (vary F from 0 to 2 in steps of 0.2) for a fixed $D = 0.5$, showing the depinning transition.

11. **Two coupled stochastic oscillators:** Simulate two coupled stochastic oscillators described by the following Langevin equations:

$$\dot{x} = \alpha x - \beta y - \gamma x(x^2 + y^2) + \sqrt{2D}\eta_x(t); \quad \dot{y} = \beta x + \alpha y - \gamma y(x^2 + y^2) + \sqrt{2D}\eta_y(t)$$

where η_x and η_y are independent white noise. Use $\alpha = 0.1$, $\beta = 1.0$, $\gamma = 0.1, D = 0.1$ to the cross-correlation function $\langle x(t)y(t+\tau)\rangle$.

```
###==================A python code for Q2.  with diff mass and dt ====
import numpy as np
import matplotlib.pyplot as plt
#===Given Quantity===
m   = 1e-15
gm = 1e-8
dt = 1e-9
#===Derived Quantity====
tau   = m / gm
dtr   = dt / tau
Ddtr = np.sqrt(2 * dtr)
#========================
Nt    = 10000
nlag = 1000
ENS   = 1000

def Update(x):
    for i in range(1, Nt):
        x[i] = x[i-1] - x[i-1] * dtr + Ddtr * np.random.randn()
    return x

def autocorr(x, nlag):
    acf = np.zeros(nlag)
    for lag in range(nlag):
        acf[lag] = np.mean(x[:Nt-lag] * x[lag:])
    return acf

Vac = np.zeros(nlag)
for _ in range(ENS):
    v_arr = np.zeros(Nt)
    v_arr = Update(v_arr)
    Vac  += autocorr(v_arr, nlag)

Vac /= ENS
t = np.arange(nlag) * dtr

fig = plt.figure(figsize = (6, 5))
plt.plot(t, Vac / Vac[0], ’o’, ms = 4, markevery = 5,
         label = ’Numerical’)
plt.plot(t, np.exp(-t), ’--’, c = ’r’, label = ’Theory’)
plt.legend()
plt.xlabel(’$t$’, size = 20)
plt.ylabel(’$\\langle v(t)v(0)\\rangle$’, size = 20)
plt.tight_layout()
plt.savefig(’Q2_fig.png’, dpi = 320)
plt.show()
###=============================================
```