

Programming Project 1: Active Brownian Particles

Felix Reiser

Abstract

In this project we want to implement a model to simulate active brownian particles and study the behavior of the system. For this we first discuss aspects of the cell list algorithm. Then study active brownian particles with and without interactions. Last we want to discuss the collective behavior of active brownian particles at high density.

Contents

1 Cell List	2
2 Langevin Equation	6
3 Interactions	11
4 MIPS	13
5 Conclusion	14

1 Cell List

a)

For the given coordinates $x = [0, 1, 1, 2]$ and $y = [0, 1, 0, 1]$. The naive algorithm returns the pairs

$$\{[0, 1], [0, 2], [1, 2], [1, 3], [2, 3]\}.$$

Answer to b) and e)

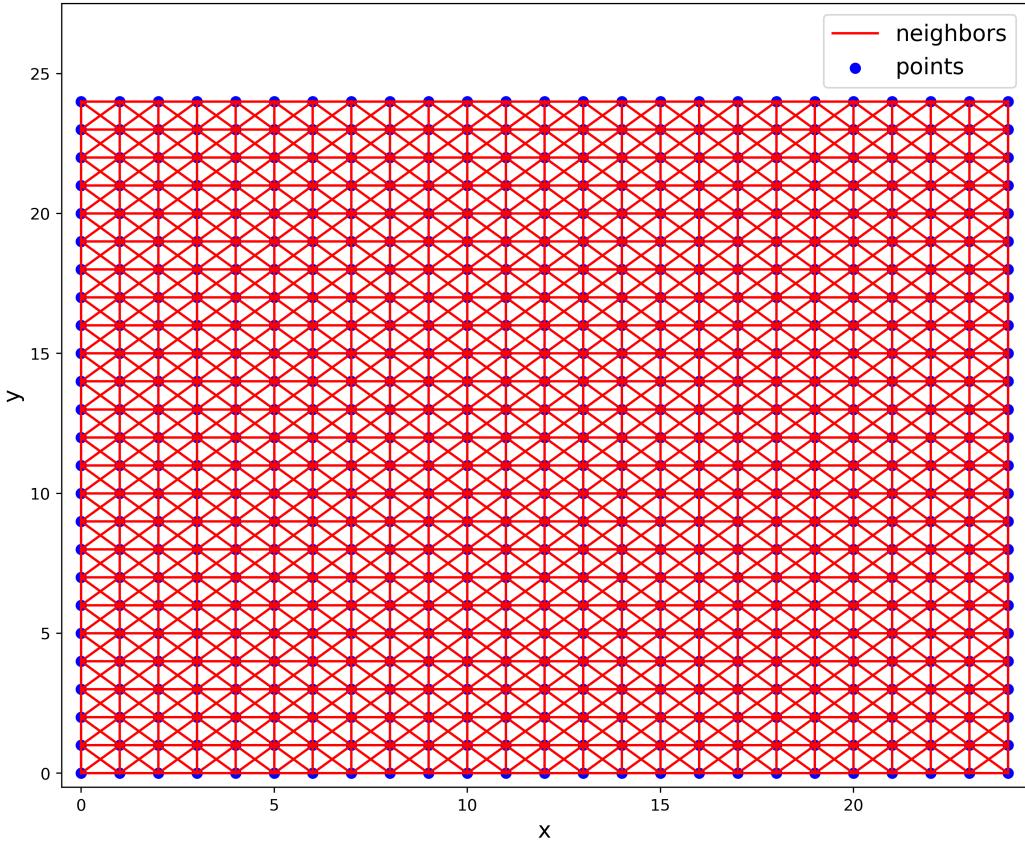


Figure 1: 25x25 grid with all connections with cut off distance $r_c = 1.5$

In Fig.1 all the connections for the cut-off distance $r_c = 1.5$ are visualized. For the expected amount of connections we simply count the connections. So starting bottom left corner the point has 3 connections from there moving to the right the next 23 points have 4 new connections we count (because we don't want to double count). Now on the right boundary the point has 2 new connections. Now we move up one row and start at the left point again. We get the same $3 + 4 \cdot 23 + 2$ connections as before. This works for the first 24 rows until we get to the upper boundary here we have just 1 new connection per point except for the upper right corner this point has no new connections. So for the upper row we simply get 24 new connections. With this we would expect there to be $(3 + 4 \cdot 23 + 2) \cdot 24 + 24 = 2352$ connections. This is exactly the

number of pairs my program for the naive neighbor algorithm returns. In Fig.2 we see the 25×25 grid and all the connections with a distances smaller than the cut-off distance $r_c = 1.3$. We see that compared to Fig.1 the difference is that all diagonal connections are not there, but rest of the connections are the same. Now we can count in a similar way the connections for $r_c = 1.3$ here we just have to be aware that there are no diagonal connections, because the diagonal points are further away than the cut off distance $1.3 < \sqrt{2}$. This leaves us with $2 \cdot 24 + 1$ connections for the first 24 rows and again 24 connections for the upper boundary. So for $r_c = 1.3$ we expect $(2 \cdot 24 + 1) \cdot 24 + 24 = 1200$ pairs. Running the cell list algorithm on this grid and with the two different cut of distances produces the same results. For $r_c = 1.5$ we find $n = 2352$ pairs and for $r_c = 1.3$ we find $n = 1200$ pairs. So the algorithms work in the expected way.

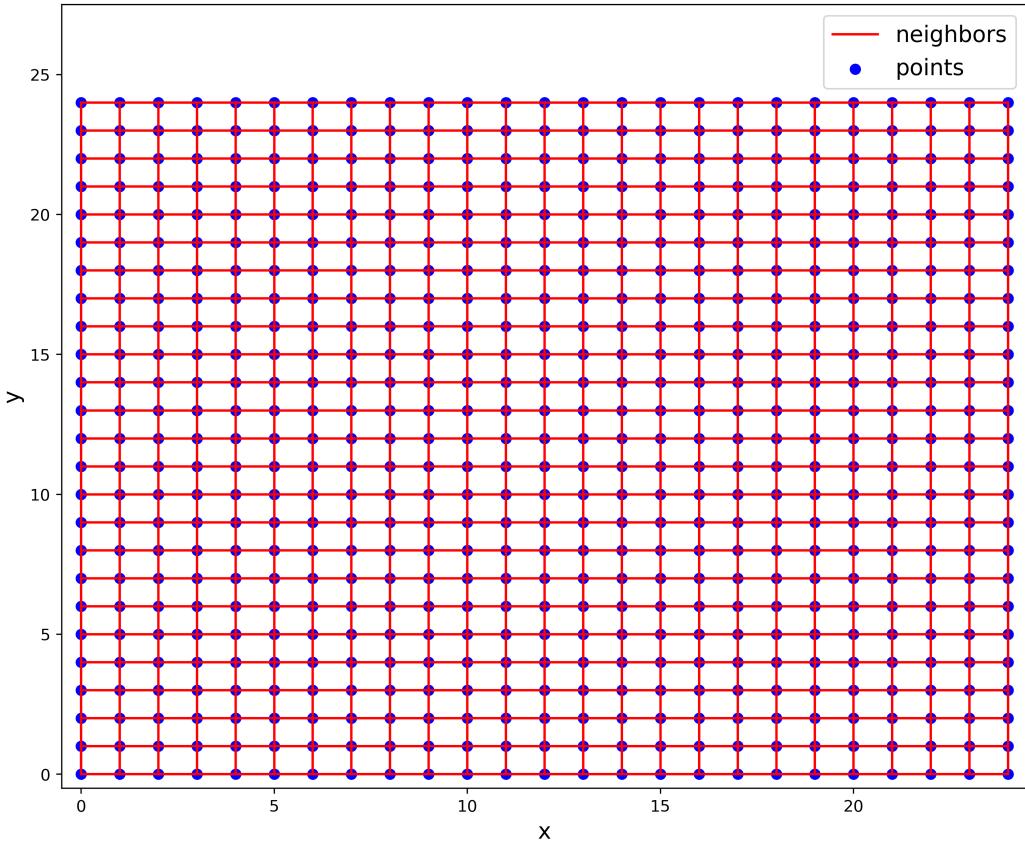


Figure 2: 25×25 grid with all connections with cut off distance $r_c = 1.3$

Answer to c) and e):

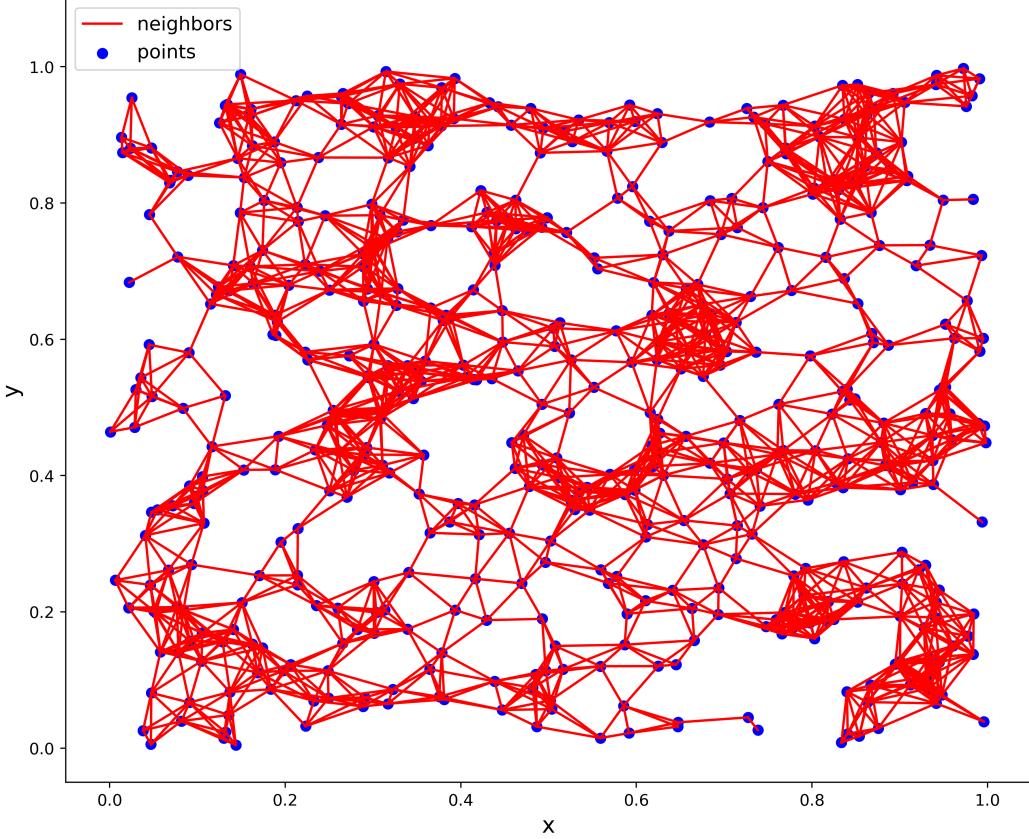


Figure 3: 500 uniformly distributed random points in the units square and the connections to neighboring points that are closer than $r_c = 0.08$.

In Fig.3 we see 500 randomly distributed points in the unit square and visualized all the connections to neighbors, points with a distance less than $r_c = 0.08$, by a red line. Here we found the neighbors with the naive algorithm and found a total of $n = 2376$ pairs. Finding the pairs on the same distribution of points with the cell list algorithm also finds $n = 2376$ pairs. So as intended the two algorithms return the same number of pairs and I have also checked that they return the same pairs not just same number of pairs.

Answer for 1 f):

In Fig.4 we observe that for small system sizes $N < 10^3$ the runtime of the two algorithms is similar and the naive is even a bit quicker for small N . For larger systems $10^3 < N \leq 10^5$ we see that the runtime of the cell list algorithm is lower than the naive algorithm. As visualized by the fitting functions plotted in Fig.4 the cell list algorithm scales linearly with system size $t \propto N$ and the naive algorithm quadratic $t \propto N^2$. The cell list algorithm has more computational overhead than the simple naive algorithm thus explaining the small N regime, where the naive algorithm is quicker. But for the larger systems the better scaling of the cell list algorithm takes over and leads to the comparatively quicker runtimes.

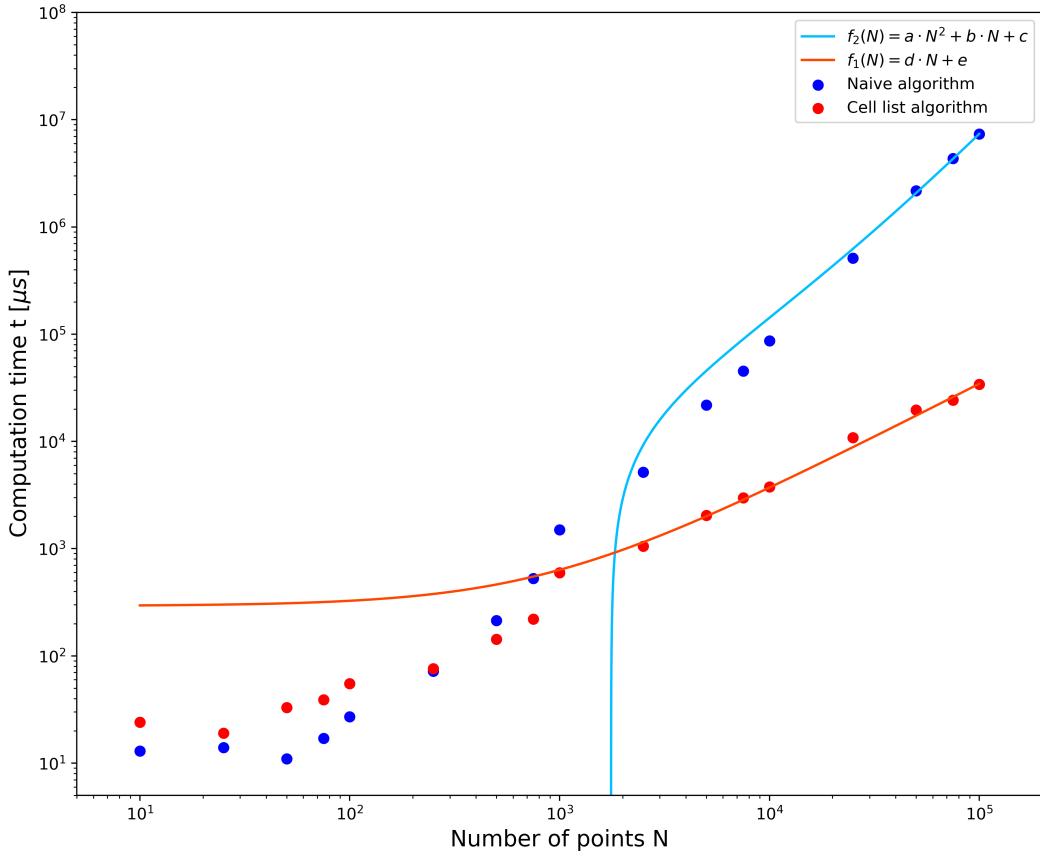


Figure 4: Runtime comparison between naive and cell list algorithm for different number of points (system size) N . The values for the runtime of the naive algorithm are fitted with a quadratic function of the form $f(N) = a \cdot N^2 + b \cdot N + c$ with the parameters $a = 6.42 \cdot 10^{-4}$, $b = 9.69$ and $c = -1.90 \cdot 10^4$. For the cell list algorithm the values were fitted by a linear function $f(N) = d \cdot N + e$ with $d = 0.34$ and $e = 292.11$

2 Langevin Equation

We describe the change in position \mathbf{r} and orientation ϕ of a single self-propelled Brownian particle with Peclet number Pe using the following Langevin equations:

$$\dot{\mathbf{r}} = Pe\mathbf{e} + \sqrt{2D}\xi \quad \mathbf{e} = \begin{pmatrix} \cos \varphi \\ \sin \varphi \end{pmatrix}$$

$$\dot{\phi} = q\sqrt{2D_r}\chi$$

The geometry factor $q \approx 1.543$ arises from Stokesian drag for spherical particles, D is the diffusion coefficient and D_r the radial diffusion coefficient, which we will from now on set to $D = D_r = 1$.

a) Explain the form of the equations and discuss the relevant Reynolds-number regime.

Answer: The Peclet number describes the speed of the particles self-propulsion and the unit vector \mathbf{e} the direction. φ is the angle between \mathbf{e} and the x -axis. ξ is a 2D vector where the x and y component are independent gaussian random variables with mean zero (and standard deviation 1 or depending on Diffusion coefficient). χ is also a independent gaussian random variable with mean zero. These random variable describe the white noise of the brownian motion. We are in the Low Reynolds number regime the dynamics are overdamped.

b) Write a program that integrates equations of motion in time for N non-interacting particles in a squared simulation box with side length L with periodic boundary conditions. Use the Euler-Maruyama scheme to discretize the equations. Build your program in a way that allows you to extract particle trajectories for further analysis.

Answer: Discretizing the equation:

$$\dot{\mathbf{r}} = Pe\mathbf{e} + \sqrt{2}\xi \Rightarrow \frac{\mathbf{r}_{k+1} - \mathbf{r}_k}{\Delta t} = Pe\mathbf{e}_k + \sqrt{2}\frac{\xi_k}{\sqrt{\Delta t}}$$

$$\mathbf{r}_{k+1} = \mathbf{r}_k + Pe\mathbf{e}_k\Delta t + \sqrt{2\Delta t}\xi_k$$

Writing this explicitly for x, y :

$$x_{k+1} = x_k + Pe \cos \varphi_k \Delta t + \sqrt{2\Delta t} \xi_{x,k}$$

$$y_{k+1} = y_k + Pe \sin \varphi_k \Delta t + \sqrt{2\Delta t} \xi_{y,k}$$

For φ

$$\dot{\varphi} = q\sqrt{2}\chi \Rightarrow \frac{\varphi_{k+1} - \varphi_k}{\Delta t} = q\sqrt{2} \frac{\chi_k}{\sqrt{\Delta t}}$$

$$\varphi_{k+1} = \varphi_k + q\sqrt{2\Delta t}\chi_k$$

With that we have our three discretized equations and can easily numerically integrate them now.

c) Choose appropriate initial conditions and find a suitable time step. An even spacing between the initial positions will help you later when there are potentials between the particles.

Answer: As starting condition we arrange the particles $N = 1000$ in a square grid of length $L = 85\sigma$ with uniform spacing and for the timestep we use $\Delta t = 10^{-4}\tau$ and end the simulation at $t = 100\tau$. This timestep is good enough for the simulation without interactions, but for the simulations with interactions we will need to use $\Delta t = 10^{-5}$ to not have particles jump around.

e)

In Fig.5 five example trajectories of the total 1000 particles simulation with $Pe = 0$ are shown and in Fig.6 five example trajectories with $Pe = 20$ are shown. As one would expect the particles with $Pe = 0$ perform a random walk and compared to the particles with $Pe = 20$ they do not move far from their initial position.

The particles in Fig.6 with $Pe = 20$ their trajectories reach regions all over the given domain and have a bigger displacement from their initial position.

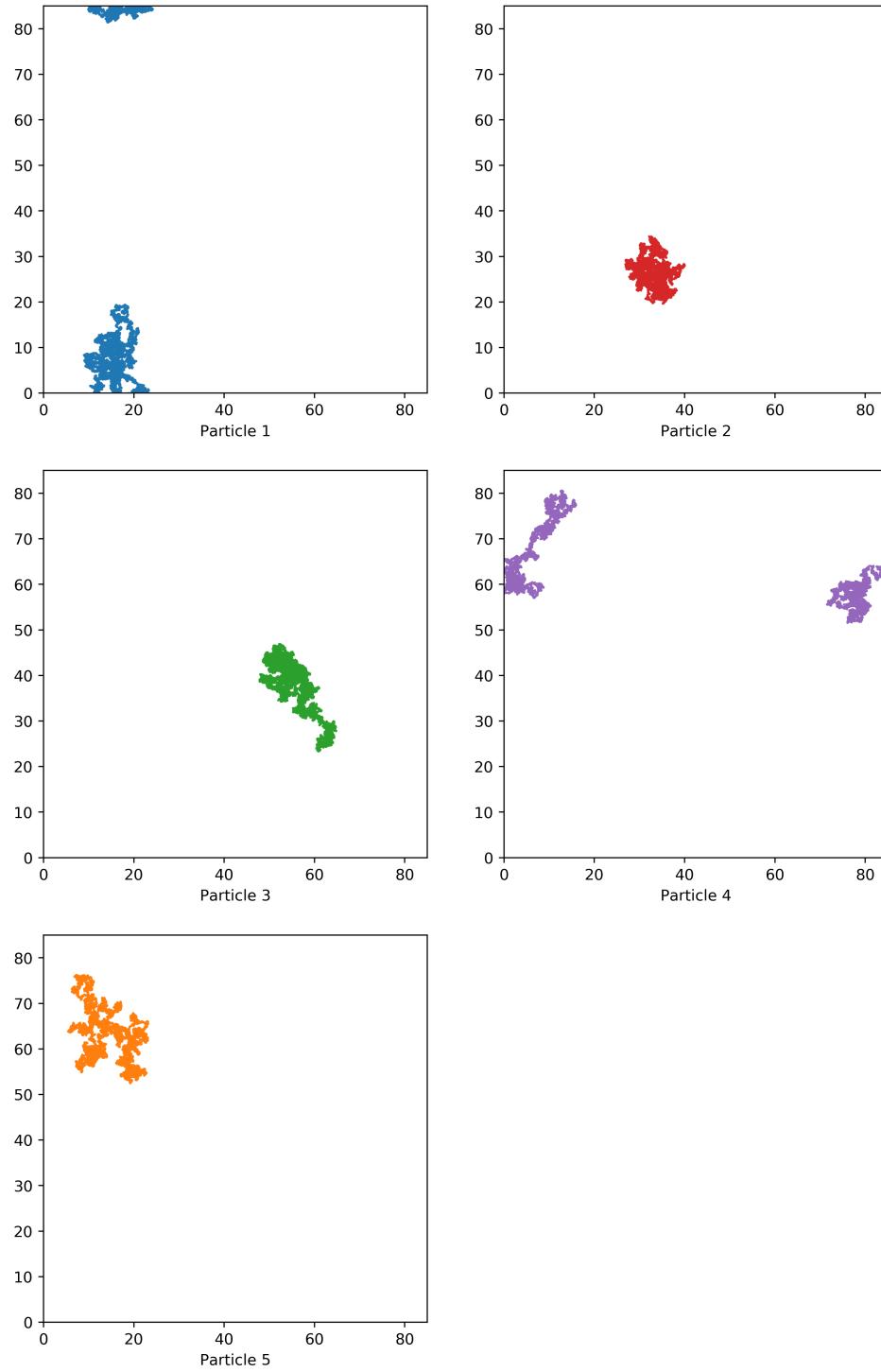


Figure 5: Example trajectories of particles for the simulation of active brownian particles without interactions with $Pe = 0$.

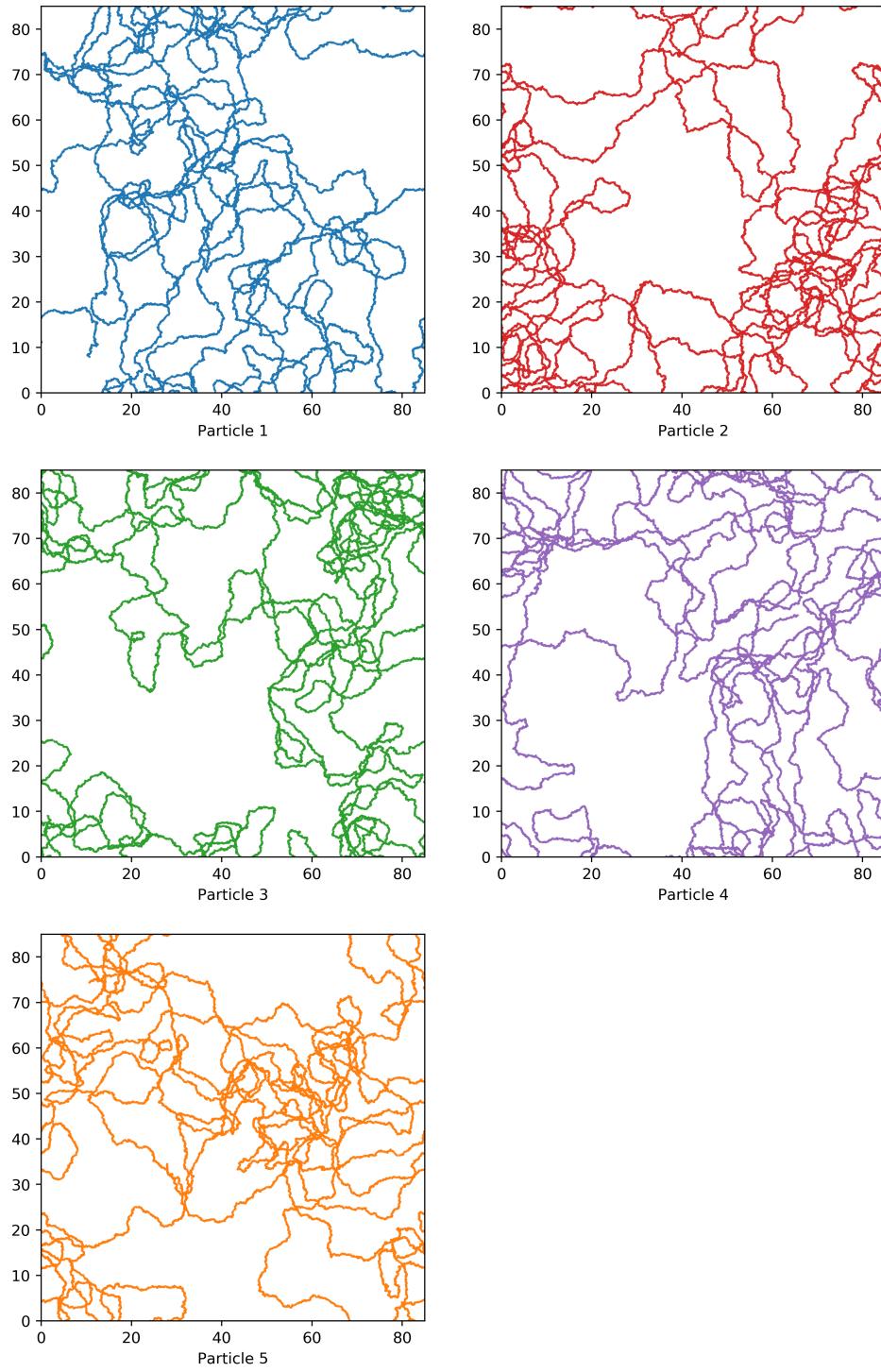


Figure 6: Example trajectories of particles for the simulation of active brownian particles without interactions with $Pe = 20$.

In Fig.7 we see the mean squared displacement (from now on referred to as MSD) of the two simulations and there we see the before mentioned bigger displacement form the initial position for $Pe = 20$. In this

graph the red line is with $Pe = 0$, so passive brownian motion and we see the theoretically expected linear scaling $\langle (x(t) - x_0)^2 \rangle = 4Dt$ of the MSD. Remembering that we set $D = 1$ for our simulations we can check if we observe that in our produced data. The coefficient that a linear fit gives us is $D = 1.0088\frac{\sigma^2}{\tau}$. For the active brownian motion $Pe = 20$ we see three different regimes:

- 1. for short times $t \ll \frac{D}{Pe^2}$ we have pure translational brownian motion $\langle (x(t) - x_0)^2 \rangle \approx 4Dt$. In the simulation here this is for $t \ll \frac{1}{400}\tau$. Again using a linear fit for our data now only including data up to $t = 5 \cdot 10^{-4}\tau$ to ensure $t \ll \frac{1}{400}\tau$ we get $D = 1.049\frac{\sigma^2}{\tau}$.
- 2. for $\frac{D}{Pe^2} \gg t \ll \frac{1}{D_r}$ we have a ballistic regime $\langle (x(t) - x_0)^2 \rangle \approx Pe^2t^2$.
- 3. for $t \gg \frac{1}{D_r}$ we have diffusive regime $\langle (x(t) - x_0)^2 \rangle \approx 2\left(\frac{Pe^2}{2D_r} + D\right)t$

In Fig.8 we see the mean squared angular displacement (MSAD) for the two simulation. The behavior of the MSAD for the two different Peclet numbers is the same because they do not depend on the the Peclet numbers and have just the expected diffusive $MSAD \propto t$ scaling. With a linear fit we get the radial diffusion coefficient $D_r = 1.189\frac{1}{\tau}$.

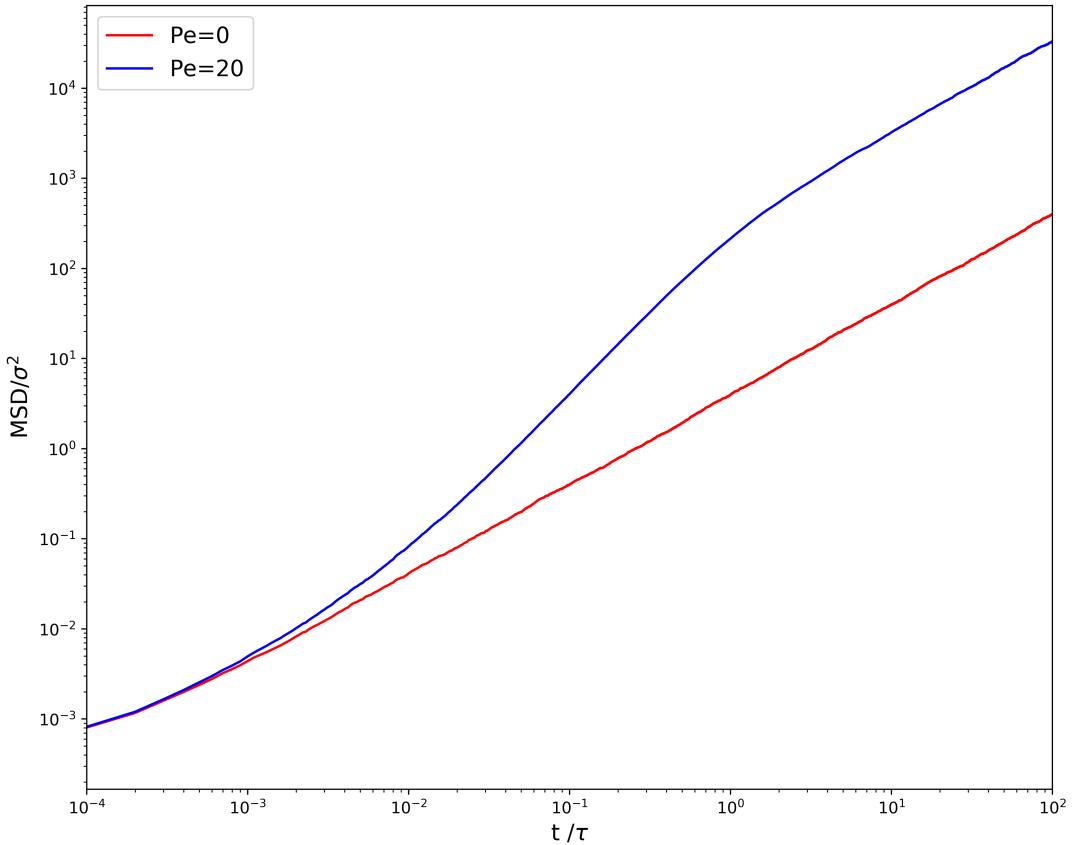


Figure 7: Mean squared displacement for the two different Peclet number $Pe = 0$ and $Pe = 20$ for the simulation of free particles.

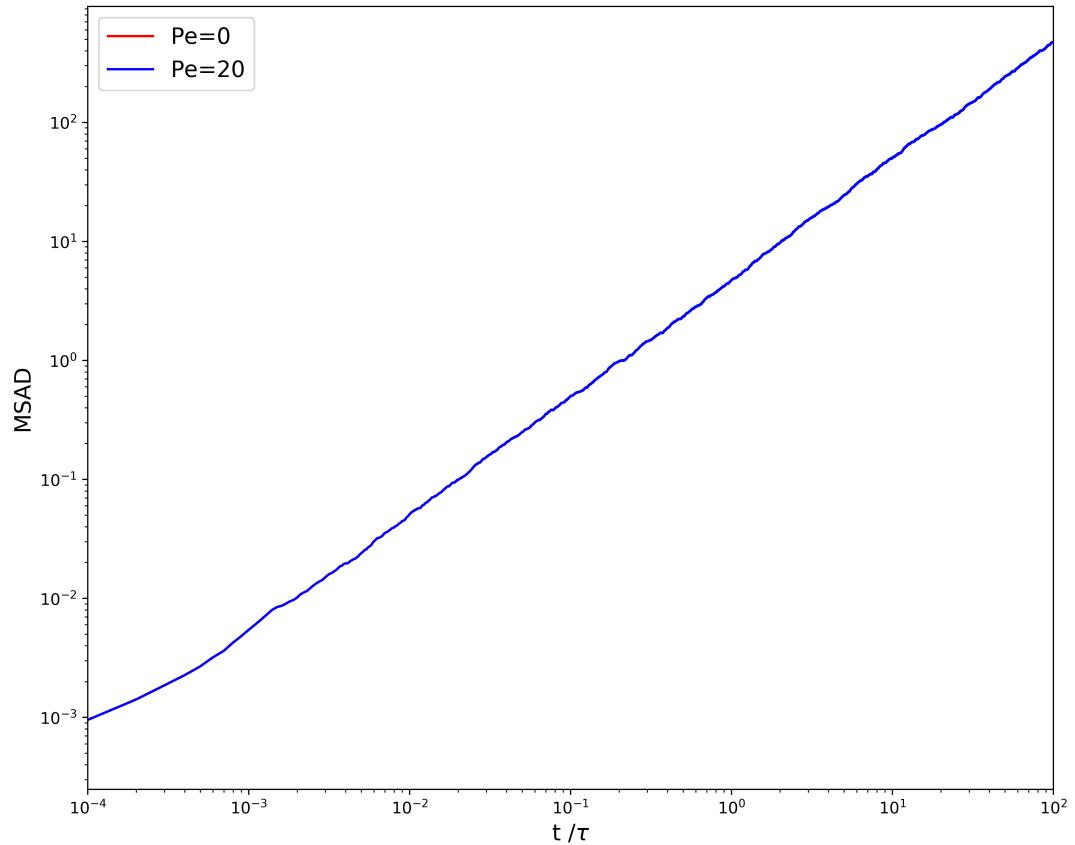


Figure 8: Mean squared angular displacement for the two different Peclet number $Pe = 0$ and $Pe = 20$ for the simulation of free particles.

3 Interactions

We now introduce excluded volume interactions and model the particles in two dimensions as soft disks by using the Weeks-Chandler-Anderson (WCA) pair potential. The diameter of the disks should be 1 in the simulation coordinates. We will use the minimum-image convention for dealing with the periodic nature of the domain. This means that particles are only interacting via their shortest connecting path.

a) Calculate the force given by the WCA potential $F_{WCA}(r_{ij})$ acting from particle i on particle j separated by the distance vector r_{ij} analytically. Use minimum-image convention to construct r_{ij} .

Answer: The Potential is:

$$V(r_{ij}) = \begin{cases} 4 \left(\left(\frac{\sigma}{r_{ij}} \right)^{12} - \left(\frac{\sigma}{r_{ij}} \right)^6 \right) & \text{if } r_{ij} < r_{cut} \\ 0 & \text{if } r_{ij} \geq r_{cut} \end{cases}$$

where r_{cut} is some defined value where the Lennard-Jones potential gets cut. We will use $r_{cut} = \sigma$ so that the particles diameter is $d = 1\sigma$. The resulting force is then easily calculated

$$F_{WCA}(r_{ij}) = \begin{cases} 48(\vec{r}_i - \vec{r}_j) \left(\left(\frac{\sigma}{r_{ij}} \right)^{14} - \frac{1}{2} \left(\frac{\sigma}{r_{ij}} \right)^8 \right) & \text{if } r_{ij} < r_{cut} \\ 0 & \text{if } r_{ij} \geq r_{cut} \end{cases}$$

b) Add the force to the Langevin equations. You can absorb any prefactors into a dimensionless energy scale $\tilde{\epsilon}$.

Answer: The equations of motion for the i th particle with the force are:

$$\dot{\mathbf{r}}_i = Pe \mathbf{e}_i + \sqrt{2}\xi + \sum_{j \neq i} F(\mathbf{r}_{ij})$$

$$\dot{\varphi}_i = q\sqrt{2}\chi$$

And than discretizing these equations again we get for the i th particle and the k th time step:

$$\begin{aligned} r_{i,k+1} &= r_{i,k} + Pe e_{i,k} \Delta t + \sqrt{2\Delta t} \xi_k + \Delta t \sum_{j \neq i} F(r_{ij,k}) \\ \varphi_{i,k+1} &= \varphi_{i,k} + q\sqrt{2\Delta t} \chi_k \end{aligned}$$

d)

Fig.9 shows the mean squared displacement of the simulation of interacting particles as in Fig.7 for two different Peclet numbers. Qualitatively we observe the behavior described above for the non interacting case. We again have the ballistic and diffusive regimes for the MSD with $Pe = 20$ and only a diffusive regime for $Pe = 0$. With the same analysis as before we find the diffusion coefficients to be $D = 0.79\frac{\sigma^2}{\tau}$ for $Pe = 0$ and $D = 1.097\frac{\sigma^2}{\tau}$ for $Pe = 20$ in the range $0 < t < 0.0005\tau$ and the MSAD is unchanged from the non interacting case. The interaction slow the displacement down which can be seen by the fact that for both Peclet numbers the MSD at $t = 100\tau$ is lower as in the free case. So the interactions with the other particles seem to be able to obstruct the free path. This will be an important feature in the next section when discussing MIPS.

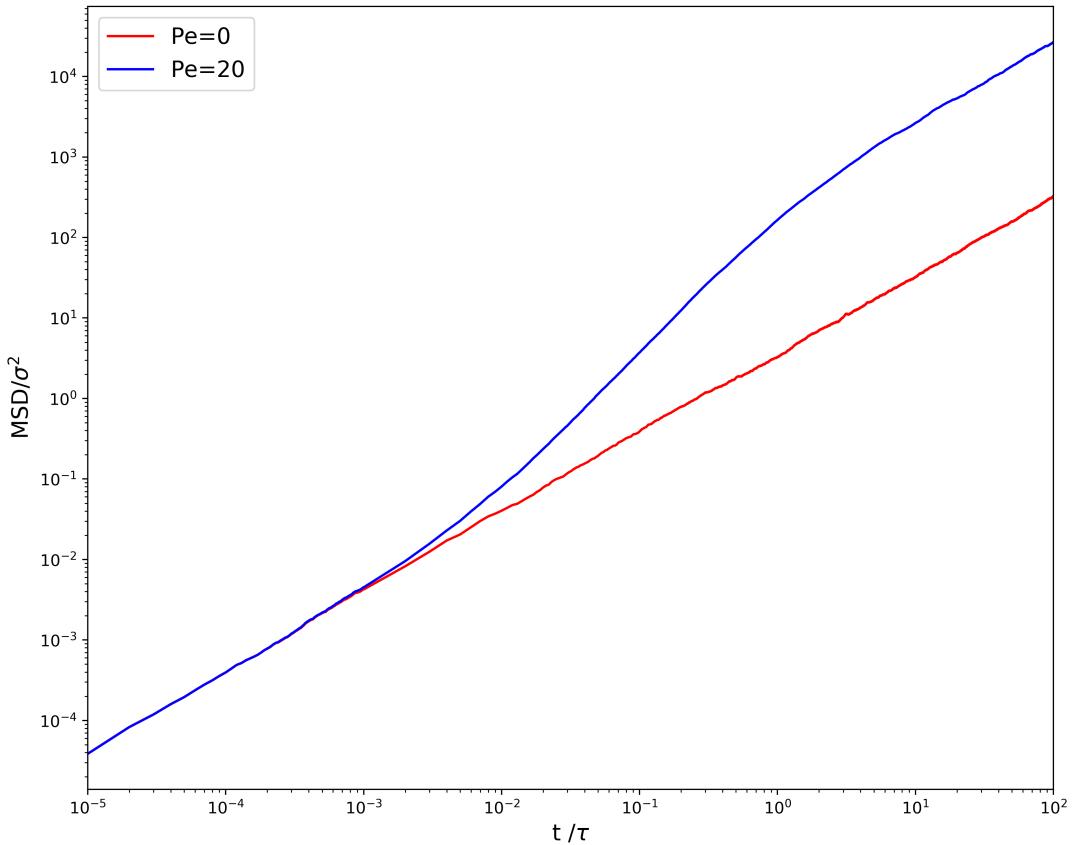


Figure 9: Mean squared displacement for the two different Peclet number $Pe = 0$ and $Pe = 20$ for the simulation of interacting particles.

4 MIPS

a)

Motility-induced phase separation is an effect that occurs in systems with high density and active brownian particles with only repulsive interactions. We there observe regions of a gas like state, i.e. freely moving particles and regions where particles group together in clusters. This grouping into clusters is in the case for passive brownian particles only possible if part of the forces are attractive, but for particles with a self propulsion (motility) it is possible even if the particles have forces that are purely repulsive. The cluster building can happen when for example 3 particles collide with each other and their self propulsion push against each other in such a way that they get stuck until they change the direction of their movement. If in that time more particles collide with these particles the cluster can grow.

We can observe the above explained in Fig.10 and Fig.11. In Fig.5 we have passive brownian particles $Pe = 0$ and we see no cluster building in the snapshots of the system at different times. The particles stay uniformly distributed over the domain. In Fig.11 we have increased the self propulsion to $Pe = 50$ and can now see that there are regions with a higher particle density and regions with lower density.

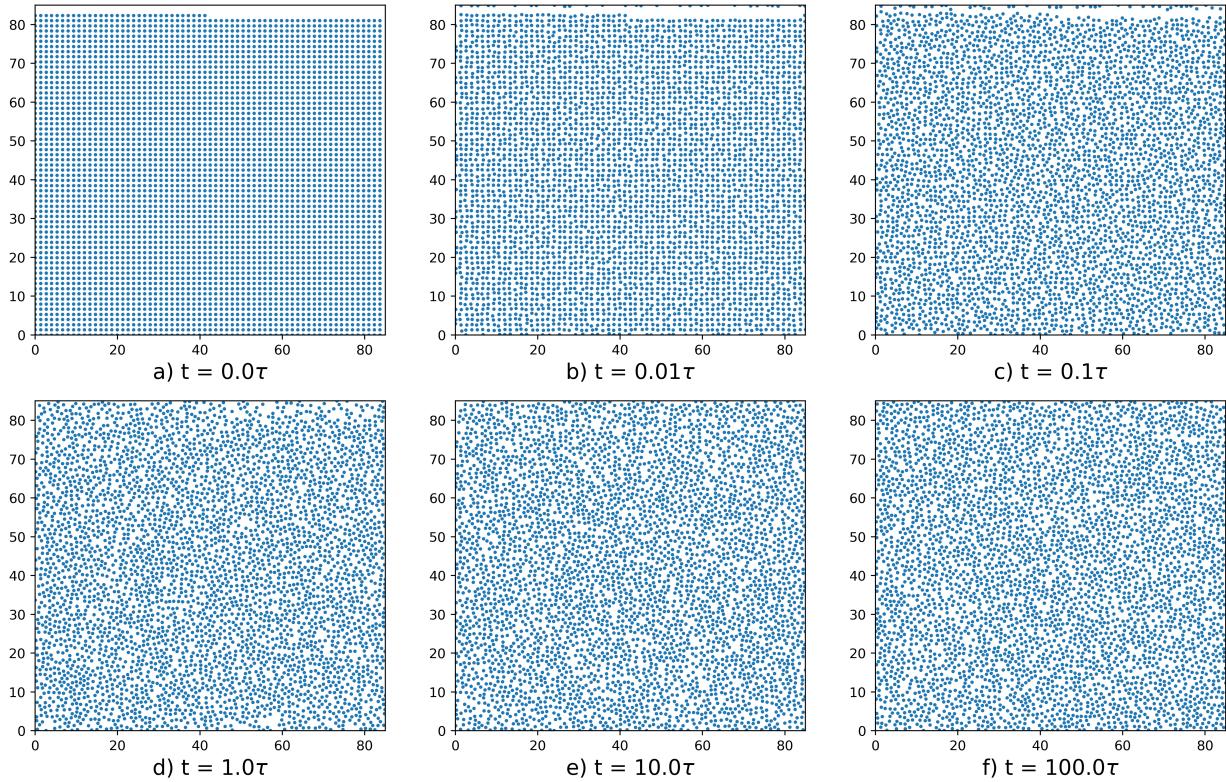


Figure 10: System snapshots for simulation of $N = 4000$ particles with $Pe = 0$

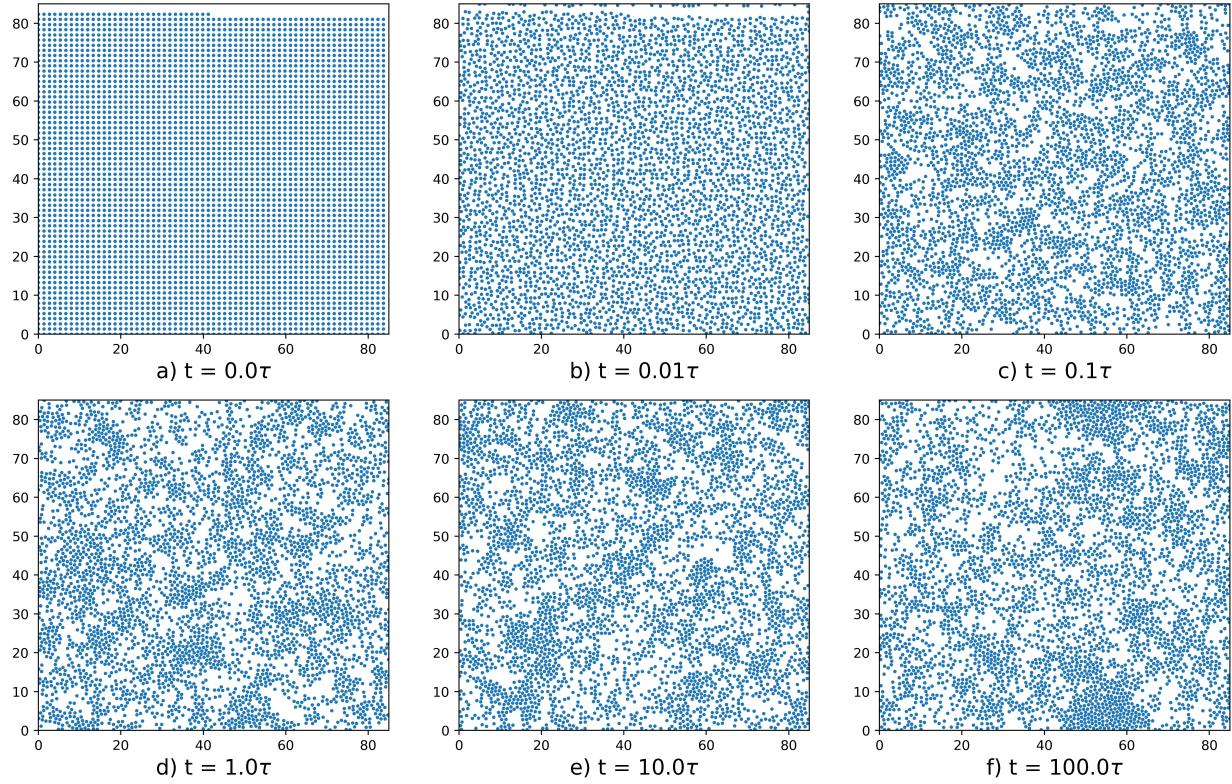


Figure 11: System snapshots for simulation of $N = 4000$ particles with $Pe = 50$

5 Conclusion

In this project we have simulated systems with passive and active brownian particles of varying system sizes. We studied how we can lower computation time by using the cell list algorithm to compute the interactions. Discussed differences of free and interacting active brownian particles, namely their MSD behavior. Lastly we discussed the effect of Motility-induced phase separation and show this effect in our simulation.