Non-Equilibrium Stat. Mech.

Project 1
Brownian Dynamics, Fluctuations and Response.

The goal of this project is to implement a Brownian Dynamics (BD) code, and put into test the Fluctuation-Dissipation Theorem in simple, paradigmatic, model random walkers. (I assume some experience with Molecular Dynamics or Monte Carlo simulations).

Consider a set of N non-interacting Brownian particles, located at positions $\mathbf{r}_i(t) = (x_i, y_i)(t)$ in a 2D $L \times L$ box with periodic boundary conditions (PBC), and described by the following set of overdamped Langevin equations:

$$\gamma \dot{\mathbf{r}}_i(t) = \boldsymbol{\eta}_i(t) \tag{1}$$

where γ is the damping coefficient and η is a stochastic process, a Gaussian white noise verifying

$$\langle \eta_i^\alpha(t) \rangle = 0, \, \forall \, i, \, t \, ; \qquad \langle \eta_i^\alpha(t) \eta_j^\beta(t') \rangle = 2\Gamma \delta(t-t') \delta^{\alpha\beta} \delta_{ij} \tag{2}$$

with $\boldsymbol{\eta}_i = (\eta_i^x, \, \eta_i^y)$ and α and β the possible orthogonal dimensions. Thus, Γ denotes the amplitude of the noise term.

1. Show that eq. 1 can be rewritten as

$$\gamma \dot{\mathbf{r}}_i(t) = \sqrt{2\Gamma} \boldsymbol{\xi}_i(t) \tag{3}$$

where now $\boldsymbol{\xi}$ is a Gaussian white noise with zero mean and unit variance.
2. Generate, from a uniform distribution of random numbers (you can une the one in the standard library), a set of $N = 10^4$ random numbers n following a normal distribution $\mathcal{N}(0, 1)$. Plot these values, their distribution $P(n)$ in lin-log and check it has the correct shape. Compute numerically their mean value and their variance.
3. From a computational standpoint, the key aspect of the Langevin equation to take into account carefully, is that the term $\boldsymbol{\xi}$ is not differentiable in any point, it is a random process: how to discretise it then? Since $\boldsymbol{\xi} = dw/dt$, where w is the *Wiener process* (for which we know a representation from stochastic calculus), one can show that the Langevin equation can be discretised as (see, for instance, [1] sec. 15.5.1.)

$$\gamma(x_i(t+\delta t) - x_i(t)) = \sqrt{2\Gamma \delta t} \, g_i^x \tag{4}$$

$$\gamma(y_i(t+\delta t) - y_i(t)) = \sqrt{2\Gamma \delta t} \, g_i^y \tag{5}$$

where g_i^α are independent random numbers picked from the normal distribution $\mathcal{N}(0, 1)$ of zero mean and unit variance. The latter discretization is the analog of the Euler algorithm for the initial value problem but in the presence of noise. It is usually called *Euler-Mayurama algorithm*. We therefore can integrate the equations of motion with the resulting first order iteration method:

$$x_i(t+\delta t) = x_i(t) + \frac{g_i^x}{\gamma}\sqrt{2\Gamma\delta t} \tag{6}$$

$$y_i(t+\delta t) = y_i(t) + \frac{g_i^y}{\gamma}\sqrt{2\Gamma\delta t}\,. \tag{7}$$

$$\tag{8}$$

Now apply this discretisation procedure to simulate a single particle in an open box and initially located at the origin $(0,0)$ with $\gamma = 1$. Plot the trajectory obtained after a time $\tau = 100\delta t$.
4. Implement periodic boundary conditions (PBC) (see [2]) and run simulations of $N = 1000$ particles, in a $L \times L$ box with $L = 100$ and random initial conditions (the initial positions of the particles are picked from a uniform distribution in the $L \times L$ box). Choose the time-step δt appropriately (justify your choice). Plot the initial configuration of the system as well as a configuration after a time $\tau = 100\delta t$.
5. Compute the Mean-square Displacement (MSD) averaged over $N = 1000$ particles:

$$\Delta^2(t) = N^{-1}\sum_{i=1}^{N}\langle(\mathbf{r}_i(t) - \mathbf{r}_i(0))^2\rangle \tag{9}$$

for $\Gamma = 0.1$, 1, 3, 10, 30, ... What is the relationship between Γ and the diffusivity D, defined as

$$D = \lim_{t\to\infty}\frac{\Delta^2(t)}{4t} \tag{10}$$

found from the simulations? [Plot D vs. Γ in log-log scale]

Random generator
The function rand generates a pseudo-random number uniformly distributed between 0 and 1. One first needs to fix the seed of the random generator using the function srand at the very beginning. For instance: 'srand(n)', then 'rand () / double(RAND_MAX)' $\in$ U(0;1). You can also use better random number generators, such as Marsenne Twister.

References

[1] M. E. Tuckerman, *Statistical mechanics: theory and molecular simulation*, Oxford university press, 2010.

[2] M. P. Allen and D. J. Tildesley, *Computer Simulation of Liquids*, Oxford university press. https://global.oup.com/booksites/content/9780198803195/