SORTING OF CHIRAL MICROSWIMMERS

A THESIS
SUBMITTED TO THE DEPARTMENT OF PHYSICS
AND THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

By
Mite Mijalkov
December, 2014

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assist. Prof. Dr. Giovanni Volpe(Advisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Hande Toffoli

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assist. Prof. Dr. Miguel Navascués

Approved for the Graduate School of Engineering and Science:

Prof. Dr. Levent Onural
Director of the Graduate School

ABSTRACT

SORTING OF CHIRAL MICROSWIMMERS

Mite Mijalkov
M.S. in Physics
Supervisor: Assist. Prof. Dr. Giovanni Volpe
December, 2014

Microscopic swimmers, for example chemotactic bacteria and cells, are capable of directed motion by exerting a force on their environment. In some cases, including bacteria and spermatozoa swimming near boundaries, or many asymmetrical artificial microswimmers, the driving force and propulsion direction are misaligned. In those situations a torque acting on the microswimmers arises, resulting in motion with a well-defined chirality which is circular in two dimensions and helicoidal in three dimensions. In this thesis, I demonstrate with numerical simulations in two dimensions, how the chirality of the circular motion can couple to chiral features present in the microswimmer environment. I show that by employing static chiral pattern of elliptical obstacles in their environment, microswimmers can be separated on the basis of their motion parameters. In particular, levogyre and dextrogyre microswimmers as small as 50nm can be separated and selectively trapped in *chiral flowers* of ellipses. *Patterned microchannels* can be used as *funnels* to rectify the microswimmer motion, as *sorters* to separate microswimmers based on their linear and angular velocities, and as *sieves* to trap microswimmers with specific parameters. I also demonstrate that these results can be extended to helicoidal motion in three dimensions.

Keywords: Active chiral microswimmers, sorting of microswimmers.

ÖZET

KIRAL MIKROYÜZÜCÜLERIN SINIFLANDIRILMASI

Mite Mijalkov
Fizik bölümü, Yüksek Lisans
Tez Yöneticisi: Yrd. Doç. Dr. Giovanni Volpe
Aralık, 2014

Kemotaktik bakteriler ve hücreler gibi yüzen mikroskopik canlılar çevrelerine kuvvet uygulayarak hareket yönlendirebilme yeteneğine sahiptirler. Bazı durumlarda, örneğin, bakteri ve sınıra yakın yüzen sperm, veya birçok asimetrik yapay yüzücülerde, itici kuvvet ve itme yönü yanlış hizalanmıştır. Böyle durumlarda; mikroyüzücüler üzerine etki eden, iki boyutta dairesel ve üç boyutta sarmal olan iyi tanımlanmış bir kiralite ile harekete neden olan bir dönme kuvveti ortaya çıkar. Bu tezde, iki boyuttaki nümerik simulasyonlar ile, dairesel hareketin kiralitesi ile mikro yüzücülerin ortamında mevcut olan kiral özelliklerin birbiri ile nasıl eşleştiği gösterilmektedir. Kendi çevrelerindeki eliptik engellerin statik kiral desenini kullanarak, mikroyüzücülerin kendi hareket parametreleri temelinde ayrılabildiği gösterilmektedir. Özellikle, 50nm kadar küçük levojir ve dextrojir mikroyüzücüler elipslerin *kiral çiçeklerinde* ayrılabilir ve seçici olarak kapana kıstırılabilirler. Mikroyüzücülerin hareketini düzeltmek amacıyla, lineer ve açısal hızına dayalı olanları ayırmak için *ayıklayıcı* olarak ve belirli parametreler ile olanları yakalamak için *süzgeç* olarak desenli mikrokanallar *huni* olarak kullanılabilirler. Ayrıca bu sonuçların üç boyuttaki sarmal harekete uzatılabildiği de gösterilmektedir.

Anahtar sözcükler: Aktif kiral mikroyüzücüler, mikroyüzücülerin sınıflandırılması.

Acknowledgement

I cannot express my gratitude to Dr. Giovanni Volpe enough, for his patience with me when I was starting to learn programming, his guidance, his willingness to listen to my ideas and discuss them and for the opportunity he gave me to work on extremely good projects, helping me develop as a person who enjoys his work. I sincerely believe that there is no-one in the academic circles that could do more for me and I feel honored that I could spend three fruitful years working with him.

A special thanks to Dr. Hande Toffoli and Dr. Miguel Navascués for their agreement to be in this thesis' comitee and provide insight to make it as good as possible. Also, I would like to express my thanks to Prof. Atilla Erçelebi for his guidance when I needed the most, upon enrolling in the physics department in Bilkent.

I would like to give my best wishes to all my friends that I met in METU and Bilkent as well as all the members of the Soft Matter group in Bilkent. You were all special to me and I will always cherish the small moments in life that I shared with you, the drinking after a hard homework, drinking in a pub, drinking in the office, picnics, coffee breaks and conference travels. I wish you all the best with your life, and hope you obtain your goals and find true happiness. I hope we could meet in future and create new and better memories.

To my uncle Sterjo, my parents Dimce and Ivanka, it goes without saying that I would not have been here without your continuous support, understanding and motivation. I will do my best to repay for all the incredible things you have done for me, but I honestly believe that one lifespan would not be enough to show my true gratefulness to you.

Finally, Merve, I hold you near and dear to my heart. The day to day support that you gave me during these past 8 months have been unimaginable and I cannot believe I was so lucky to meet such a kind and understanding person. You impress me more and more with every passing day and I hope to see your beautiful face and charming smile for the long years to come.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Most of the events that we frequently encounter are inherently random. Randomness takes place in many sciences including game and information theory, and is also used to model stock markets, games of chance, etc. [1]. Due to its widespread use, there have been great interest to model and gain an intuitive understanding of these phenomena. However, solving the corresponding equations is a tough task due to the requirement of complex mathematical understanding. Often, processes are simplified by doing some justifiable approximations and solving simpler models that serve as a paradigm to understand more complex behavior.

One of the simplest stochastic processes that can be studied is the Brownian motion [2]. This is the omnipresent and continuous random motion of any microscopical particle suspended in a fluid that arises due to the continuous collisions with the surrounding fluid molecules. As such, it has been a model and inspiration for many discoveries, in particular the development of the microscopic theory for the matter structure [3]. Moreover, since the Brownian motion exists in thermal equilibrium as a fluctuation, it can be used as model for systems in the scope of equilibrium statistical mechanics.

In contrary to the Brownian particles, which fluctuate randomly while their average position remains at zero, active Brownian particles, also referred to as microswimmers, are able to take up energy from the environment and use it to

navigate through the environment [4]. They have a well directed motion, which is disturbed by random fluctuations due to their Brownian nature. Due to this property, active swimmers possess the potential to be used in many applications which would include any kind of pick up and delivery of molecules at the nanoscale, then to localize pollutants in soils or to perform tasks in lab-on-a-chip devices [5, 6, 7, 8]. Additionally, their ability to self-propel means that active swimmers are out of equilibrium, so their behavior could be used to model such systems [4]. All these promising applications led to a great interest to study microswimmers and many artificial swimming methods to solve the problem of self-propulsion have been proposed [9].

However, often microswimmers are designed to be asymmetric and the self-propulsion direction does not align with the driving force [10]. In these cases microswimmers become chiral and they undergo a circular motion in two and helicoidal motion in three dimensions. Such examples are vast, and they also exist in nature, for example the bacterium E. *coli* and spermatozoa swim in circles when they are near a boundary [11, 12, 13, 14, 15, 16, 17].

Sorting chiral microswimmers based on their swimming properties is of topmost importance in science and engineering. For example, in order to increase the probability of success of the artificial fertilization, velocity based selection of the spermatozoa might be employed [18]. Separation and sorting of some genetically engineered bacteria can be achieved using the morphological variations in the motion parameters [19]. Moreover, the capability of microswimmers to perform a specific tasks, such as bioremedication or drug delivery, can be significantly increased just by selecting the swimmers that possess the most appropriate swimming properties. Finally, the need to separate the chiral (levogyre and dextrogyre) molecules is very pronounced because generally only one specific chirality is needed by the pharmaceutical and chemical industry [20]. Molecules can be made active by coupling them with a microscopic chiral propellers, and then remove the propellers after the separation is completed. The extreme importance in the methods presented in this thesis, lies in the fact that due to the small Reynolds number of the environment in which the molecules exist, their separation due to chirality is very difficult to achieve by mechanical means [21].

In this thesis, I demonstrate by performing numerical simulations, that active

chiral swimmers can be sorted and separated due to their chirality, linear and angular velocity by placing some static patterns in their environment [22]. I explain how to numerically solve the Langevin equation for the motion in homogeneous environment (chapters 2-5), complex environment (chapter 6) and finally I present my results of how to sort chiral microswimmers (chapter 7). First, in chapter 2, I go over some simple definitions of the random walk before outlining the finite difference method and show how to numerically simulate the white noise. Using these concepts, I demonstrate how to numerically simulate a Brownian particle (chapter 3), active Brownian motion (chapter 4) and chiral microswimmers (chapter 5). Then, in chapter 6, I show how to treat the most commonly encountered boundary conditions which arise when a microscopic particle encounters any obstacle in its environment. In here, I first outline the procedure of treatment of periodic and reflective boundary conditions and then I present the sliding boundary conditions illustrated in the examples of particle in a spherical well and microswimmer encountering an elliptical obstacle. Finally, I introduce my results and show that a racemic mixture of two dimensional chiral microswimmers can be separated by placing the microswimmers inside a *chiral flower* which traps microswimmers with specific chirality. Moreover, I show that a *patterned microchannel* can be used as a *funnel* which rectify the motion of the swimmers, as a *sorter* on the basis of the linear and angular velocities of the microswimmers and as a *sieve* to trap the microswimmers with certain properties. Moreover, I demonstrate that all the results can be extended to the three dimensional case. All these results are scalable down to smaller microparticles as long as the Péclet number of the motion is kept constant.

Chapter 2

Random Walk

Random walks are used to model many phenomena occurring at different scales, such as scattering of the light at the nanoscale, the motion of biomolecules and nanodevices at microscale or the financial status of a gambler, stock markets or food foraging. Two properties of the random walk are of utmost importance [23]: *scale invariance*, the random walks look the same at all scales, and *universality*, random walks have the same properties regardless of their nature. Therefore, I discuss some general properties of the random walk in this chapter as an outline of the physical view of Brownian motion that I introduce in Chapter 3. First, I consider the binary unbiased random walk and derive average quantities and the diffusion equation for this model. Then I introduce the finite difference method and go over the procedure of numerically simulating the effects on white noise, before I end this chapter by describing the generalized version of random walk and show the properties of this stochastic process via numerical simulations.

2.1 Binary unbiased random walk in 1D

Binary random walk in 1D occurs when the walker can choose to go in one of the two available directions, making steps of equal length in the process. If the two choices can occur with same probabilities the walk is said to be unbiased, the

walk is biased if one event is more likely to occur then the other (i.e. the walker experiences a kind of *drift* in either dimension).
In particular, consider the case in which we have a walker in 1 dimension who makes a decision to go left or right based on the outcome of a simple coin flip. All the steps are of equal length, L, as depicted in figure 2.1.

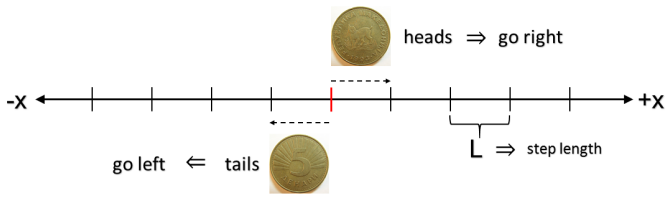

Figure 2.1: Illustration of binary random walk. The walker makes a decision based on a coin flip: he goes right if the result is heads and left if tails. All the steps are of equal length, L. This is a unbiased random walk since the coin is fair, there is 50% probability to go in either direction.

In order to show how the motion looks, I simulate three random walkers that take 15 steps each and show them in figure 2.2. Any individual step is denoted by the black markers, the colors indicate the distinctive paths of the walkers. From figure 2.2, we observe that all the walks are rugged, unlevel, independent and different from each other. Subsequently, the individual displacements after N steps, D_N vary from walker to walker. As a result, some variables need to be derived that will answer the questions about how far the walker goes *on average* in a given time, or how much time does the walker need to get to some certain distance.

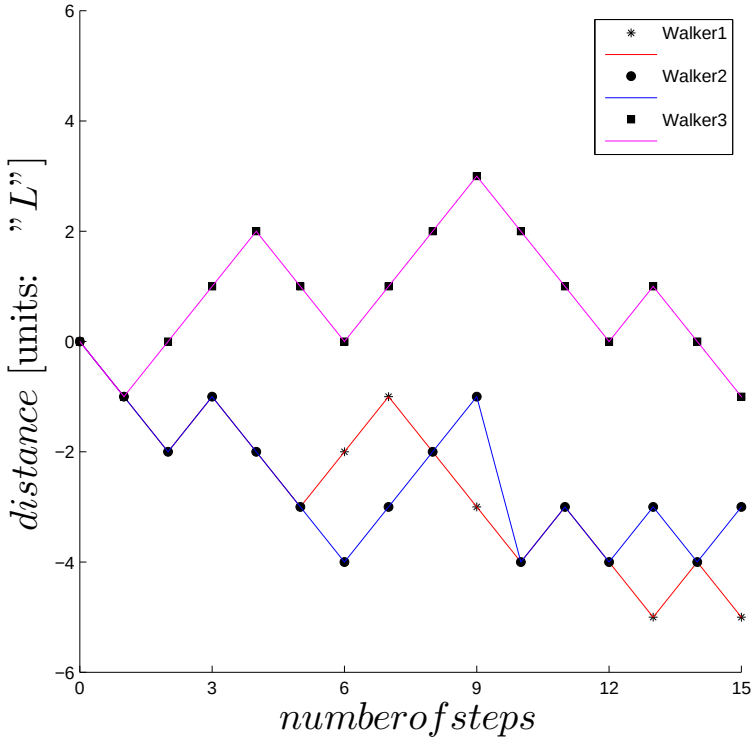

Figure 2.2: The path that three walkers take (in units of L) as a function of the number of steps taken in the fair coin toss game. Black markers denote each individual step of the walkers.

2.2 Average quantities

Consider the binary random walker that can make steps with length $\pm L$. Then, if D_N is the distance covered in N steps, we can express this distance as a sum of the individual steps,

$$D_N = L_1 + L_2 + L_3 + ... + L_N = \sum_{i=1}^{N} L_i \tag{2.1}$$

Taking the average and keeping in mind that the average for a single step in an unbiased random walk is 0, we get

$$\overline{D_N} = \overline{L_1 + L_2 + L_3 + ... + L_N} = \overline{L_1} + \overline{L_2} + \overline{L_3} + ...\overline{L_N} = 0 \tag{2.2}$$

Therefore, the **average position of an unbiased random walk is zero**. This means that, if we plot the probability distribution of finding the walker at some

distance, it will be centered at the origin and it will have its tails expanding as the number of steps increases.

The quantity that gives the information of how much the walker traveled from the initial point after some certain number of steps N is the square of the distance, D_N^2. To find the expectation value of D_N^2, we simply square 2.1. Then,

$$\overline{D_N^2} = \overline{(L_1 + L_2 + L_3 + ... + L_N)^2} = \sum_{i=1}^{N} {L_i}^2 + \sum_{i=1}^{N}\sum_{j\neq i}^{N} L_i L_j \tag{2.3}$$

Now, calculating both terms in the sum one by one,

$$(L_i)^2 = L_i L_i = \frac{1}{2}[(+L)(+L) + (-L)(-L)] = L^2 \tag{2.4}$$

$$L_i L_j = \frac{1}{4}[(+L)(+L) + (-L)(-L) + (+L)(-L) + (-L)(+L)] = 0 \tag{2.5}$$

Therefore, D_N^2, or the mean square displacement from the starting point, we have [23]

$$\langle D_N^2 \rangle = \sum_{i=1}^{N} L^2 = NL^2 \tag{2.6}$$

So, the **mean-square displacement of an unbiased walker is proportional to the number of steps**.

One more measure for a random walk, is the information of how much time it takes to the walker to reach a certain distance. Consider that it takes the walker t_0 seconds to make a single step with some velocity v. Then, during the total time t, the walker will make $N = \frac{t}{t_0}$ steps with length $L = vt_0$. Then,

$$\langle D_N^2 \rangle = NL^2 = \frac{t}{t_0}(vt_0)^2 = (vt_0)vt = (vL)t \tag{2.7}$$

Therefore, we conclude that the **the mean-square displacement of an unbiased walker is proportional to the time.**

Figure 2.3 shows the calculated MSD (mean-squared displacement) for 4 different step lengths. It can be seen that the MSD is proportional to the number of steps (also proportional to time, since $t = Nt_0$) and it also illustrates the fact that MSD increases as the step length increases.

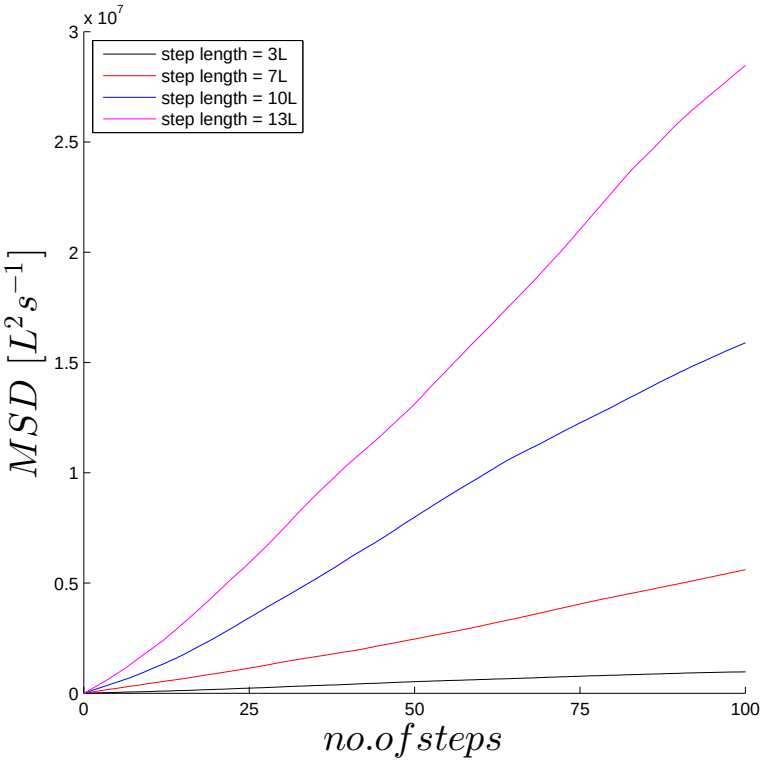

Figure 2.3: MSD as a function of time (or number of steps) for 4 different walkers. Each walker takes a step with different length and they all walk for $100L$ steps. It shows that MSD depends greatly on the step size and is directly proportional to the time elapsed in motion.

An important point to note here is that the above properties were obtained by taking *time averages*, i.e. considering a particular trajectory of a single random walker and averaging it over time. However, the random walk represents an *ergodic* system, so if we take *ensemble* averages, i.e. considering a lot of random walkers and averaging their trajectories at the same time, the obtained results will coincide. To illustrate this point, I calculate the average distance over a lot of random walkers and plot its evolution over time in figure 2.4. This figure illustrates the point that if we take larger and larger ensembles of walkers, the average of their trajectory approaches the equilibrium position.

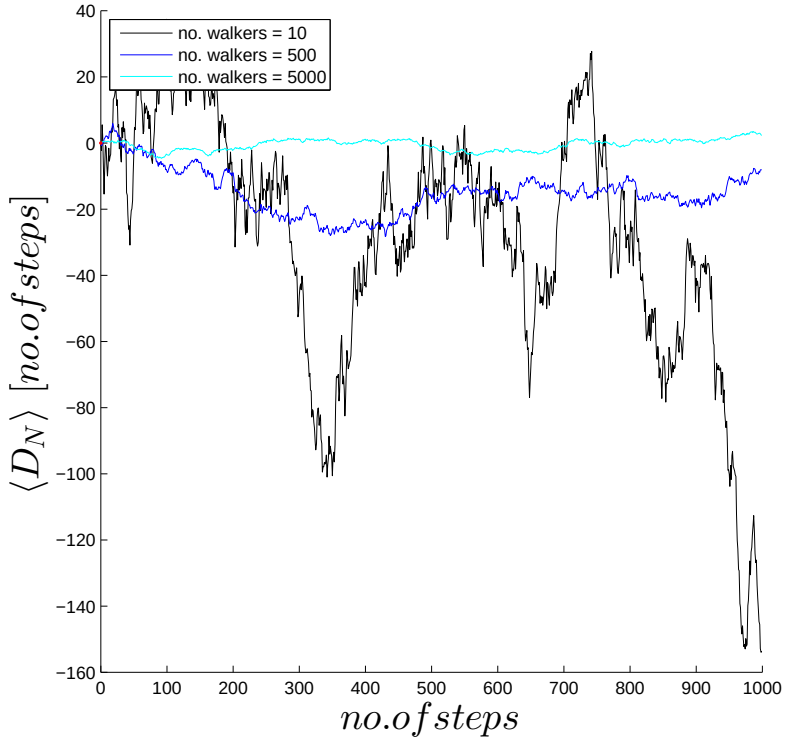

Figure 2.4: Average distance as a function of time when averaged over 10 (the black line), 500 (the blue line) and 5000 (cyan line). The average distance goes to zero as we average over larger ensembles. Due to the fact that the system is ergodic, averaging over ensembles gives the same results as averaged over time as shown in section 2.2.

2.3 Continuum limit

In the previous sections, I described the random walk and showed some of its properties by using individual trajectories and analyzing their behavior. However, another way to look at the random walk, is to describe the behavior in terms of the probability to find the walker at a certain position and analyzing how this probability evolves over time. Before outlining these properties, I first explain the finite difference method, that is used thoroughly throughout this thesis to simulate various motions.

2.3.1 Numerical simulations: The Finite difference method

In this method [24, 25], I use finite differences in order to approximate the derivatives in the stochastic equations. In particular, the following expression approximates the first derivative in backward difference:

$$f^{'}(x) \approx \frac{f(x) - f(x - \Delta x)}{\Delta x} \tag{2.8}$$

In order to approximate the second derivative, again in the backward direction, I use the following:

$$f^{"}(x) \approx \frac{f(x) - 2f(x - \Delta x) + f(x - 2\Delta x)}{(\Delta x)^2} \tag{2.9}$$

2.3.2 The Diffusion equation

Consider a random walker that has a probability p to take a step to the right and probability q to take a step to the left. Let $P_N(n)$ be the probability to find the walker at position $D_N = nL$ after time $t = Nt_0$. This random walk has the properties that the probabilities to go from one position to another, in this case p and q are constant in time, do not depend on the history of the system and obey the sum rule. Due to these properties, the following equation follows for the probabilities:

$$P_{N+1}(n) = pP_N(n-1) + qP_N(n+1) \tag{2.10}$$

This equation represents the following: If the particle is at position $D_{N+1}(n) = nL$ at time $t = (N+1)t_0$, then there is p chance that the particle came with a step to the right from $D_N(n-1) = (n-1)L$ at time $t = (N)t_0$ and there is q chance that the particle came with a step to the left from $D_N(n+1) = (n+1)L$ at time $t = (N)t_0$ as outlined in figure 2.5

From this point on, I will continue the derivation for the specialized case of unbiased random walk. i.e. $p = q = \frac{1}{2}$. However, the diffusion equation that will be obtained at the end holds in general as I show in the following chapters of this

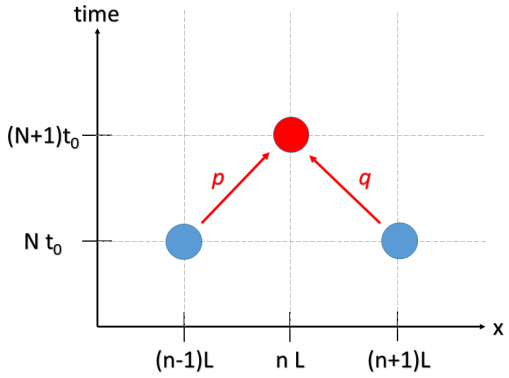

Figure 2.5: Probability transitions and sum rule. The particle can reach its final state (the red circle) from two possible positions that could occur 1 time step earlier (the blue circles). Therefore the total probability for the particle to be found at the red circle's position is the sum of the probabilities for the particle to be found at the positions of the left (right) blue circle and make a next step to the right (left).

thesis.

In this special case, the equation 2.10 becomes:

$$P_{N+1}(n) = \frac{1}{2}P_N(n-1) + \frac{1}{2}P_N(n+1) \tag{2.11}$$

Subtracting $P_N(n)$ from both sides yields

$$P_{N+1}(n) - P_N(n) = \frac{1}{2}(P_N(n-1) + P_N(n+1) - 2P_N(n)) \tag{2.12}$$

Consider the expression on the left. It gives the difference between the probabilities obtained at two consecutive times. In accordance to the discussion in section 2.3.1, in the limit when N is large (thus $t_0 \ll t$) this finite difference represents the time derivative of the probability distribution. In order to preserve the units, this has to be multiplied by t_0.

By going through the same arguments, the finite difference on the right represents the second derivative of the probability with respect to position, multiplied by L^2. Therefore, equation 2.12 turns into,

$$t_0\frac{\partial P}{\partial t} = \frac{1}{2}L^2\frac{\partial^2 P}{\partial x^2} \tag{2.13}$$

Finally, this can be rewritten as,

$$\frac{\partial P}{\partial t} = D\frac{\partial^2 P}{\partial x^2} \tag{2.14}$$

where D is the coefficient of diffusion given by $D = \frac{L^2}{2t_0}$. Equation 2.14 is known as **free diffusion equation that describes the evolution of the probability for a random walk**. For the case when all the particles start at the origin, the diffusion equation has the following solution,

$$P(x,t) = \frac{1}{\sqrt{4\pi\Delta t}} e^{-\frac{x^2}{4\Delta t}}; \tag{2.15}$$

where, $x = nL$ and $t = Nt_0$

2.4 Random walk as a stochastic process

Until now, in the 1D random walk that I considered, the walker was able to move left or right on a lattice with equal steps of length L. However, in the most general case, the lengths do not need to be same, in fact they can also be chosen to be random. Therefore, the random walk can be described as a stochastic process by the following simple free diffusion equation:

$$\frac{\partial x}{\partial t} = \xi(t) \tag{2.16}$$

$\xi(t)$ is called white noise and it is the term that gives the randomness of the system. Intuitively, this equation represents the straightforward interpretation of a random walk: the increments the walker takes over time are variables that are chosen at random.
In the following, I will first discuss the properties of the white noise and how to simulate it numerically, then I illustrate some of the properties discussed above by numerical simulations of random walk.

2.4.1 White noise properties and simulation

The *white noise* carries the following properties: [25]

1. The mean of the white noise is zero $\Rightarrow \langle \xi(t) \rangle = 0$ for all t

2. Two instances of the noise that occur at different times are independent of each other

3. $\langle \xi(t)^2 \rangle = 1$ for all (t)

In order to represent this noise within the finite difference approach, we need a discrete sequence of random numbers that will have the same properties as *white noise*, thus imitating its behavior. As argued in [25], due to property 1 and 2 described above, we require a sequence of uncorrelated random numbers that has a mean zero. As authors also point out, if we further require the condition $\frac{\langle (\xi(t)\Delta t)^2 \rangle}{\Delta t} = 1$, due to the fact that the power of $\xi(t)$ is unitary, the random number sequence should have a variance of $\frac{1}{\Delta t}$. Since the random number sequences generated by most programming languages are Gaussian sequences with zero mean and unit variance, we need to scale down the random number sequence by dividing it by $\sqrt{\Delta t}$ to obtain the correct variance.
Finally, the white noise term $\xi(t)$ is described in the finite difference approach as the

$$\xi(t) = \frac{\xi_i}{\sqrt{\Delta t}} \tag{2.17}$$

where ξ_i is a sequence of random numbers generated by any programming language. One particular instance of the white noise is shown in figure 2.6

2.4.2 Numerical simulation of general random walk

In view of the discussion above about the treatment of white noise and the derivative within the finite difference approach in section 2.3.1, equation 2.16 takes the following form:

$$\frac{x_{i+1} - x_{i+1}}{\Delta t} = \frac{\xi(t)}{\sqrt{\Delta t}} \Rightarrow x_{i+1} = x_i + \sqrt{(\Delta t)}\xi_i \tag{2.18}$$

If we would like to consider random walk in 2 or 3 dimensions, it is sufficient to notice that the motion along different Cartesian coordinates can be simulated independently. Therefore, the set of equations that simulate 2D random walk are

$$x_{i+1} = x_i + \sqrt{(\Delta t)}\xi_{x_i} \tag{2.19}$$

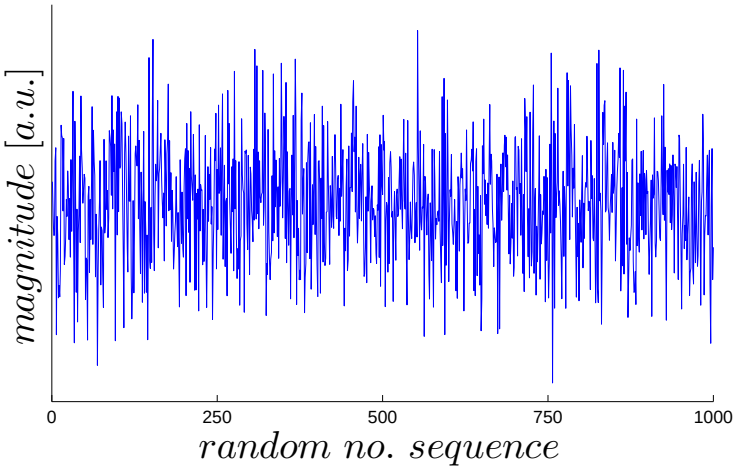

Figure 2.6: One particular occurrence of white noise simulated by using a sequence of 1000 random numbers generated by MATLAB scripting language with a time step of $\Delta t = 0.5$

$$y_{i+1} = y_i + \sqrt{(\Delta t)}\xi_{y_i} \tag{2.20}$$

where ξ_{x_i} and ξ_{y_i} are mutually independent random noises.

In order to illustrate the previous discussions, I plot a single instance of the random walk in figure 2.7. Parts (a-c) show the particular realizations of the white noise needed to simulate the random walks in (d-f) in one and (h-j) in two dimensions respectively. All trajectories are 10s long and we observe that as the time step is decreased, the random walk becomes more jagged and we need a larger magnitude of the white noise to simulate the motion. However, due to the fact that all random walks look similar to each other at all scales, this simulation does not depend on the time step chosen and there is no optimal time step for this free diffusion equation. What is more, random walks in x and y direction in figure 2.7 (d-f) are obtained by separating the two components of the two dimensional motion shown in (h-j) thus demonstrating that the Cartesian components of two and three dimensional random walks behave independently of each other.

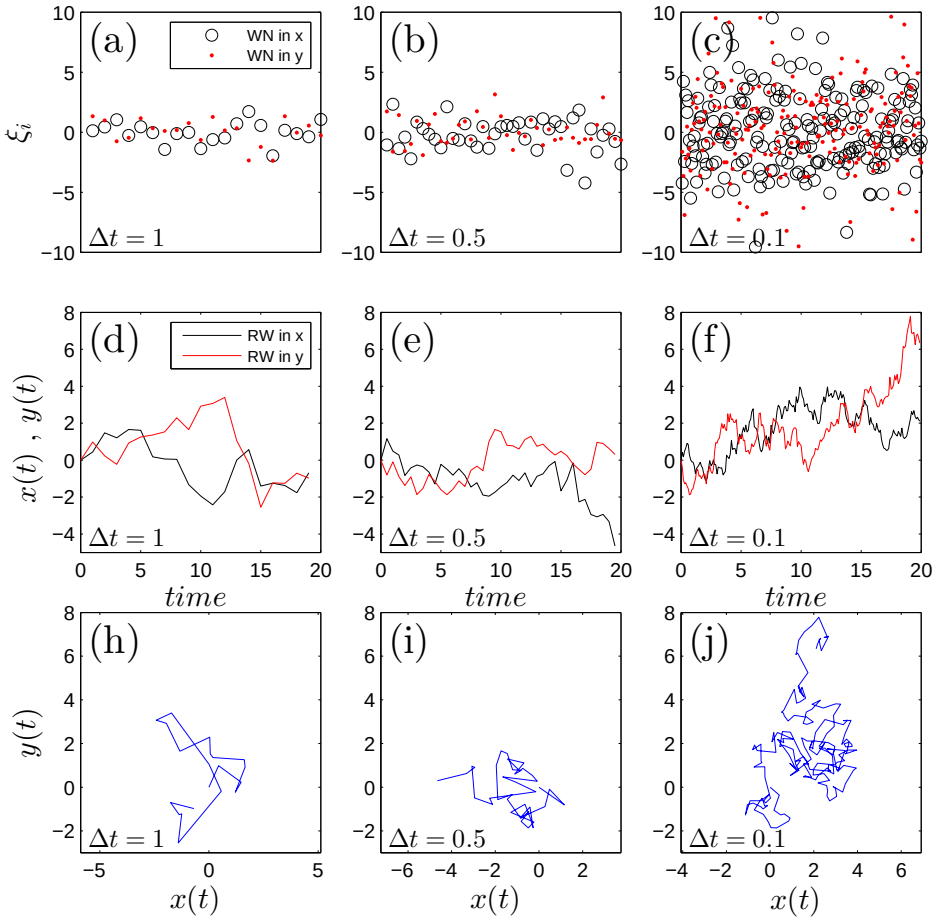

Figure 2.7: Simulation of one particular realization of random walk with trajectories that are 10s long.As the time step decreases the trajectories look more ragged and white noise of larger magnitude is needed to replicate the motion. (a-c) The instances of the white noise needed to simulate the one dimensional (d-f) and two dimensional (h-j) random walk. The trajectories in (h-j) are constructed by concatenating both one dimensional trajectories in (d-f) showing that in two dimensional random walk, each of the x and y components performs an independent random walk itself.

Chapter 3

Brownian Motion

Brownian motion is the everlasting zigzag motion of the entities present at the microscale. From its earliest observation in the early 19^{th} century, Brownian motion has constantly gained much attention from scientists all around the world. The complete formulation of its theory was done in the early 20^{th} century and from that point on, the many studies of this motion resulted in its widespread application. In this chapter, after going through a very brief historical overview, I present an outline of the theory of Brownian motion as developed by Einstein and Langevin. Finally, I explain how to simulate this motion by using the finite difference method explained in Chapter 2 and demonstrate some of its properties by numerical simulations. As the generally accepted picture of the Brownian motion is the one of a microscopic colloidal particle performing random motion due to the collisions with the surrounding molecules, there will be many similarities between the concepts presented here and the ones presented in Chapter 2.

3.1 History

Robert Brown was a botanist who first observed the irregular motion of the pollen grains when suspended in water. He thought that this motion was due to the fact that the grains were living and he was looking for the *"vital forces"*,

interpreting his observations as a motion of small living creatures. However, after experimenting with a vast range of plants of different age as well as inorganic grains, he concluded that the motion was an intrinsic property of the microscopic entities, not connected with biology, rather with physics itself.

Afterwards, around 1880s, a series of experiments done by Leon Gouy showed that the motion cannot be due to external forces, but is a property of the fluid itself. Then, the Brownian motion gained its theoretical interpretations. It all started with the works of Einstein who derived the expressions for the diffusion constant and mean-squared displacement of a suspended particle. Then, Smoluchowski obtained the same results as Einstein but with a different numerical coefficient and finally, Langevin showed a mistake in Smoluchowski's assumptions and provided a much simpler derivation of the diffusion coefficient. These results provided a good framework for Jean Perrin to make his careful, precise and systematic experiments that showed the first quantitative observations of the Brownian motion. [26]

3.2 Basic properties

Figure 3.1 below shows three trajectories of Brownian particles recorded by Perrin. He was tracking small granules (the radius of the granules being $0.52\mu m$) suspended in water by recording the positions of the granules every 30 seconds and then connecting the positions (black dots in the figure) by straight line segments in what is one of the earliest experimental systematic study of the Brownian motion.

All careful work on Brownian motion led to the conclusion that the following main properties hold:

- The motion consists of straight translations and rotations at random angles. It is extremely irregular and the trajectory does not have a tangent (the velocity of the motion is not well defined).
- Even when many particles approach each other very closely, they still do

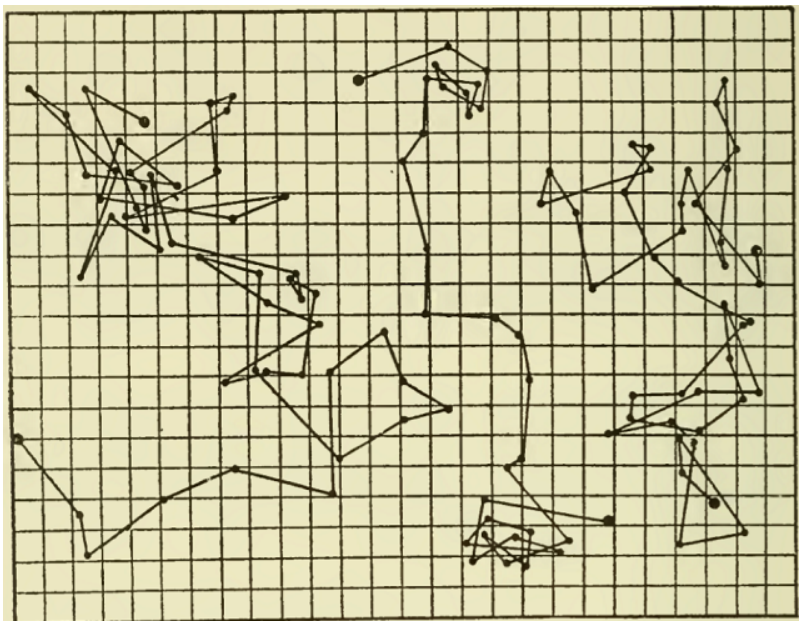

Figure 3.1: Three independent trajectories of the Brownian motion of granules with radius $0.52\mu m$ as recorded in 30s intervals by Jean Perrin in one of the earliest systematic experimental works to measure the properties of Brownian motion. Adapted from [27]

.

not affect each other and move independently.

- The motion becomes more vigorous as: the particles get smaller, the fluid gets less viscous and the temperature increases.
- The composition of the particles has no effect on the motion.
- The motion is omnipresent at microscale and does not ever stop.

3.3 Einstein's theory of Brownian motion

Einstein's development of the theory consists of two main results. The first one connects the diffusion coefficient to the other physical parameters describing the system, while the second result is the derivation of the free diffusion equation which describes the evolution of the probability distribution of a particle. In the following, I will go over the main points of both derivations, using the same

notation as used in [3]. All the following arguments are derived by using 1D systems, however the extension to 2D and 3D is straightforward.

3.3.1 Diffusion constant

According to Einstein's view, the suspended colloids in a solution perform random walk due to the random impulses these colloids experience because of numerous collisions with the solvent's molecules. He also asserts that the suspended particles and the surrounding molecules are indistinguishable with respect to the osmotic pressure, expressed as

$$p = k_B T \nu \tag{3.1}$$

where ν represents the number of suspended particles in unit volume. The argument runs along the following lines: Let ν amount of particles be suspended in a liquid in equilibrium. Furthermore, assume that they are acted upon some external force K (the origin of this external force need not be specified) and the system is in equilibrium. In this configuration, the external force is balanced by the force due to the osmotic pressure of the suspension. Therefore,

$$K\nu - \frac{\partial p}{\partial x} = 0 \tag{3.2}$$

From equation 3.1, substituting for the pressure and taking into account that only the quantity ν depends on x,

$$K\nu - k_B T \frac{\partial \nu}{\partial x} = 0 \tag{3.3}$$

The next step is to consider unit area and calculate how many particles pass through that area in unit time due to the motion of the suspended colloids. This motion can be analyzed as an interplay between two processes that occur in opposite directions: the external force K pushing a single particle and imparting a certain velocity to it opposed by a process of diffusion produced by thermal molecular movement.

According to Stokes' law, if the suspended particles are spheres with radii R in a liquid with viscosity η, upon the influence of the force K each particle gains

a velocity $\frac{K}{6\pi\eta R}$. Therefore, a total of $\frac{\nu K}{6\pi\eta R}$ particles pass the unit area in unit time. In addition, if we denote the coefficient of diffusion by D, then a total of $-D\frac{\partial\nu}{\partial x}$ particles pass through the same are in unit time. Since the particles are at equilibrium,

$$\frac{\nu K}{6\pi\eta R} + (-D\frac{\partial\nu}{\partial x}) = 0 \tag{3.4}$$

Substituting the value for νK from equation 3.3, equation 3.4 becomes

$$(\frac{k_B T}{6\pi\eta R} - D)\frac{\partial\nu}{\partial x} = 0 \tag{3.5}$$

Then, the diffusion constant is obtained as

$$D = \frac{k_B T}{6\pi\eta R} \tag{3.6}$$

which is the first main result derived originally by Einstein in ref. [3].

3.3.2 Probability distribution

Let us assume that the probability of a particle to be at a position x at time t is given by $f(x,t)$. Then, in the subsequent very short time interval τ, suppose the particle takes a very short step of magnitude Δ. There will be a certain probability law that will hold for Δ, $\phi(\Delta)$, representing the probability of the jump with specific length Δ happening. In terms of ϕ, the total probability is expressed as,

$$\int_{-\infty}^{+\infty} \phi(\Delta) = 1 \tag{3.7}$$

In addition, there will be equal probabilities of going left and right, so $\phi(\Delta) = \phi(-\Delta)$ will hold.

Then, the aim is to calculate the number of particles that are located between x and $x + dx$ at time $t + \tau$.

$$f(x, t+\tau)dx = \int f(x+\Delta)\phi(\Delta)d\Delta \tag{3.8}$$

Since we consider a very small increment of time τ, we can expand the left hand side to first order,

$$f(x, t+\tau) = f(x,t) + \tau\frac{\partial f(x,t)}{\partial t} \tag{3.9}$$

Similarly, because Δ is also very small,

$$f(x+\Delta,t) = f(x,t) + \Delta\frac{\partial f(x,t)}{\partial x} + \frac{\Delta^2}{2!}\frac{\partial^2 f(x,t)}{\partial x^2}\dots \tag{3.10}$$

Bringing this expression under the integral (bearing in mind that $f(x,t)$ does not depend on Δ) and combining the two equations,

$$\begin{aligned} f(x,t) + \tau\frac{\partial f(x,t)}{\partial t} = & \\ & f(x,t)\int \phi(\Delta)d\Delta + \frac{\partial f(x,t)}{\partial x}\int \Delta\phi(\Delta)d\Delta \\ & + \frac{\partial^2 f(x,t)}{\partial x^2}\int \frac{\Delta^2}{2}\phi(\Delta)d\Delta\dots \end{aligned} \tag{3.11}$$

Consider the right hand side: The second, fourth, sixth and all consecutive terms will vanish because of the property that the probability law is an even function. The fifth, seventh and next terms are ignored because of the high powers of Δ, and the integral in the first term is equal to 1. Therefore, identifying the diffusion coefficient as the second moment of the probability law, in other words the variance,

$$D = \frac{1}{\tau}\int_{-\infty}^{+\infty}\frac{\Delta^2}{2}\phi(\Delta)d\Delta \tag{3.12}$$

the diffusion equation is obtained,

$$\frac{\partial f(x,t)}{\partial t} = D\frac{\partial^2 f(x,t)}{\partial x^2} \tag{3.13}$$

In the discussion until now, all of the particles' positions were taken relative to a single coordinate system. However, due to the randomness and independence of the particle trajectories from each other, Einstein argues that it is possible to consider every particle as an independent system whose origin corresponds to the center of the particle initially, at $t = 0$. In that case, the solution of the diffusion equation is expressed as,

$$f(x,t) = \frac{n}{\sqrt{4\pi D}}\frac{e^{-\frac{x^2}{4Dt}}}{\sqrt{t}} \tag{3.14}$$

where n is the total number of particles. From here, we are able to read off the mean square displacement of the motion as

$$\langle x^2\rangle = 2Dt \tag{3.15}$$

3.3.3 Fokker-Planck equation

The previously derived free diffusion equation is the equation of "motion" of the probability distribution of the Brownian motion. The generalization of this equation in the case in which there is also an external force acting on the particle is known as Fokker-Planck equation. This equation can be derived by introducing the *diffusion current*, which satisfies the continuity relation and is expressed as,

$$J_{diff}(x,t) = -D\frac{\partial f(x,t)}{\partial x} \tag{3.16}$$

If there is any external force applied to the particle,$F(x,t)$, the velocity of the particle can be written as $v(x,t) = \frac{F(x,t)}{\gamma}$, γ being the friction coefficient of the suspended particle in the liquid. The total current of particles can be written as

$$J(x,t) = J_{diff}(x,t) + J_{ext}(x,t) = -D\frac{\partial f(x,t)}{\partial x} + v(x,t)f(x,t) \tag{3.17}$$

Then, by taking one position derivative of the right side and invoking the continuity relation, the Fokker-Planck equation is obtained:

$$\frac{\partial f(x,t)}{\partial t} = \frac{\partial F(x,t)p(x,t)}{\partial x} - D\frac{\partial^2 f(x,t)}{\partial x^2} \tag{3.18}$$

3.4 Langevin description of Brownian motion

As seen from the previous section, there are 2 ways to describe the Brownian motion: in terms of probability density distribution of an ensemble of particles that evolves over time (its evolution is given by Fokker-Plank equation) or as a stochastic trajectory of a single particle. Both approaches are intrinsically connected, since the probability can be obtained by averaging over many trajectories, or on the other hand, the statistical properties of the random forces depend strongly on the probability density distributions. Einstein's derivation of the diffusion equation and its connection to the diffusion constant are connected with the 1st approach. The Langevin approach is concerned with the derivation of an equation whose solution describes a stochastic trajectory of a single Brownian swimmer. In this section I outline the derivation of the Langevin equation and present a way to solve it numerically, via the method of finite differences.

3.4.1 Langevin equation

Consider a spherical particle with radius R and velocity v suspended in a liquid with a viscosity η. The motion of the particle can be described by the Newton's equation:

$$m\frac{\partial^2 x}{\partial t^2} = F_{total}(t) \tag{3.19}$$

The force on the right side represents the total force due to all external interactions that the particle feels at time t. Therefore, by virtue of knowing this force as a function of time exactly, the motion of the particle would be completely deterministic. However, owing to the fact that the force is due to collisions with a large number of molecules present in the liquid, a closed from of this force is unknown. Instead, the force is generally broken down into components that model and represent the effects the particle feels. First, as the particle moves through the liquid, there is a frictional force opposing its motion. This force is given by $-\gamma v$, where γ is the friction coefficient given by Stokes' law, $\gamma = 6\pi\eta R$. So, the equation now takes the form,

$$m\frac{\partial^2 x}{\partial t^2} = -\gamma v(t) \tag{3.20}$$

The solution to this equation gives an exponentially decaying velocity. However, the actual velocity of the particle cannot remain at zero, so it follows that the friction force is not the only contribution. Rather, another term in the shape of random or fluctuating force is added to model the random collisions with the molecules of the solvent and the equation attains the following form, [28]

$$m\ddot{x} = -\gamma\dot{x} + \sqrt{2k_B T\gamma}\xi(t) \tag{3.21}$$

where $\xi(t)$ again represents a white noise.
This equation represents the Langevin equation to describe the Brownian motion. Note that, both the fluctuation and friction term in the equation originate from the interactions of the Brownian particle with its environment, as such they are closely connected by the Einstein relation for diffusion $D = \frac{k_B T}{\gamma}$.

3.4.2 Numerical Solution

The finite difference method can be readily applied to solve the Langevin equation numerically. By using the definitions of the derivatives explained in section 2.3.1 and treating the white noise as explained in section 2.4.1, the Langevin equation takes the following discretized form:

$$m(\frac{x_i - 2x_{i-1} + x_{i-2}}{(\Delta t)^2}) = -\gamma(\frac{x_i - x_{i-1}}{\Delta t}) + \sqrt{2k_B T\gamma}\frac{1}{\sqrt{\Delta t}}\xi_i \tag{3.22}$$

Solving this algebraically to obtain the solution for x_i, one obtains [25]

$$x_i = \frac{2 + \Delta t(\frac{\gamma}{m})}{1 + \Delta t(\frac{\gamma}{m})}x_{i-1} - \frac{1}{1 + \Delta t(\frac{\gamma}{m})}x_{i-2} + \frac{\sqrt{2k_B T\gamma}}{m[1 + \Delta t(\frac{\gamma}{m})]}(\Delta t)^{\frac{3}{2}}\xi_i \tag{3.23}$$

This way of writing this equation places emphasis on the ratio $\frac{\gamma}{m}$. It has units of $\frac{Ns}{m^2 kg} = s^{-1}$. Therefore, the time $\tau = \frac{m}{\gamma}$ is the characteristic time scale of the equation (momentum relaxation time). It means that one needs to be particularly careful when choosing the time step of discrete simulation Δt. It has to be small with respect to the total observation time, nevertheless it has to be chosen sufficiently larger than τ, so that two succeeding updates of the position can be considered independent of each other. This is in contrast to the random walk discussed in chapter 2, due to the fact that there is not a particular time scale of the random walker considered in that chapter.

3.4.3 Inertial vs Non-inertial solution

Reynold's number is defined as the ratio between inertial and viscous forces and its value is an indication of how important the inertial forces are in the medium. The value can be approximated as $\frac{Rv\rho}{\eta}$, where ρ, η are fluid density and viscosity respectively and R is the radius of the particle moving with velocity v. For the systems that I consider throughout this work, the Reynolds number is very low [29]. Therefore, for these kinds of swimmers the inertia plays a negligible role, friction forces are overwhelmingly dominant. In addition, since in typical experiments the time scale is much larger than the τ defined above and the instantaneous velocity and the ballistic regime are not probed, [25] the inertia

term can be dropped. Therefore, a good approximation to equation 3.21 is the following version of Langevin equation

$$\dot{x} = \sqrt{2D}\xi(t) \tag{3.24}$$

which has a simpler numerical solution,

$$x_i = x_{i-1} + \sqrt{2D\Delta t}\xi_i \tag{3.25}$$

An extensive discussion of the validity of the non-inertial approximation vs the inertial solution is done in [25] and their conclusions are summarized in figure 3.2. In figures 3.2(a) and 3.2(b) the authors simulate 1D Brownian motion solving both inertial, eq. 3.22 and non-inertial, eq. 3.25 solutions and plot them as functions of time. In figure 3.2(a) the time steps used in the simulations are much smaller than the characteristic time of the motion and therefore we can observe a big difference in both trajectories. However, when the time steps are large enough, as in figure 3.2(b), both trajectories look the same. In these cases, the microscopic details are not observable, therefore the effect of inertia is not noticeable and both trajectories look jagged.

In figure 3.2(c) plots of the velocity auto-correlation function are presented, which give information on how the velocity of the particle at time $t^{'}$ influences the velocity of the particle at some later time $t + t^{'}$ and is calculated as

$$C_v(t) = \overline{v(t')v(t+t')} \tag{3.26}$$

We can observe that while the correlation function of the velocity decays to zero with some time scale in the case of the inertial solution, it goes immediately down to zero when we apply the non-inertial solution demonstrating that it does not have a characteristic time scale in this case. Finally, the authors plot the mean square displacement for both cases. Theoretically, it is expected for ballistic motion to have an MSD proportional to t^2 and for diffusive motion MSD is proportional to t. The plot in figure 3.2(d) demonstrates that both solutions have the same MSD for sufficiently long times, therefore making them equivalent in this limit.

Due to the discussions above, in the following chapters I will always set $m \approx 0$ and solve the non-inertial version of the Langevin equation.

3.5 Numerical solution of the free diffusion equation

The free diffusion equation, 3.13, can be solved numerically by using the method of finite differences. Instead of solving for $f(x,t)$, I solve for the discrete series $f(x_n,t_n)$ therefore creating a grid that discretizes the position and time independently. In this solution, the index referring to position is denoted with n and the one denoting the time with m.
Approximate the first time derivative in the forward direction:

$$\frac{\partial f(x,t)}{\partial t}|_{x_n,t_m} \approx \frac{f_{n,m+1}-f_{n,m}}{\Delta t} \tag{3.27}$$

To approximate the second derivative of the position, I use the backward derivative:

$$\frac{\partial f(x,t)^2}{\partial^2 x}|_{x_n,t_m} \approx \frac{f_{n,m}-2f_{n-1,m}+f_{n-2,m}}{(\Delta x)^2} \tag{3.28}$$

After rearranging the terms, I obtain for the probability distribution

$$f_{n,m+1} = f_{n,m} + \frac{D\times\Delta t}{(\Delta x)^2}(f_{n,m}+2f_{n-1,m}+f_{n-2,m}) \tag{3.29}$$

The unit for the diffusion constant is m^2s^{-1}. In the figure 3.3 I plot the probability distribution at different times and use $D = 0.5m^2s^{-1}$ for the diffusion coefficient. Note that for the Brownian motion of the molecules, the translational diffusion coefficient is of the order $\approx 1\ \mu^2s^{-1}$. A comparison between equations 3.25 and 2.18 we see that the value used in this calculation corresponds to macroscopic random walk. In the figure we observe that starting the particles from very sharp Gaussian distribution centered at the origin will result to the flattening of the probability distribution for longer times and asymptotically it becomes uniform (over the region in position that is simulated in the grid. In this case it is the region between $x = -0.5$ and $x = 0.5$ m). Therefore, a Brownian particle moving for an infinite amount of time, goes over every point in space.

3.6 Brownian motion in 2D square well

In order to further illustrate this probability distribution behavior, I confine a Brownian particle in a two dimensional square well with reflective boundary conditions. The method for treating the boundary conditions is explained in section 6.2. In figure 3.4 we see the time evolution of the probability distribution of an ensemble of 100 particles randomly distributed around the origin at $t = 0s$. We see that while in the beginning most of the particles are stacked near the origin, after 1000000 seconds the probability to find a particle inside any position in the well is uniform. Although these results were calculated using ensembles of 100 particles, due to the *ergodicity* of the system, the same results could be obtained by calculating the probability due to a single very long trajectory.

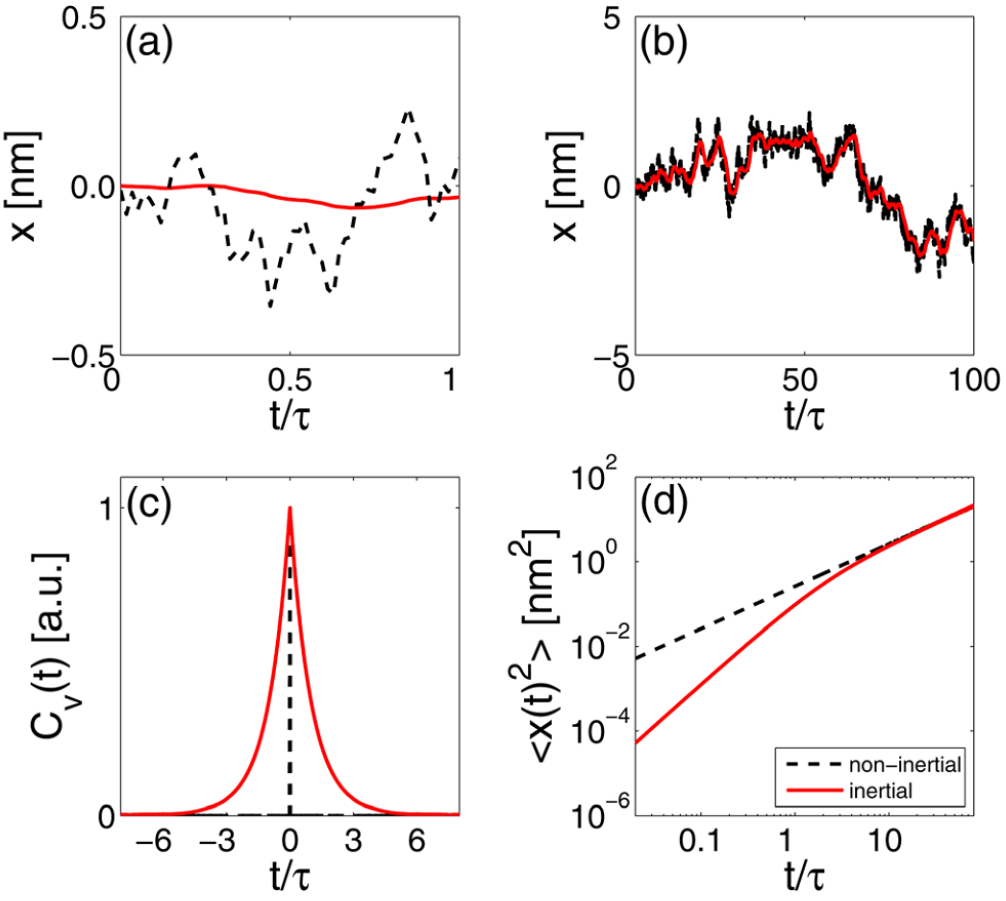

Figure 3.2: Comparison between the inertial and non-inertial regime. The inertial and non-inertial solutions are plotted for small times in (a) which leads to difference in their behavior. The difference vanishes in the limit of larger time steps, as illustrated in (b). (c) Plot of velocity auto-correlation function in both cases. The fact that the values of this function drop to zero immediately show that velocity has no characteristic time scales. (d) Plot of the MSD in both cases which shows agreement between the two when long time steps are used. Reproduced from [25].

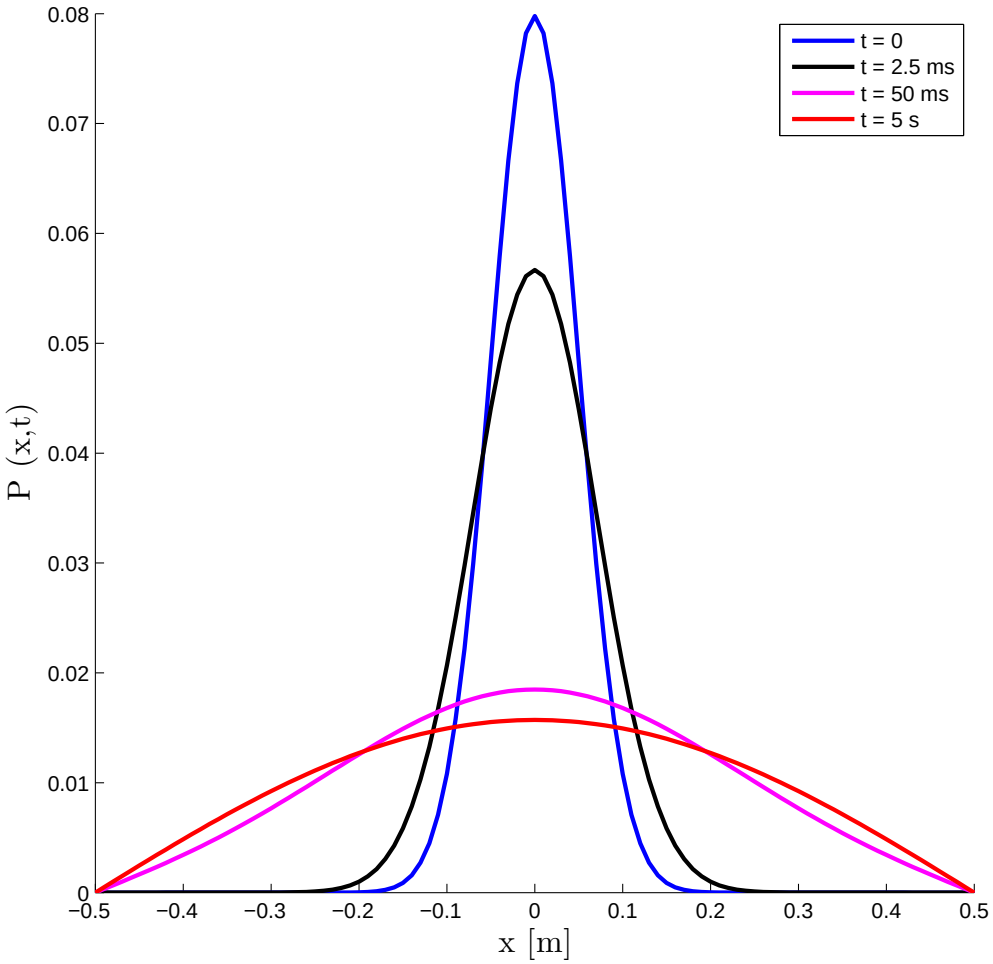

Figure 3.3: Probability distribution as a function of position in the range [-0.5m,0.5m] for different times. The initial distribution of the particles is a sharply peaked Gaussian centered at the origin. As the time passes, the probability distribution becomes flatter, asymptotically becoming uniform.

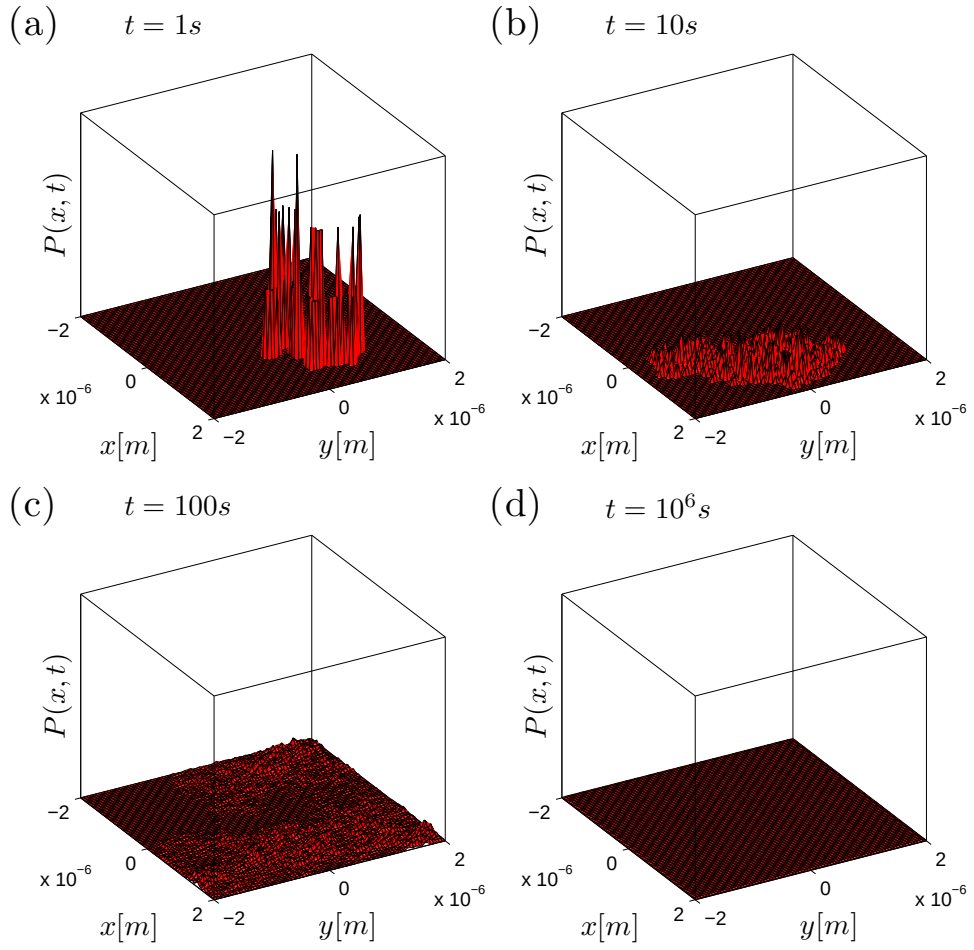

Figure 3.4: Evolution of probability distribution averaged over 100 particles in time in square well with side length $4\mu m$. Initially, the particle position is randomized near the origin. As the time passes, the probability distribution spreads out, becoming uniform after 1000000s. The radius of all particles is 1 μm.

Chapter 4

Active swimmers

Active swimmers, or microswimmers, are able to convert the energy they pick up from the environment into kinetic energy, resulting in a directed motion. The range of agents that can be considered active is vast, from insects and birds on the macroscale, to flagellated bacteria or sperm cells in the microscale. Due to their potential for many applications, many artificial microswimmers were devised (for example, Janus particles) which are able to propel themselves by various mechanisms. Another reason to study these microswimmers is the fact that they can serve as a model system for out of equilibrium phenomena and thus, drawing a comparison between the systems outlined in this and previous chapter can give a great insight of these phenomena [4]. In here, I first go over some examples of the active motion, starting from the flagellated bacteria to the artificially engineered Janus particles, then I describe the model I use to simulate the active motion before finally I present the numerical solution and illustrate some properties of the motion by simulations.

4.1 Examples of Active motion

There are two main examples of active motion at the microscale: the run-and-tumble motion which is a property of flagellated bacteria and active Brownian

motion exhibited by the artificially designed microswimmers. In this section I briefly go over the different mechanisms of motion citing some examples of both.

4.1.1 Run-and-tumble of the bacteria

A particular trajectory taken by H. Berg [19] while studying the movement E. *coli* in gradients is shown in figure 4.1(a).

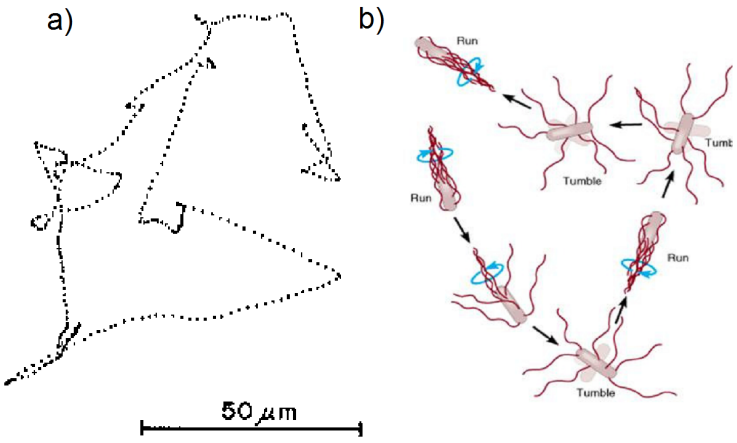

Figure 4.1: (a) An example of the run-and-tumble motion of *Escherichia coli*. The dots denote time steps of 0.1 second. Adapted from [19] (b) An illustration of the behavior of the flagella of E.coli. During a run, they rotate together in counter-clockwise direction and propel the bacteriaum. Once in a while, the flagella change their direction of rotation and they start working out of synch, therefore tumbling the bacterum. Adapted from [30]

The E. *coli* bacterium moves by virtue of a bundle of flagella that rotate together in the counter-clockwise direction while doing the "runs". To conserve angular momentum, the body of bacterium rotates in the opposite direction and moves through space. However, once in a while, the flagella turn to rotate in the opposite direction, they stop working together and the bacteria "tumbles". The main property of the bacterium is that E. *coli* can sense the environment around itself. Therefore it can go around and change the length of the runs (in other words, the frequency of tumbling) depending on whether the run is directed to more favorable conditions or not. Therefore, as it can be seen in figure 4.1, all the runs of the bacterium trajectory are not equal in length.

4.1.2 Active Brownian motion due to Janus particles

The artificially engineered active Brownian motion is generally made possible by the production and use of Janus particles. Their name comes from Janus, the two faced Roman god, and they are particles which have two or more distinctive physical properties as shown schematically in figure 4.2(a).

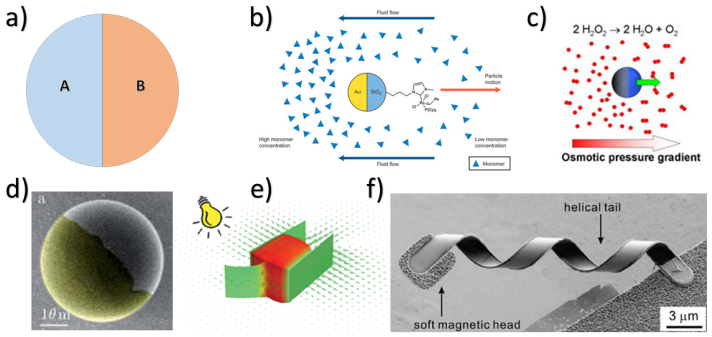

Figure 4.2: (a) A schematic representation of Janus particles. Parts A and B have different physical and chemical properties and therefore can react differently to outside stimuli. (b-c) Methods of propulsion using Janus particles inside a gradient [31, 32] (d) A particular type of Janus particle, half silica half gold. Adapted from [33] (e-f) Different type of self-propulsion including responsive gel body and artificial flagella [34, 35]

.

Therefore, since the parts A and B have different properties, they can react differently in a specifically prepared medium, therefore creating gradient that is able to push the particles in some direction, as is the case in 4.2(b) and (c). Apart from creating gradients, many researches have found propulsion mechanisms using artificial flagella, shown in 4.2(f) or creating a periodically shrinking and expansive gel body, 4.2(e). An extensive overview of the many different propulsion mechanisms can be found in [9]. In the next section I briefly explain a particular method that is using light to tune the motion of the swimmer.

4.1.3 Active motion tunable by light

This method was studied in detail in [36] and it uses silica particles half-coated with a gold cap, figure 4.2(d), placed in a critical mixture of water and lutidine. The mixture is kept at a temperature which is very near to the critical temperature of the mixture. Therefore, when light shines upon the particle the two sides of the particle get heated up differently. This results in a local demixing of the critical mixture, therefore creating gradient that propels the particle. The whole process is summarized in figure 4.3.

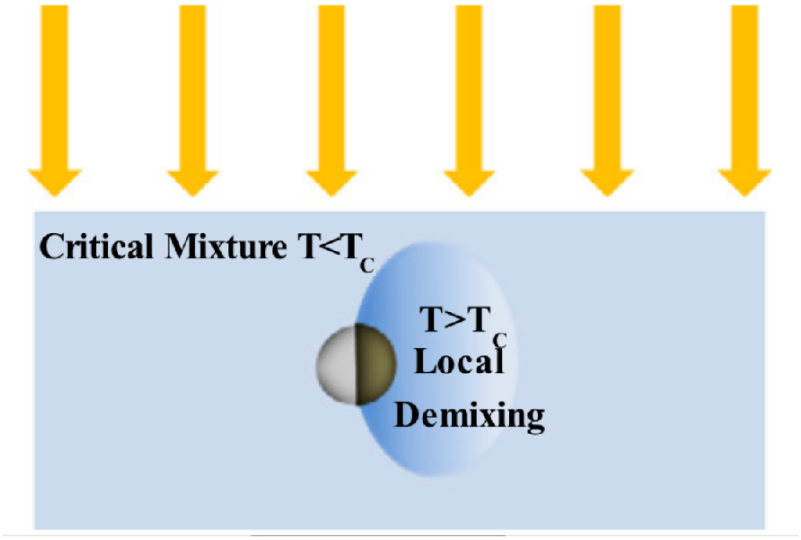

Figure 4.3: Active Brownian motion tunable by light. A half silica half gold coated particle is placed in a critical mixture. Upon shining of light onto it, there is local demixing, a gradient is created and the particle can be propelled. Adapted from [36]

.

Here it is important to note that the orientation of the particle is due to the rotational diffusion and has no effect of the propulsion of the motion. Therefore, the frequency of runs of the particle is connected to the characteristic time scale of the system, irrespective of the external force or environment the particle is in.

4.2 Model

Active Brownian motion can be viewed as an interplay between Brownian fluctuations, both in the translational and rotational direction, and a self-propelling force that results in a velocity of propulsion v, assumed to be constant [4]. To be more precise, the position of the active swimmers undergoes a translational Brownian diffusion with a coefficient,

$$D_T = \frac{k_B T}{6\pi\eta R} \tag{4.1}$$

On the other hand, the orientation of the particle is characterized by angle $\phi(t)$ which performs rotational diffusion with rotational diffusion coefficient,

$$D_R = \frac{k_B T}{8\pi\eta R^3} \tag{4.2}$$

At the same time, $\phi(t)$ specifies the direction of the velocity of the motion as illustrated in figure 4.4(a). Since the examples of microswimmers that will be considered further in this text, move within the low Reynolds number regime, in writing these equation I drop the inertial term. Therefore, the set of Langevin equations describing the motion of an active microswimmer are the following,

$$\frac{d}{dt}\phi(t) = \sqrt{2D_R}\xi_\phi \tag{4.3}$$

$$\frac{d}{dt}x(t) = v\cos\phi(t) + \sqrt{2D_T}\xi_x \tag{4.4}$$

$$\frac{d}{dt}y(t) = v\sin\phi(t) + \sqrt{2D_T}\xi_y \tag{4.5}$$

In these equations, ξ_ϕ, ξ_x, ξ_y are independent white noise terms.

4.3 Numerical simulations

These equations are solved by using the finite difference methods explained in section 2.3.1. By making the appropriate discretized version of the derivatives and the white noise terms, the above equations take the following form:

$$\phi_i = \phi_{i-1} + \sqrt{2D_R\Delta t}\xi_{\phi,i} \tag{4.6}$$

$$x_i = x_{i-1} + v\cos\phi_i\Delta t + \sqrt{2D_T\Delta t}\xi_{x,i} \tag{4.7}$$

$$y_i = y_{i-1} + v\sin\phi_i\Delta t + \sqrt{2D_T\Delta t}\xi_{y,i} \tag{4.8}$$

In figure 4.4 I plot 10s trajectories for different parameters of active particles in order to demonstrate how the motion changes depending on the propulsion velocity of the particles as well as their radius. It can be seen that as the particles become smaller, the trajectories become more and more similar to the ones for passive Brownian particles. This is due to the fact that the rotational diffusion scales down as R^{-3} and for smaller particles the rotational diffusion plays a dominant role over the propulsion. On the contrary, as expected, the increase in the velocity leads to trajectories well spread and particles are able to travel over longer distances. The MSD of this motion is quadratic with respect to time in the short time scales, but then becomes linear with time on the long time scales with enhanced diffusion coefficient.

4.4 Active Brownian swimmers in square well

In order to draw a comparison between the active and passive motion, I plot the active Brownian motion of a particle in the same well used to calculate the probability shown in figure 3.4. Again I use ensemble of 100 particles with the same size, in this case they are propelled with velocity of $v = 1\mu m$ for figure 4.5 and velocity of $v = 50\mu m$ for figure 4.6. While we were observing uniform probability distribution in the case for the Brownian motion, the active particles tend to spend some more time near the walls of the well. In particular, the time they spend there increases as the velocity of propulsion of the active particles increases.

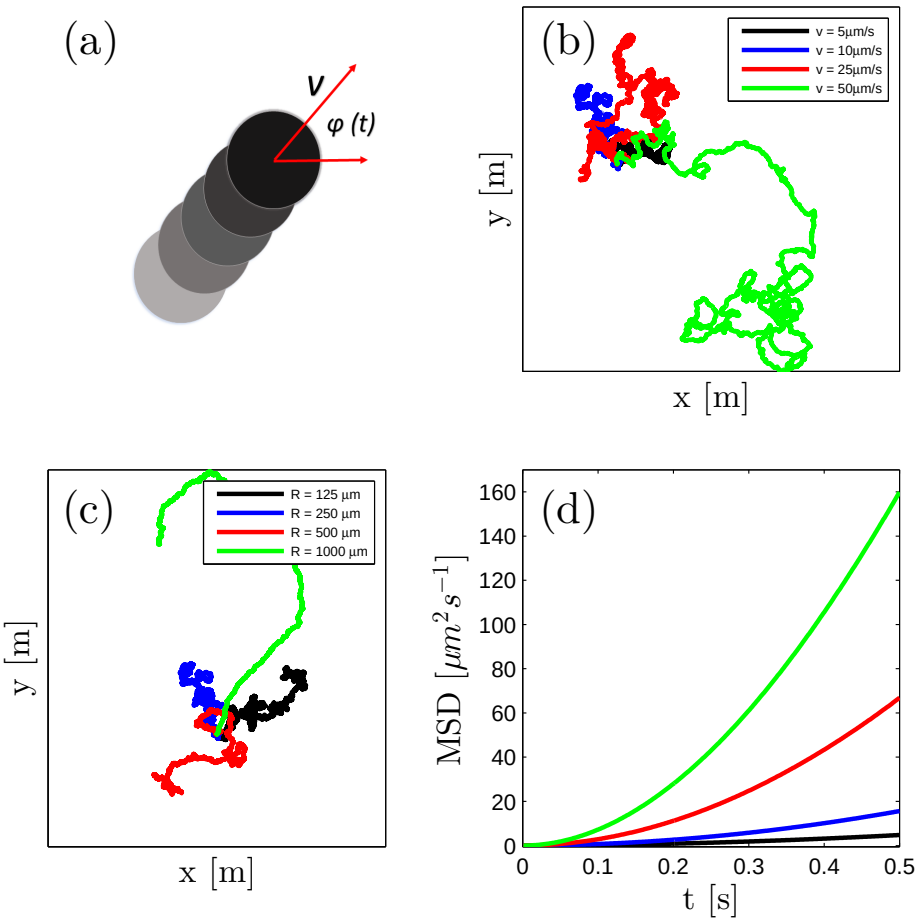

Figure 4.4: (a) The angle between the velocity and x axis is given by $\phi(t)$ and it performs rotational Brownian diffusion therefore reorienting the particle as it is propelled through space with velocity v. (b) Active Brownian motion as a function of the velocity for particles of radius $R = 250nm$ (c) Active Brownian motion as a function of the radius of the particles for velocity $v = 10\mu m/s$. (d) MSD for active particles of radius $R = 1000nm$ as a function of velocity. All velocities in (d) are analogous to the ones in part (b). All trajectories are 10s long.

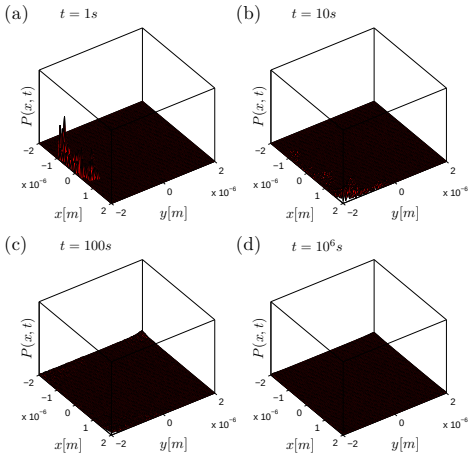

Figure 4.5: Evolution of probability distribution averaged over 100 active particles in time in square well with side length $4\mu m$. The radius of all particles is 1 μm and their velocity is $1\mu m/s$. As the time passes, the probability distribution spreads out and eventually becomes uniform except at the walls, where particles tend to spend some more time.

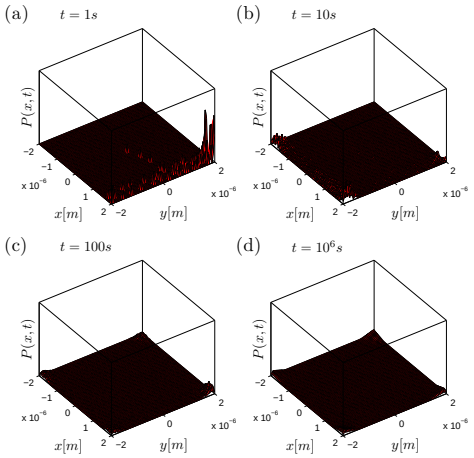

Figure 4.6: Evolution of probability distribution averaged over 100 active particles in time in square well with side length $4\mu m$. The radius of all particles is 1 μm and their velocity is $50\mu m/s$. As the time passes, the probability distribution spreads out and eventually becomes uniform except at the walls, where particles tend to spend some more time. The probability of an active particle to be found near the wall of the well increases with the increasing of the propulsion velocity of the particle.

Chapter 5

Chiral Active Swimmers

In the process of taking up energy from the environment in order to perform active motion, a microswimmer exerts a force on its surroundings. In the case of highly symmetric microswimmers moving in a symmetric environment with a driving force acting exactly along the direction of motion, the microswimmer performs a motion along a straight line which is just perturbed by the Brownian fluctuations. More often than not, the swimmers are asymmetric and the force is not aligned with the propulsion direction. This results in a net torque acting on the particle, and the particle starts moving in circular (in 2D) or helicoidal (in 3D) fashion, thus becoming chiral [22].
In this chapter, I give the model that I use to simulate chiral active swimmers and obtain all the results that will be presented later in the thesis, then I solve the set of Langevin equation numerically and discuss the motion of the chiral microswimmers in homogenous environments.

5.1 Model for chiral microswimmers

Since chiral motion arises when a net torque from outside starts acting on the particle, we can think of it as a motion of active microswimmer of the kind considered in chapter 4, but with a bias in the rotational angle. Denoting the

angular velocity as Ω, with units $rad\ s^{-1}$, the equations of motion for a chiral swimmer are obtained straightforward from equations 4.3, 4.4 and 4.5 as

$$\frac{d}{dt}\phi(t) = \Omega + \sqrt{2D_R}\xi_\phi \tag{5.1}$$

$$\frac{d}{dt}x(t) = v\cos\phi(t) + \sqrt{2D_T}\xi_x \tag{5.2}$$

$$\frac{d}{dt}y(t) = v\sin\phi(t) + \sqrt{2D_T}\xi_y \tag{5.3}$$

All of the symbols retain their meanings from Chapter 4. Here it should be noted that the sign of Ω defines the type of chirality (levogyre or dextrogyre) of the swimmer.

5.2 Numerical Solution

Using the same methods as described in earlier chapters, the following equations are solutions of the discretized Langevin equation for a chiral swimmer:

$$\phi_i = \phi_{i-1} + \Omega\Delta t + \sqrt{2D_R\Delta t}\xi_{\phi,i} \tag{5.4}$$

$$x_i = x_{i-1} + v\cos\phi_i\Delta t + \sqrt{2D_T\Delta t}\xi_{x,i} \tag{5.5}$$

$$y_i = y_{i-1} + v\sin\phi_i\Delta t + \sqrt{2D_T\Delta t}\xi_{y,i} \tag{5.6}$$

The black line in figure 5.1(c) is a single instance of a dextrogyre swimmer. The radius of the swimmer is $R = 1000nm$ and it is seen that when swimming in obstacle-free environment this swimmer bends clockwise, tracing out an almost exact circular trajectories just perturbed by Brownian fluctuations due to the translational and rotational diffusion of the particle. Just by changing the sign of Ω we change the chirality of the particle, and thus we obtain the red line in figure 5.1(c) for the case of levogyre swimmer. The general properties of the motion are scalable and in particular we observe similar trajectories in figure figure 5.1(d - g) for particles with radii $R = 500, 250, 125$ and $50nm$ respectively. The qualitative resemblance between the trajectories is retained provided the Péclet number, given by $P_e = \frac{Rv}{D_T}$, is kept constant. This is accomplished by

scaling the velocity v, angular velocity Ω and time t by the factor of R^{-2}. The whole set of parameters used to produce these trajectories is given in Table 1.
I note here that although there is a resemblance between the trajectories for different swimmers, as the particles become smaller, their motion become more vigorous and the trajectories become less deterministic. As discussed in chapter 4, the reason for this lies in the equations 4.1 and 4.2 which show that when the particle is small, the rotational diffusion, which is responsible for the reorientation of the particle, becomes more prominent since it scales down as R^{-3} while the translational diffusion only scales down as R^{-1}. Therefore, in order to gain insight into the characteristic motion of the chiral swimmers, we should consider the ensemble average of many swimmers. In figure 5.1(h), I plot the average of 10^5 trajectories for particles with radius $R = 1000nm$ starting from the origin, red line represents levogyre chirality swimmers while black line shows the dextrogyre swimmers. The average trajectory is a *spira mirabilis* with orientation dependent on the chirality of motion. This *spira mirabilis* has dimensions that set the length scale for the swimmers rotations which becomes important for the motion of the swimmers in patterned environments.

Table 5.1: Microswimmer parameters used in the simulations. From the radius R the rotational diffusion coefficient D_R and the translational diffusion coefficient D_T are obtained using Eqs. 4.1 and 4.2 respectively. The linear velocity v and the angular velocity ω are rescaled in order to maintain the Péclet number $Pe = \frac{Rv}{D_T}$ constant. Reproduced from [22]

$R\,(\mathrm{nm})$	$D_R\,(\mathrm{rad^2\,s^{-1}})$	$D_T\,(\mu\mathrm{m^2\,s^{-1}})$	$v\,(\mu\mathrm{m\,s^{-1}})$	$\Omega\,(\mathrm{rad\,s^{-1}})$	P_e
1000	0.16	0.22	3.13×10^1	± 3.14	142
500	1.32	0.44	1.25×10^2	$\pm 1.26 \times 10^1$	142
250	10.54	0.88	5.00×10^2	$\pm 5.03 \times 10^1$	142
125	84.4	1.76	2.00×10^3	$\pm 2.00 \times 10^2$	142
50	1320	4.4	1.25×10^4	$\pm 1.25 \times 10^3$	142

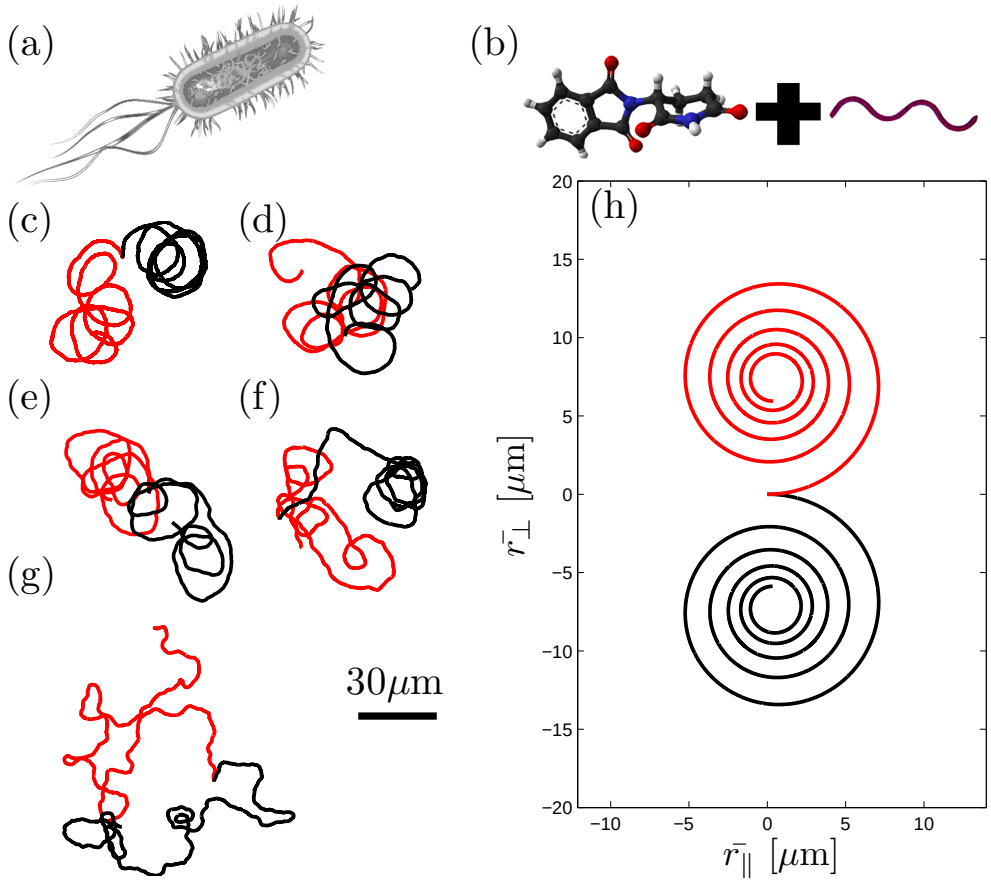

Figure 5.1: Chiral microswimmers. (a) *Escherichia coli* bacteria perform a characteristic chiral motion in the proximity of a surface. (b) Active chiral molecules can be obtained by chemically attaching a chiral molecule with a chiral propeller, e.g., a flagellum. ($c-g$) The trajectories of chiral levogyre (red) and dextrogyre (black) microswimmers with different radii ($R = 1000$, 500, 250, 125, and 50 nm for (c), (d), (e), (f) and (g) respectively, see Tab. 5.1 for the other parameters) are qualitatively similar as long as the Péclet number is kept constant and the time is scaled accordingly ($t = 10$ s, 2.5 s, 625 ms, 157 ms and 25 ms for (c), (d), (e), (f) and (g) respectively). See also the supplementary movies 1, 2 and 3 corresponding to (c), (d) and (e) respectively. (h) Average of 10^5 trajectories starting at $[x(0), y(0)] = [0\,\mu\text{m}, 0\,\mu\text{m}]$ with $\varphi(0) = 0$ for levogyre (red) and dextrogyre (black) microswimmers as in (c). Reproduced from [22]

.

Chapter 6

Boundary Conditions

In this chapter I simulate the motion of the particles in complex environments and show how to treat the motion of the particle when the particle approaches some impenetrable boundary that restricts its motion. Let the current position of the particle be given by $\mathbf{r}_n = (x_n, y_n)$. We would like to find the position appropriate with the boundary conditions after one time step, $\mathbf{r}_{n+1}(x_{n+1}, y_{n+1})$. This task can be achieved by applying the following algorithm to the motion of the particle:

1. Temporarily update the position of the particle for one time step, to $\widetilde{r}_{n+1} = (\widetilde{x}_{n+1}, \widetilde{y}_{n+1})$.
2. Check whether the boundary condition is satisfied by the point $\widetilde{r}_{n+1}$.
3. Do one of the following:
   - If the point satisfies the condition, accept it as a final point, and accept $\mathbf{r}_{n+1}(x_{n+1}, y_{n+1}) = \widetilde{r}_{n+1}$.
   - If the point does not satisfy the condition, calculate the new point that satisfies the boundary conditions. Therefore, we can write $\mathbf{r}_{n+1}(x_{n+1}, y_{n+1}) = \mathbf{F}(\widetilde{r}_{n+1})$, where $\mathbf{F}$ is some action to be taken depending on the nature of the boundary conditions.

4. Continue with the simulation of the next time step, going through this algorithm during every update.

This algorithm is general and it does not change with the nature of the boundary conditions. However, each set of boundary conditions needs distinctive treatment in order to find the right form of the action F. Over the next sections in this chapter, I go over some of the most commonly used boundary condition and outline the procedure of treating them retaining the notation I used this section.

6.1 Periodic boundary conditions

This type of boundary conditions is generally used to model an infinite space which can be constructed by series of translations of a single unit cell, figure 6.1(b). In this case, one imagines that the particle is confined in the unit square of length L. We require that the coordinates, x and y for 2D motion, have the periodicity of the length of the unit cell.
The updated coordinates are always calculated as the remainder of the division of themselves by the length of the unit cell. This ensures the following:

- If after the update, the particle's position is inside the unit cell, $\widetilde{r}_{n+1} < L$, therefore the remainder of the division of particle position to the unit cell is the position itself, $\mathbf{r}_{n+1} = \widetilde{r}_{n+1}$.
- If after the update, the particle's position is outside of the unit cell, $\widetilde{r}_{n+1} > L$, then $\mathbf{r}_{n+1}$ is the remainder of the division of particle position to the unit cell length. This satisfies the periodicity requirement, as a particle who has gone out of the unit cell through some wall, enters back into the same unit cell from a wall directly opposite, as illustrated in figure 6.1(a).

One visualization problem arises when plotting the trajectories. If we directly plot the trajectory as obtained from the simulation program, all the steps will

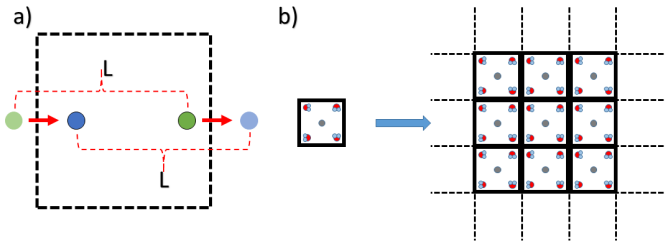

Figure 6.1: Treatment of periodic boundary conditions.(a) Application of the boundary conditions. If the particle is at the position of dark green circle and in the next step the position is calculated to be outside of the unit cell (light blue circle), the particle is assumed to have came from the opposite wall and the final updated position will be the dark blue circle. (b) If we take a unit cell to be a square, the whole two dimensional space can be constructed by translations of the unit cell.

appear as if they have been taken within the unit cell (due to the boundary conditions, the simulated particle never escapes the unit cell, just makes jumps between opposite walls). Therefore, when visualizing, we need to be aware of the nature of the trajectory, therefore *"unwrap"* it so that we see the original motion in the periodic space.

Note: Some programming languages have a built in function that does the visualization of the trajectory properly. In particular, MATLAB has a built-in function *unwrap* that, having taken as input x and y gives the unwrapped trajectory X and Y as

$$X = unwrap(\frac{x}{2\pi L})/(2\pi L) \tag{6.1}$$

$$Y = unwrap(\frac{y}{2\pi L})/(2\pi L) \tag{6.2}$$

Figure 6.2 shows the difference between the unwrapped (black line) and original trajectory (yellow line) for a simulated time of $t = 5s$. The unit cell is denoted by the red square and consists of 9 impenetrable circular obstacles.

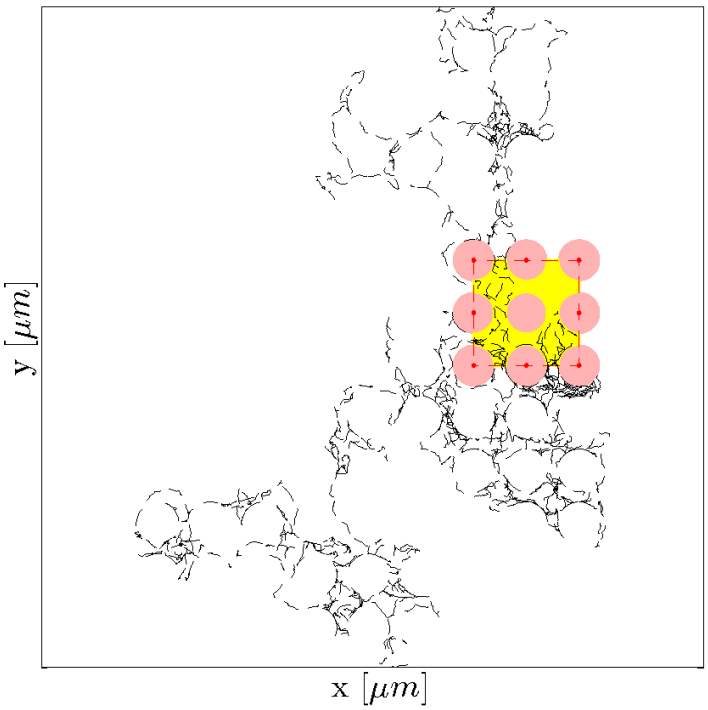

Figure 6.2: Difference between the original (yellow line) and unwrapped (black line) trajectory for an active Brownian particle swimming for 5s. The original trajectory always stays withing the unit cell, denoted by the red square in the figure. On contrary, the unwrapped trajectory gives the "real" motion in the space that is obtained by consecutive translations of the unit cell.

6.2 Reflective boundary conditions in a square

Reflective boundary conditions are used when **the x and y coordinates of the particle are reflected with respect to the wall of the boundary**. In particular, since we are simulating motion in 2 dimensions, the walls are represented by lines with a given thickness. Denote with $\mathbf{r}_n$ the initial point, $\widetilde{r}_{n+1}$ the tentative final point and $\mathbf{r}_{n+1}$ the final point after one time step update. Therefore, if $\widetilde{r}_{n+1}$ is outside of the square well, we do the following:

- In the case of vertical wall,

$$x_{n+1} = -\widetilde{x}_{n+1} \ \ \& \ \ y_{n+1} = +\widetilde{y}_{n+1} \tag{6.3}$$

- In the case of horizontal wall,

$$x_{n+1} = +\widetilde{x}_{n+1} \ \& \ y_{n+1} = -\widetilde{y}_{n+1} \tag{6.4}$$

- In the case when wall is given by a generic function, $y_{wall} = mx_{wall} + b$, see appendix A for the detailed derivation of the formula.

6.3 Sliding boundary conditions in a circular well

This type of boundary condition follow closely the type explained in the previous section, with the difference that instead of reflecting of the coordinates with respect to the wall, **the particle is placed at the rim of the wall**. The point on the rim at which the particle is placed is the intersection point between the line that connects the origin to $\widetilde{r}_{n+1}$ and the rim of the circle, as illustrated in figure 6.3. In this case, I kill off the perpendicular component of the motion and retain only the parallel one, thus having the effect of particle sliding along the circumference of the circle. This method is applicable to the circular well (it needs modification in the case when circle is ellipse, as I show in the next section) due to the symmetry of the circle.
Let the circular well have a radius R_{well} with a center placed at $[x_{well}, y_{well}]$. Then, the intersection point will have a certain polar angle, call it m, with respect to the x - axis. The angle is found as,

$$m = \arctan\left(\frac{\widetilde{y}_{n+1} - y_{well}}{\widetilde{y}_{n+1} - y_{well}}\right) \tag{6.5}$$

Having obtained that, the final point is,

$$x_{n+1} = x_{well} + R_{well}\cos(m) \tag{6.6}$$

$$y_{n+1} = y_{well} + R_{well}\sin(m) \tag{6.7}$$

The derivation for the boundary conditions when the particle is confined to move only on the *outside* of the circular boundary go along the same lines. However,

it should be noted that, in that case the time step of the simulation should be chosen carefully so that the **spatial increment of the trajectory is less than the size of the obstacle**, therefore avoiding a number of unphysical situations, for example a particle jumping over to the opposite side of unpenetrable obstacle.

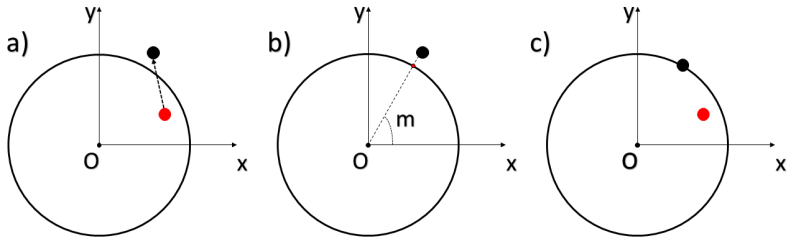

Figure 6.3: Illustration of sliding boundary conditions along a circular well. (a) Particle is initially at the position of the red circle and the tentative update of the position is the black circle. (b) Find the polar angle that the tentative point has with respect to the x axis. (c) Place the point at the rim of the circle by setting its radial position to R_{well} and find the x and y coordinates as discussed in equations 6.6 and 6.7

6.4 Sliding boundary conditions for ellipsoidal obstacles

The sliding boundary condition when the particle encounters an ellipsoidal obstacle are more complex to calculate than the corresponding ones for the circular obstacle because of the breaking of the symmetry due to the unequal major and minor axis. The additional feature of the ellipses is that they can have particular orientation in the medium, expressed as angle between the major axis and x axis. We start by realizing that an ellipse with any arbitrary position and orientation can be obtained from an ellipse placed in the center with its major axis and x axis aligned by a suitable translation followed by rotation. Therefore, referring

to figure 6.4, I outline the process of implementation of the boundary conditions when a particle encounters an arbitrary elliptical boundary:

- Perform a suitable rotation and translation of the general ellipse, such that after these processes the ellipse is centered at the origin and its major axis is aligned with the x axis. (figure 6.4(a) and (b))
- Find the line that connects $\mathbf{r}_n$ (red dot in figure 6.4(b)) to $\widetilde{r}_{n+1}$ (black dot in figure 6.4(c)). Find the intersection point between this line and the ellipse.
- Draw the tangent line to the ellipse passing through the intersection point. (figure 6.4(c))
- Project $\widetilde{r}_{n+1}$ to the tangent line. The projected point is $\mathbf{r}_{n+1}$. (figure 6.4(d-e))
- Transform the system back to the initial state by performing the inverses of the actions undertaken in step one. Take note that rotations and translations do not commute. (figure 6.4(f))

In appendix B, I present detailed calculation for the treatment of the boundary conditions and obtain the corresponding equations for the x and y coordinates of the final point $\mathbf{r}_{n+1}$.

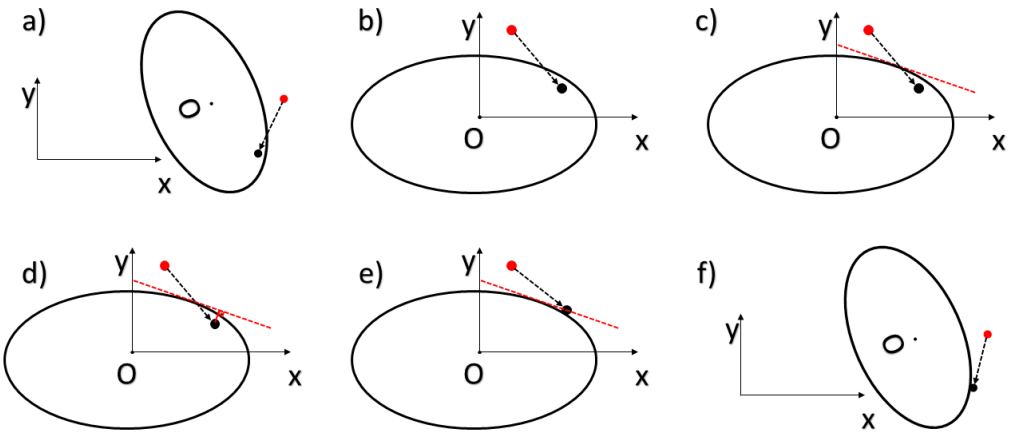

Figure 6.4: Illustration of the boundary conditions when the microswimmer encounter an elliptical obstacle. (a) Any general ellipse can be brought to the center of the standard coordinate system by a suitable rotation and translation. (b-e) Find the tangent line to the ellipse through a point that is intersection between the black and the red circle, project the black circle onto this tangent to find the final point. (f) Rotate the whole system back to the initial position.

Chapter 7

Results

In this chapter I numerically show how to sort chiral microswimmers on the basis of their swimming properties, such as chirality, linear and angular velocity. The microswimmers are simulated using the model I presented in chapter 5 and all the discussions follow closely the ones in ref. [22]. Most of the results I show are obtained by simulating two dimensional swimmers moving in two dimensional patterned environment, however at the end I provide a brief explanation on how these results can be extended to microswimmers performing helicoidal motion in three dimensions. In the simulations I show in this chapter microswimmers frequently encounter obstacles on their way. They interact with the obstacles by sliding along them until the orientation of the particle points away from the obstacle as discussed in [33]. Numerically, I implement this fact by using the procedure presented in section 6.4. Despite neglecting the hydrodynamic effects between the microswimmers and the obstacles, in ref. [33] the authors showed that the model I use in this thesis accurately describes the interaction trajectories near an obstacle.

7.1 Chirality separation

The microswimmers can be selected on the basis of their chirality by placing some chiral patterns in their swimming environment, therefore coupling the chirality of the motion to the chirality of the obstacle pattern. The pattern we propose, the *chiral flower*, is an arrangement of tilted ellipses along a circle, depicted as the shaded areas in 7.1(a). In here, the circle has a radius of $11\mu m$, specifically chosen in this way by comparing it to the length scale of the *spira mirabilis* shown in figure 5.1. Since the chirality of the motion couples with the chirality of the *flower* we observe different behavior for the opposite chiralities. Therefore, while the levogyre swimmer (the red line in 7.1(a)) is able to move in and out of the *chiral flower* the dextrogyre microswimmer (black line) is always trapped inside the *flower*. To gain greater understanding of this behavior, I plot the probability distribution in the swimmers' radial position. The probability distribution is averaged over 10^5 trajectories that are 10s long after all the particles have been released from the origin. Figure 7.1(b)) shows the results: the levogyre swimmers (the red histogram) are outside of the *chiral flower* while the dextrogyre particles (the black histogram) are trapped inside.

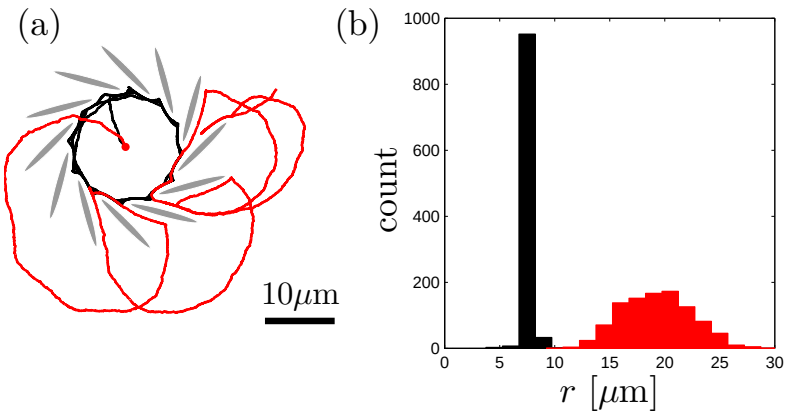

Figure 7.1: Chiral microswimmers in a chiral environment. (*a*) 10 s trajectories of a levogyre (red line) and dextrogyre (black line) microswimmer with radius $R = 1000\,\text{nm}$ (for the whole list of parameters, see Tab. 5.1) in a chiral flower of ellipses (shaded areas). (*b*) Radial position distribution after 10 s of 1000 levogyre (red histogram) and dextrogyre (black histogram) microswimmers starting from the center of the chiral flower at $t = 0\,\text{s}$. Reproduced from [22].

The method outlined until now manages to trap only the dextrogyre microswimmers, in this configuration we do not have the control over where the levogyre swimmers end up after escaping the flower. However, improving on the single flower configuration, I propose a simple device to separate and trap microswimmers with different chiralities. As it is shown in figure 7.2, the separation can be achieved by placing two chiral flowers with opposite chiralities in a box in which the particles move freely. I start by placing a racemic mixture inside each of the chiral flowers at $t = 0s$ (figure 7.2a)). What I observe is that, each chirality couples with one of the two flowers, therefore we have most of the levogyre (dextrogyre) microswimmers going out of the right (left) flower while the opposite chirality remains trapped. The microswimmers that escaped are free to roam and explore the box, until the time they are trapped by the corresponding chiral flower (figure 7.2(c-f)). In particular, note that most of the swimmers are trapped and the separation is finished by $t = 500s$.

In order to make these observations more quantitative, I introduce the parameter

$$Q = \frac{1}{2}\Big[\frac{L_L}{L_L + D_L} + \frac{D_D}{L_D + D_D}\Big] \tag{7.1}$$

where $L_L(D_L)$ is the number of levogyre (dextrogyre) microswimmers in the levogyre flower and $D_D(L_D)$ is the number of dextrogyre (levogyre) microswimmers in the dextrogyre flower, both calculated after $t = 500s$ have elapsed. As shown in figure 7.3, I easily obtain $Q \approx 100\%$ for a great range of parameters, including variations of the radius of the flower (figure 7.3(b)), the length of the elliptical obstacles (figure 7.3(c)) and the chirality of the flower (figure 7.3(c)). The time needed to separate the microswimmers depends mainly on the size of the box that encloses the chiral flower and does not depend strongly on the parameters of the flowers themselves. Due to the increasing randomness in the trajectories, the coefficient becomes slightly lower for smaller particles, however high sorting efficiency can still be obtained for small particles down to $50nm$.

7.2 Sorting by velocity

Sorting of microswimmers on the basis of their linear velocity can be done by using a patterned microchannel. This channel is achieved by placing a series of elliptical structures on the walls of the channel, tilting the ellipses at some angle and slightly shifting the obstacles on the top and bottom walls, as shown by the grey areas in figure 7.4. A channel devised by this way is chiral and can couple with the chirality of the motion, and figure 7.4 shows that the microswimmers with different linear velocities show different behavior when placed inside this channel. Indeed, after starting at $x = 0\mu m$ at $t = 0s$, the levogyre microswimmers with linear velocity of $v = 70\mu m$ are able to propagate several hundreds microns to the right, while the slower microswimmers are trapped near the initial position (figure 7.4(a)). Both sets of microswimmers have the same chirality and same angular velocity of $\Omega = +3.1\ rad\ s^{-1}$. The channel has a multiple functions when considered how the microswimmers behave when placed inside it. In particular, the channel may act as a *funnel* and it rectifies the motion of the microswimmers to the right. Another usage of the channel is as a *sieve*, since from figure 7.4 we can see that the particles below certain velocity cannot propagate to the right. What is more, by changing the structure of the channel (length and angle of the ellipses), the threshold velocity above which particles propagate can be changed. Also, by reorientation of the ellipses, we can obtain the same results for dextrogyre particles. Finally, the microchannel is a *sorter* of the microswimmers with separation efficiency increasing over time. Figure 7.4(b-e) show that, as time progresses, the probability distributions of the particles separate, the faster particles move further down the channel while the slower ones are constantly kept back.

7.3 Sorting by angular velocity

The microchannel I discussed in the previous section can be also used to separate the microswimmers on the basis of their angular velocity. Same as the previous

case, the probability distribution of the microswimmers can separate over time, figure 7.5(b-e) and as shown by the histograms, swimmers with angular velocities $\Omega = 2.2, 2.5, 2.8$ and $3.1 rad\ s^{-1}$ can be successfully separated, with the ones with smaller angular velocity propagating further to the right in the channel. In particular, as shown in 7.5(a), the particles with $\Omega = 2.2 rad\ s^{-1}$ (red dots) can propate while the ones with $\Omega = 3.1 rad\ s^{-1}$ (black dots) are trapped near to the initial position.

7.4 Sorting of 3D chiral microswimmers

All the results discussed in the previous cases, can be extended to 3-dimensional swimmers that perform helicoidal motion. A particular example of this is illustrated in figure 7.6. In here, the microswimmers are made to enter from the bottom with a given initial velocity. As such, they perform helicoidal motion whose orientation depends on the chirality of the swimmer. In a microchannel patterned with a chiral flower at a certain height, as the one showed in 7.6, levogyre microswimmers (red line) tend to escape from the channel while the dextrogyre have the tendency to remain in the tube and the efficiency of the process that I was able to obtain is $Q = 93\%$. A straightforward generalization can also be made from the microchannels shown in figures 7.4 and 7.5 just by decorating the inner walls of the 3D microchannel with some elliptical obstacles.

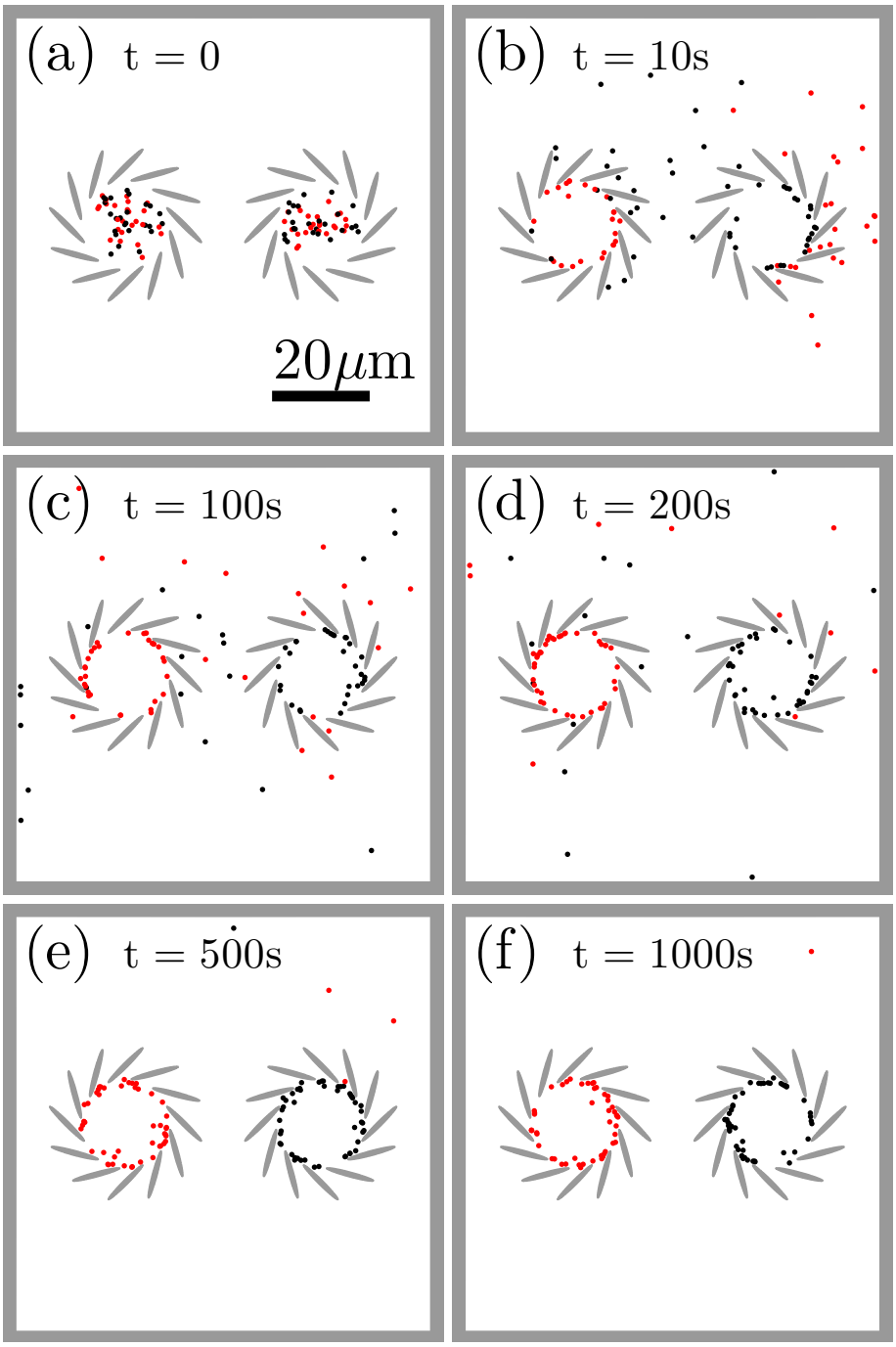

Figure 7.2: Separation of levogyre and dextrogyre microswimmers. (a) At $t = 10\,\mathrm{s}$, the microswimmers ($R = 1000\,\mathrm{nm}$, see Tab. 5.1) are released inside two chiral flowers with opposite chirality. ($b - f$) As time progresses, the levogyre (black symbols) microswimmers are trapped in the right chiral flower and the dextrogyre (red symbols) ones in the left chiral flower. Reproduced from [22].

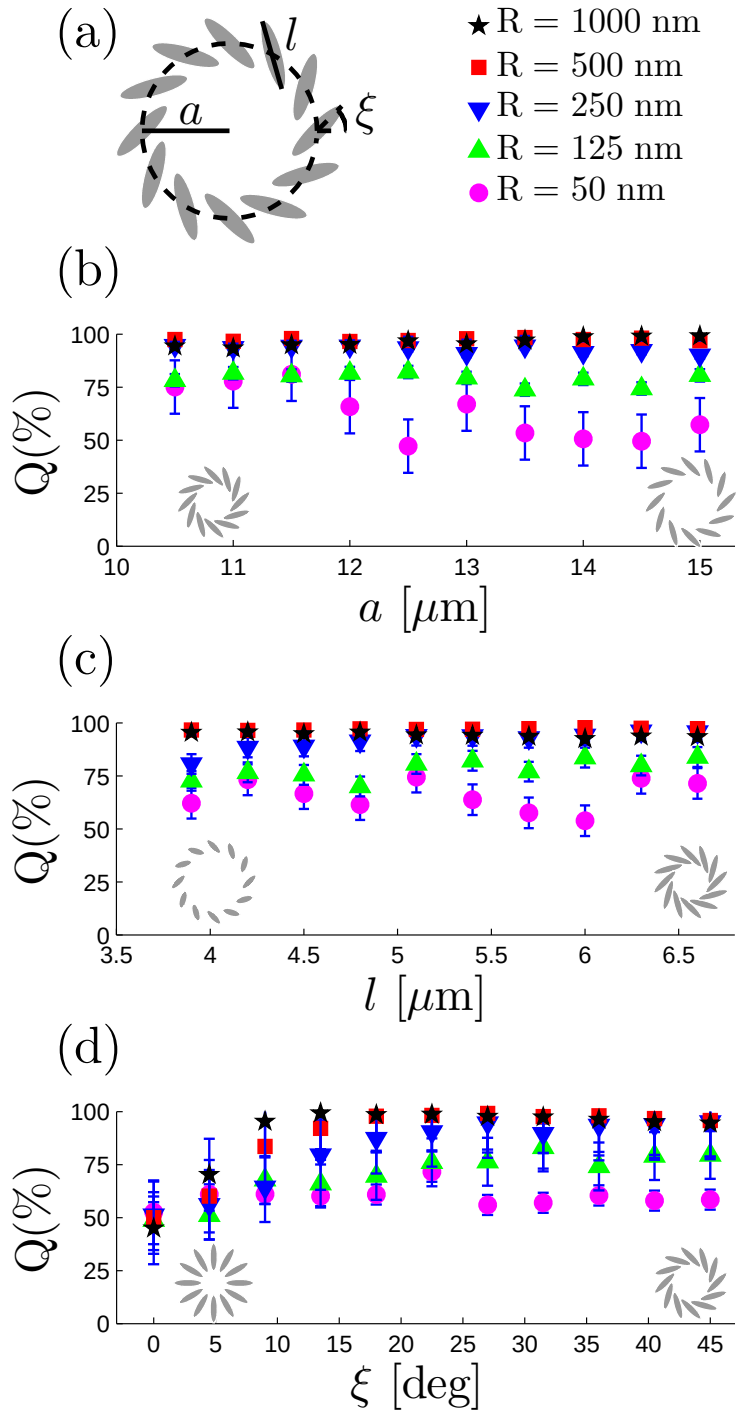

Figure 7.3: Sorting efficiency. (a) Parameters of the chiral flower: a is the radius of the flower; l is the length of the ellipses; and ξ is the angle of the ellipses. (b-d) Sorting efficiency Q (Eq. (7.1)) as a function of (b) a, (c) l, and (d) ξ for various particle sizes (Tab. 5.1) using the configuration in Fig. 2.3. Each datapoint is calculated using 50 levogyre and 50 dextrogyre particles placed inside the chiral flowers. The error bars represent one standard deviation repeating the numerical simulations 10 times. Reproduced from [22].

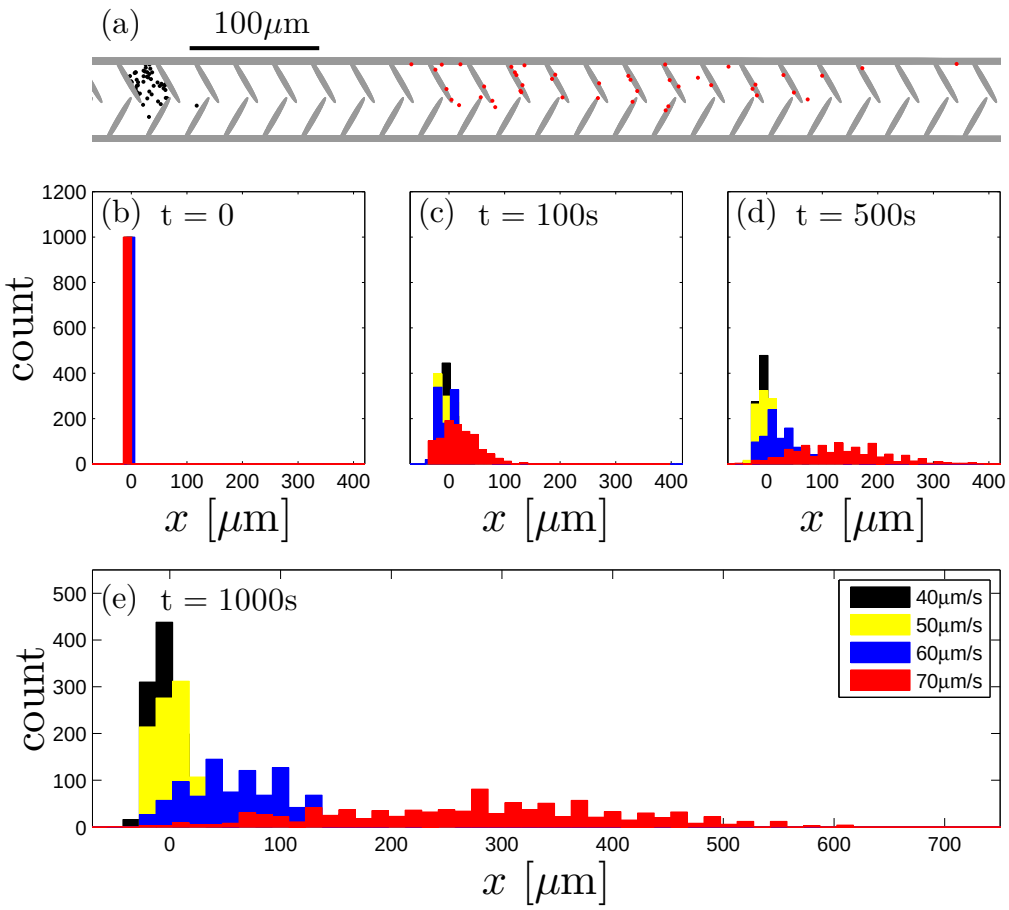

Figure 7.4: Linear velocity based sorting in a microchannel. (a) Patterned microchannel (grey areas) and position of levogyre 1000 nm microswimmers with $v = 40$ (black symbols) and 70 μm/s (red symbols) 1000 s after they have been released from position $x = 0\,\mu$m (other parameters as in Tab. 5.1).($b-e$) Histograms at various times of 1000 particles with $v = 40$, 50, 60 and 70 μm/s released at $t = 0$ s from $x = 0\,\mu$m. Reproduced from [22].

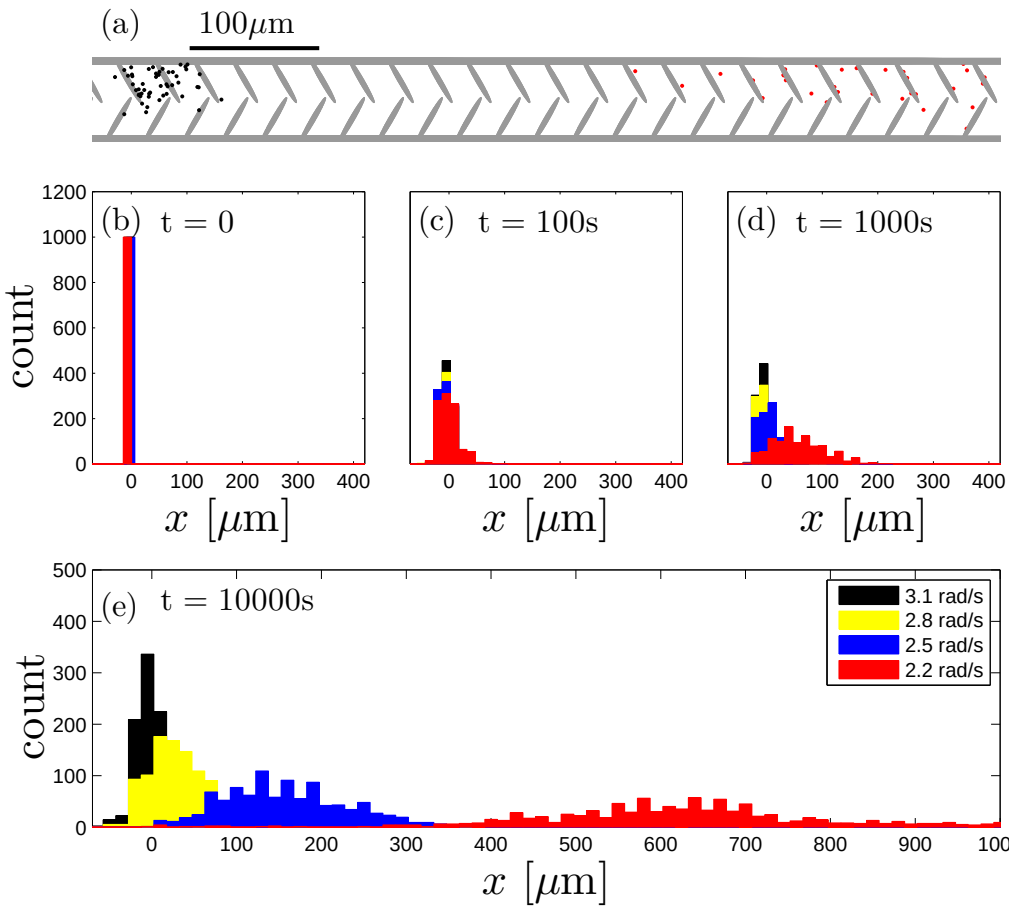

Figure 7.5: Angular velocity based sorting in a microchannel. (a) Patterned microchannel (grey areas) and position of levogyre 1000 nm microswimmers with $\Omega = 2.2$ (black symbols) and 3.1 rad/s (red symbols) 1000 s after they have been released from position $x = 0\,\mu$m ($v = 40\,\mu$m/s, other parameters as in Tab. 5.1). ($b - e$) Histograms at various times of 1000 particles with $\Omega = 2.2$, 2.5, 2.8 and 3.1 rad/s released at $t = 0$ s from $x = 0\,\mu$m. Reproduced from [22].

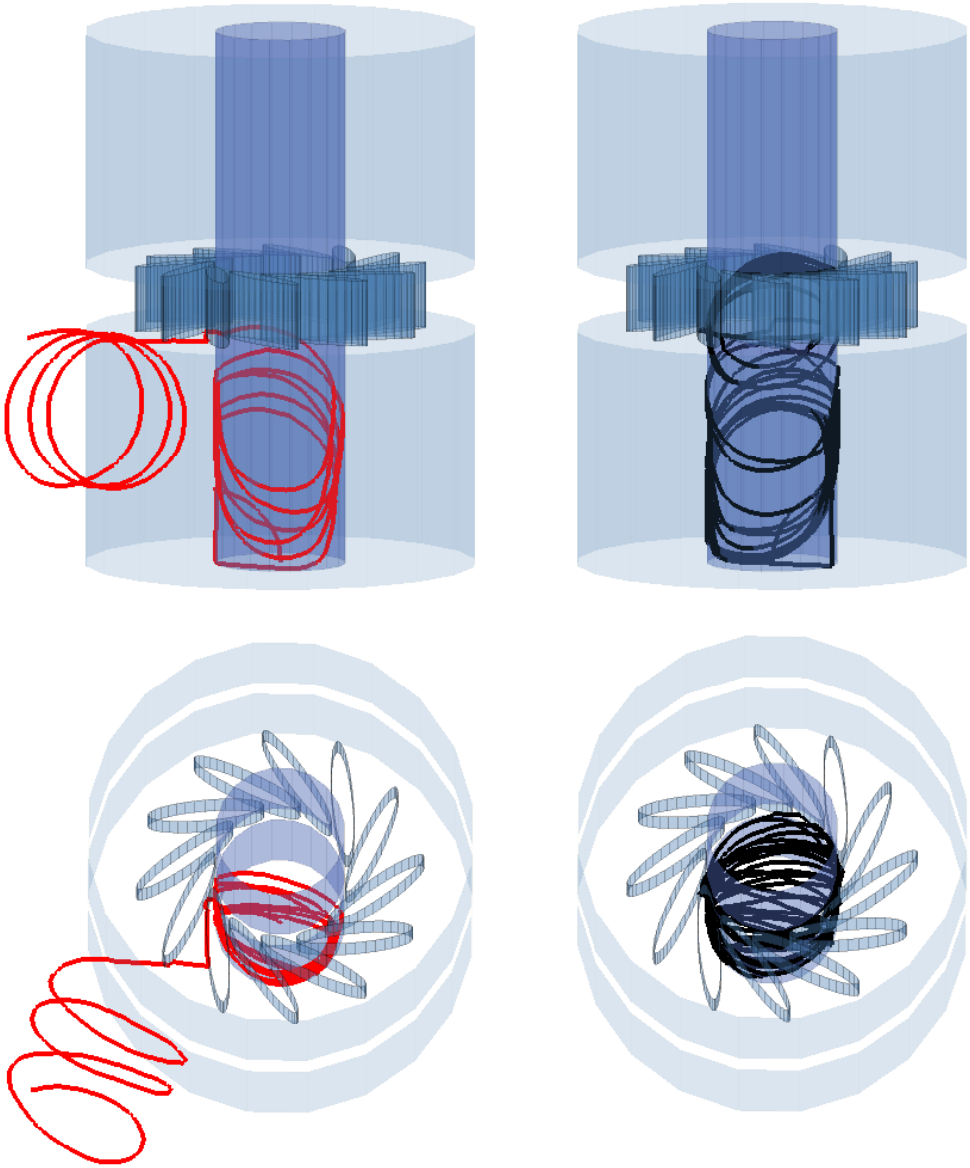

Figure 7.6: Sorting of three-dimensional chiral microswimmers. The transparent structures represent a circular microchannel (inner diameter 10 μm) with an engraved chiral flower. Views from the side (top) and from above (bottom) are presented. When levogyre microswimmers ($R = 1000\,\mathrm{nm}$, $v = 3.2\mathrm{e}{+2}\,\mu\mathrm{m/s}$, $\Omega = +50\,\mathrm{rad/s}$, red line on the left) are made to enter the channel from the lower end to break the reflection symmetry of the channel, they tend to exit the channel as soon as they reach the chiral flower, in contrast to dextrogyre microswimmers ($\Omega = -50\,\mathrm{rad/s}$, black line on the right), which are prevented from escaping. The sorting efficiency is $Q = 93\%$. Reproduced from [22].

Chapter 8

Conclusion and outlook

Due to the ability to propel themselves, active Brownian particles hold great promises for applications in many different tasks in science and engineering or as model to study out of equilibrium systems. Taking inspiration from the active microorganisms existing naturally, many artificial swimmers that can propel themselves have been designed.

If these swimmers are engineered to be anti-symmetric, they follow characteristic circular trajectories in two dimensional environments, therefore becoming chiral. Sorting of this kind of microswimmers is of extreme importance due to the fact that selecting only the best microswimmers for a certain task can result in greater chance to accomplish that task with improved efficiency. In this thesis, I showed that such separation is possible by placing a static chiral patterns that can be readily fabricated by standard micro fabrication techniques in the environment of the microswimmer.

I showed that *chiral flower* can trap microswimmers with certain chirality, a *patterned microchannel* can sort the microswimmers based on their linear and angular velocity, can trap the microswimmers with certain parameters and can serve to rectify their motion. The main advantage of the proposed methods is that they rely on *static* environmental structures and on the microswimmers' own self-propulsion, without the need of moving parts, fluid flows or external forces. These techniques can be exploited to separate naturally occurring chiral active

particles, for example bacteria and spermatozoa, or separate chiral passive particles, for example chiral molecules, by coupling them to some chiral motors to make them active.

Bibliography

[1] S. H. Strogatz, *Nonlinear dynamics and chaos, 1994*, Reading: Perseus Books, 1996

[2] E. Nelson, *Dynamical theories of Brownian motion*, 17, Princeton university press Princeton 1967,

[3] A. Einstein, *Investigations on the Theory of the Brownian Movement*, Courier Dover Publications, 1956

[4] G. Volpe, S. Gigan and G. Volpe, *Simulation of the active Brownian motion of a microswimmer*, American Journal of Physics, v.82,7, 659–664, 2014

[5] D. B. Weibel, P. Garstecki, D. Ryan, W. R. DiLuzio, M. Mayer, J. E. Seto and G. M. Whitesides *Microoxen: Microorganisms to move microscale loads*, Proc. Natl. Acad. Sci. USA v. 102, 11963-11967, 2005

[6] R. M. Ford and R. W. Harvey, *Role of chemotaxis in the transport of bacteria through saturated porous media*, Adv. Water Resour, v.30, 1608-1617, 2007

[7] W. Yang, V. R. Misko, K. Nelissen, M. Kong and F. M. Peeters, *Using self-driven microswimmers for particle separation*, Soft Matter, v.8, 5175-5179, 2012

[8] C. D. Chin, V. Linder and S. K. Sia, *Lab-on-a-chip devices for global health: Past studies and future opportunities*, Lab Chip, v.7, 41-57, 2007

[9] S. J. Ebbens and J. R. Howse, *In pursuit of propulsion at the nanoscale*, Soft Matter, v. 6, 4, 726–738, 2010, Royal Society of Chemistry

[10] S. van Teeffelen and H. Lwen, *Dynamics of a Brownian circle swimmer*, Phys. Rev. E, v. 78, 28, 28, 2008,

[11] W. R. DiLuzio, L. Turner, M. Mayer, P. Garstecki, D. B. Weibel, H. C. Berg and G. M. Whitesides, *Escherichia coli swim on the right-hand side*, Nature, v.435, 1271-1274, 2005

[12] E. Lauga, W. R. DiLuzio, G. M. Whitesides and H. A. Stone, *Swimming in Circles: Motion of Bacteria near Solid Boundaries* , Biophys. J., v.90, 400-412, 2006

[13] B.M. Friedrich and F. Jülicher, *Steering Chiral Swimmers along Noisy Helical Paths*, Phys. Rev. Lett., v. 103, 068102, 2009

[14] L. Lemelle, J. F. Palierne, E. Chatre and C. Place, *Counterclockwise Circular Motion of Bacteria Swimming at the Air-Liquid Interface*, J. Bacteriol., v.192, 6307-6308, 23, 2010

[15] P. Denissenko, V. Kantslerc, D. J. Smitha and J. Kirkman-Brownb, *Human spermatozoa migration in microchannels reveals boundary-following navigation*, Proc. Natl. Acad. Sci. USA,

[16] R. Di Leonardo, D. Dell'Arciprete, L. Angelani and V. Iebba, *Swimming with an Image*, Phys. Rev. Lett., v.106, 038101, 2011

[17] T.-W. Su, L. Xue and A. Ozcan, *High-throughput lensfree 3D tracking of human sperms reveals rare statistics of helical trajectories*, Proc. Natl. Acad. Sci. USA, v.109, 16018-16022, 2012

[18] D. S. Guzick, J. W Overstreet, P. Factor-Litvak, C. K Brazil, S. T. Nakajima, C. Coutifaris, S. A. Carson, P. Cisneros, M. P. Steinkampf, J. A Hill, D. Xu, D. L. Vogel, *Sperm Morphology, Motility, and Concentration in Fertile and Infertile Men*, New England Journal of Medicine, 345, 1388-1393, 2001,

[19] H.C. Berg, *E. coli in Motion*, Springer Verlag, 2004, New York

[20] *Chiral Separation Methods for Pharmaceutical and Biotechnological Products*, John Wiley and Sons, Inc., S. Ahuja, Hoboken, New Jersey, 2011

[21] Marcos, H. C. Fu, T. R. Powers and R. Stocker, *Separation of Microscale Chiral Objects by Shear Flow* , Phys. Rev. Lett., v.102, 15, 2009

[22] M. Mijalkov and G. Volpe, *Sorting of chiral microswimmers*, Soft Matter, 9, 28, 6376–6381, 2013, Royal Society of Chemistry

[23] J. P Sethna *Statistical mechanics: entropy, order parameters, and complexity*, Oxford University Press Oxford, 2006

[24] P. Kloeden and E. Platen, *Numerical solutions of stochastic differential equation*, Springer Verlag, Berlin- Heidelberg, 1992

[25] G. Volpe and G. Volpe, *Simulation of a Brownian particle in an optical trap*, Am. J. Phys., v. 81, 224-230, 2013

[26] M. D Haw, *Colloidal suspensions, Brownian motion, molecular reality: a short history*, Journal of physics: condensed matter, v.14, 33, 7769, 2002, IOP Publishing

[27] J. Perrin, *Brownian movement and molecular reality*, Courier Dover Publications,2013

[28] R. Zwanzig, *Nonequilibrium statistical mechanics*, Oxford University Press, 2001

[29] E. M Purcell, *Life at low Reynolds number*, Am. J. Phys, v. 45, 3–11, 1977

[30] A. Pototsky, *Dynamics and rectification of active Brownian particles*, url:http://www.newton.ac.uk/files/seminar/20130618110012001-153602.pdf, urldate: 2013-06-18

[31] J. R Howse, R. A. L. Jones, A. J. Ryan, T. Gough, R. Vafabakhsh and R. Golestanian, *Self-Motile Colloidal Particles: From Directed Propulsion to Random Walk*, Phys. Rev. Lett., v. 99, 048102, 2007,

[32] B. Feringa, *Molecular motors: fuelling movement at the nanoscale*, Nature Chemistry, v. 3, 12, 915-916, 2011, Nature Publishing Group

[33] G. Volpe, I. Buttinoni, D. Vogt, H. J. Kümmerer and C. Bechinger, *Microswimmers in patterned environments*, Soft Matter, v. 7, 19, 8810–8815, 2011, Royal Society of Chemistry

[34] H. Masoud and B. I Bingham and A. Alexeev, *Designing maneuverable micro-swimmers actuated by responsive gel*, Soft Matter, v.8, 34, 8944-8951, 2012, Royal Society of Chemistry

[35] L. Zhang, J. J Abbott, L. Dong, K. E Peyer, B. E Kratochvil, H. Zhang, C. Bergeles and B. J Nelson, *Characterizing the swimming properties of artificial bacterial flagella*, Nano letters, v. 9, 10, 3663-3667, 2009, ACS Publications

[36] I. Buttinoni, G. Volpe, F. Kümmel, G. Volpe and C. Bechinger *Active Brownian motion tunable by light*, Journal of Physics: Condensed Matter, v.24, 28, 284129, 2012, IOP Publishing

Appendix A

Reflection about a generic wall

Let the wall be defined by the equation $y_{wall} = mx_{wall} + b$. The swimmer's current position is denoted by $\mathbf{r}_n$, I am tentatively updating its position after one time step to point $A \equiv \widetilde{r}_{n+1}$ and supposing that this position cannot be accepted in the simulation, the following derivation will show how to find the final updated point, $B \equiv \mathbf{r}_{n+1}$ which is reflection of $\widetilde{r}_{n+1}$ about the wall given by the above equation.

- Find the slope of the line of reflection (this is the line which is perpendicular to the wall)

$$m_{ref} \times m = -1 \Longrightarrow m_{ref} = -\frac{1}{m} \tag{A.1}$$

- Compute the equation of the reflection line that passes through the point A

$$y_{ref} - \widetilde{y}_{n+1} = m_{ref}(x_{ref} - \widetilde{x}_{n+1}) \Longrightarrow y_{ref} = m_{ref}x_{ref} + (\widetilde{y}_{n+1} - m_{ref}\widetilde{x}_{n+1}) \tag{A.2}$$

  Call $d = \widetilde{y}_{n+1} - m_{ref}\widetilde{x}_{n+1}$.

- Find the point of intersection (use the subscript *int* to denote this point)

$$y_{int} = mx_{int} + b = m_{ref}x_{int} + d \Longrightarrow x_{int} = \frac{b-d}{m_{ref} - m} \tag{A.3}$$

So, the point of intersection,$P_{int}[x_{int}, y_{int}]$ is given by the coordinates

$$x_{int} = \frac{b-d}{m_{ref}-m} \;\; \& \;\; y_{int} = mx_{int} + b \tag{A.4}$$

- Finally, we find B by noting that it lies on the line of reflection at equal distance from P_{int} as A. Therefore, we can write

$$B = A + 2 \times \overrightarrow{AP_{int}} \tag{A.5}$$

Since we can write the x and y components of the vector as $[x_{int} - \widetilde{x}_{n+1}, y_{int} - \widetilde{y}_{n+1}]$, for the final point we have

$$\mathbf{x(n+1)} = 2x_{int} - \widetilde{x}_{n+1} \quad \& \quad \mathbf{y(n+1)} = 2y_{int} - \widetilde{y}_{n+1}$$

Appendix B

Boundary conditions in the presence of an elliptical obstacle

Let the point $A \equiv \mathbf{r}_n$ be the initial point and the point $B \equiv \mathbf{r}_{n+1}$ be the final point after one time step increment. These points are given with respect to the standard coordinate system. As discussed in section 6.4 I perform translation then rotation of the system consisting of an arbitrary ellipse plus the two points in a manner that I obtain ellipse centered at the origin of the standard coordinate system and two points, (x_{t_0}, y_{t_0}) and (x_{t_1}, y_{t_1}) respectively. Let a and b be the axes of the ellipse, (xc, yc) be the coordinates of the center of the ellipse and the angle ξ be the angle of orientation of the ellipse. By using this notation, we can find the relation that helps us go from one point to another.

- Given the point $(x(n), y(n))$ we get (x_{t_0}, y_{t_0}) from the following transformation:

$$x_{t_0} = (x(n) - xc) * \cos(-\xi) - (y(n) - yc) * \sin(-\xi)$$

$$y_{t_0} = (x(n) - xc) * \sin(-\xi) + (y(n) - yc) * \cos(-\xi)$$

  Exactly the same equations can be used to obtain the point (x_{t1}, y_{t_1}) from the point $(x(n+1), y(n+1)$.

- In here I discuss the case of a particle that is restricted to enter into the ellipse. The check to be performed for this condition is,

$$\left(\frac{x_{t_1}}{a}\right)^2 + \left(\frac{y_{t_1}}{b}\right)^2 < 1$$

If this condition returns true, I continue to calculate the sliding point.

- Find the intersection point between the line that connects the points (x_{t_0}, y_{t_0}) and (x_{t_1}, y_{t_1}) and the ellipse.

1. The equation of the line connecting the two points is the following:

$$y = y_{t_1} + m * (x - x_{t_1})$$

where "m" is the slope of this line and can be expressed as

$$m = \frac{y_{t_1} - y_{t_0}}{x_{t_1} - x_{t_0}}$$

2. To simplify the notation, define $y_0 = y_{t_1} - m * x_{t_1}$, to get the equation of the line as,

$$y = y_0 + m * x$$

3. Find the intersection between this line and the ellipse.

$$\left(\frac{x}{a}\right)^2 + \left(\frac{y_0 + xm}{b}\right)^2 = 1$$

Expanding this and grouping the like terms,

$$\left(\frac{1}{a^2} + \frac{m^2}{b^2}\right) x^2 + \left(\frac{2y_0 m}{b^2}\right) x + \left(\frac{y_0^2}{b^2} - 1\right) = 0$$

4. This equation yields two solutions for the x coordinate of the intersection points.

$$x_1 = \frac{-(\frac{y_0 * m}{b^2}) + sqrt((\frac{y_0 * m}{b^2})^2 - (\frac{1}{a^2} + \frac{m^2}{b^2})(\frac{y_0^2}{b^2} - 1))}{(\frac{1}{a^2} + \frac{m^2}{b^2})}$$

$$x_2 = \frac{-(\frac{y_0 * m}{b^2}) - sqrt((\frac{y_0 * m}{b^2})^2 - (\frac{1}{a^2} + \frac{m^2}{b^2})(\frac{y_0^2}{b^2} - 1))}{(\frac{1}{a^2} + \frac{m^2}{b^2})}$$

The values for the y coordinate of the same point are found from the equation of the line derived earlier,

$$y_1 = y_0 + x_1 * m$$

$$y_2 = y_0 + x_2 * m$$

5. From the two solution points, chose only the point which is closer to (x_{t_1}, y_{t_1}). Decide this by comparing the distances between those points.

$$(x_1 - x_{t_1})^2 + (y_1 - y_{t_1})^2 < (x_2 - x_{t_1})^2 + (y_2 - y_{t_1})^2$$

and if this is true then I choose the point (x_1, y_1) and if this is not correct the point (x_2, y_2) is chosen.
For the purpose of this derivation I use the notation (x_i, y_i) for any of these points.

- Find the tangent line passing through this point. The slope of the tangent line can be found by implicit differentiation of the equation of the ellipse.

$$\frac{2x}{a^2} + \frac{2ym_{tan}}{b^2} = 0$$
$$\frac{ym_{tan}}{b^2} = -\frac{x}{a^2}$$

Then, at the particular point $y = y_i$ and $x = x_i$, the slope of the tangent line is given as:

$$m_{tan} = -\frac{x_i}{y_i} * \frac{b^2}{a^2}$$

- Find the equation of the line that is normal to this tangent line and passes through the point (x_{t_1}, y_{t_1}).

1. The slope of this line is found by $m_n = -\frac{1}{m_{tan}}$, and for this particular case

$$m_n = \frac{y_i}{x_i} * \frac{a^2}{b^2}$$

2. The equation for the line:

$$y = y_{t_1} + m_n(x - x_{t_1})$$

- Find the point at which this line intersects the tangent line by equating the y coordinates of both equations:

$$y_i + m_{tan}x - m_{tan}x_i = y_{t_1} + m_n x - m_n x_{t_1}$$
$$x_f = \frac{y_{t_1} - y_i + m_{tan}x_i - m_n x_{t_1}}{m_{tan} - m_n}$$
$$y_f = y_i + m_{tan}x_f - m_{tan}x_i$$

- Finally, project this point back into the original configuration. This means, rotate the point for an angle ξ and then translate for xc (yc). This gives the final point $(x(n+1), y(n+1)$ as:

$$x(n+1) = x_f * \cos\xi - y_f * \sin\xi + xc$$

$$y(n+1) = xf * \sin\xi + y_f * \cos\xi + yc$$

Appendix C

Code

C.1 Random Walk in 3D

```
function result = RWin3D(position,parameter,simulation)

%   RWin3D(position,increment parameter) simulates the motion of a particle
%   performing random walk in three dimensions

%   Inputs:

%   position = Structure with information about the initial position of the
%   particle
%   position.x0 = initial position in x direction
%   position.y0 = initial position in y direction
%   position.z0 = initial position in z direction

%   parameter = Structure with information about the multiplication factor
%   that controls the strength of the white noise
%   parameter.PX = multiplication factor of the increment in the x direction
%   parameter.PY = multiplication factor of the increment in the y direction
```

```
18 %   parameter.PZ = multiplication factor of the increment in the z
       direction
19
20 %   simulation = Structure with information about the simulation
       parameters
21 %   simulation.N = number of samples
22 %   simulation.dt = time step of simulation
23
24 %   Outputs:
25
26 %   result.x = x coordinate
27 %   result.y = y coordinate
28 %   result.z = z coordinate
29
30 %   Copyright 2012 Soft Matter Lab @ Bilkent University
31 %   Author: Mite Mijalkov
32 %   $Revision: 1.0 $  $Date: 2012/04/17$
33
34 % Simulation parameters
35
36 x0 = position.x0;
37 y0 = position.y0;
38 z0 = position.z0;
39
40 PX = parameter.PX;
41 PY = parameter.PY;
42 PZ = parameter.PZ;
43
44 N = simulation.N;
45 dt = simulation.dt;
46
47 %Simulation
48 x = zeros(1,N);
49 y = zeros(1,N);
50 z = zeros(1,N);
51
52 x(1) = x0;
53 y(1) = y0;
54 z(1) = z0;
55
```

```
for m = 2:N

    x(m) = x(m-1)+ PX * sqrt(dt) * randn(1);
    y(m) = y(m-1)+ PY * sqrt(dt) * randn(1);
    z(m) = z(m-1)+ PZ * sqrt(dt) * randn(1);

end

%   Output:

result.x = x;
result.y = y;
result.z = z;
```

Listing C.1: Random walk in 3D

C.2 Brownian motion in 2D

```
function result = BM(particle, simulation)

%   BM(particle,simulation) simulates the motion of a particle
    performing
%   Brownian motion

%   Inputs:

%   particle = Structure with information about the physical
    properties of
%   the particle
%   particle.R = particle radius [m]
%   particle.eta = fluid viscosity [Ns/m^2] -- water = 0.001
%   particle.T = temperature [K]

%   simulation = Structure with information about the simulation
    parameters
%   simulation.dt = timestep [s]
%   simulation.N = number of samples
%   simulation.x0 = initial x-coordinate [m]
%   simulation.y0 = initial y-coordinate [m]

```

```
20 %   Outputs:
21
22 %   result = Structure with the simualted trajectory
23 %   result.particle = Structure with information about the physical
       properties of the particle
24 %   result.simulation = Structure with information about the
       simulation parameters
25 %   result.x = x coordinate [m]
26 %   result.y = y coordinate [m]
27
28 %   Copyright 2012 Soft Matter Lab @ Bilkent University
29 %   Author: Mite Mijalkov
30 %   $Revision: 1.0 $  $Date: 2012/03/08$
31
32 % Constants:
33 kB = 1.38e-23; % Boltzmann constant [J/K]
34
35 % Physical Parameters
36 R = particle.R; % radius [m]
37 eta = particle.eta; % fluid viscosity [Ns/m^2] -- water = 0.001
38 T = particle.T; % temperature [K]
39
40 % Derived quantities
41 gamma = 6*pi*eta*R; % friction coefficient [Ns/m]
42 Dt = kB*T/gamma; % diffusion [m^2/s]
43
44 % Simulation Parameters
45 dt = simulation.dt; % timestep [s]
46 N = simulation.N; % number of samples
47 x0 = simulation.x0; % initial x-coordinate [m]
48 y0 = simulation.y0; % initial y-coordinate [m]
49
50 % Simulation
51
52 x = zeros(1,N);
53 x(1) = x0;
54 y = zeros(1,N);
55 y(1) = y0;
56
57 for n = 1:1:N-1
```

```
58
59      x(n+1) = x(n) + sqrt(2*Dt*dt)*randn(1);
60      y(n+1) = y(n) + sqrt(2*Dt*dt)*randn(1);
61
62  end
63
64  % Output
65
66  result.particle = particle;
67  result.simulation = simulation;
68  result.x = x;
69  result.y = y;
```

Listing C.2: Brownian motion in 2D

C.3 Active Brownian motion in 2D

```
1  function result = ABM_2D(particle,simulation)
2  %   ABM(particle,simulation) simulates the motion of an active
       brownian
3  %   particle
4
5  %   Inputs:
6
7  %   particle = Structure with information about the physical
       properties of the particle
8  %   particle.R = particle radius [m]
9  %   particle.eta = fluid viscosity [Ns/m^2] -- water = 0.001
10 %   particle.T = temperature [K]
11 %   particle.v = velocity [m/s]
12
13 %   simulation = Structure with information about the simulation
       parameters
14 %   simulation.dt = timestep [s]
15 %   simulation.N = number of samples
16 %   simulation.x0 = initial x-coordinate [m]
17 %   simulation.y0 = initial y-coordinate [m]
18 %   simulation.phi0 = initial angle with in-plane orientation of the
19 %   particle [rad]%
```

```
20 %   simulation.theta0 = initial angle with out of plane orientation
       of the
21 %   particle [rad]
22
23 %   Outputs:
24
25 %   result = Structure with the simualted trajectory
26 %   result.particle = Structure with information about the physical
       properties of the particle
27 %   result.simulation = Structure with information about the
       simulation parameters
28 %   result.x = x coordinate [m]
29 %   result.y = y coordinate [m]
30 %   result.phi = angle with in-plane orientation of the particle [
       rad]
31 %   result.theta = angle with out of plane orientation of the
       particle [rad]
32
33 %   Copyright 2012 Soft Matter Lab @ Bilkent University
34 %   Author: Mite Mijalkov
35 %   $Revision: 1.0 $  $Date: 2012/03/08$
36
37 % Constants
38 kB = 1.38e-23; % Boltzmann constant [J/K]
39
40 % Physical Parameters
41 R = particle.R; % radius [m]
42 eta = particle.eta; % fluid viscosity [Ns/m^2] -- water = 0.001
43 T = particle.T; % temperature [K]
44 v = particle.v;    % velocity [m/s]
45
46 % Derived quantities
47 gamma = 6*pi*eta*R; % friction coefficient [Ns/m]
48 Dt = kB*T/gamma; % diffusion [m^2/s]
49 Dr = kB*T/(8*pi*eta*(R^3)); % rotational diffusion [1/s]
50 % F = v*gamma;    % force [N]
51
52 % Simulation Parameters
53 dt = simulation.dt; % timestep [s]
54 N = simulation.N; % number of samples
```

```
55 x0 = simulation.x0; % initial x-coordinate [m]
56 y0 = simulation.y0; % initial y-coordinate [m]
57 phi0 = simulation.phi0; % initial angle with in-plane orientation of
       the particle [rad]
58
59 % Simulation
60
61 x = zeros(1,N);
62 x(1) = x0;
63 y = zeros(1,N);
64 y(1) = y0;
65 phi = zeros(1,N);
66 phi(1) = phi0;
67
68 for n = 1:1:N-1
69
70     %creating random increments
71     nphi = randn;
72     nx = randn;
73     ny = randn;
74
75     %angular part performs simple Brownian motion
76     phi(n+1) = phi(n) + sqrt(2*Dr*dt)*nphi;
77
78     % x and y components depend on the parameter v which in turn
       depend on
79     % the propulsion force
80     x(n+1) = x(n) + v*cos(phi(n+1))*dt + sqrt(2*Dt*dt)*nx;
81     y(n+1) = y(n) + v*sin(phi(n+1))*dt + sqrt(2*Dt*dt)*ny;
82
83 end
84
85 % Output
86 result.particle = particle;
87 result.simulation = simulation;
88 result.x = x;
89 result.y = y;
90 result.phi = phi;
```

Listing C.3: Active Brownian motion in 2D

C.4 Chiral active Brownian motion in complex environments

```
function result = CABMellipses(particle,ellipses,simulation)
% CABMellipses(particle, ellipses, simulation) simulates the motion of
% Chiral Active Brownian particle in the presence of ellipses

%   Inputs:
%
%   particle = Structure with information about the physical properties of the particle
%   particle.R = particle radius [m]
%   particle.eta = fluid viscosity [Ns/m^2] -- water = 0.001
%   particle.T = temperature [K]
%   particle.v = velocity [m/s]
%   particle.alpha = angular velocity [rad/s]
%
%   simulation = Structure with information about the simulation parameters
%   simulation.dt = timestep [s]
%   simulation.N = number of samples
%   simulation.x0 = initial x-coordinate [m]
%   simulation.y0 = initial y-coordinate [m]
%   simulation.phi0 = initial angle with in-plane orientation of the
%   particle [rad]
%
%   ellipses = Structure with the information about the ellipses that give
%   the boundary conditions
%   ellipses.a = axis of the ellipse that corresponds to x coordinate [m]
%   ellipses.b = axis of the ellipse that corresponds to y coordinate [m]
%   ellipses.xc = x coordinate of the center of the ellipse [m]
%   ellipses.yc = y coordinate of the center of the ellipse [m]
%   ellipses.csi = the angle about which the ellipse is rotated [rad] it is
%   measured in counterclockwise direction from the x axis
```

```
30 %   ellipses.io = parameter which specifies which boundary condition
       to check,
31 %   where +1 means that particle is confined inside; -1 stands for
       outside
32 %
33 %   Output:
34 %
35 %   result = Structure with the simualted trajectory
36 %   result.particle = Structure with information about the physical
       properties of the particle
37 %   result.simulation = Structure with information about the
       simulation parameters
38 %   result.x = x coordinate [m]
39 %   result.y = y coordinate [m]
40 %   result.phi = angle with in-plane orientation of the particle [
       rad]
41
42 % Constants
43 kB = 1.38e-23; % Boltzmann constant [J/K]
44
45 % Physical Parameters
46 R = particle.R; % radius [m]
47 eta = particle.eta; % fluid viscosity [Ns/m^2] -- water = 0.001
48 T = particle.T; % temperature [K]
49 v = particle.v;    % velocity [m/s]
50 alpha = particle.alpha; % angular velocity [rad/s]
51
52 % Derived quantities
53 gamma = 6*pi*eta*R; % friction coefficient [Ns/m]
54 Dt = kB*T/gamma; % diffusion [m^2/s]
55 Dr = kB*T/(8*pi*eta*(R^3)); % rotational diffusion [1/s]
56 % F = v*gamma;    % force [N]
57
58 % Simulation Parameters
59 dt = simulation.dt; % timestep [s]
60 N = simulation.N; % number of samples
61 x0 = simulation.x0; % initial x-coordinate [m]
62 y0 = simulation.y0; % initial y-coordinate [m]
63 phi0 = simulation.phi0; % initial angle with in-plane orientation of
       the particle [rad]clear simualtion;
```

```
64
65 % Simulation
66 x = zeros(1,N);
67 x(1) = x0;
68 y = zeros(1,N);
69 y(1) = y0;
70 phi = zeros(1,N);
71 phi(1) = phi0;
72
73 for n = 1:1:N-1
74     phi(n+1) = phi(n) + alpha*dt + sqrt(2*Dr*dt)*randn;
75     x(n+1) = x(n) + v*sin(phi(n+1))*dt + sqrt(2*Dt*dt)*randn;
76     y(n+1) = y(n) + v*cos(phi(n+1))*dt + sqrt(2*Dt*dt)*randn;
77
78
79     for ellipse = ellipses
80         a = ellipse.a;
81         b = ellipse.b;
82         xc = ellipse.xc;
83         yc = ellipse.yc;
84         csi = ellipse.csi;
85         io = ellipse.io;
86
87         % Final point in standard coordinate system
88         xt1 = (x(n+1)-xc)*cos(-csi) - (y(n+1)-yc)*sin(-csi);
89         yt1 = (x(n+1)-xc)*sin(-csi) + (y(n+1)-yc)*cos(-csi);
90
91         if ( io>0 & (xt1/a)^2+(yt1/b)^2>1 ) || ( io<0 & (xt1/a)^2+(
    yt1/b)^2<1 )
92
93             % Initial point in standard coordinate system
94             xt0 = (x(n)-xc)*cos(-csi) - (y(n)-yc)*sin(-csi);
95             yt0 = (x(n)-xc)*sin(-csi) + (y(n)-yc)*cos(-csi);
96
97             % Slope of line between initial and final point
98             m = (yt1-yt0)/(xt1-xt0);
99
100            % Intersections with ellipse
101            y0 = yt1-xt1*m;
102
```

```
103             xi1 = ( -y0*m/b^2 + sqrt( (y0*m/b^2)^2 - (a^-2+m^2/b^2)
    *(y0^2/b^2-1) ) )/(a^-2+m^2/b^2);
104             xi2 = ( -y0*m/b^2 - sqrt( (y0*m/b^2)^2 - (a^-2+m^2/b^2)
    *(y0^2/b^2-1) ) )/(a^-2+m^2/b^2);
105             yi1 = y0+xi1*m;
106             yi2 = y0+xi2*m;
107 
108             % Chooses the closest intersection to the final point
109             if (xi1-xt1)^2+(yi1-yt1)^2 < (xi2-xt1)^2+(yi2-yt1)^2
110                 xi = xi1;
111                 yi = yi1;
112             else
113                 xi = xi2;
114                 yi = yi2;
115             end
116 
117             % Slope of the tangent and normal to the ellipse at the
    intersection point
118             m_tan = -xi/yi*b^2/a^2;
119             m_norm = -m_tan^-1;
120 
121             % Calculate the projection on the ellipse
122             xf = (yt1-yi+m_tan*xi-m_norm*xt1)/(m_tan-m_norm);
123             yf = yi+m_tan*xf-m_tan*xi;
124 
125             % Projection in intial coordinate system
126             x(n+1) = xf*cos(csi) - yf*sin(csi) + xc;
127             y(n+1) = xf*sin(csi) + yf*cos(csi) + yc;
128         end
129     end
130 
131 end
132 
133 % Output
134 result.particle = particle;
135 result.simulation = simulation;
136 result.x = x;
137 result.y = y;
```

```
138 result.phi = phi;
```

Listing C.4: Chiral active Brownian motion in complex environments (elliptical obstacles) in 2D