Recreation of Numerical Simulations of a Brownian Particle in an Optical Trap

Marc Rodriguez Salazar

Introduction

The objective of this task is to recreate the numerical experiments presented in the paper *Simulation of a Brownian Particle in an Optical Trap* (`https://doi.org/10.1119/1.4772632`).

The paper explores different scenarios of a Brownian particle confined in an optical trap. A Brownian particle is a microscopic particle suspended in a medium (liquid or gas), whose movement is subject to random collisions with the surrounding molecules, which cause it to move in a stochastic manner. An optical trap is a high-focused laser beam that holds particles due to momentum transfer from the light to the particle.

The Langevin equation is a stochastic differential equation that describes the evolution of a system under the influence of deterministic and random forces. In this case, the deterministic force is the optical force of the optical trap, and the random force is due to Brownian motion. The most general Langevin equation of a Brownian particle in an optical trap in one dimension is:

$$\underbrace{m\ddot{x}(t)}_{\text{Inertia}} = \underbrace{-\gamma\dot{x}(t)}_{\text{Friction}} + \underbrace{kx(t)}_{\text{Restoring force}} + \underbrace{\sqrt{2k_BT\gamma}W(t)}_{\text{white noise}} \tag{1}$$

Solving this equation is difficult due to the white noise term, which is non-smooth, discontinuous and has infinite variation. These characteristics make it difficult to treat analytically. Therefore, to solve equation 1, the finite difference algorithm is used.

The finite difference algorithm is a simple numerical technique that approximates derivatives with finite differences. The time domain is discretized in Δt time steps. Then, if Δt is sufficiently small, $x_i \approx x(t_i)$, and we can define the derivatives as:

$$\dot{x}(t) \approx \frac{x_i - x_{i-1}}{\Delta t} \tag{2}$$

$$\ddot{x}(t) \approx \frac{x_i - 2x_{i-1} + x_{i-2}}{\Delta t^2} \tag{3}$$

All terms in equation 1 can be approximated using the finite difference method, except for the white noise term $W(t)$. To approximate $W(t)$, a discrete sequence $W_i = w_i/\sqrt{\Delta t}$ is used, where w_i are uncorrelated Gaussian numbers with zero mean and unit variance, resulting in W_i with 0 mean and $1/\Delta t$ variance.

To recreate the numerical simulations of the paper, I have worked with Jupyter Notebook (Python 3), because it allows quick image visualization and real-time adjustments. All code can be found at my GitHub repository: `https://github.com/marcrodriguez2001/Simulation-of-a-Brownian-particle-in-an-optical-trap`.

There is a notebook for each section. The NumPy and Matplotlib libraries have been used.

Numerical Experiments and Results

0.1 Simulation of White Noise

White noise can be simulated using the simplest form of the free diffusion equation:

$$\dot{x}(t) = W(t) \tag{4}$$

Then, the random walk is numerically implemented using the finite difference algorithm and the definition of $W(t)$:

$$x_i = x_{i-1} + \sqrt{\Delta t} w_i \tag{5}$$

To simulate particle trajectories with white noise, we run the simulation up to time $T = 30$, using different time steps $\Delta t = 1.0, 0.5$, and 0.1 to analyze the effect of discretization. For each time step Δt, the number of points is calculated as $N = T/\Delta t$, and the corresponding time vector t is generated using a linear spacing. A sequence of N independent and identically distributed Gaussian random variables $w_i \sim \mathcal{N}(0, 1)$ is generated using the NumPy random normal function, and it is used to represent the white noise.

The trajectory x_i is initialized at zero and computed iteratively in a loop using the finite difference approximation of random walk equation 5. 10,000 trajectories were also simulated to calculate the mean and standard deviation.

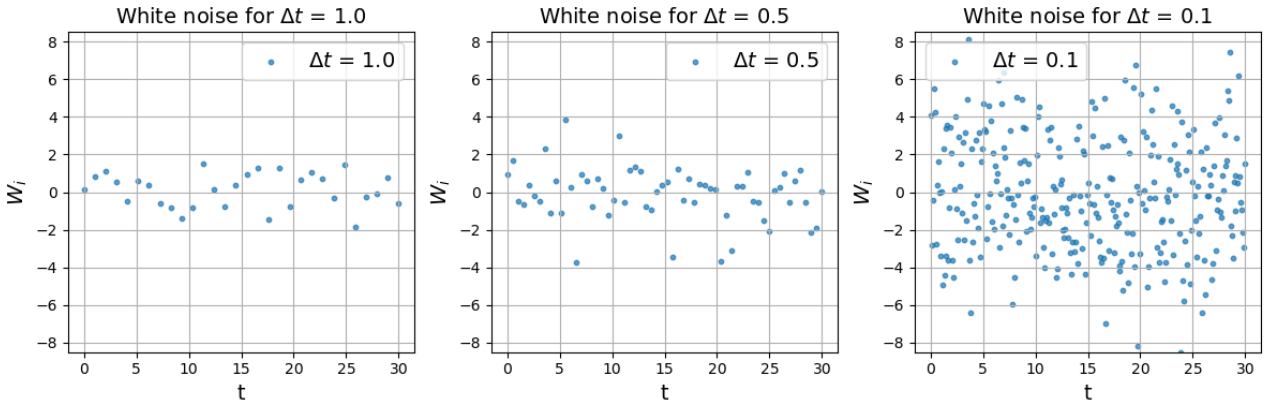


Figure 1: Scatter plot of white noise W_i for $\Delta t = 1.0$ (left), $\Delta t = 0.5$ (center), and $\Delta t = 0.1$ (right)

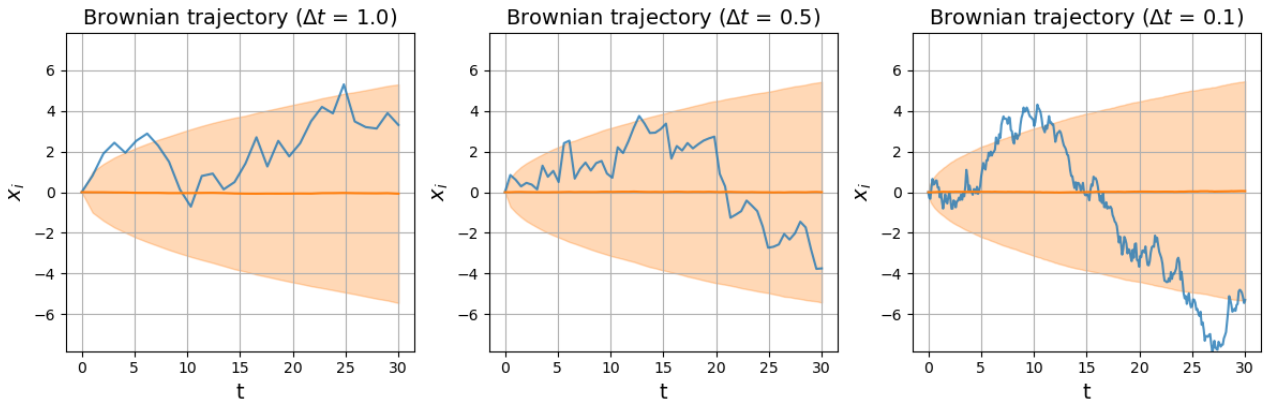


Figure 2: Brownian trajectory as a function of time for $\Delta t = 1.0$ (left), $\Delta t = 0.5$ (center), and $\Delta t = 0.1$ (right). The blue line is the Brownian trajectory, the orange line is the mean and the shaded area is the variance.

Figure 1 shows three scatter plots of W_i for each Δt. The plots illustrate that the spread of W_i values increases when Δt decreases due to discretization. Figure 2 shows a Brownian trajectory for each Δt, along with the mean and variance computed from the 10,000 trajectories. The blue lines correspond to the individual Brownian trajectories, which become more jagged as Δt decreases. The orange lines represent the mean of the simulations, which

remains at zero due to the properties of white noise. Finally, shaded area represents the variance. Each Brownian trajectory behaves similarly but differs from the others because each trajectory is a random process. However, the statistical properties, the mean and the variance of the 10,000 trajectories, are consistent for different values of Δt.

0.2 From Ballistic Motion to Brownian Diffusion

To simulate Brownian motion of real particles, diffusion must be considered. The Langevin equation for this scenario is:

$$m\ddot{x}(t) = -\gamma\dot{x}(t) + \sqrt{2k_BT\gamma}W(t) \tag{6}$$

Dropping the inertial term (the term with mass) from equation 6, we obtain the following equation:

$$\dot{x}(t) = \sqrt{2D}W(t) \tag{7}$$

Then, using the finite difference algorithm and the definition of $W(t)$, the solution of both equations 6 and 7 can be obtained:

- **Inertial regime:** solution of equation 6, corresponds to ballistic motion.

$$x_i = \frac{(2+\Delta t\gamma/m)}{(1+\Delta t\gamma/m)}x_{i-1} - \frac{1}{(1+\Delta t\gamma/m)}x_{i-2} + \frac{\sqrt{2k_BT\gamma}}{m(1+\Delta t\gamma/m)}\Delta t^{3/2}w_i \tag{8}$$

- **Non-inertial regime:** solution of equation 7, corresponds to Brownian diffusion behavior.

$$x_i = x_{i-1} + \sqrt{2D\Delta t}w_i \tag{9}$$

The momentum relaxation time τ is the time scale of the transition between both regimes.

A silica microparticle in water with radius $R = 1$ μm, mass $m = 11$ pg, viscosity $\eta = 0.001$ Ns/m^2, friction coefficient $\gamma = 6\pi\eta R$, temperature $T = 300$ K, diffusion coefficient $D = (K_BT)/\gamma$ and $\tau = m/\gamma = 0.6$ μs is simulated using a time step $\Delta t = 10$ ns.

The trajectory x_i is initialized at zero and computed iteratively in a loop for both inertial motion (equation 8) and non-inertial motion (equation 9). Two simulations were performed, one with a total duration equal to τ, and another with a duration of 100τ. In both simulations, the dynamics were studied in both the inertial and non-inertial regimes.

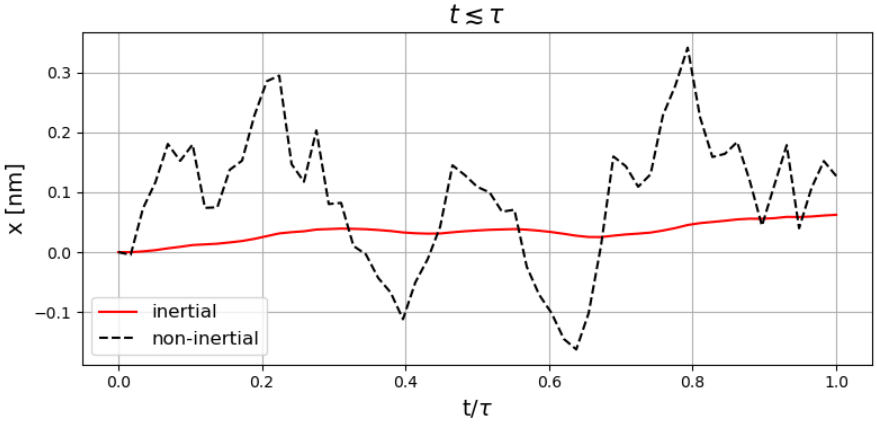


Figure 3: Brownian trajectories for short times.

Figure 3 represents Brownian trajectories for times smaller than or comparable to τ. The inertial motion (red line) is smoother than the non-inertial (black dashed line). This is because, for times shorter than or comparable to τ, the inertial regime maintains ballistic behavior.

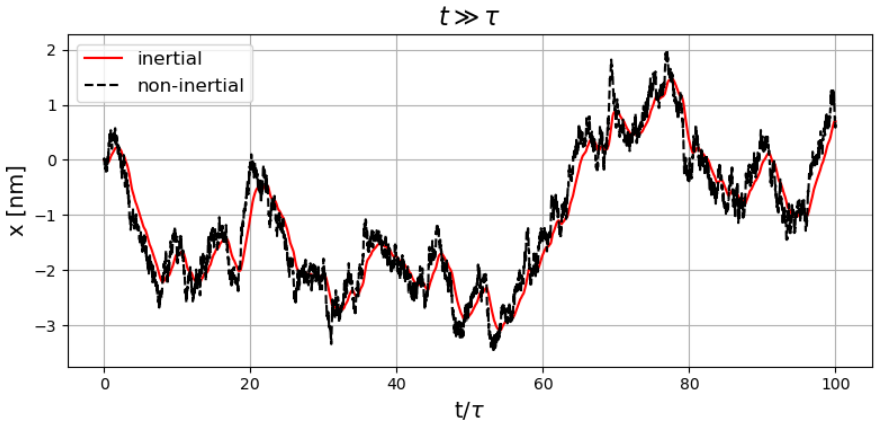


Figure 4: Brownian trajectories for long times.

Figure 4 represents Brownian trajectories for times larger than τ. Both regimes, inertial and non-inertial, are jagged. This is because, for times $t \gg \tau$, the system loses memory of its initial velocity, and even in the inertial case, the motion ceases to be ballistic and becomes diffusive.

To better understand the differences between the inertial and non-inertial regimes, the velocity autocorrelation function and the mean square displacement of the particle position were calculated.

The velocity autocorrelation function can be implemented as:

$$C_{v,n} = \overline{v_{i+n} v_i} = \sum_{i=0}^{N-n-1} v_i v_{i+n} \tag{10}$$

And the mean square displacement:

$$\langle x_n^2 \rangle = \overline{[x_{i+n} - x_i]^2} = \sum_{i=0}^{N-n-1} (x_{i+n} - x_i)^2 \tag{11}$$

Both equations 10 and 11 are implemented through functions that perform the calculations shown and normalize the results. In the case of the autocorrelation function, the velocity is computed as $v = (x_{i+1} - x_i)/\Delta t$.

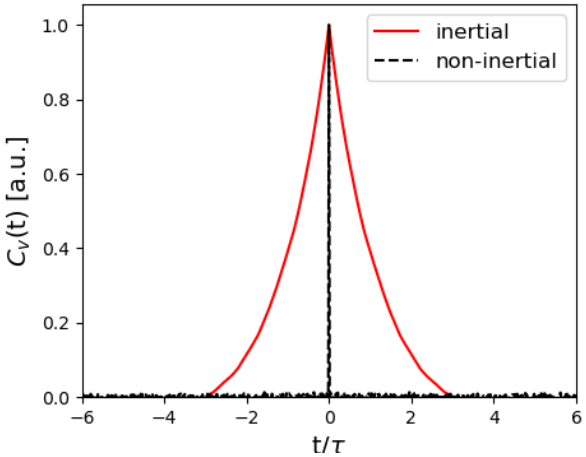


Figure 5: The velocity autocorrelation function.

In Figure 5 we can see the velocity autocorrelation function. The red line corresponds to inertial motion, while the black dashed line represents non-inertial motion. In the non-inertial regime (diffusive), the velocity decays to zero immediately. However, for the inertial (ballistic) regime, this process takes some time. This illustrates how the system loses memory of its initial velocity, even in the inertial motion.

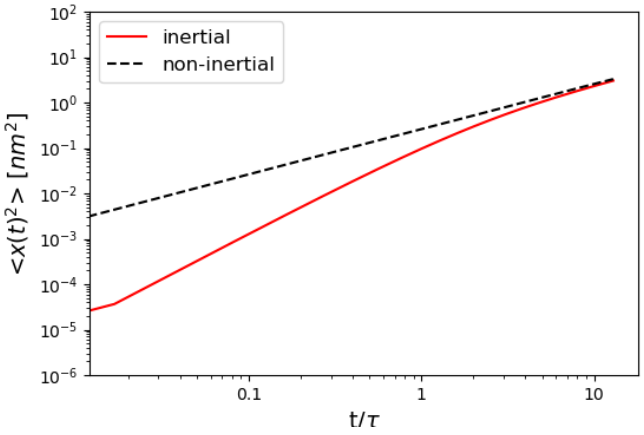


Figure 6: Log-log plot of the mean square displacement.

In Figure 6 we can see the log-log plot of the mean square displacement. For inertial motion, the mean square displacement is initially quadratic, but later becomes linear. For non-inertial motion, it is always linear. When both lines converge, it represents the transition from ballistic to diffusive motion, and it can be seen that this occurs for times $t \gg \tau$.

0.3 Optical Traps

To simulate a Brownian particle in an Optical Trap, we need to consider equation 1 without the inertial term. The solution of this equation is:

$$\vec{r_i} = \vec{r_{i-1}} - \frac{1}{\gamma}\vec{k} \cdot \vec{r_{i-1}}\Delta t + \sqrt{2D\Delta t}\vec{w_i} \tag{12}$$

$\vec{r_i}$ is a vector of (x_i, y_i, z_i), so we need to compute an iterative loop for each direction. $\vec{w_i}$ is a vector of normally distributed Gaussian random numbers. And $\vec{k_i}$ is the stiffnesses of the trap $(k_x, k_y, k_z) = (1, 1, 0.2)$fN/nm.

The simulated particle is the same of the previous section, with identical values for radius, temperature, diffusion coefficient, and so on, but with the new values of the stiffnesses of the optical trap, a time step $\Delta t = 0.001s$, and number of points N.

In Figure 7 can be seen a trajectory of a Brownian particle in the optical trap with stiffnesses $(k_x, k_y, k_z) = (1, 1, 0.2)$ fN/nm. The particle explores an ellipsoidal volume around the center of the trap, where the surface represents an equiprobable region.

The probability distribution represents the probability density of finding the particle at a specific position in the plane. To compute the probability distribution in a plane, the general form of the 2D Gaussian distribution is used:

$$P_{xy} = \frac{1}{2\pi r_x r_y} \exp\left(-\frac{1}{2}\left(\frac{X^2}{r_x^2} + \frac{Y^2}{r_y^2}\right)\right) \tag{13}$$

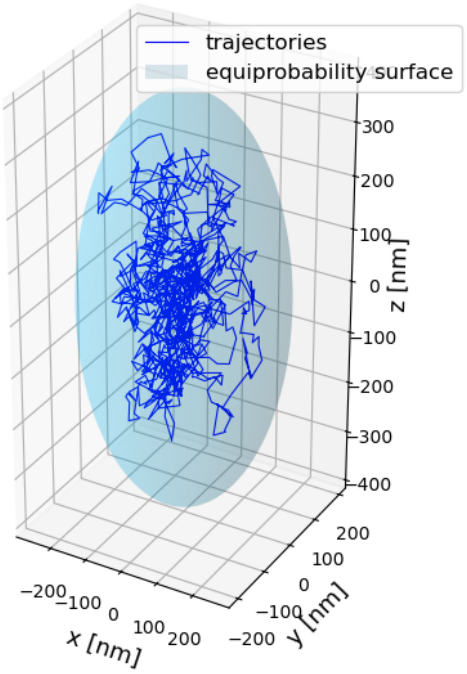
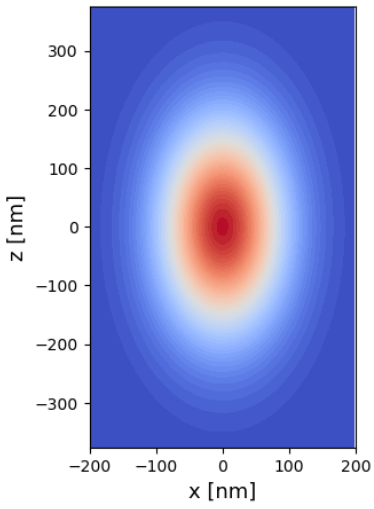
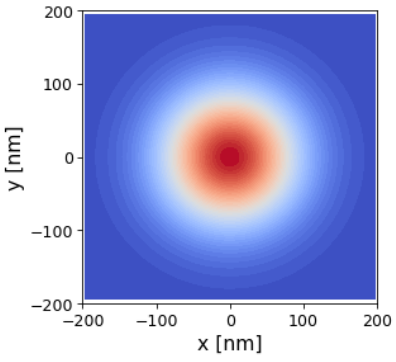


Figure 7: Trajectory of a Brownian particle in an optical trap.

Figure 8a and Figure 8b show the probability distributions of finding the particle in the xz and xy planes.



(a) Probability distribution of finding the particle in the xz plane



(b) Probability distribution of finding the particle in the xy plane

Figure 8

Let's study the effect of the stiffness value k_{xy}. To do this, 12 simulations were performed, each with a different value of $k_{xy} = 0.1, 0.2, 0.5, 1, 2, 3, 4, 5, 6, 7, 8$ and 9 fN/nm.

Theoretically, it is known that $\sigma_{xy}^2 \propto 1/k_{xy}$. It is also computed σ_{xy}^2 as a function of k_{xy} to verify that the theoretical relationship aligns with the results obtained from the simulations.

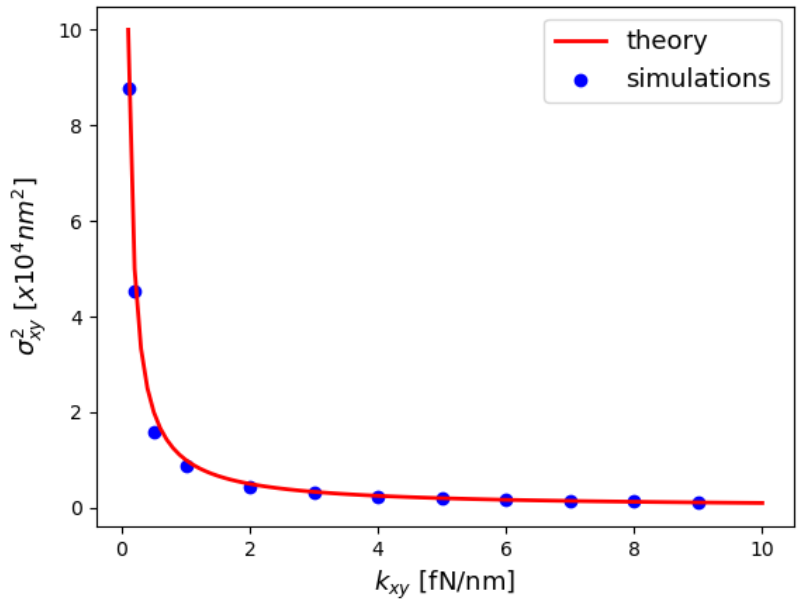


Figure 9: Variance as function of stiffness. Red line corresponds to the theoretical relation. Blue points correspond to the simulations performed.

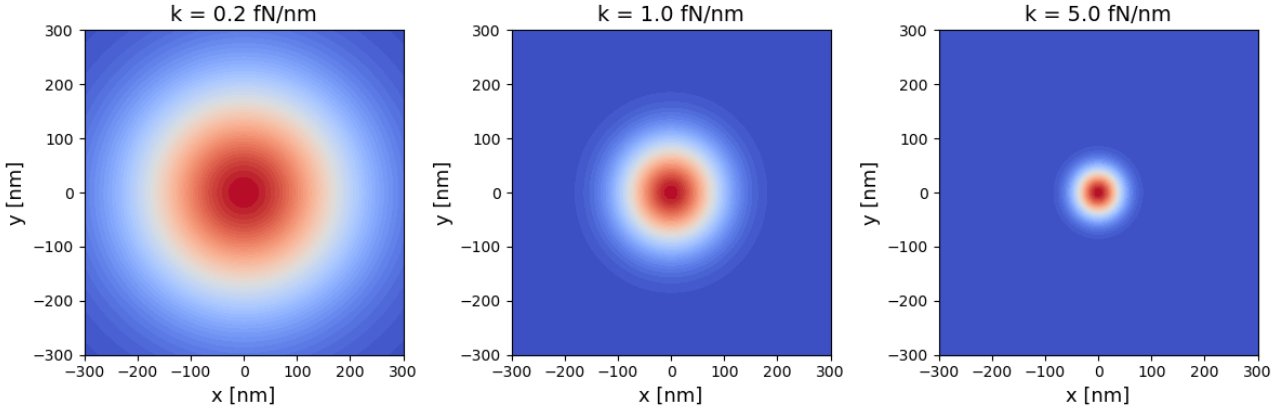


Figure 10: Probability distribution of finding the particle in the xy plane for different values of k_{xy}

Figure 9 shows that the computed simulations align well with the theoretical relationship. It can be seen that, as the trap stiffness k_{xy} increases, the particles become more confined and the variance of their positions decreases. This behavior is also observed in Figure 10, where the probability distribution functions become more confined around the center of the trap when k_{xy} increases.

Finally, the autocorrelation function of the position and the mean square displacement are calculated. Both are calculated as explained in the previous section, the only difference is that the autocorrelation function is now for position and not for velocity as shown in equation 14.

$$C_{x,n} = \overline{x_{i+n} x_i} = \sum_{i=0}^{N-n-1} x_i x_{i+n} \tag{14}$$

In Figure 11 can be seen that as the trap stiffness increases, which means that restoring force also increases, the decay time of the position decreases. This occurs because the particle has a smaller phase-space to explore.

In Figure 12 can be seen that independently of the value of k_{xy}, the system reaches a plateau. Unlike in the previous section, where the mean square displacement grew without limit, the addition of an optical trap causes it to reach a plateau due to the confinement imposed by the trap.

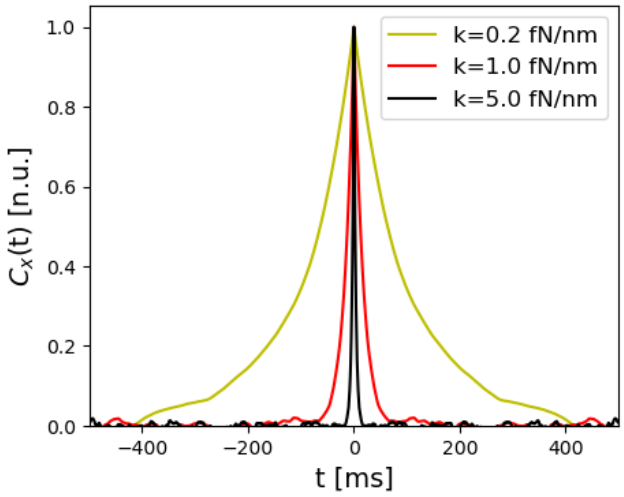


Figure 11: The position autocorrelation function for different values of k_{xy}

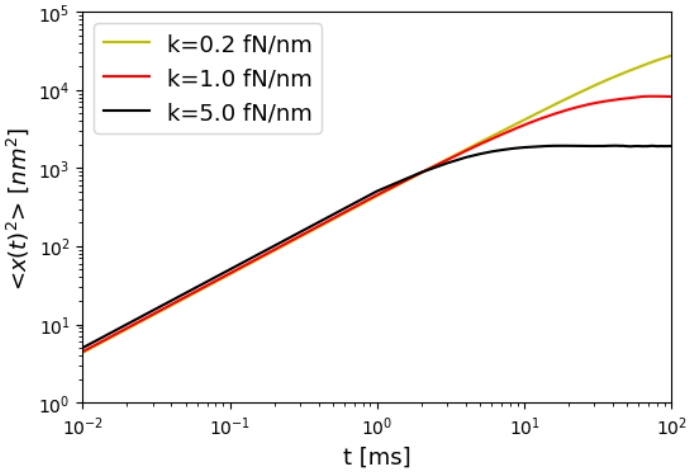


Figure 12: Log-log plot of the mean square displacement for different values of k_{xy}.

0.4 Further Numerical Experiments

Other phenomena can also be studied using this approach. For a general force $F(x(t), t)$ the Langevin equation is:

$$\dot{x}(t) = \frac{1}{\gamma} F(x(t), t) + \sqrt{2D} W(t) \tag{15}$$

Let's study three different cases with different forces acting on the particle. In all cases the values of mass, temperature, and other coefficients are the shown in previous sections.

0.4.1 Constant force F_c

Adding a constant force $F_c = 200$ fN, the total force is $F(x(t), t) = -kx(t) + F_c\ h(t)$, where $h(t)$ is the Heaviside step function. So, the resulting equation of motion is:

$$x_i = x_{i-1} - \frac{1}{\gamma} \Delta t\ k_x\ x_{i-1} + \frac{1}{\gamma} F_c\ h(t)\ \Delta t + \sqrt{2D\Delta t}\ w_{x\ i} \tag{16}$$

Iterating in a loop, we obtain the trajectory of the particle under the influence of both, the optical trap and the constant external force. The constant force causes a displacement of the particle's equilibrium position, shifting it from the center of the trap, as can be seen in Figure 13. As shown, this displacement Δx can be calculated.

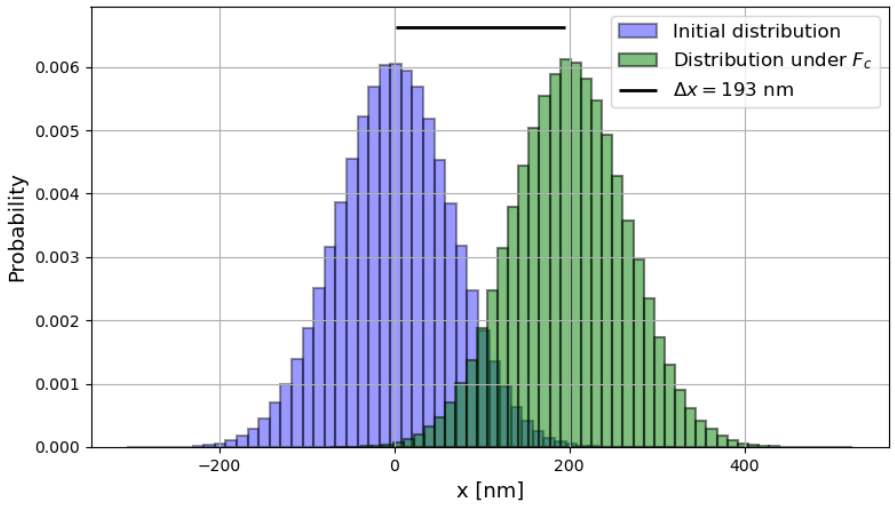


Figure 13: Probability distribution of an optically trapped particle shifts in response to an external force.

0.4.2 Rotational force

Now, a simple rotational force field is applied:

$$\vec{F}(x,y) = -\begin{pmatrix} k & \gamma\Omega \\ -\gamma\Omega & k \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} \tag{17}$$

Obtaining the following equations of motion:

$$x_i = x_{i-1} - \frac{1}{\gamma}k_x\, x_{i-1}\ \Delta t - \Omega\, y_{i-1}\Delta t + \sqrt{2D\Delta t}\, w_{x\,i} \tag{18}$$

$$y_i = y_{i-1} - \frac{1}{\gamma}k_y\, y_{i-1}\ \Delta t + \Omega\, x_{i-1}\Delta t + \sqrt{2D\Delta t}\, w_{y\,i} \tag{19}$$

By implementing both equations 18 and 19 in the same loop, we can obtain the trajectories of $x(t)$ and $y(t)$ using $\Omega = 132.6\ \mathrm{s}^{-1}$. The autocorrelation function C_x is calculated, as well as the position cross-correlation C_{xy}. Figure 14 shows C_x and C_{xy} as function of time.

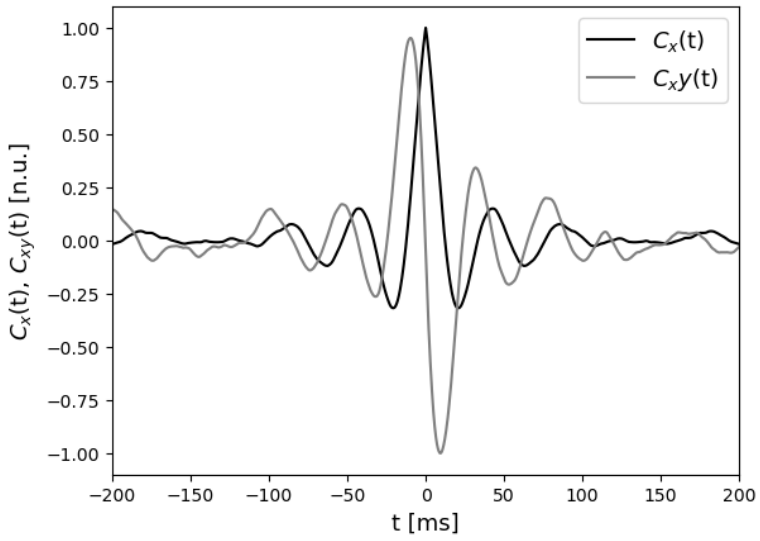


Figure 14: The position autocorrelation and cross-correlation functions of the trapped particle.

0.4.3 Double-well potential

Finally, we study the statistical properties of a particle in a double-well potential $U(x) = ax^4/4 - bx^2/2$ which produces a force $F(x) = -ax^3 + bx$. Obtaining the following equation of motion:

$$x_i = x_{i-1} + \frac{1}{\gamma}\Delta t \left(-a\, x_{i-1}^3 + b\, x_{i-1}\right) + \sqrt{2D\Delta t}\, w_{x\,i} \tag{20}$$

The trajectory is simulated using the values $a = 1.0 \times 10^7 \mathrm{N/m}^3$ and $b = 1.0 \times 10^{-6} \mathrm{N/m}^3$. In Figure 15 we can see how the trajectory transitions between the two equilibrium positions of the double-well potential, located approximately at 300 and -300 nm.

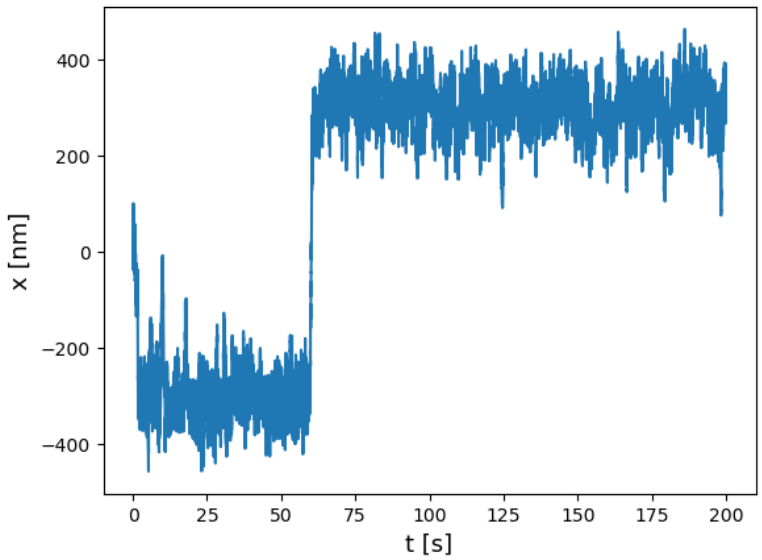


Figure 15: Brownian trajectory of a particle in a double-well potential.

Conclusions

The replication of the original paper has been successfully achieved, demonstrating that with very basic implementations, it is possible to simulate Brownian trajectories driven by randomness, as well as their confinement in optical traps. These results contribute to a deeper understanding of the underlying physical phenomena and showcase the effectiveness of simple models in describing complex systems.