Arnold Sommerfeld Center for Theoretical Physics
Ludwig-Maximilians-Universität München

Prof. Dr. S. Rulands
A. Hashtroud, Dr. R. Mukhamadiarov

Stochastic simulation of active Brownian motion

Brownian motion underlies many biophysical processes and it is the fundamental building block for the theory of stochastic processes. Its out-of-equilibrium counterpart, active Brownian motion, is a theoretical model for swimming bacteria. Such bacteria convert chemical energy into rotational motion of protein bundles. This rotation gives rise to a kinematic force that propels the bacterium forward. In this practical, we will familiarize ourselves with the concept of stochastic differential equations and then implement stochastic simulations of such active swimmers. By letting many of such bacteria interact with each other we will investigate collective phenomena in groups of active swimmers.

1 Preparation before you start the practical

(a) In order to prepare for the practical, please read Sections 1.1 and 2.1.1 of the introductory material which you can download from the web page of the practical. The remaining sections might be helpful in case you are not familiar with statistics.

(b) On the day of the practical, you will do some coding. You can choose a programming language of your choice, such as Python. The coding we will do is not difficult, but in case you do not have prior experience with programming please familiarize yourself with the basics of Python or Matlab.

(c) You will be most flexible if you use your own laptop for programming. If you cannot bring you own laptop please let us know in advance and we will organize access to a computer.

2 Persistent random walk of a single particle

We will begin by implementing a simple simulation of a single swimmer undergoing active Brownian motion.

(a) Use Python, Matlab, C++, or any language of your choice to implement a bacterial swimmer in two spatial dimensions. To this end, we denote the position of the swimmer by $\vec{x}$, its velocity by v_0 and the angle of its movement by θ. Then, at each time step of the simulation, we update the position of the swimmer according to

$$\vec{x}(t+\Delta t) = \vec{x}(t) + \Delta t\, v_0\, \vec{n}(t), \tag{1}$$

where the vector $\vec{n}$ is a unit vector pointing in the direction of movement of the swimmer. It follows form the angle θ by

$$\vec{n}(t) = \hat{\vec{e}}_1 \cos\theta(t) + \hat{\vec{e}}_2 \sin\theta(t), \tag{2}$$

where $\hat{\vec{e}}_{1,2}$ are the unit vectors. We also update the angle by adding at each time step a random variable $\xi(t)$ to it

$$\theta(t+\Delta t) = \theta(t) + \sqrt{\Delta t \Gamma_0}\xi(t). \tag{3}$$

We assume that this random variable (called noise) is drawn from a Gaussian distribution with zero mean and unit variance. We further assume that this noise is uncorrelated, i.e.

$$\langle \xi(t)\xi(t')\rangle = \delta(t-t'). \tag{4}$$

The parameter Γ_0 is the amplitude of the noise. We recommend that you use the following parameter values: $\theta(0) = 0$, $\vec{x}(0) = 0$, $\Delta t = 0.05$, $\Gamma_0 = 0.1$, and $v_0 = 1$.

Question: Why does the prefactor of the noise term in Eq. (3) scale with the square root of Δt?

(b) Now run the simulation you wrote 100 times and save the results. We would now like to analyse the trajectory of the simulated swimmer. To this end, calculate the average position as a function of time, $\langle \vec{x}(t)\rangle$, and the mean-squared displacement, $\langle \vec{x}(t)^2\rangle - \langle \vec{x}(t)\rangle^2$. The average, $\langle\cdot\rangle$, is to be taken over independent simulation runs.

In a first step we would now like to check whether the simulation you wrote is correct. To this end, compare your results to the analytical predictions that are shown below, and plot both of them together. The analytical findings read

$$\langle \vec{x}(t) \rangle = 0, \tag{5}$$

for the average position and

$$\langle \vec{x}(t)^2 \rangle = 2\frac{v_0^2}{\omega^2}(\omega t - 1 + e^{-\omega t}) = \begin{cases} 2v_0^2\, t^2 & \text{for } \omega t \ll 1, \\ 2v_0^2 t/\omega & \text{for } \omega t \gg 1. \end{cases} \tag{6}$$

where $\omega = v_0^2/\Gamma_0$.
Hint: Use double logarithmic plots.

(c) How would you define the diffusion constant D, using the parameters above? Using definition of the Péclet number, provide your interpretation of the limits in Eq. (6).

3 Brownian dynamics of a collection of interacting particles

Now that you can be confident that your code is working so far, we can add another layer of complexity. To this end, we will now see what happens when we put many active particles together and let them interact.

(a) Assume now that the particles interact via some force $\vec{F}$. How do the equations of motion Eq. (1) and Eq. (3) change?

(b) If two particles are closer than a distance $2R$, these particles should align their direction of movement. We further assume that interactions are harmonic, such that the force on particle i that comes from particle j with positions $\vec{x}_i$ and $\vec{x}_j$ reads

$$\vec{F}_{ij} = \begin{cases} \hat{r}_{ij} k(2R - |\vec{x}_i - \vec{x}_j|) & \text{for } |\vec{x}_i - \vec{x}_j| < 2R \\ 0 & \text{for } |\vec{x}_i - \vec{x}_j| \geq 2R \end{cases} . \tag{7}$$

k is the repulsion strength coefficient, $\hat{\vec{r}}_{ij} = (\vec{x}_i - \vec{x}_j)/|\vec{x}_i - \vec{x}_j|$ is a unit vector pointing from $\vec{x}_2$ to $\vec{x}_2$, and $|\vec{x}_i - \vec{x}_j|$ is the Eucledian distance between the two vectors. To implement your simulation, choose a radius R, and fill your simulation box so that the overall density is about 10%. Use periodic boundary conditions.

(c) Run simulations for different particle densities and compute the mean squared displacement by averaging over different particle trajectories.
Question: How do you interpret your observations?