Phase Separation and Demixing in ABP Models

Eloise Lardet

One Population

We follow Redner et al.'s paper for modelling a system of spherical active Brownian particles in two dimensions using the overdamped Langevin SDEs [5]:

$$\dot{\mathbf{r}}_i = \frac{D_T}{k_B T}\mathbf{F}(\{\mathbf{r}_i\}) + v_p\hat{\boldsymbol{\nu}}_i + \sqrt{2D_T}\boldsymbol{\xi}_i, \tag{1}$$

$$\dot{\theta}_i = \sqrt{2D_R}\xi_\theta, \tag{2}$$

where $\mathbf{r}_i = (x_i, y_i)$ and θ_i are the position and orientation angle of particle i. Here, k_B is the Boltzmann constant, T is temperature, and $v_p\hat{\boldsymbol{\nu}}_i$ is the self-propulsion velocity in which v_p is the magnitude and $\hat{\boldsymbol{\nu}}_i = (\cos\theta_i, \sin\theta_i)$ is the direction. We also have Gaussian white noise variables $\boldsymbol{\xi}_i, \xi_\theta$. The translational and rotational diffusion constants are given by D_T and D_R, respectively. There is a repulsive excluded volume force, calculated from the WCA potential,

$$V_{WCA}(r_{ij}) = \begin{cases} 4\varepsilon\left[\left(\frac{\sigma}{r_{ij}}\right)^{12} + \left(\frac{\sigma}{r_{ij}}\right)^{6}\right] + \varepsilon, & r_{ij} < 2^{(1/6)}\sigma, \\ 0, & r_{ij} \geq 2^{(1/6)}\sigma, \end{cases} \tag{3}$$

where r_{ij} is the magnitude of the distance between two particles i and j, σ is the diameter of the particles, and we set $\varepsilon = k_B T$.

By nondimensionalizing as Redner et al. [5] describe, with σ as the unit of length, $k_B T$ as the unit of energy, $\tau = \sigma^2/D_T$ as the unit of time, as well as the relation between translational and rotational diffusion constants, $D_R = \frac{3D_T}{\sigma^2}$, we get

$$\dot{\mathbf{r}}_i = \tilde{\mathbf{F}}(\{\mathbf{r}_i\}) + \mathrm{Pe}\hat{\boldsymbol{\nu}}_i + \sqrt{2}\boldsymbol{\xi}_i, \tag{4}$$

$$\dot{\theta}_i = \sqrt{6}\xi_\theta, \tag{5}$$

where $\mathrm{Pe} = \frac{v_p\tau}{\sigma}$ is the nondimensional velocity. The nondimensional version of the excluded-volume repulsive force $\tilde{\mathbf{F}}$ is given by:

$$\tilde{\mathbf{F}}(\{\mathbf{r}_i\}) = \sum_{i\neq j}\mathbf{F}_{j\to i} = \sum_{i\neq j}\frac{24\varepsilon}{r_{ij}^2}\left[2\left(\frac{\sigma}{r_{ij}}\right)^{12} - \left(\frac{\sigma}{r_{ij}}\right)^{6}\right]\mathbf{r}_i, \tag{6}$$

which was found by differentiating the WCA potential with respect to $\mathbf{r}_i$. Here, r_{ij} is the distance between particle i and j. Calculation of the force can also be sped up by using Newton's third law: $\mathbf{F}_{j\to i} = -\mathbf{F}_{i\to j}$.

Numerical methods

Simulations were generally carried out with a timestep of $\Delta t = 2e^{-5}\tau$; however, slightly larger timesteps up to $\Delta t = 5e^{-5}\tau$ did not lead to instability when Péclet was not too large. Periodic boundary conditions were employed throughout.

Euler method

A first order Euler method can be derived for the above SDEs.

$$\mathbf{r}_i(t+\Delta t) = \mathbf{r}_i(t) + \Delta t\left[\tilde{\mathbf{F}}(\{\mathbf{r}_i\}) + \mathrm{Pe}\hat{\boldsymbol{\nu}}_i(t)\right] + \sqrt{2\Delta t}\boldsymbol{w}_i, \tag{7}$$

$$\theta_i(t+\Delta t) = \theta_i(t) + \sqrt{6\Delta t}w_{\theta,i}, \tag{8}$$

where $\boldsymbol{w}_i = (w_{x,i}, w_{y,i})$ and $w_{x,i}, w_{y,i}, w_{\theta,i}$ are uncorrelated sequences of random numbers taken from the standard normal distribution. This method is first order accurate and requires calling the force function once.

Ref [8] was a useful resource for getting started with this for a non-interacting system. MATLAB codes for the dimensionalised model were also provided.

SRK method

Following the approach of the second-order Stochastic Runge-Kutta method given in [3], here is an SRK method for the above model:

$$\mathbf{r}_i(t+\Delta t) = \mathbf{r}_i(t) + \frac{\Delta t}{2}\left[\mathbf{G}_i^a + \mathbf{G}_i^b\right] + \sqrt{2\Delta t}\boldsymbol{w}_i, \tag{9}$$

$$\theta_i(t+\Delta t) = \theta_i(t) + \sqrt{6\Delta t}w_{\theta,i}. \tag{10}$$

The force is calculated in two stages:

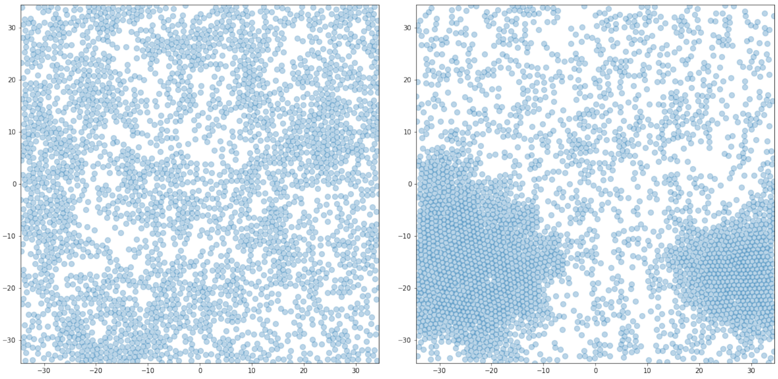
$$\mathbf{G}_i^a = \tilde{\mathbf{F}}(\{\mathbf{r}_i\}) + \mathrm{Pe}\hat{\boldsymbol{\nu}}_i(t), \tag{11}$$

$$\mathbf{G}_i^b = \tilde{\mathbf{F}}(\{\mathbf{R}_i\}) + \mathrm{Pe}\hat{\boldsymbol{\nu}}_i(t+\Delta t), \tag{12}$$

where $\mathbf{R}_i = \mathbf{r}_i + \Delta t\mathbf{G}_i^a + \sqrt{2\Delta t}\boldsymbol{w}_i$. This method is second order accurate but requires calling the force function twice (and therefore takes around twice as long as the Euler method above). For this reason, most of the remaining simulations were carried out using the Euler method for its faster computation times.

MIPS

Discussed in length by Cates and Tailleur [4], motility induced phase separation (MIPS) is an emergent out of equilibrium in which the motility of the active particles with repulsive interactions drives the phase separation of particles into dense and dilute phases. It is useful to check that we indeed see MIPS for any code for the one population ABP model above (see Figure 1 below). Figure 3 in [5] provides a phase diagram for MIPS as a function of volume fraction ϕ and Péclet number Pe.



(a) $N = 3025; \phi = 0.5; \mathrm{Pe} = 40$ (b) $N = 3025; \phi = 0.5; \mathrm{Pe} = 100$

Figure 1: One population model snapshots at $t = 100\tau$ with different Péclet numbers. Left: no MIPS. Right: MIPS.

Two Populations

Phase separation and demixing has previously been investigated in models with two populations of particles with slightly different properties: active and passive particles [7]; differing speeds [2] (note that this model has a slightly different setup without ABPs); different (translational) diffusivities [9].

However, the effect of persistence alone has not been explored, as far as we are aware. To model a change in persistence, we redefine the rotational diffusion as $D_R = \frac{\alpha D_T}{\sigma^2}$, where there is now a new persistence controlling parameter, α. The persistence time of a particle is defined as $\tau_p = 1/D_R$, and persistence length as $\ell_p = v_p \tau_p$. Thus, as the value of α is decreased, τ_p increases and persistence increases. This may be easier to understand by considering the new nondimensionalised Langevin equations:

$$\dot{\mathbf{r}}_i = \tilde{\mathbf{F}}_{ex}(\{\mathbf{r}_i\}) + \mathrm{Pe}\hat{\boldsymbol{\nu}}_i + \sqrt{2}\boldsymbol{\xi}_i, \tag{13}$$

$$\dot{\theta}_i = \sqrt{2\alpha}\xi_\theta. \tag{14}$$

A smaller α means that the rate of change of θ is generally smaller, meaning the particle takes longer to change direction, so has a larger displacement, and is therefore more persistent. In a sense, more persistent particles are more 'active' than less persistent particles.

In terms of a two population model, we define two populations A and B, with an equal number of particles ($N_A = N_B = N_{particles}/2$). Then we define an alpha value for each, α_A, α_B.

Phase Separation

First, it must be noted that when I use the term *phase separation* in the context of this report, I mean it in the sense of thinking of a single population of particles in an external environment as above, with a dense phase cluster in a sea of particles in a dilute phase. The term *demixing* is used instead in the context of particles separating into their distinct types (which is a kind of phase separation in a sense too).

Under the normal circumstances for the one population model with $\alpha = 3$, $\phi = 0.5$ and $\mathrm{Pe} = 40$, the phase diagram in Fig. 3 of [5] tells us that no phase separation would be expected. However, defining $\alpha_A = 3, \alpha_B = 0.3$ in our two population model, we do see phase separation (see Figure 2). In contrast to MIPS in the one population above, this two population model seemed to require a minimum system size to show any phase separation. For $\phi = 0.5, \mathrm{Pe} = 40$, this seemed to be around $N = 3000$. However, clusters were still fairly unstable for this system size so I have generally used a system size of $N = 10000$ for the remainder of the simulations.

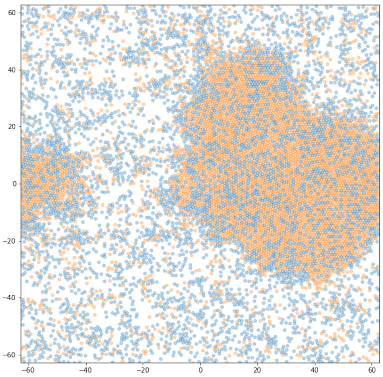


Figure 2: Two population, constant persistences model snapshot at $t = 500\tau$. Type A particles are blue, and type B particles are orange. [$N = 10000; \phi = 0.5; \mathrm{Pe} = 40; \alpha_A = 3; \alpha_B = 0.3$]

Figure 3 shows the fraction of particles in a gas phase over time, to demonstrate the nucleation point and period of growth before a steady state is reached for phase separation, similar to Figure 5 in Redner, Hagan, and Baskaran [5].

Figure 4 plots the local densities of particles as a probability density (a kernel density estimation from histogram). This shows the transition from a unimodal to a bimodal distribution in the local densities as the ratio α_A/α_B is increased, indicating the presence of phase separation.

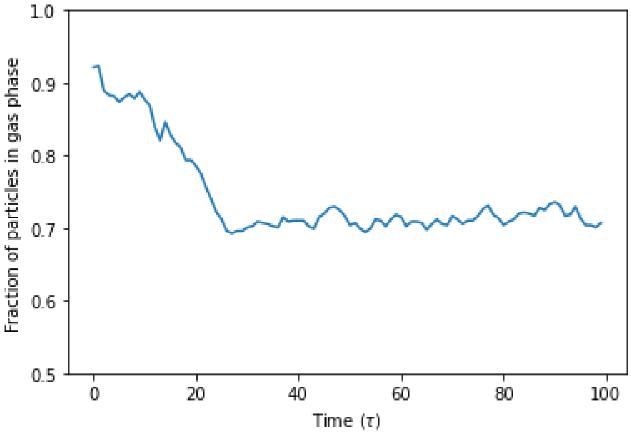


Figure 3: Fraction of particles in gas fraction (defined here as a number density $N/V < 0.9$) over time. Measured using a sub-box method of edge length ≈ 5 and measuring the local number density within each sub-box. Nucleation time for phase separation seems to occur around $t = 0 - 10\tau$, before a period of growth and then reaching a steady state at around $t = 30\tau$. $[N = 10000; \phi = 0.5; \mathrm{Pe} = 40; \alpha_A = 3; \alpha_B = 0.3]$

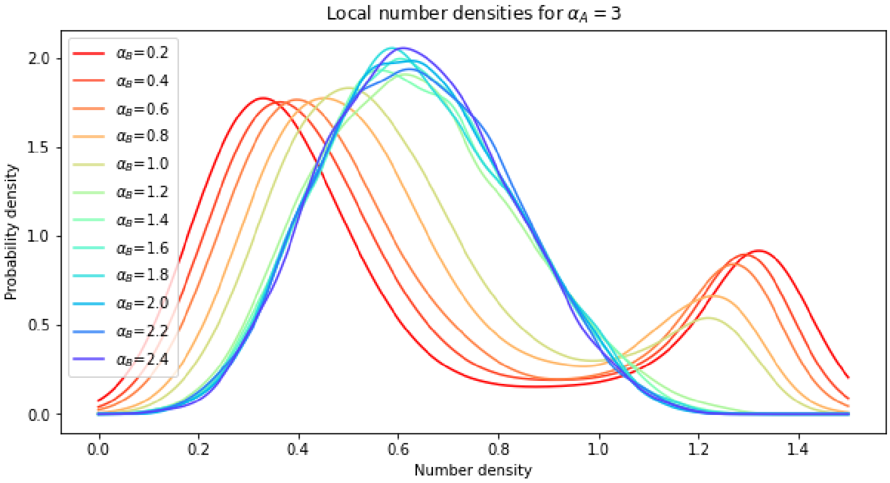


Figure 4: Local number densities of particles averaged over $200 - 300\tau$ (samples every 10τ) for different values of α_B with $\alpha_A = 3$ fixed. Measured using a sub-box method of edge length ≈ 5 and measuring the local number density (N/V) within each sub-box. At larger α_B values (i.e. smaller ratio α_A/α_B), the distribution is unimodal indicating no phase separation. At smaller α_B values (i.e. larger ratio α_A/α_B), the distribution is bimodal indicating there is phase separation between a dense phase and a dilute phase. $[N = 10000; \phi = 0.5; \mathrm{Pe} = 40; \alpha_A = 3]$

Demixing

Demixing is the emergence of particles separating into their separate types of populations (e.g. a cluster of orange particles in a sea of blue particles). One measure for quantifying demixing is the following demixing parameter (DP) from [6]:

$$DP = \langle DP_i \rangle = \left\langle 2\left(\frac{N_s}{N_t} - \frac{1}{2}\right)\right\rangle . \tag{15}$$

Here, N_s and N_t are the number of same type and total number of neighbouring particles from a Voronoi tessellation. A completely mixed system should have $DP = 0$ and a completely demixed (sorted) system would have $DP = 1$ in an infinite system size (note that negative DP values are also possible, although they generally stay fairly close to 0 for a mixed system). In practice, demixed systems generally have a value of DP much lower than 1, especially for smaller systems. However, a demixed system can be easily identified by looking at the value of DP over time. In a mixed system, this generally fluctuates around 0. In a demixed system, the value of DP starts at 0 until a nucleation time when it increases until the system reaches a steady state, and it fluctuates around a larger value of DP (see Figure 5). These fluctuations can be accounted for by averaging over multiple different snapshot times. Figure 6 shows the effect of increasing the ratio of α_A/α_B (in this case with $\alpha = 3$ fixed) on the demixing parameter (15).

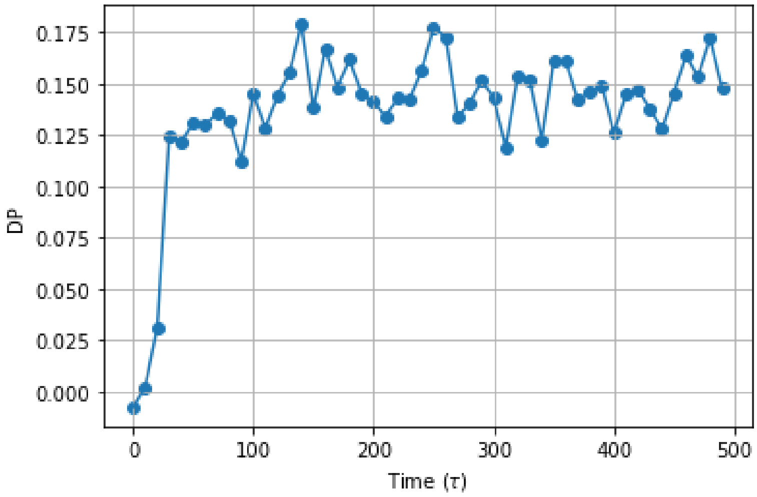


Figure 5: Value of DP over time up to 500τ. There is a nucleation time around the same time that phase separation starts. The value of DP then quickly increases until it appears to reach a steady state (I will run some longer simulations to check this is indeed a steady state). $[N = 10000; \phi = 0.5; \mathrm{Pe} = 40; \alpha_A = 3; \alpha_B = 0.3]$

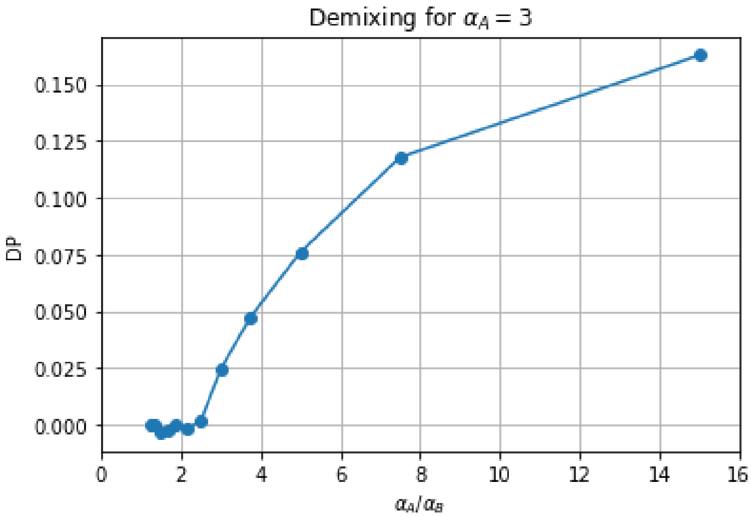


Figure 6: Demixing parameter averaged over 200-300τ, with samples every 10τ for various α_A/α_B ratios, with $\alpha_A = 3$ fixed. (I will add more α_A and α_B values to this plot later.) $[N = 10000; \phi = 0.5; \mathrm{Pe} = 40; \alpha_A = 3]$

Collisions and Persistence

The original motivation for investigating different persistences was from some observations from some experimentalists looking at the movement of specific cells that resulted in striped pattern formation in organisms such as drosophila and zebrafish. By measuring the MSD of cells after collisions, it was seen that homotypic collisions between A/A and B/B did not lead to any change compared to normal for the MSD, but for hetertypic collisions between A/B, the MSD of type B cells increased more than usual for a short time after collision (i.e. an increase in persistence).

To model this, we assume particles are 'in contact' if they are within the interacting distance of $r = 2^{(1/6)}\sigma$. To keep track of which particle pairs A-B are in contact, there is a matrix M of size $(\frac{N}{2}, \frac{N}{2})$, which has entry 1 in M_{ij} if A_i and B_j are in contact, and 0 if they are not. At every timestep, we find all the particles pairs of type A and B in contact and update matrix M accordingly. The time at which an entry of matrix M changes from a 1 to 0 is defined as t_{coll}. Each particle has its latest t_{coll} time stored in a vector t^*. At time t_{coll}, the α value of the particle of type B involved in the collision is set to a smaller value of α'_B. Over a set period of time t_{rec}, this α value increases linearly back to its normal value of α_B (see Figure 7). If another collision were to happen before the α value recovering to its normal value, it would simply jump back down to α'_B before increasing linearly again over time t_{rec}.

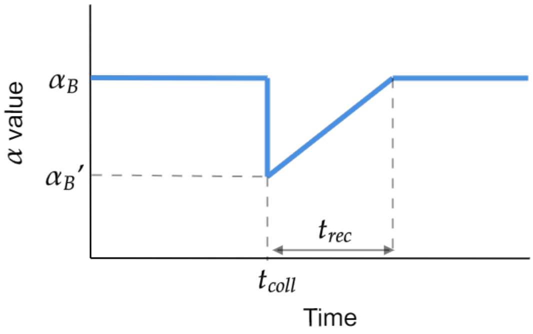


Figure 7: Sketch of α value over time of a particle of type B involved in a collision with a type A particle. After collision at t_{coll}, α immediately decreases to α'_B, before linearly increasing back to its normal α_B value over a recovery time of t_{rec}.

Further work

Phase diagrams

Create full phase diagram for phase separation with α_A vs α_B, and full phase diagram for demixing with α_A vs α_B.

Understanding the demixing

It clearly appears from my simulations that demixing cannot occur without phase separation. However, it seems like we may be able to find phase separation without demixing (i.e. the dense cluster being a random mix of type A/B particles). Is this controlled by the ratio of the α values? Or maybe my simulations have simply not been run for long enough to see demixing yet.

Demixing seems to be controlled by the fact that the more persistent B particles are in the MIPS region, but the less persistent A particles are not. But what if we set the α_A value to be in the MIPS region? In some case they do demix where we have a cluster of B particles surrounded by a sea of A particles. One theory could be the fact that when the B particles are in the MIPS region, they create a dense cluster. These B particles may then push the less persistent out of the cluster. This then leaves a space with a lower effective volume fraction to the initial ϕ, which may mean that the less persistent A particles are no longer in the MIPS region, even if they would have been before.

But what if we set α_A to be even lower, such that they should still undergo MIPS? Maybe some sort of ring pattern could form of B particles in a cluster surrounded by A particles, or A particles could form their own separate cluster, or maybe they do not demix and simply form one large cluster. My simulations so far seem to show this final option in which demixing does not occur (e.g. $\alpha_A = 0.1, \alpha_B = 0.01$). We also demixing to a lesser extent when the α values are too far apart (e.g. $\alpha_A = 3, \alpha_B = 0.03$) Although this could again be down to the fact that the simulations have not been run for long enough and have not reached steady state for phase separation and/or demixing (I am currently testing this) as, due to their much higher persistence, the B particles now take longer to change direction.

Changing Persistence Model

In the end, this model was not investigated in as much detail compared to the constant (but different) perisistences model. This is because to find any phase separation or demixing in this model, we would certainly expect to see it in a model where all particle B α values stay constant at the lower value of α'_B. From some preliminary testing, it seems like the same sort of effects of phase separation and demixing are seen in the model. It would be useful to play around with the recovery time t_{rec}, as I found that due to the sheer number of collisions occurring, the B particles' α values often stayed very low close to α'_B unless t_{rec} is extremely short (which might not make much physical sense). It could be helpful to display the value of α for each particle as a colour map when plotting snapshots/ animations. It would be useful to compare some of the data from this model to the experimental data found, for example MSDs after collisions. This could also help to fine tune the parameters to be more comparable with the experimental data.

References

[1] M. P. Allen and D. J. Tildesley, *Computer Simulation of Liquids.* Oxford University Press, Nov. 23, 2017, vol. 1, ISBN: 978-0-19-880319-5. DOI: 10.1093/oso/9780198803195.001.0001.

[2] G. Baglietto, A. Seif, T. S. Grigera, and W. Paul, "Otherwise identical particles with differing, fixed speeds demix under time-reversible dynamics," *Physical Review E*, vol. 101, no. 6, p. 062 606, Jun. 18, 2020. DOI: 10.1103/PhysRevE.101.062606.

[3] A. C. Brańka and D. M. Heyes, "Algorithms for brownian dynamics computer simulations: Multivariable case," *Physical Review E*, vol. 60, no. 2, pp. 2381–2387, Aug. 1, 1999. DOI: 10.1103/PhysRevE.60.2381.

[4] M. E. Cates and J. Tailleur, "Motility-induced phase separation," *Annual Review of Condensed Matter Physics*, vol. 6, no. 1, pp. 219–244, Mar. 2015. DOI: 10.1146/annurev-conmatphys-031214-014710.

[5] G. S. Redner, M. F. Hagan, and A. Baskaran, "Structure and dynamics of a phase-separating active colloidal fluid," *Physical Review Letters*, vol. 110, no. 5, p. 055 701, Jan. 31, 2013. DOI: 10.1103/PhysRevLett.110.055701.

[6] P. Sahu, D. M. Sussman, M. Rübsam, A. F. Mertz, V. Horsley, E. R. Dufresne, C. M. Niessen, M. C. Marchetti, M. L. Manning, and J. M. Schwarz, "Small-scale demixing in confluent biological tissues," *Soft Matter*, vol. 16, no. 13, pp. 3325–3337, 2020. DOI: 10.1039/C9SM01084J.

[7] J. Stenhammar, R. Wittkowski, D. Marenduzzo, and M. E. Cates, "Activity-induced phase separation and self-assembly in mixtures of active and passive particles," *Physical Review Letters*, vol. 114, no. 1, p. 018 301, Jan. 6, 2015. DOI: 10.1103/PhysRevLett.114.018301.

[8] G. Volpe, S. Gigan, and G. Volpe, "Simulation of the active brownian motion of a microswimmer - MATLAB code," *American Journal of Physics*, vol. 82, no. 7, pp. 659–664, Jul. 2014. DOI: 10.1119/1.4870398.

[9] S. N. Weber, C. A. Weber, and E. Frey, "Binary mixtures of particles with different diffusivities demix," *Physical Review Letters*, vol. 116, no. 5, p. 058 301, Feb. 4, 2016. DOI: 10.1103/PhysRevLett.116.058301.

Appendix: A note about neighbour lists

The most computationally expensive part of most of the codes was the inter-particle force calculations. At first a straight forward 'brute force' method was applied in which we check every pair of particles to determine whether they are within the interacting distance. Even with some further optimisations, and making use of Newton's third law to halve the number of distance calculations, this method is still $O(N^2)$.

Since computation time was a significant limitation of my codes, especially for larger system sizes, I tested and implemented three new, more efficient methods: Verlet, linked list, and a combined method making use of both of the first two. These all make use of neighbour lists. A useful starting point for Verlet and linked neighbour lists was Chapter 5 in [1].

The combined method employed here uses the linked list method to assign each particle to smaller grid (of edge length at least r_{skin}) within the box, and store this information as a linked list. At the very start of the code, all the possible grid neighbours for each grid (of which there are nine, including this grid itself) are calculated. This linked list is then used to create a Verlet neighbour list with a total skin radius of r_{skin}. We search all nine neighbouring grids, but discount any particles with an index lower than the current particle's index, as this particle pair will already have been accounted for. At each timestep of the numerical method, we only have to search the neighbouring particles in the Verlet neighbour list to find the particles within the interacting distance, r_{cut} (note $r_{cut} < r_{skin}$). After a certain number of timesteps, the particles will have travelled far enough that the Verlet list will need to be updated, at which point the linked list is recalculated from which a new Verlet list is generated. This update can be automatic by doing the neighbour list update when the sum of the magnitudes of the two greatest particles' total displacements is greater than $r_{skin} - r_{cut}$.

Figure 8 shows the computation time comparison for the different neighbour list methods, as well as the brute force method. Even for fairly small system sizes, the neighbour list methods are much more efficient. For very large system sizes, the combined method outperforms both the Verlet and linked list method significantly.

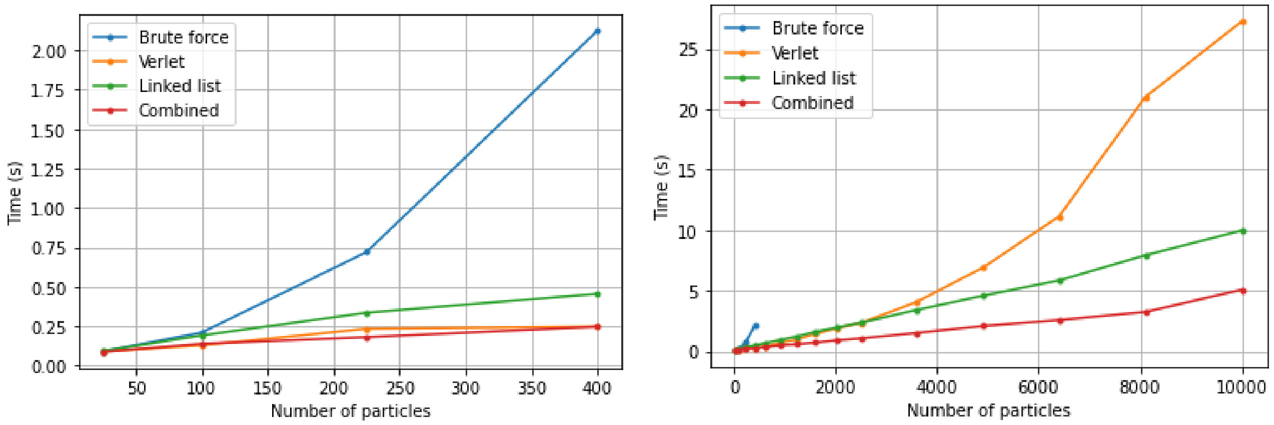


Figure 8: Computation time comparison for different neighbour list methods for various different system sizes. All simulations were run until $t = 0.1\tau$ with the same seed, and without saving any data to csv files. [$\Delta t = 5e^{-5}; \phi = 0.6; \mathrm{Pe} = 20$]