Abstract of thesis entitled

Run-and-tumble motion and differential dynamic microscopy

submitted by

Zhao, Yongfeng

for the Degree of Doctor of Philosophy
at The University of Hong Kong
in July 7, 2017

Systems biology focuses on how the biochemical regulations determine the functions of biological systems. Nowadays we are able to test many systems biology hypotheses with synthetic biology, and push the knowledge of biological systems into a new level. Pattern formation and collective motion are important functions for some biological systems, as they help explain the morphogenesis of multicellular organisms. Previous work has shown that patterns can emerge from the regulations of cell motion. However, the precise underlying mechanisms remain an open question.

I attacked this question for *E. coli* systems. I first investigated the motion of *E. coli*, which can be described as run-and-tumble motion. It consists of straight runs alternating with frequent random tumbling. The run-and-tumble motion in liquid has been well studied. I revised the previous results, and showed the effect of density-dependent motility regulations on *E. coli* collective motion. We found density-enhanced motility causes cells to segregate from signals, while density-inhibited motility causes cells to co-migrate with signals. We also found two species with mutual activation of motility can form segregating patterns, while mutual inhibition of motility helps form co-migrating patterns.

However, the environment for *E. coli* motion is often some polyporous materials, with a lot of obstacles. How the obstacles affect the macroscopic motion of microbes is still an open question. Thus a simplified stochastic model described by master equation had been established to describe cells moving in agar, and lattice Monte-Carlo simulations have been carried out. This model showed the same feature as the observations in experiments that cells with higher tumbling rate can move faster in semi-solid agar, in contrast to the case in liquid. The model suggested the mechanism to be that running cells are more easily blocked by the obstacles, while tumbling events help cells escape from the blocking obstacles.

Another problem in studying motion of *E. coli* is how to measure their motion in experiments. A novel method called differential dynamic microscopy (DDM), first used to characterize the diffusion of colloids, was adapted to measure parameters of run-and-tumble motion. DDM makes use of the information contained in the auto-correlation function of Fourier transformed time-lapse images, which can be compared with their theoretical predictions for run-and-tumble particles to extract relevant experimental parameters. The difficulty of lacking a mathematical expression of the solution in real time was overcome by a numerical inverse Laplace transform, making use of the classic Weeks' method.

To test this method, data generated from simulations were used, showing that for ideal run-and-tumble particles, both swimming speed and tumbling rate can be measured. Experimental data using AB1157 *E. coli* have been analysed, and the corresponding parameters were measured. The wild type and CheY knock-out strains were compared, and their difference supports the utility of our new development.

(An abstract of 451 words)

Run-and-tumble motion and differential dynamic microscopy

by

Zhao, Yongfeng

B. Sc., Peking University, China

A thesis submitted in partial fulfilment of the requirements for

the Degree of Doctor of Philosophy

at The University of Hong Kong

July 7, 2017

To my dearest Mengfei,

who gives me a whole world

and enlightens my life in the darkness.

Declaration

I hereby declare that this thesis represents my own work, except where due acknowledgement is made, and that it has not been previously included in a thesis, dissertation or report submitted to this University or to any other institution for a degree, diploma or other qualification.

Zhao, Yongfeng

Acknowledgement

Firstly, I would like to thank my supervisors Prof. Zhang Fu-Chun and Prof. Cheng Kwong-Sang. They were always supportive in my projects, providing the freedom for me to do projects in biophysics.

Special thank to Prof. Huang Jian-Dong, and Dr. Huang Wei, who introduced me to the group, and provided the experimental platform to my research. They also provided many inspiring discussions. The projects cannot proceed without their support.

Special thank to Dr. Julien Tailleur, who brought me back to statistical physics, gave me the opportunity to do the DDM project, and helped review this thesis. And special thank to Dr. Adrian Daerr, who helped me with experimental setup, and taught me a lot in experimental physics. It's a pleasure to work with them.

Thanks to Dr. Hugues Chaté, Prof. Shen Shun-Qing, and Dr. Zhang Shi-Zhong, who are kind to be my thesis examiners. They helped review this thesis and provided many useful suggestions.

I also need to thank Prof. Terrence Hwa, Prof. Peter Lenz, Dr. Guo Chin-Lin and Prof. Tang Lei-Han. They helped a lot in my early research work, and their ideas were valuable. I learned how to do science from them.

Thank to my undergraduate supervisor Dr. Li Fang-Ting, who not only brought me to the area of biophysics, but also gave some suggestions in my PhD research.

Thank to all the current and former members in JD lab. Thanks to Dr. Yu Bin, Dr. Xue Wen-Qian, Dr. Ni Yan-Xiang, Dr. Lin Qiu-Bin, Dr. Lu Xi-Bin, Dr. Hu Jian-Jiang, Dr. Wang Xiao-Lei, Dr. Liu Li-Zhong, Dr. Tang Xu-Ming, Mr. Zhang Bao-Zhong, Ms. Xu Chen, Mr. Fan Wei-Min, Mr. Duan Zhi-Gang, Ms. Smriti Arya, Mr. Feng Zhen-Dong. Special thank to Mr. Zhou Nan for all the discussions and the *E. coli* experiments. Special thank to our technician Mr. Anthony Tsang, who spent a lot of time in helping us buy the stuff that was not easy to find in a biological lab.

Thank to Mr. Zhang Dan-Bo and Mr. Tang Gao-Min from Department of Physics for many helps in department and being my room mates. Thanks to Ms. Michelle Lo from department office for all the helps. I Also need to thank to my room mates Mr. Li Nai-Lin and Mr. Li Liang-Hao.

Thank to all current and former members in MSC. Thanks to Dr. Alexandre Solon, Ms. Agnese Curatolo, Mr. Mourtaza Kourbane-Houssene, Ms. Solene Song, Dr. Lina Hamouche, who helped me a lot during my visits in Paris.

Thank to my girlfriend Dr. Liu Mengfei, whose support helped me go through the most tough days during my PhD study.

Thank to all my friends back in PKU. They are always helpful and supportive.

Thank to my family, who are always supportive in my PhD study.

Contents

List of figures

List of tables

Abbreviations and symbols

E. coli	*Escherichia coli*
CW	Clockwise
CCW	Counter clockwise
DDM	Differential dynamic microscopy
ISF	Intermediate scattering function
RDP	Run-and-diffuse with a distribution on velocity
RT	Run-and-tumble
RTD	Run-and-tumble with Brownian diffusion
RTDP	Run-and-tumble with Brownian diffusion and a distribution on velocity
RTDTT	Run-and-tumble with Brownian diffusion and tumbling duration
RTDPTT	Run-and-tumble with Brownian diffusion, a distribution on velocity, and tumbling duration

$\boldsymbol{a}$, a_i	Vector
$\hat{\boldsymbol{a}}$, $\hat{a}_i$	Unit vector, with the same direction of $\boldsymbol{a}$
∇, ∇_i	Gradient operator
∂_i	Differential operator $\partial/\partial x_i$
$\boldsymbol{a} \cdot \boldsymbol{b}$, $a_i b_i$	Scaler production
$\hat{L}$	Operator

α	Fraction of running particles
v	Speed of particle
$\bar{v}$	Mean speed of particles
σ	Standard deviation of particle speed
D	Brownian diffusion coefficient
λ	Tumbling rate
δ	Tumbling duration (tumbling time)
D_r	Rotational diffusion coefficient
D_{ra}	Active rotational diffusion coefficient

Chapter 1

Introduction

1.1 Systems biology and synthetic biology

The development of molecular biology has brought the study of biological systems into a new era. Since the discovery of the central dogma of molecular biology by Francis Crick in 1956 [1], biologists have devoted a lot of efforts in finding the mechanisms of underlying biological processes at the molecular level. Researchers found that regulation pathways comprising protein-protein interactions and protein-DNA interactions are critical in understanding biological process [2]. Until now, many protein interactions and regulation pathways have been discovered, and have been recorded in databases for researchers to study [3, 4, 5]. And the number of discovered pathways is still increasing.

However, the study of biology will not end up with finding all the pathways in biological systems. The pathways are coupled and connected, forming complex network with feedbacks and feedforwards, which cannot be understood simply and intuitively. This is where the **systems biology** begins to play a role. Instead of finding new specific pathways, systems biology aims to find the universal principles behind biological systems, to understand how the topology of regulation network determines its function [6].

Studying complex networks is not easy, especially for large networks, as they are usually many-body problems with strong and complex interactions. A natural idea is borrowed from electrical engineering to deal with such networks. In much the same way as one treats electric circuits, we try to decompose the networks into small but functional modules called **motifs** [6]. With the knowledge of motifs, we can analyse the complex network easily, and even design artificial biological systems to have advanced functions. Although one may argue the existence of motifs in biological system from theoretical aspect [7], this idea is still worth exploring.

Theoretical work is needed in studying motifs, as even for small motifs, coupled feedbacks may be hard for people to understand intuitively. And theory can also help us to find the possible network topologies for a given function. Some motifs have already been identified in previous theoretical work [8, 9, 10]. Several quantitative methods have been developed, involving mathematical modeling and physics. Nonlinear dynamics was often used in describing biochemical reactions [6]. Probability theory played an important role in studying noise in biological systems [11]. To understand topics like immunology [12] and collective behaviour of cells [13], statistical physics is also needed.

As science, systems biology cannot live without experimental evidence. Observation and bioengineering can provide some of the evidence. These methods can be considered as reverse-engineering, as they are trying to understand the unknown functions of an existing system. But we can also perform forward-engineering to biology, which is called **synthetic biology** [14]. In synthetic biology, we design artificial genetic circuits to perform the functions we require. Such circuits can be simple and decoupled from the natural ones, and are easier to study in experiments. Hence synthetic biology is powerful in studying the network motifs, providing direct evidence to systems biology, and helping us to understand the natural biological systems. The potential of synthetic biology is vast [15].

1.2 Morphogenesis and pattern formation

Morphogenesis is one of the topics that systems biologists and synthetic biologists are interested in. Almost every multicellular organism develops from a single cell into a large group of cells with spatial inhomogeneous structure with regular patterns, such as digits, sweat pores, lung, and kidney [16, 17]. How the symmetry is broken during morphogenesis is of great interest of biologists and physicists.

In physics, such transition from a spatial homogeneous state into an inhomogeneous state is called **pattern formation**, and is particularly rich in systems far away from equilibrium [18]. As dissipative systems, pattern formation systems need to exchange energy and entropy with environment to remain stable. Besides the biological systems, patterns are quite common in convection systems, diffusion-reaction systems, and nonlinear optics [18]. A well-known example is the Rayleigh-Bénard convection, which appears when a fluid is placed between two flat horizontal plates of constant temperature, but with a finite temperature difference between the plates. The temperature difference and gravity drive fluid convection, which results in cell-like patterns.

Alan Turing in his famous paper in 1952 [19] provided a possible mechanism for pattern formation in morphogenesis, which is the famous Turing pattern. The Turing pattern contains an "inhibitor" and an "activator", forming a negative feedback loop. If the diffusion of the inhibitor is slower than the activator, the spatial homogeneous state may lose its stability and a pattern will emerge. Recently, a group reported that the formation of digits may be a result of the Turing pattern [20].

Besides the Turing pattern, the regulation of cell motion can also result in a spatial inhomogeneous structure, like the previous work in M.E.Cates, 2010 [21] and in Chenli Liu, 2011 [22]. If the motion of a given cell depends on the rest of the population via a signalling molecule, and gets inhibited in the high density region, the spatial homogeneous state will lose its stability. A ring pattern has been observed in an engineered *E. coli* system in experiments [22]. Another example for the mammalian cell is the formation of a stripe pigment pattern in zebrafish. Researchers [23] suggested the stripe pattern was formed via cell-cell interactions of motility. Following these ideas, the final objective of this work is to explore the relation between motility regulations and patterns in *E. coli* systems. But firstly, we need to understand the motion of *E. coli*.

1.3 *E. coli* motion and run-and-tumble model

The study on the motion of *E. coli* in liquid dates back to 1972 [24]. It is often described as **run-and-tumble motion**. H.C. Berg tracked the motion of *E. coli* in three dimensions in liquid. He found that the trajectory of the cell consists of successive line segments. In each segment, *E. coli* moves with almost constant speed and direction, and then randomly "tumbles" to change its direction. He also measured the mean speed and the mean run length, and found them to be dependent on the specific strain of *E. coli*. He reported the mean speed to be of the order of 10µm/s, the mean time between two tumbles of the order of 1s, and the tumbling duration to be around 0.1s. In the later research, people found that the change of the cell direction in tumbling events is biased, with a peak around 60° [25].

The biochemical process controling the *E. coli* motion has been identified in early years, as byproducts of studying bacterial chemotaxis [26]. *E. coli* is propelled by its flagella, which consists of several helical flagellar filaments [27]. Each filament connects with a molecular motor which can spin either clockwise (CW) or counter clockwise (CCW). If the motors spin CCW, the filaments can form a bundle and propel the cell along a relatively straight trajectory. Otherwise, the filaments cannot form a bundle, and the forces exerted by the filaments pushing the fluid compete randomly, distracting the cell and resulting in a sudden tumble, as shown in Figure 1.1.

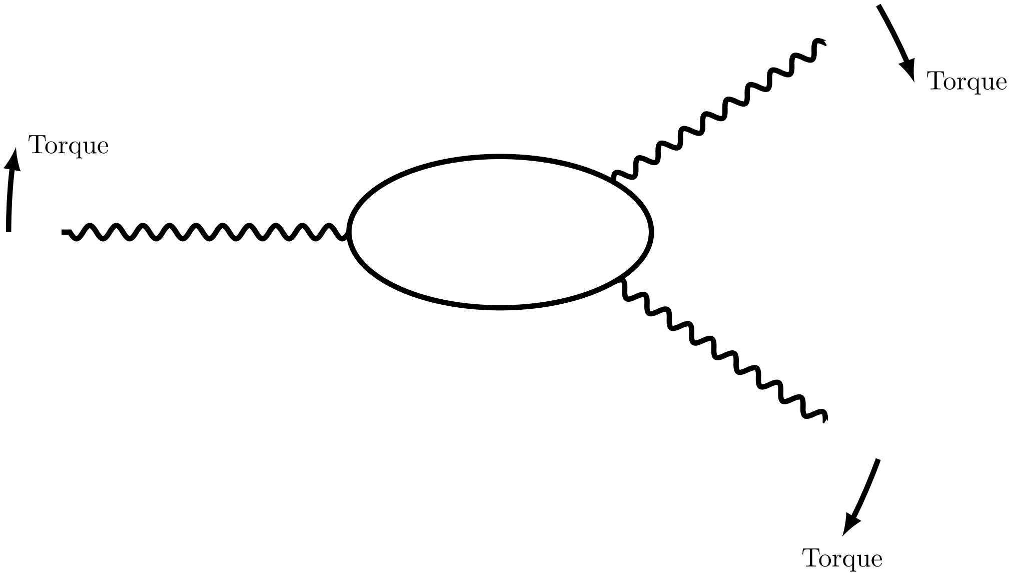


Figure 1.1: A sketch of a tumbling *E. coli* cell. The clockwise spinning filaments provide a global torque, causing the cell to rotational diffuse actively.

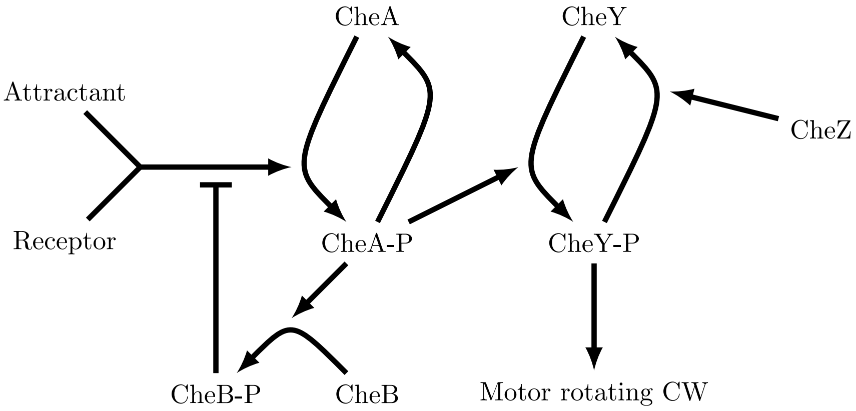


Figure 1.2: A sketch of biochemical pathways of *E. coli* chemotaxis.

The underlying pathways are sketched in Figure 1.2. The switch of the rotating direction of the motors between CW and CCW is controlled by a signalling protein called CheY [27, 28, 29]. The phosphorylated CheY can bind to the motors and change their direction from CCW to CW, switching the cell from running to tumbling. The CheY phosphorylation process can be activated by the phosphorylated CheA, and CheA is phosphorylated by the upstream signal from chemoattractants or chemorepellents. The phosphorylated CheA can also activate another protein CheB to inhibit the activation of CheA, forming a negative feedback loop. This kind of negative feedback loop with buffer node has been identified as a network motif for an adaptive response [8], making the phosphorylation of CheY to be a transient process [30]. Another chemotactic signalling protein CheZ can help dephosphorylate CheY and make the cell recover quickly from tumbling. Thus the tumbling events last for a short time with the mean duration around 0.1s.

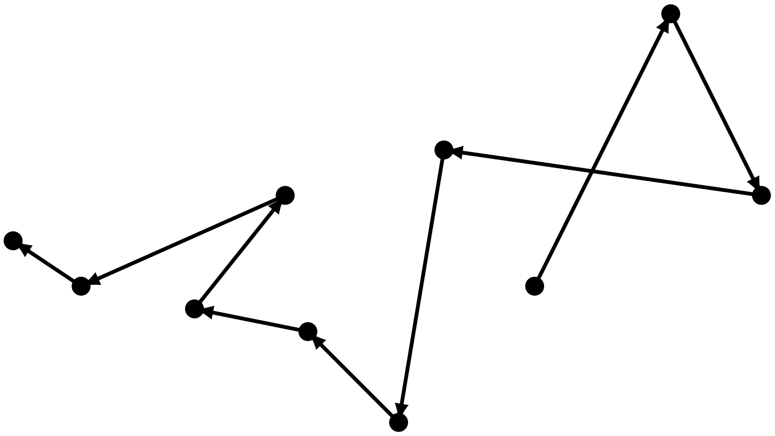


Figure 1.3: A sketch of run-and-tumble model.

According to these findings, a simplified run-and-tumble microscopic model can be established, as sketched in Figure 1.3. In this model, the particle is running with a constant speed, and keeping the direction during the running phase. The tumble events are measured by Berg, et al, 1972 [24] to be Poisson process in time. The tumbling events may take some time, or in a simple model be assumed as instantaneous. The distribution of the cell direction change after tumbling can be assumed isotropic in a simple model, or to obey some distribution function. With the microscopic models, we can coarse grain to have macroscopic models to describe the time evolution of particle density, and to study the pattern formation induced by motility regulations. The microscopic models also enable us to measure the parameters such as speed and tumbling rate, with the differential dynamic microscopy method.

1.4 Differential dynamic microscopy

How to quantitatively measure the microscopic motion of *E. coli* is an important problem in both the theoretical and the experimental study. Particle tracking has been developed as a feasible method [31]. However, it has certain limitations. Firstly, due to the difficulty of three-dimensional imaging, the tracking system is often restricted to two-dimensional systems, making it hard to directly study the three-dimensional systems. Secondly, the high density of particles will increase the chance of intersections of particle trajectories, making the tracking results inaccurate. But studying the high density systems can be important to understand their collective behaviors [22]. Finally, the constraints in cell density, image size, and computational cost will limit the sampling number to $\sim 10^3$, which may give a poor statistics. Especially for the three dimensional systems, the tracking microscopes need to track single cells, and the statistics will be horrible.

Differential dynamic microscopy (DDM) is a novel way of measuring the microscopic parameters of the motion of particles, first introduced in 2008 [32] to study diffusive colloidal particles, and later generalized to the study of bacterial motion by L. G. Wilson, et al [33] and V. A. Martinez, et al [34]. This method makes use of the information encoded in the auto-correlation function of time-lapse images, thus has a lot of advantages compared with the particle tracking. As long as nonlinear effect in the light absorption hasn't appeared, this method is valid for arbitrary high density of particles. So we can expect a higher sampling number ($> 10^4$) and a better statistics. The special equipment for this method is just a high speed camera with the frame rate of the order of 10^2. Then the processing of the data can be accomplished with our algorithm in any computer.

In this thesis I adapted the DDM method for the study of the bacterial motion, especially for *E. coli*. The previous works have successfully measured the speed of bacteria along with their diffusion constant due to thermal Brownian motion. In this thesis, I will present a method to also characterize the tumbling rate of *E. coli*.

1.5 Project objective and thesis structure

The final objective of this study is to investigate how the cell-cell interactions of motility regulate the bacterial pattern formation. In studying this question, we need to investigate the microscopic models of bacterial motion, and coarse grain the microscopic models to get macroscopic models for analysis. We also need to develop the experimental techniques for observation and measurement. Solving these problems is the major content of this thesis.

I have already introduced the background and the general concepts of this work. In Chapter 2, I will start from summarizing the previous work of a simple model for the run-and-tumble motion to present the techniques used in Chapter 3. In Chapter 3, I will provide new models to analyse the influence of tumbling duration and rotational diffusion to the run-and-tumble motion.

In Chapter 4, I will present some theoretical results of the run-and-tumble particles moving in the environment with obstacles, which hasn't been reported in the literature. As in pattern formation experiments, the cells are in the semi-solid agar which can trap the cells while moving, the influence of the obstacles need to be taken care of.

Chapter 5 will present the current progress and some preliminary results we have in studying the effects of the cell-cell interactions to the pattern formation. The originality of the theoretical work belongs to our collaborators. I worked for the fluorescence imaging setup in the experiments. As the project is still on-going when this thesis is finished, this chapter will be a brief introduction.

To characterize the motion of our engineered *E. coli* used in this study, I adapted the differential dynamic microscopy method to measure the tumbling rate and tumbling duration. Although it has been used for measuring Brownian particles and straight swimmers, I didn't find a rigid proof for the theory of the differential dynamic microscopy, so I will provide one in Chapter 6. This chapter will need the results in Chapter 2 and 3. Chapter 7 will present the tests of this method using the simulation data, and the preliminary experimental tests will be present in Chapter 8.

Finally, in Chapter 9 I will discuss the results and present the future plan of this study. The appendix will be the supplementary figures and the technical reference for the DDM algorithm and codes.

Chapter 2

A simple model for run-and-tumble particles

2.1 The master equation

We first consider the non-interacting particles, so that the probability distribution reduce to computing $p(\boldsymbol{x}, \hat{\boldsymbol{v}}, t)$ of finding a single particle at the position $\boldsymbol{x}$ and the time t, with the velocity pointing in the direction of the unit vector $\hat{\boldsymbol{v}}$, as the state of a particle is fully determined by its position and momentum. The motion of the particle can be considered as a continuous Markov process, so the time evolution of the probability density will satisfy the corresponding differential equation, called master equation.

For a simple model of the run-and-tumble particles with Brownian diffusion, in which we assume instantaneous and isotropic tumbling, the master equation is

$$\frac{\partial p(\boldsymbol{x}, \hat{\boldsymbol{v}}, t)}{\partial t} = \nabla\cdot[D\nabla p(\boldsymbol{x}, \hat{\boldsymbol{v}}, t)] - \nabla\cdot[\boldsymbol{v} p(\boldsymbol{x}, \hat{\boldsymbol{v}}, t)] - \lambda p(\boldsymbol{x}, \hat{\boldsymbol{v}}, t) + \frac{\lambda}{\Omega}\int p(\boldsymbol{x}, \hat{\boldsymbol{v}}', t)\, d\Omega' , \tag{2.1}$$

where ∇ acts on the spatial coordinates. The $\nabla \cdot (D\nabla p)$ term is given by the Fick's law of diffusion, with diffusion coefficient to be D. $-\nabla \cdot (\boldsymbol{v}p)$ is the contribution of the straight motion. Here $\boldsymbol{v}$ may be a function of the spatial coordinates. $-\lambda p$ is the rate at which a particle changes its heading into another one, and $\frac{\lambda}{\Omega}\int p\, d\Omega'$ is the rate at which the particles from all other directions changing back to the direction of $\boldsymbol{v}$. Here $\lambda \int p\, d\Omega'$ is the fraction of particles that is tumbling at position $\boldsymbol{x}$ and time t. As we are assuming isotropic tumble, the tumbling particles has uniform probability to enter a random direction, so this term is divided by Ω, which denotes the total solid angle, for the rate of a specific direction $\hat{\boldsymbol{v}}$. $d\Omega$ is the infinitesimal element of solid angle. In three-dimensional space with spherical coordinates, $\Omega = 4\pi$ and $d\Omega = \sin\theta \, d\theta d\phi$.

By giving an initial probability distribution and a proper boundary condition, the time evolution of the probability distribution function is determined by the master equation, and gives the description of the particle motion. For N particles, we can formally define the macroscopic density of the particles ρ_M to be

$$\rho_M(\boldsymbol{x}, t) = Np(\boldsymbol{x}, t) = N \int p(\boldsymbol{x}, \hat{\boldsymbol{v}}', t) \, d\Omega' \,, \tag{2.2}$$

where $p(\boldsymbol{x}, t)$ is the spatial probability distribution. If the direction of v is labelled in three-dimensional spherical coordinates, the last integration reads

$$\int_0^{2\pi} d\phi \int_0^{\pi} p(\boldsymbol{x}, \theta, \phi, t) \sin\theta \, d\theta \,. \tag{2.3}$$

2.2 Moment expansion and diffusive limit

In practice, we are often more interested in the time evolution of the macroscopic particle density, rather than the full microscopic probability distribution. Direct integration of (2.1) is difficult, and an explicit closed exact equation for the density field does not exist. But for the large spatio-temporal scales, we can seek for a diffusive limit of (2.1) to obtain an effective equation of the density field.

Here I follow the method in M.E. Cates, 2013 and A.P. Solon, 2015 [35, 36]. The idea is to expand the angular dependence of $p(\boldsymbol{x}, \hat{\boldsymbol{v}}, t)$ with respect to $\hat{\boldsymbol{v}}$ to get an equation for the first order moment of p, and then do the approximation. The expansion reads (summation on the repeated indices)

$$p = \frac{\rho_M}{\Omega} + J_i \hat{v}_i + Q_{ij} \left(\hat{v}_i \hat{v}_j - \frac{\delta_{ij}}{d} \right) + \Theta[p] \,. \tag{2.4}$$

The basis tensors (the bracket in the subscript means summation over all the permutations of the indices inside)

$$M_{i_1 i_2 \cdots i_l} = \hat{v}_{i_1} \hat{v}_{i_2} \cdots \hat{v}_{i_l} + \sum_{j=1}^{[l/2]} \frac{(-1)^j \Gamma\left(l + \frac{d}{2} - j - 1\right)}{j!(l-2j)! 4^j \Gamma\left(l + \frac{d}{2} - 1\right)} \delta_{(i_1 i_2} \cdots \delta_{i_{2j-1} i_{2j}} \hat{v}_{i_{2j+1}} \cdots \hat{v}_{i_l)} \tag{2.5}$$

are orthogonalized from the l-th power functions of $\hat{\boldsymbol{v}}$, and are linear combinations of the d-dimensional spherical harmonics with the same degree of l. Thus they are eigenfunctions of the Laplace-Beltrami operator on the $d - 1$ dimensional unit sphere $\Delta_{\hat{\boldsymbol{v}}}$ with the eigenvalues $l(l + d - 2)$.

Noticing the following relations:

$$\int \hat{v}_i \, d\Omega = 0 \,, \tag{2.6}$$

$$\int \hat{v}_i \hat{v}_j \, d\Omega = \frac{\Omega}{d} \delta_{ij} \,, \tag{2.7}$$

$$\int \hat{v}_i \hat{v}_j \hat{v}_k \, d\Omega = 0 \,, \tag{2.8}$$

$$\int \hat{v}_i \hat{v}_j \hat{v}_k \hat{v}_l \, d\Omega = \frac{\Omega}{d(d+2)} (\delta_{ij}\delta_{kl} + \delta_{jk}\delta_{il} + \delta_{ik}\delta_{jl}) \,, \tag{2.9}$$

$$\cdots$$

$$\int \hat{v}_{i_1} \hat{v}_{i_2} \cdots \hat{v}_{i_{2l-1}} \, d\Omega = 0 \,, \tag{2.10}$$

$$\int \hat{v}_{i_1} \hat{v}_{i_2} \cdots \hat{v}_{i_{2l}} \, d\Omega = \frac{\Gamma\left(d/2\right) \Gamma\left(l + 1/2\right) \Omega}{(2l)! \sqrt{\pi} \Gamma(l + d/2)} \delta_{(i_1 i_2} \cdots \delta_{i_{2l-1} i_{2l})} \,, \tag{2.11}$$

we can multiply $M_{i_1 i_2 \cdots i_l}$ to the master equations (2.1) and do the integration on all the directions of $\hat{\boldsymbol{v}}$. For the first three terms in the expansion (2.4), we have

$$\frac{\partial \rho_M}{\partial t} = - \frac{\Omega}{d} \nabla_i \left(v J_i\right) + \nabla_i (D \nabla_i \rho_r) \,, \tag{2.12}$$

$$\frac{\partial J_i}{\partial t} = - \frac{1}{\Omega} \nabla_i (v \rho_M) - \frac{2}{d+2} \nabla_j \left(v Q_{ij}\right) + \nabla_j \left(D \nabla_j J_i\right) - \lambda J_i \,, \tag{2.13}$$

$$\frac{\partial Q_{ij}}{\partial t} = - \frac{d+2}{2} B_{ijkl} \nabla_k \left(v J_l\right) + \nabla_k \left(D \nabla_k Q_{ij}\right) - \lambda Q_{ij} - \nabla_k \left(v \chi_{ijk}\right) \,, \tag{2.14}$$

with

$$B_{ijkl} = \frac{1}{d+2} \left(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk} - \frac{2}{d} \delta_{ij}\delta_{kl} \right) \,, \tag{2.15}$$

$$\chi_{ijk} = \int M_{ij} \hat{v}_k \Theta[p] \, d\Omega \,. \tag{2.16}$$

In the large temporal scales ($t \gg 1/\lambda$), we can assume the moments of order higher than one, which are the rapid modes, to reach their steady states. $\partial Q_{ij}/\partial t = \partial J_i/\partial t = 0$. Then from (2.14), we have $Q_{ij} \sim o(\nabla)$. From (2.13), we have

$$\begin{aligned} J_i =& - \frac{1}{\lambda\Omega} \nabla_i (v \rho_M) - \frac{2}{(d+2)\lambda} \nabla_j (v Q_{ij}) + \frac{1}{\lambda} \nabla_j (D \nabla_j J_i) \\ =& - \frac{1}{\lambda\Omega} \nabla_i (v \rho_M) + o(\nabla^2) \,. \end{aligned} \tag{2.17}$$

In the large spatial scales, the spatial derivatives of order higher than $o(\nabla^2)$ can also be neglected. Thus we can get a closed partial differential equation for the density field

$$\frac{\partial \rho_M}{\partial t} = \nabla_i \left[\frac{v}{d\lambda} \nabla_i (v \rho_M) \right] + \nabla_i (D \nabla_i \rho_r) \,, \tag{2.18}$$

which is the well-known [37] diffusive limit of the original master equation.

If all the parameters are independent of the spatial coordinates, we find the effective diffusion coefficient of the run-and-tumble particles in d-dimensional space

$$D_{\text{eff}} = D + \frac{v^2}{\lambda d}\,. \tag{2.19}$$

Without tumbling, the particles should show ballistic motion with the mean square displacement $\langle \Delta x^2 \rangle \propto t^2$, and have an infinite effective diffusion coefficient. However, the tumbling events slow down the motion of the particles in the large temporal scales from $\langle \Delta x^2 \rangle \propto t^2$ to $\langle \Delta x^2 \rangle \propto t$, and a finite effective diffusion coefficient appears. Thus a run-and-tumble particle resembles a Brownian particle in the large spatio-temporal scales, with an effective diffusion coefficient dependent on the dimension of the space.

2.3 Chemotaxis flux

Here I would like to present an example to show what information we can obtain from the diffusive limit. Comparing (2.18) with the continuity equation

$$\frac{\partial \rho_M}{\partial t} = -\nabla \cdot \boldsymbol{J}\,, \tag{2.20}$$

we have the flux of the particles to be

$$\boldsymbol{J} = -D\nabla\rho_M - \frac{v\nabla(v\rho_M)}{\lambda d}\,. \tag{2.21}$$

The first term is still the contribution from the Fick's law of diffusion. The second term, if v depends on the concentration of some chemical $a(\boldsymbol{x})$, which is a function of the spatial coordinates $v = v(a) = v(a(\boldsymbol{x}))$, can be written as

$$\frac{v\nabla(v\rho_M)}{\lambda d} = \frac{v^2}{\lambda d}\nabla\rho_M + \frac{vv'(a)\rho_M}{\lambda d}\nabla a \tag{2.22}$$

The first term is a contribution to diffusion from the run-and-tumble mechanism. However, the second term shows a flux of particles proportional to the gradient of an external field

$$\boldsymbol{J}_{\text{chemo}} = -\frac{vv'(a)\rho_M}{\lambda d}\nabla a\,. \tag{2.23}$$

If $v'(a) > 0$, the chemical increases the speed of particles, we have $\boldsymbol{J}_{\text{chemo}} \propto -\nabla a$. The particle will move in the opposite direction with the increasing chemical concentration, causing particles to move towards the low concentration area. Similarly, if $v'(a) < 0$, the particles will move towards the high concentration area. Hence, it shows a possible mechanism of chemotaxis. Whether a chemical can increase or decrease the particle speed determines whether the chemical is chemoattractant or chemorepellent.

Alternatively, a direction dependent tumbling rate $\lambda = \lambda(\hat{v}) = \lambda_0 + \lambda_1 \hat{\boldsymbol{v}} \cdot \nabla a$ can also bring a flux proportional to the gradient of the chemotactant and provide a possible mechanism for chemotaxis.

2.4 Solution in three-dimensional free space

A solution of (2.1) can give us all the information of the system, but it is hard to solve the general equation without assumptions. Luckily, If D, v, and λ in (2.1) are constants, we can get a solution of the master equation in the free space, which means the size of the system is taken to be infinity and $\lim_{\boldsymbol{x}\to\infty} p(\boldsymbol{x}, \hat{\boldsymbol{v}}, t) = 0$. We take the isotropic initial condition to be proportional to the Dirac delta function, $p(\boldsymbol{q}, \hat{\boldsymbol{v}}, t = 0) = \delta(x)/\Omega$. The initial condition is not only for simplicity. The practical significance of this initial condition will be seen in Chapter 6 in the theory of the differential dynamic microscopy.

Solving the master equation directly is often difficult. In the following discussion, we will consider the Fourier transformed distribution defined as

$$p(\boldsymbol{q}, \hat{\boldsymbol{v}}, t) = \int p(\boldsymbol{x}, \hat{\boldsymbol{v}}, t) e^{-i\boldsymbol{q}\cdot\boldsymbol{x}}\, d^3x\,, \tag{2.24}$$

$$p(\boldsymbol{x}, \hat{\boldsymbol{v}}, t) = \frac{1}{(2\pi)^3} \int p(\boldsymbol{q}, \hat{\boldsymbol{v}}, t) e^{i\boldsymbol{q}\cdot\boldsymbol{x}}\, d^3q\,, \tag{2.25}$$

and the Laplace transform defined as

$$p(\boldsymbol{x}, \hat{\boldsymbol{v}}, s) = \int_0^\infty e^{-st} p(\boldsymbol{x}, \hat{\boldsymbol{v}}, t)\, dt\,. \tag{2.26}$$

The functions in the real space and the Fourier/Laplace space are distinguished by their arguments.

After Fourier transform to the spatial coordinates and Laplace transform to the time, the master equation becomes

$$sp(\boldsymbol{q}, \hat{\boldsymbol{v}}, s) - \frac{1}{\Omega} = -Dq^2 p(\boldsymbol{q}, \hat{\boldsymbol{v}}, s) - i\boldsymbol{v}\cdot\boldsymbol{q}p(\boldsymbol{q}, \hat{\boldsymbol{v}}, s) - \lambda p(\boldsymbol{q}, \hat{\boldsymbol{v}}, s) + \frac{\lambda}{\Omega}\rho_M(\boldsymbol{q}, s)\,. \tag{2.27}$$

where $1/\Omega = p(\boldsymbol{q}, \hat{\boldsymbol{v}}, t)|_{t=0}$ is the initial condition. The technique of solving $\rho_M(\boldsymbol{q}, s) = \int p(\boldsymbol{q}, \hat{\boldsymbol{v}}, s)\, d\Omega'$ is similar to K. Martens, 2012 [38]. Firstly we rearrange the terms into

$$p(\boldsymbol{q}, \hat{\boldsymbol{v}}, s) = \frac{\lambda}{(s + Dq^2 + i\boldsymbol{q}\cdot\boldsymbol{v} + \lambda)\Omega}\rho_M(\boldsymbol{q}, s) + \frac{1}{(s + Dq^2 + i\boldsymbol{q}\cdot\boldsymbol{v} + \lambda)\Omega}\,. \tag{2.28}$$

Acting the operator $\int d\Omega$ on both side, and denoting

$$P = \frac{1}{\Omega} \int \frac{1}{s + Dq^2 + i\boldsymbol{q} \cdot \boldsymbol{v} + \lambda} \, d\Omega \,, \tag{2.29}$$

we can solve $\rho_M(\boldsymbol{q}, s)$ as

$$\rho_M(\boldsymbol{q}, s) = \frac{P}{1 - \lambda P} \,. \tag{2.30}$$

Then in the three-dimensional space, we set the direction of $\boldsymbol{q}$ to be the z-axis and the integration becomes $\int d\Omega = \int \sin\theta \, d\theta d\phi$, we can evaluate P to be

$$\begin{aligned} P &= \frac{1}{2} \int_{-1}^{1} \frac{1}{s + Dq^2 + iqv\cos\theta + \lambda} \, d\cos\theta \\ &= \frac{i}{2qv} \ln\left(\frac{s + Dq^2 + \lambda - iqv}{s + Dq^2 + \lambda + iqv} \right) \\ &= \frac{\arctan(qv/(s + Dq^2 + \lambda))}{qv} \,. \end{aligned} \tag{2.31}$$

So we have

$$\rho_M(\boldsymbol{q}, s) = \frac{\arctan(qv/(s + Dq^2 + \lambda))}{qv - \lambda \arctan(qv/(s + Dq^2 + \lambda))} \,, \tag{2.32}$$

which is independent of the direction of $\boldsymbol{q}$. Hence we get the Fourier-Laplace transformed solution of the particle density from the master equation, which is of more interest than the full probability distribution.

In addition, the full (but less interesting) probability distribution is given by

$$\begin{aligned} p(\boldsymbol{q}, \hat{\boldsymbol{v}}, s) =& \frac{1}{(s + Dq^2 + i\boldsymbol{q} \cdot \boldsymbol{v} + \lambda)\Omega} \cdot \frac{1}{1 - \lambda P} \\ =& \frac{1}{4\pi(s + Dq^2 + i\boldsymbol{q} \cdot \boldsymbol{v} + \lambda)} \cdot \frac{qv}{qv - \lambda \arctan(qv/(s + Dq^2 + \lambda))} \,. \end{aligned} \tag{2.33}$$

However, we cannot invert the Fourier-Laplace transformed solution analytically. If needed, we can implement the numerical inverse Laplace transform. The numerical details can be found in Section B.3.

2.5 Properties of the solution

In this section I will present the properties of solution (2.32), to give some insights of the run-and-tumble motion.

2.5.1 Mean square displacement

How far from the origin the particles move after the time t can be characterized by the its mean square displacement $\langle \Delta x^2(t) \rangle$. As

$$\langle \Delta x^2(s) \rangle = - \left. \frac{\partial^2 \rho_M(q,s)}{\partial q^2} \right|_{q=0} , \tag{2.34}$$

we have

$$\langle \Delta x^2(s) \rangle = \frac{2D}{s^2} + \frac{2v^2}{3s^2(s+\lambda)} \tag{2.35}$$

and the inverse Laplace transform gives

$$\langle \Delta x^2(t) \rangle = 2Dt + 2\left(\frac{v^2}{3\lambda} t + \frac{v^2}{3\lambda^2} e^{-\lambda t} \right) - \frac{2v^2}{3\lambda^2} . \tag{2.36}$$

If $t \ll 1/\lambda$, $\langle \Delta x^2(t) \rangle \approx v^2 t^2/3 + 2Dt$, apart from the Brownian diffusion term, the particles show ballistic motion. If $t \gg 1/\lambda$, the exponential term will vanish and the motion is diffusive as we expected. The diffusion coefficient is same with in Section 2.2.

2.5.2 Small length scale limit and straight swimmers

If q is sufficient large such that $Dq^2 \gg \lambda$ and $qv \gg 2\pi\lambda$, which means we are considering the system in the very short length scales, and the tumbles can be neglected. The particles that cannot tumble are called "straight swimmers". In this case, the solution reduces to

$$\rho_M(q,s) = \frac{1}{qv} \arctan \frac{qv}{s + Dq^2} , \tag{2.37}$$

We can do the inverse Laplace transform to it to get

$$\rho_M(q,t) = e^{-Dq^2 t} \frac{\sin(qvt)}{qvt} . \tag{2.38}$$

This is simply the solution of

$$\frac{\partial p(\boldsymbol{q}, \hat{\boldsymbol{v}}, \tau)}{\partial \tau} = -Dq^2 p(\boldsymbol{q}, \hat{\boldsymbol{v}}, \tau) - i\boldsymbol{v} \cdot \boldsymbol{q} p(\boldsymbol{q}, \hat{\boldsymbol{v}}, \tau) . \tag{2.39}$$

This result can also be obtained and understood in another way [33]. Apart from the contribution from diffusion, the microscopic density of a single swimmer is simply

$$\rho(\boldsymbol{x}, t) = \delta(\boldsymbol{x} - \boldsymbol{v}t) , \tag{2.40}$$

$$\rho(\boldsymbol{q}, t) = e^{-i\boldsymbol{q} \cdot \boldsymbol{v}t} . \tag{2.41}$$

Then we average on the direction of $\boldsymbol{q}$ in the three-dimensional space to give

$$\rho(q,t) = \frac{1}{4\pi}\int e^{-iqvt\cos\theta}\sin\theta\, d\theta d\phi = \frac{\sin(vqt)}{vqt}\,. \tag{2.42}$$

Then we multiply e^{-Dq^2t} which is the solution of the diffusion equation to get the solution, as the Brownian diffusion merely introduces a Gaussian spreading into the probability distribution.

2.5.3 Large length scale limit

Now we consider the system in the large length scales ($q \to 0$). We first rewrite (2.32) as

$$\rho_M(q,s) = \frac{1}{qv/\arctan(qv/(s+Dq^2+\lambda)) - \lambda} \tag{2.43}$$

and expand the term in the denominator with respect q, to get

$$\frac{qv}{\arctan(qv/(s+Dq^2+\lambda))} = s+\lambda+\left(D+\frac{v^2}{3(\lambda+s)}\right)q^2+o(q^4)\,, \tag{2.44}$$

and

$$\rho_M(q,s) = \frac{1}{s+(D+v^2/3(\lambda+s))\,q^2+o(q^4)}\,. \tag{2.45}$$

The inverse Laplace transformation can be carried out to get

$$\rho_M(q,t) = \frac{(\Delta+\lambda-Dq^2)e^{-t(Dq^2+\lambda-\Delta)/2}+(\Delta-\lambda+Dq^2)e^{-t(Dq^2+\lambda+\Delta)/2}}{2\Delta}\,. \tag{2.46}$$

with

$$\Delta = \sqrt{(\lambda-Dq^2)^2-\frac{4q^2v^2}{3}}\,. \tag{2.47}$$

It is valid for the sufficient small q (in practice, $q \sim 0.01\mu\mathrm{m}^{-1}$). As $\Delta < |\lambda - Dq^2| < \lambda + Dq^2$, we got a composition of two exponential decays with the different time scales. Considering $\Delta = \lambda - Dq^2 - 2v^2q^2/3\lambda + o(q^4)$, the rapid decay term is proportional to $e^{-\lambda t + v^2q^2t/3\lambda}$. For $q \ll \sqrt{3}\lambda/v$, the rapid decay term can be neglected for the long time scales. Thus we have the slow decay term which dominates in the long time scales,

$$\rho_M(q,t) \sim e^{-(D+v^2/3\lambda)q^2t}\,, \tag{2.48}$$

which is exactly the diffusive limit we have in Section 2.2, with the effective diffusion coefficient

$$D_{\text{eff}} = D + \frac{v^2}{3\lambda}\,. \tag{2.49}$$

The same result can be obtained by comparing the expansion

$$\rho_M(q,t) = \frac{1}{s + (D + v^2/3\lambda)q^2 + o(sq^2)} \tag{2.50}$$

and the Laplace-Fourier transformed solution for the diffusion equation $\frac{\partial\rho(\boldsymbol{x},t)}{\partial t} = D_{\text{eff}}\nabla^2\rho(\boldsymbol{x},t)$

$$\rho(\boldsymbol{q},t) = \frac{1}{s + D_{\text{eff}}q^2} \,. \tag{2.51}$$

Again we can be convinced that the run-and-tumble particles behave like the Brownian particles in the large temporal-spacial scales.

2.5.4 Sensitivity to parameters

As we will see in Chapter 6, we can compare the solution (2.32) with the experimental data to measure the parameters in the run-and-tumble motion. Thus a sensitivity analysis may be helpful to find a proper q range for measuring different parameters. We need (2.32) to be sensitive to the parameters in the q range that can be reached in the experiments, so that we can fit the experimental data efficiently.

To do the local sensitivity analysis, we follow the method developed by K.S. Brown, et al [39, 40, 41, 42]. Generally, for an arbitrary function of time $f(t;\boldsymbol{p})$, where $\boldsymbol{p}$ is the vector of the parameters, we define the cost function

$$C(\boldsymbol{p}) = \int_0^\infty [f(\tau;\boldsymbol{p}) - f(\tau;\boldsymbol{p}_0)]^2 d\tau \tag{2.52}$$

to study the sensitivity around the specific parameters $\boldsymbol{p}_0$. The idea is that the slow changing direction of $C(\boldsymbol{p})$ in the parameter space means the function is insensitive to the parameters change in that direction. So we expand $C(\boldsymbol{p})$ in the vicinity of $\boldsymbol{p}_0$, noticing $C(\boldsymbol{p}_0) = 0$ and $\partial C/\partial\boldsymbol{p}|_{\boldsymbol{p}=\boldsymbol{p}_0} = \boldsymbol{0}$, to get a quadratic form

$$C(\boldsymbol{p}_0 + \Delta\boldsymbol{p}) = \frac{1}{2}\sum_{i,j} \left.\frac{\partial^2 C}{\partial p_i \partial p_j}\right|_{p_i=p_{i0},p_j=p_{j0}} \Delta p_i \Delta p_j + o(\Delta p^3) \tag{2.53}$$

with the matrix

$$H_{ij} = \left.\frac{\partial^2 C}{\partial p_i \partial p_j}\right|_{p_i=p_{i0},p_j=p_{j0}} \,. \tag{2.54}$$

If an orthogonal matrix $\boldsymbol{V}$ bring $\boldsymbol{H}$ into a diagonal matrix through congruent transformation, defining $\boldsymbol{\eta} = \boldsymbol{V}\boldsymbol{p}$, and the cost function can be written as

$$C(\boldsymbol{p}_0 + \Delta\boldsymbol{p}) = \frac{1}{2}\sum_i (\boldsymbol{V}\boldsymbol{H}\boldsymbol{V}^T)_{ii}\Delta\eta_i^2 = \frac{1}{2}\sum_i \xi_i \Delta\eta_i^2 \,, \tag{2.55}$$

so the eigenvalue ξ_i of $\boldsymbol{H}$ will determine how sensitive the function is to the change in the direction of $\hat{\boldsymbol{\eta}}_i$, and the eigenvector corresponding to the least eigenvalue will determine the most insensitive direction in the parameter space.

As all the parameters p_i are non-negative and have the different order of magnitude, we may replace Δp_i by $p_{i0}\Delta p_i$ and rescale the matrix into

$$H_{ij} = p_{i0}p_{j0}\frac{\partial^2 C}{\partial p_i \partial p_j}\bigg|_{p_i=p_{i0},p_j=p_{j0}} \tag{2.56}$$

in the following discussion.

For many models we are interested in, an explicit expression of time t doesn't exist, but a Laplace transformed function of s exists. By direct substitution and noticing

$$[f(\tau;\boldsymbol{p}) - f(\tau;\boldsymbol{p}_0)]\frac{\partial^2 f(\tau;\boldsymbol{p})}{\partial p_i \partial p_j}\bigg|_{p_i=p_{i0},p_j=p_{j0}} = 0\,, \tag{2.57}$$

we have

$$\begin{aligned} H_{ij} &= p_{i0}p_{j0}\int_0^\infty \frac{\partial f(\tau;\boldsymbol{p})}{\partial p_i}\bigg|_{p_i=p_{i0}} \frac{\partial f(\tau;\boldsymbol{p})}{\partial p_j}\bigg|_{p_j=p_{j0}} d\tau \\ &= p_{i0}p_{j0}\mathcal{L}\left\{ \frac{\partial f(\tau;\boldsymbol{p})}{\partial p_i}\bigg|_{p_i=p_{i0}} \frac{\partial f(\tau;\boldsymbol{p})}{\partial p_j}\bigg|_{p_j=p_{j0}} \right\}(s=0) \\ &= \frac{p_{i0}p_{j0}}{2\pi i}\int_{-i\infty}^{i\infty} \frac{\partial F(s;\boldsymbol{p})}{\partial p_i}\bigg|_{p_i=p_{i0}} \frac{\partial F(-s;\boldsymbol{p})}{\partial p_j}\bigg|_{p_j=p_{j0}} ds \\ &= \frac{p_{i0}p_{j0}}{2\pi}\int_{-\infty}^{\infty} \frac{\partial F(ix;\boldsymbol{p})}{\partial p_i}\bigg|_{p_i=p_{i0}} \frac{\partial F(-ix;\boldsymbol{p})}{\partial p_j}\bigg|_{p_j=p_{j0}} dx\,. \end{aligned} \tag{2.58}$$

In the last two equations, $F(s;\boldsymbol{p})$ is the Laplace transformed function of $f(t;\boldsymbol{p})$. We made use of the multiplication formula of Laplace transform. Assuming all the singularities are left of the imaginary axis (which is true for the functions we are interested in), we take the integration counter along the imaginary axis.

For the model (2.1), we have a solution of the Laplace transformed macroscopic density ρ_M as

$$\rho_M(q,s;\boldsymbol{p}) = \frac{\arctan(qv/(s+Dq^2+\lambda))}{qv - \lambda\arctan(qv/(s+Dq^2+\lambda))}\,. \tag{2.59}$$

But it's difficult to calculate H_{ij} explicitly. So I used a short Mathematica program to numerically evaluate H_{ij}. I investigated the H_{ij} for the parameters $v = 13\mu\text{m/s}$, $\lambda = 1\text{s}^{-1}$, $D = 0.4\mu\text{m}^2/\text{s}$ which were similar with the real values in experiments. The results are listed in Table 2.1, 2.2, 2.3, 2.4.

Eigenvector			Eigenvalue
$\hat{v}$	$\hat{\lambda}$	$\hat{D}$	
-0.9964	0.0841	-0.0087	3.73×10^{-1}
-0.0823	-0.9882	-0.1293	2.43×10^{-3}
-0.0195	-0.1281	0.9916	2.26×10^{-5}

Table 2.1: The eigenvalues and corresponding eigenvectors of H_{ij} for $q =$ 0.5µm^{-1}. The least sensitive direction is almost along with $\hat{D}$. Here the hat denotes the unit vector.

Eigenvector			Eigenvalue
$\hat{v}$	$\hat{\lambda}$	$\hat{D}$	
-0.9992	0.0373	-0.0133	2.25×10^{-1}
-0.0271	-0.8887	-0.4577	6.64×10^{-4}
-0.0289	-0.4569	0.8890	4.96×10^{-5}

Table 2.2: The eigenvalues and corresponding eigenvectors of H_{ij} for $q =$ 1.0µm^{-1}. The least sensitive direction is almost in the plane of $\hat{\lambda}$ and $\hat{D}$.

Eigenvector			Eigenvalue
$\hat{v}$	$\hat{\lambda}$	$\hat{D}$	
-0.9995	0.0238	-0.0185	1.50×10^{-1}
-0.0007	0.5958	0.8031	4.11×10^{-4}
-0.0301	-0.8028	0.5955	3.91×10^{-5}

Table 2.3: The eigenvalues and corresponding eigenvectors of H_{ij} for $q =$ 1.5µm^{-1}. The least sensitive direction is almost in the plane of $\hat{\lambda}$ and $\hat{D}$.

Eigenvector			Eigenvalue
$\hat{v}$	$\hat{\lambda}$	$\hat{D}$	
-0.9995	0.0177	-0.0249	1.04×10^{-1}
0.0171	-0.3518	-0.9359	3.93×10^{-4}
-0.0253	-0.9359	0.3514	1.96×10^{-5}

Table 2.4: The eigenvalues and corresponding eigenvectors of H_{ij} for $q =$ 2.0µm^{-1}. The least sensitive direction is almost along $\hat{\lambda}$.

Comparing the results in Table 2.1, 2.2, 2.3, and 2.4, we found that the speed v is the most sensitive one among the three parameters in a large q range, which means we are able to measure it in the experiments with high accuracy.

Diffusion coefficient D is sensitive only for the high q. As indicated in Section 2.2, in the large spatial scales (small q), the Brownian diffusion will be mixed with the diffusive effect of the run-and-tumble motion, and distinguish them would be hard. Thus we need data in $q >\sim 1.0\mu\text{m}^{-1}$ to measure D in the experiments.

Tumbling rate λ is sensitive only for the small q. In the small spatial scales (large q), tumbling events would be rare and the particles would exhibit ballistic motion, as indicated in Section 2.5.2. So data in $q <\sim 1.0\mu\text{m}^{-1}$ are needed to measure λ efficiently in the experiments. We will make use of this result in Chapter 7.

Chapter 3

More detailed models

The simple model in Chapter 2 has captured the most important features of many realistic run-and-tumble particles. But we should mention we have neglected several ”ingredients” in this model, which may have an observable effect. For example, the bacteria needs some time during tumbling [24], and the direction change after tumbling may not be isotropic [25]. Now we are at the stage to add more ingredients to the simplest model, and see how these ingredients affect the motion of the run-and-tumble particles.

3.1 Finite tumbling duration

The model in Chapter 2 corresponds to the instantaneous tumbling, but it cannot be true in the experiments. To implement a finite tumbling duration in the simplest model, we separate the population into the running cells and the tumbling cells, described by $p_r(\boldsymbol{q}, \hat{\boldsymbol{v}}, t)$ and $p_t(\boldsymbol{q}, t)$ correspondingly. Then the total macroscopic density should be

$$\rho_M(\boldsymbol{q}, t) = \int p_r(\boldsymbol{q}, \hat{\boldsymbol{v}}, t)\, d\Omega + p_t(\boldsymbol{q}, t)\,. \tag{3.1}$$

As for the model (2.1), the swimming particles have the probability $\lambda\, dt$ to tumble in the time interval dt. But instead of a fixed time delay, we assume the tumbling cells has probability dt/δ to recover in time interval dt, where δ has the dimension of a time and can be regarded as a characteristic tumbling duration (or a mean tumbling duration). Then the master equations read

$$\frac{\partial p_r(\boldsymbol{x},\hat{\boldsymbol{v}},t)}{\partial t} = \nabla\cdot\left[D\nabla p_r(\boldsymbol{x},\hat{\boldsymbol{v}},t)\right] - \nabla\cdot\left[\boldsymbol{v}p_r(\boldsymbol{x},\hat{\boldsymbol{v}},t)\right] - \lambda p_r(\boldsymbol{x},\hat{\boldsymbol{v}},t) + \frac{1}{\delta\Omega}p_t(\boldsymbol{x},t)\,, \tag{3.2}$$

$$\frac{\partial p_t(\boldsymbol{x},t)}{\partial t} = \nabla\cdot\left[D\nabla p_t(\boldsymbol{x},t)\right] + \lambda\int p_r(\boldsymbol{x},\hat{\boldsymbol{v}}',t)\,d\Omega' - \frac{1}{\delta}p_t(\boldsymbol{x},t)\,. \tag{3.3}$$

3.1.1 Moment expansion and diffusive limit

Again, we first investigate the diffusive limit of (3.2) and (3.3) to get an effective equation for the density field ρ_M, with the procedure described in Section 2.2. The moment expansion of (3.2) can be written as

$$p_r = \frac{\rho_r}{\Omega} + J_i\hat{v}_i + Q_{ij}\left(\hat{v}_i\hat{v}_j - \frac{\delta_{ij}}{d}\right) + \Theta[p_r]\,. \tag{3.4}$$

And as in Section 2.2, we can get the equations for the first three moments. For the first three terms in the expansion (3.4), we have

$$\frac{\partial\rho_r}{\partial t} = -\frac{\Omega}{d}\nabla_i\left(vJ_i\right) + \nabla_i(D\nabla_i\rho_r) - \lambda\rho_r + \frac{1}{\delta}p_t\,, \tag{3.5}$$

$$\frac{\partial J_i}{\partial t} = -\frac{1}{\Omega}\nabla_i(v\rho_r) - \frac{2}{d+2}\nabla_j\left(vQ_{ij}\right) + \nabla_j\left(D\nabla_j J_i\right) - \lambda J_i\,, \tag{3.6}$$

$$\frac{\partial Q_{ij}}{\partial t} = -\frac{d+2}{2}B_{ijkl}\nabla_k\left(vJ_l\right) + \nabla_k\left(D\nabla_k Q_{ij}\right) - \lambda Q_{ij} - \nabla_k\left(v\chi_{ijk}\right)\,, \tag{3.7}$$

where

$$B_{ijkl} = \frac{1}{d+2}\left(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk} - \frac{2}{d}\delta_{ij}\delta_{kl}\right)\,, \tag{3.8}$$

$$\chi_{ijk} = \int M_{ij}\hat{v}_k\Theta[p_r]\,d\Omega\,. \tag{3.9}$$

Assuming the time derivatives of the $l \geq 1$ moments and the ∇^3 terms to be vanished, we have

$$J_i = -\frac{1}{\Omega\lambda}\nabla\left(v\rho_r\right)\,, \tag{3.10}$$

and

$$\frac{\partial\rho_M}{\partial t} = \nabla\cdot\left(D\nabla\rho_M\right) + \nabla\left[\frac{v}{d\lambda}\nabla\left(v\rho_r\right)\right]\,. \tag{3.11}$$

To have more information about ρ_r, we need one more assumption that the fixed ratio of the density is reached everywhere between the running particles and the tumbling particles. The fixed ratio means

$$\frac{\partial}{\partial t}\frac{\rho_r(\boldsymbol{x},t)}{p_t(\boldsymbol{x},t)} = 0\,. \tag{3.12}$$

Making use of (3.2) and (3.3), noticing $\rho_r(\boldsymbol{x},t) \propto p_t(\boldsymbol{x},t)$, and assuming the isotropic probability distribution, we have

$$\begin{aligned}(p_t(\boldsymbol{x},t)+\rho_r(\boldsymbol{x},t))\left(\frac{p_t(\boldsymbol{x},t)}{\delta}-\lambda\rho_r(\boldsymbol{x},t)\right) &= 0\,,\\ p_t(\boldsymbol{x},t) &= \delta\lambda\rho_r(\boldsymbol{x},t)\,,\\ \rho_r(\boldsymbol{x},t) = \frac{1}{1+\delta\lambda}(p_t+\rho_r) &= \frac{1}{1+\delta\lambda}\rho_M(\boldsymbol{x},t)\,. \end{aligned} \tag{3.13}$$

With this approximation, we have

$$\frac{\partial\rho_M}{\partial t} = \nabla\cdot(D\nabla\rho_M) + \nabla\left[\frac{v}{d\lambda}\nabla\left(\frac{v}{1+\lambda\delta}\rho_M\right)\right]\,. \tag{3.14}$$

If all the parameters are independent of the spatial coordinates, we find the effective diffusion coefficient of the run-and-tumble particles in the d-dimensional space

$$D_{\text{eff}} = D + \frac{v^2}{d\lambda(1+\lambda\delta)}\,. \tag{3.15}$$

Comparing with the continuity equation

$$\frac{\partial\rho_M}{\partial t} = -\nabla\cdot\boldsymbol{J}\,, \tag{3.16}$$

we have the flux of particles to be

$$\boldsymbol{J} = -D\nabla\rho_M - \frac{v}{d\lambda}\nabla\left(\frac{v}{1+\delta\lambda}\rho_M\right)\,. \tag{3.17}$$

And if δ or λ depend on some external chemical, we can also have a chemotaxis flux. So regulating speed is not the only way to have chemotaxis phenomenon. As for the bacteria, the chemotaxis system seems to be regulating the tumbling rate λ and the tumbling duration δ instead of the speed v, but we need more experimental data to confirm this.

The same result can be obtained by rescaling time. The finite tumbling duration delays the fraction of the running time from 1 to $1/(1+\lambda\delta)$. If we rescale both v and λ by a factor of $1/(1+\lambda\delta)$, we can have the same result. Hence the finite tumbling duration only results in a rescaling in isotropic tumbling in the large spatial scale.

3.1.2 Solution in three-dimensional free space

To get a similar solution as in Section 2.4, we assume the constant parameters and do Fourier-Laplace transform to get

$$\begin{aligned} sp_r(\boldsymbol{q},\hat{\boldsymbol{v}},s) - p_{r0} =& - Dq^2 p_r(\boldsymbol{q},\hat{\boldsymbol{v}},s) - i\boldsymbol{v}\cdot\boldsymbol{q}p_r(\boldsymbol{q},\hat{\boldsymbol{v}},s) \\ & - \lambda p_r(\boldsymbol{q},\hat{\boldsymbol{v}},s) + \frac{1}{\delta\Omega}p_t(\boldsymbol{q},s)\,, \qquad (3.18) \\ sp_t(\boldsymbol{q},s) - p_{t0} =& - Dq^2 p_t(\boldsymbol{q},s) + \lambda\int p_r(\boldsymbol{q},\hat{\boldsymbol{v}}',s)\,d\Omega' - \frac{1}{\delta}p_t(\boldsymbol{q},s)\,. \qquad (3.19) \end{aligned}$$

Here p_{r0} and p_{t0} are the initial conditions which should satisfy

$$p_{t0} + \int p_{r0}\,d\Omega' = p_{t0} + \Omega p_{r0} = 1\,. \qquad (3.20)$$

The first equation is from the assumption that the initial probability distribution is isotropic. In general we have no further information about the initial conditions p_{r0} and p_{t0}, the solution will depend on the initial ratio of the swimming particles to the tumbling particles. But in practice, many measurements are often taken a few minutes after the preparation of the sample. The sample should have adequate time to reach an invariant ratio of the swimming cells to the tumbling cells.[1] By setting

$$\frac{\partial}{\partial t}\frac{\int dV \int p_r(\boldsymbol{x},\hat{\boldsymbol{v}},t)\,d\Omega}{\int p_t(\boldsymbol{x},t)dV} = \frac{\partial}{\partial t}\frac{\int p_r(\boldsymbol{q}=0,\hat{\boldsymbol{v}},t)\,d\Omega}{p_t(\boldsymbol{q}=0,t)} = 0 \qquad (3.21)$$

and making use of (3.2) and (3.3) after Fourier transform, we have

$$\left(p_t(0,t) + \int p_r(0,\hat{\boldsymbol{v}},t)\,d\Omega\right)\left(\frac{p_t(0,t)}{\delta} - \lambda\int p_r(0,\hat{\boldsymbol{v}},t)\,d\Omega\right) = 0\,, \qquad (3.22)$$

$$p_t(0,t) = \delta\lambda\int p_r(0,\hat{\boldsymbol{v}},t)\,d\Omega\,. \qquad (3.23)$$

As $p_{r,t}(\boldsymbol{x},t=0) \propto \delta(x)$, $p_{r,t}(\boldsymbol{q},0) = p_{r,t}(\boldsymbol{q}=0,0)$, we can solve (3.20) to get

$$p_{r0} = \frac{1}{\Omega(\delta\lambda+1)}\,, \qquad\qquad p_{t0} = \frac{\delta\lambda}{\delta\lambda+1}\,. \qquad (3.24)$$

Now equation (3.19) gives

$$p_t(\boldsymbol{q},s) = \frac{\delta\lambda}{\delta(s+Dq^2)+1}\left(\int p_r(\boldsymbol{q},\hat{\boldsymbol{v}}',s)\,d\Omega' + \frac{\delta}{\delta\lambda+1}\right)\,. \qquad (3.25)$$

[1] This assumption is much weaker than (3.12). Here we only assume a fixed ratio of the total number in the initial condition, not a fixed ratio of the density everywhere and for any time.

Substituting it into (3.18), we have

$$\begin{aligned} p_r(\boldsymbol{q}, \hat{\boldsymbol{v}}, s) =& \frac{1}{\Omega(s + Dq^2 + i\boldsymbol{v}\cdot\boldsymbol{q} + \lambda)} \left(\frac{\delta(s + \lambda + Dq^2) + 1}{(\delta\lambda + 1)(\delta(s + Dq^2) + 1)} \right. \\ & \left. + \frac{\lambda}{\delta(s + Dq^2) + 1} \int p_r(\boldsymbol{q}, \hat{\boldsymbol{v}}', s)\, d\Omega' \right) . \end{aligned} \tag{3.26}$$

Denoting $\rho_r(\boldsymbol{q}, s) = \int p_r(\boldsymbol{q}, \hat{\boldsymbol{v}}, s)\, d\Omega\,$, with the same trick with which we get (2.32), we have in three dimensions

$$\begin{aligned} \rho_r(\boldsymbol{q}, s) =& \frac{\arctan(qv/(s + \lambda + Dq^2))}{qv} \left[\frac{\delta(s + \lambda + Dq^2) + 1}{(\delta\lambda + 1)(\delta(s + Dq^2) + 1)} \right. \\ & \left. + \frac{\lambda}{\delta(s + Dq^2) + 1} \rho_r(\boldsymbol{q}, s) \right] , \end{aligned} \tag{3.27}$$

$$\rho_r(\boldsymbol{q}, s) = \frac{\delta(s + \lambda + Dq^2) + 1}{\delta\lambda + 1} \frac{\arctan(qv/(s + \lambda + Dq^2))}{qv(\delta(s + Dq^2) + 1) - \lambda \arctan(qv/(s + \lambda + Dq^2)} . \tag{3.28}$$

Then we have

$$\begin{aligned} \rho_M(\boldsymbol{q}, s) =& \rho_r(\boldsymbol{q}, s) + p_t(\boldsymbol{q}, s) \\ =& \frac{\delta(s + \lambda + Dq^2) + 1}{\delta(s + Dq^2) + 1} \rho_r(\boldsymbol{q}, s) + \frac{\delta^2\lambda}{(\delta(s + Dq^2) + 1)(\delta\lambda + 1)} \\ =& \frac{qv\delta^2\lambda + [1 + \delta(Dq^2 + 2\lambda + s)] \arctan(qv/(Dq^2 + \lambda + s))}{(1 + \delta\lambda)[qv(1 + \delta Dq^2 + \delta s) - \lambda \arctan(qv/(Dq^2 + \lambda + s))]} . \end{aligned} \tag{3.29}$$

Again, $\rho_M(\boldsymbol{q}, s)$ is independent of the direction of $\boldsymbol{q}$. And by the limitation $\delta \to 0$, we can recover (2.32).

The mean square displacement of the particles satisfies

$$\langle \Delta x^2(s) \rangle = - \left. \frac{\partial^2 \rho_M(q, s)}{\partial q^2} \right|_{q=0} , \tag{3.30}$$

we have

$$\langle \Delta x^2(s) \rangle = \frac{2D}{s^2} + \frac{2v^2}{3s^2(1 + \lambda\delta)(s + \lambda)} \tag{3.31}$$

and the inverse Laplace transform gives

$$\langle \Delta x^2(t) \rangle = 2Dt + \frac{2}{1 + \lambda\delta} \left(\frac{v^2}{3\lambda} t + \frac{v^2}{3\lambda^2} e^{-\lambda t} \right) - \frac{2v^2}{3\lambda^2(1 + \lambda\delta)} . \tag{3.32}$$

Comparing with the result of the particles with instantaneous tumbling (2.36), the finite tumbling duration just introduces a factor of $1/(1 + \lambda\delta)$ into the run-and-tumble terms. The ballistic motion in the small time scales is also affected, and exhibits an effective speed of $v/\sqrt{1 + \lambda\delta}$.

For the large spatial scales, we can expand $1/\rho_M$ in terms of q to have

$$\rho_M(q,s) = \frac{1}{s + \left[D + v^2/3(1+\lambda\delta)(\lambda+s)\right] q^2 + o(q^4)} \,. \tag{3.33}$$

The inverse Laplace transform can be carried out to get

$$\rho_M(q,t) = \frac{(\Delta + \lambda - Dq^2) e^{-t(Dq^2+\lambda-\Delta)/2} + (\Delta - \lambda + Dq^2) e^{-t(Dq^2+\lambda+\Delta)/2}}{2\Delta} \,. \tag{3.34}$$

with

$$\Delta = \sqrt{(\lambda - Dq^2)^2 - \frac{4q^2v^2}{3(1+\lambda\delta)}} \,. \tag{3.35}$$

Similar with the model with $\delta \to 0$, it's also a composition with two exponential decays, and the slow mode shows an effective Brownian diffusion.

By expanding $1/\rho_M$ in terms of q and s, we can also get

$$\frac{1}{\rho_M} = s + \left(D + \frac{v^2}{\lambda d(1+\delta\lambda)}\right) q^2 + o(sq^2) \,, \tag{3.36}$$

and find an effective diffusion coefficient same with (3.15).

3.1.3 Sensitivity analysis

In Chapter 6, we will fit (3.29) with the experimental data to measure the parameters. Since we have introduced a new parameter, the new parameter may change the q range we need to find the old parameters. We also need to know the proper q range for measuring tumbling duration. Hence I investigated the sensitivity of (3.29) to all the parameters. Same with Section 2.5.4, I investigated the H_{ij} for the parameters $v = 13$µm/s, $\lambda = 1\text{s}^{-1}$, $D = 0.4$µm^2/s, $\delta = 0.14$s which is similar with the real values in experiments. The results are listed in Table 3.1, 3.2, 3.3, and 3.4.

Comparing the results in Table 3.1, 3.2, 3.3, and 3.4, we found that the conclusion in Section 2.5.4 still holds: v is still the most sensitive parameter and should be easily measured. We can find D in the high q range and λ in the low q range in the experimental data. In addition, we found the sensitivity of the tumbling duration δ is complicated. There is a relative sensitive eigenvector having an angle with the direction of $\hat{\delta}$. Together with the fitting results of the simulation data in Section 7.7, we found the fitting of δ depends little on q.

Eigenvector				Eigenvalue
$\hat{v}$	$\hat{\lambda}$	$\hat{D}$	$\hat{\delta}$	
-0.9867	0.1496	-0.0116	0.0628	3.16×10^{-1}
-0.1611	-0.8853	-0.0568	-0.4325	4.96×10^{-3}
-0.0015	-0.3945	-0.3420	0.8529	1.60×10^{-4}
0.0226	0.1956	-0.9379	-0.2856	1.47×10^{-5}

Table 3.1: The eigenvalues and corresponding eigenvectors of H_{ij} for $q =$ 0.5µm^{-1}. The least sensitive direction is almost along with $\hat{D}$. Here the hat denotes the unit vector.

Eigenvector				Eigenvalue
$\hat{v}$	$\hat{\lambda}$	$\hat{D}$	$\hat{\delta}$	
-0.9951	0.0867	-0.0184	0.0440	1.73×10^{-1}
-0.0937	-0.7812	-0.0774	-0.6123	2.67×10^{-3}
0.0011	-0.4468	-0.6168	0.6480	2.32×10^{-4}
0.0317	0.4272	-0.7831	-0.4509	4.03×10^{-5}

Table 3.2: The eigenvalues and corresponding eigenvectors of H_{ij} for $q =$ 1.0µm^{-1}. The least sensitive direction is the subspace of $\hat{D}$, $\hat{\lambda}$, and $\hat{\delta}$.

Eigenvector				Eigenvalue
$\hat{v}$	$\hat{\lambda}$	$\hat{D}$	$\hat{\delta}$	
-0.9969	0.0653	-0.0252	0.0369	1.15×10^{-1}
-0.0707	-0.7211	-0.0757	-0.6850	1.73×10^{-3}
0.01635	-0.3181	-0.8464	0.4268	2.62×10^{-4}
-0.0316	-0.6120	0.5265	0.5893	4.33×10^{-5}

Table 3.3: The eigenvalues and corresponding eigenvectors of H_{ij} for $q =$ 1.5µm^{-1}. The least sensitive direction is the subspace of $\hat{D}$, $\hat{\lambda}$, and $\hat{\delta}$.

Eigenvector				Eigenvalue
$\hat{v}$	$\hat{\lambda}$	$\hat{D}$	$\hat{\delta}$	
-0.9972	0.0567	-0.0333	0.0355	8.08×10^{-2}
0.0627	0.6859	0.0593	0.7225	1.22×10^{-3}
0.0292	-0.2084	-0.9389	0.2723	2.98×10^{-4}
-0.0282	-0.6949	0.3374	0.6344	3.22×10^{-5}

Table 3.4: The s and corresponding eigenvectors of H_{ij} for $q = 2.0\mu\text{m}^{-1}$. The least sensitive direction is the subspace of $\hat{D}$, $\hat{\lambda}$, and $\hat{\delta}$.

3.2 Tumble by active rotational diffusion

Now we take into account the anisotropic tumbles. Instead of inserting a phenomenological kernel $T(\hat{\boldsymbol{v}}, \hat{\boldsymbol{v}}')$ in the integration term $\frac{\lambda}{\Omega}\int p(\boldsymbol{x}, \hat{\boldsymbol{v}}', t)\, d\Omega'$ in the simple model (2.1), we can consider a more detailed model that assumes the tumbles to be active rotational diffusion, which is a result of the unbalanced global torque from the repelling force of the unbundled flagella, as shown in Figure 1.1. Due to the rod shape of the particles, there may also be a passive rotational diffusion effect that causing the drift of the direction during cell running. The active and the passive rotational diffusion coefficients may be different.

With both effects taken into account, and the similar notations with Section 3.1, we have the master equation for the running particles $p_r(\boldsymbol{x}, \hat{\boldsymbol{v}}, t)$ and the tumbling particles $p_t(\boldsymbol{x}, \hat{\boldsymbol{v}}, t)$

$$\frac{\partial p_r}{\partial t} = \nabla(D\nabla p_r) + D_r \Delta_{\hat{\boldsymbol{v}}} p_r - \nabla\cdot(\boldsymbol{v} p_r) - \lambda p_r + \frac{1}{\delta} p_t\,, \tag{3.37}$$

$$\frac{\partial p_t}{\partial t} = \nabla(D\nabla p_t) + D_{ra} \Delta_{\hat{\boldsymbol{v}}} p_t + \lambda p_r - \frac{1}{\delta} p_t\,. \tag{3.38}$$

Here D_r and D_{ra} are the two rotational diffusion coefficients (subscript a stands for *active*), and $\Delta_{\hat{\boldsymbol{v}}}$ is the Laplace-Beltrami operator on the $d-1$ dimensional unit sphere. In three dimensions, $\Delta_{\hat{\boldsymbol{v}}} = -\hat{L^2}$ where

$$\hat{L^2} = -\frac{1}{\sin\theta}\frac{\partial}{\partial\theta}\left(\sin\theta\frac{\partial}{\partial\theta}\right) - \frac{1}{\sin^2\theta}\frac{\partial^2}{\partial\phi^2} \tag{3.39}$$

is the well-known total angular momentum operator acting on $\hat{\boldsymbol{v}}$.

3.2.1 Moment expansion and diffusive limit

To get an effective equation for the density field ρ_M, with the procedure described in Section 2.2, we expand the angular dependence of p_r and p_t with respect to $\hat{\boldsymbol{v}}$ as

$$p_{r,t} = \frac{\rho_{r,t}}{\Omega} + J_i^{(r,t)}\hat{v}_i + Q_{ij}^{(r,t)}\left(\hat{v}_i\hat{v}_j - \frac{\delta_{ij}}{d}\right) + \Theta[p_{r,t}]\,. \tag{3.40}$$

For the first three terms in the expansion (3.40), we have

$$\begin{aligned}
\frac{\partial\rho_r}{\partial t} =& -\frac{\Omega}{d}\nabla_i\left(vJ_i^{(r)}\right) + \nabla_i(D\nabla_i\rho_r) - \lambda\rho_r + \frac{1}{\delta}\rho_t\,, && (3.41)\\
\frac{\partial\rho_t}{\partial t} =& \nabla_i(D\nabla_i\rho_t) + \lambda\rho_r - \frac{1}{\delta}\rho_t\,, && (3.42)\\
\frac{\partial J_i^{(r)}}{\partial t} =& -\frac{1}{\Omega}\nabla_i(v\rho_r) - \frac{2}{d+2}\nabla_j\left(vQ_{ij}^{(r)}\right) + \nabla_j\left(D\nabla_j J_i^{(r)}\right)\\
& -(d-1)D_r J_i^{(r)} - \lambda J_i^{(r)} + \frac{1}{\delta}J_i^{(t)}\,, && (3.43)\\
\frac{\partial J_i^{(t)}}{\partial t} =& \nabla_j\left(D\nabla_j J_i^{(t)}\right) - (d-1)D_{ra}J_i^{(t)} + \lambda J_i^{(r)} - \frac{1}{\delta}J_i^{(t)}\,, && (3.44)\\
\frac{\partial Q_{ij}^{(r)}}{\partial t} =& -\frac{d+2}{2}B_{ijkl}\nabla_k\left(vJ_l^{(r)}\right) + \nabla_k\left(D\nabla_k Q_{ij}^{(r)}\right) - 2dD_r Q_{ij}^{(r)}\\
& -\lambda Q_{ij}^{(r)} + \frac{1}{\delta}Q_{ij}^{(t)} - \nabla_k\left(v\chi_{ijk}^{(r)}\right)\,, && (3.45)\\
\frac{\partial Q_{ij}^{(t)}}{\partial t} =& \nabla_k\left(D\nabla_k Q_{ij}^{(t)}\right) - 2dD_{ra}Q_{ij}^{(t)} + \lambda Q_{ij}^{(r)} - \frac{1}{\delta}Q_{ij}^{(t)}\,, && (3.46)
\end{aligned}$$

where

$$\begin{aligned}
B_{ijkl} =& \frac{1}{d+2}\left(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk} - \frac{2}{d}\delta_{ij}\delta_{kl}\right)\,, && (3.47)\\
\chi_{ijk}^{(r)} =& \int M_{ij}\hat{v}_k\Theta[p_r]\,d\Omega\,. && (3.48)
\end{aligned}$$

Assuming the time derivatives of the $l \geq 1$ moments and the ∇^3 terms to be vanished, we have

$$\begin{aligned}
J_i^{(r)} =& -\frac{1 + (d-1)D_{ra}\delta}{\Omega(d-1)[D_r + D_{ra}\lambda\delta + (d-1)D_r D_{ra}\delta]}\nabla\left(v\rho_r\right)\,, && (3.49)\\
J_i^{(t)} =& -\frac{\lambda\delta}{\Omega(d-1)[D_r + D_{ra}\lambda\delta + (d-1)D_r D_{ra}\delta]}\nabla\left(v\rho_r\right)\,. && (3.50)
\end{aligned}$$

Noticing $\rho_M = \rho_r + \rho_t$, we have the diffusion equation

$$\frac{\partial \rho_M}{\partial t} = \nabla \cdot (D\nabla\rho) + \frac{1}{d(d-1)}\nabla\left[\frac{v(1+(d-1)D_{ra}\delta)}{D_r + D_{ra}\lambda\delta + (d-1)D_r D_{ra}\delta}\nabla\left(v\rho_r\right)\right] . \tag{3.51}$$

With the same approximation with Section 3.1.1, we have

$$\rho_r = \frac{1}{1+\lambda\delta}\rho_M \tag{3.52}$$

and

$$\frac{\partial \rho_M}{\partial t} = \nabla\cdot(D\nabla\rho) + \frac{1}{d(d-1)}\nabla\left[\frac{v(1+(d-1)D_{ra}\delta)}{D_r + D_{ra}\lambda\delta + (d-1)D_r D_{ra}\delta}\nabla\left(\frac{v}{1+\lambda\delta}\rho_M\right)\right] . \tag{3.53}$$

The effective diffusion coefficient is

$$D_{\text{eff}} = D + \frac{v^2[1+(d-1)D_{ra}\delta]}{d(d-1)(1+\lambda\delta)[D_r + D_{ra}\lambda\delta + (d-1)D_r D_{ra}\delta]} . \tag{3.54}$$

If the rotational diffusion during running can be neglected, by taking $D_r = 0$ we have

$$D_{\text{eff}} = D + \frac{v^2[1+(d-1)D_{ra}\delta]}{d(d-1)(1+\lambda\delta)D_{ra}\lambda\delta} , \tag{3.55}$$

which looks like a rescaling of v and λ by two factors: $1/(1+\lambda\delta)$ and $[1+(d-1)D_{ra}\delta]/(d-1)D_{ra}\delta$ in (2.19). As a small D_{ra} makes particles persist on its original direction, it will enhance the diffusivity of the particles.

For the particles that have isotropic tumbling, we have $D_{ra} \to \infty$ and

$$D_{\text{eff}} = D + \frac{v^2}{d(1+\lambda\delta)[\lambda + (d-1)D_r]} , \tag{3.56}$$

which is already known for the active Brownian particles [35, 36]. We notice except for the factor $1/(1+\lambda\delta)$, the effect of the passive rotational diffusion resembles an effective tumbling rate $\lambda' = (d-1)D_r$.

If the particles don't have randomization in direction during tumbling, $D_{ra} = 0$ and

$$D_{\text{eff}} = D + \frac{v^2}{d(d-1)(1+\lambda\delta)D_r} , \tag{3.57}$$

which is the diffusivity of the active Brownian particles without tumbling, but with a rescaling in the time due to the duration δ for the particles to stop running.

If the particles don't have active rotational diffusion mechanism, but randomize their direction with the passive rotational diffusion. Then the particles just stop running with the rate λ. Taking $D_{ra} = D_r$ we have

$$D_{\text{eff}} = D + \frac{v^2[1 + (d-1)D_r\delta]}{d(d-1)D_r(1+\lambda\delta)[1 + \lambda\delta + (d-1)D_r\delta]} \,. \tag{3.58}$$

As we can see introducing anisotropic tumbling complicates the diffusion coefficient of the run-and-tumble particles. The effects of the active and the passive rotational diffusion can mix up in a non-trivial form.

3.2.2 Formal solution

The direct solutions of (3.37) and (3.38) are very difficult to get. We can obtain a formal solution using the following procedure in three dimensions ($d = 3$). After the Fourier-Laplace transform, we expand the $p_{r,t}(\boldsymbol{q}, \hat{\boldsymbol{v}}, s)$ in terms of the spherical harmonics

$$p_{r,t}(\boldsymbol{q}, \hat{\boldsymbol{v}}, s) = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} b_{lm}^{(r,t)}(\boldsymbol{q}, s) Y_{lm}(\theta, \phi) \,, \tag{3.59}$$

where

$$Y_{lm}(\theta, \phi) = (-1)^m \sqrt{\frac{(2l+1)(l-m)!}{4\pi(l+m)!}} P_l^m(\cos\theta) e^{im\phi} \tag{3.60}$$

are the spherical harmonics, and

$$P_l^m(x) = (-1)^{l+m} \frac{(l+m)!}{(l-m)!} \frac{1}{2^l l! (1-x^2)^{m/2}} \frac{d^{l-m}}{dx^{l-m}} [(1-x^2)^l] \tag{3.61}$$

are the associated Legendre polynomials. Noticing that

$$\begin{aligned} Y_{lm}(\theta, \phi) \cos\theta = & \sqrt{\frac{(l-m+1)(l+m+1)}{(2l+1)(2l+3)}} Y_{l+1,m}(\theta, \phi) \\ & + \sqrt{\frac{(l-m)(l+m)}{(2l+1)(2l-1)}} Y_{l-1,m}(\theta, \phi) \,, \end{aligned} \tag{3.62}$$

we can define the raising and the lowering operators $\hat{a}_+$ and $\hat{a}_-$ as

$$\hat{a}_+ b_{lm}(\boldsymbol{q}, s) = \sqrt{\frac{(l-m+1)(l+m+1)}{(2l+1)(2l+3)}} b_{l+1,m}(\boldsymbol{q}, s) \,, \tag{3.63}$$

$$\hat{a}_- b_{lm}(\boldsymbol{q}, s) = \sqrt{\frac{(l-m)(l+m)}{(2l+1)(2l-1)}} b_{l-1,m}(\boldsymbol{q}, s) \,. \tag{3.64}$$

As the spherical harmonics are the eigenvectors of the total angular momentum operator,

$$\hat{L}^2 Y_{lm}(\theta,\phi) = l(l+1) Y_{lm}(\theta,\phi)\,, \tag{3.65}$$

(3.37) and (3.38) become

$$sb^{(r)}_{lm} - b^{(r)}_{lm,0} = -Dq^2 b^{(r)}_{lm} - D_r l(l+1) b^{(r)}_{lm} - iqv(\hat{a}_+ + \hat{a}_-) b^{(r)}_{lm} - \lambda b^{(r)}_{lm} + \delta^{-1} b^{(t)}_{lm}\,, \tag{3.66}$$

$$sb^{(t)}_{lm} - b^{(t)}_{lm,0} = -Dq^2 b^{(t)}_{lm} - D_{ra} l(l+1) b^{(t)}_{lm} + \lambda b^{(r)}_{lm} - \delta^{-1} b^{(t)}_{lm}\,. \tag{3.67}$$

Here $b^{(r,l)}_{lm,0}$ is the initial condition of $b^{(r,l)}_{lm}(\boldsymbol{q},t)$ before Laplace transform. If we take the initial condition to be the Dirac delta function in space, then $b^{(r,l)}_{lm,0} = 0$ for $l \neq 0$, but is constant for $l = 0$. Defining operators $\hat{\psi}$ and $\hat{\psi}'$ as

$$\hat{\psi} = \frac{1}{s + Dq^2 + D_r l(l+1) + iqv(\hat{a}_+ + \hat{a}_-) + \lambda}\,, \tag{3.68}$$

$$\hat{\psi}' = \frac{1}{s + Dq^2 + D_{ra} l(l+1) + \delta^{-1}}\,, \tag{3.69}$$

the linear equations can be formally solved as

$$b^{(r)}_{lm} = \left(1 - \frac{\lambda}{\delta}\hat{\psi}\hat{\psi}'\right)^{-1} (\hat{\psi} b^{(r)}_{lm,0} + \delta^{-1}\hat{\psi}\hat{\psi}' b^{(t)}_{lm,0})\,, \tag{3.70}$$

$$b^{(t)}_{lm} = \left(1 - \frac{\lambda}{\delta}\hat{\psi}'\hat{\psi}\right)^{-1} (\hat{\psi}' b^{(t)}_{lm,0} + \lambda\hat{\psi}'\hat{\psi} b^{(r)}_{lm,0})\,. \tag{3.71}$$

Using the identity

$$\hat{A}(1 - c\hat{B}\hat{A})^{-1} = (1 - c\hat{A}\hat{B})^{-1}\hat{A}\,, \tag{3.72}$$

we have

$$\begin{aligned} b_{lm} &= b^{(r)}_{lm} + b^{(t)}_{lm} \\ &= (1 + \lambda\hat{\psi}')\left(1 - \frac{\lambda}{\delta}\hat{\psi}\hat{\psi}'\right)^{-1} \hat{\psi} b^{(r)}_{lm,0} + \left(1 + \frac{\hat{\psi}}{\delta}\right)\left(1 - \frac{\lambda}{\delta}\hat{\psi}'\hat{\psi}\right)^{-1} \hat{\psi}' b^{(t)}_{lm,0}\,. \end{aligned} \tag{3.73}$$

However, the evaluation of this expression need to solve infinity dimensional linear equations. We may seek an approximation by a suitable truncation, which is left for future work.

The same formal solution can also be constructed in another way, following the method developed by F. Detcheverry [43]. Using the joint probability distributions of the state jump $J_{r\to t}(\boldsymbol{x}, \hat{\boldsymbol{v}}, t)$ and $J_{t\to r}(\boldsymbol{x}, \hat{\boldsymbol{v}}, t)$, we can construct the solution in real time and space

$$p_r = \int_0^t dt'\,\psi(t-t')\int \sin\theta'\,d\theta' d\phi' d^3x\, g_{rs}(\boldsymbol{x}-\boldsymbol{x}',\hat{\boldsymbol{v}}-\hat{\boldsymbol{v}}',t-t')J_{t\to r}(\boldsymbol{x}',\hat{\boldsymbol{v}}',t') \\ +\psi_\alpha(t)\int \sin\theta'\,d\theta' d\phi' d^3x\, g_{rs}(\boldsymbol{x}-\boldsymbol{x}',\hat{\boldsymbol{v}}-\hat{\boldsymbol{v}}',t)p_{r0}(\boldsymbol{x}',\hat{\boldsymbol{v}}')\,, \tag{3.74}$$

$$p_t = \int_0^t dt'\,\psi'(t-t')\int \sin\theta'\,d\theta' d\phi' d^3x\, g_{rd}(\boldsymbol{x}-\boldsymbol{x}',\hat{\boldsymbol{v}}-\hat{\boldsymbol{v}}',t-t')J_{r\to t}(\boldsymbol{x}',\hat{\boldsymbol{v}}',t') \\ +\psi'_\alpha(t)\int \sin\theta'\,d\theta' d\phi' d^3x\, g_{rd}(\boldsymbol{x}-\boldsymbol{x}',\hat{\boldsymbol{v}}-\hat{\boldsymbol{v}}',t)p_{t0}(\boldsymbol{x}',\hat{\boldsymbol{v}}')\,, \tag{3.75}$$

$$J_{r\to t} = \int_0^t dt'\,\psi_\beta(t-t')\int \sin\theta'\,d\theta' d\phi' d^3x\, g_{rs}(\boldsymbol{x}-\boldsymbol{x}',\hat{\boldsymbol{v}}-\hat{\boldsymbol{v}}',t-t')J_{t\to r}(\boldsymbol{x}',\hat{\boldsymbol{v}}',t') \\ +\psi_\gamma(t)\int \sin\theta'\,d\theta' d\phi' d^3x\, g_{rs}(\boldsymbol{x}-\boldsymbol{x}',\hat{\boldsymbol{v}}-\hat{\boldsymbol{v}}',t)p_{r0}(\boldsymbol{x}',\hat{\boldsymbol{v}}')\,, \tag{3.76}$$

$$J_{t\to r} = \int_0^t dt'\,\psi'_\beta(t-t')\int \sin\theta'\,d\theta' d\phi' d^3x\, g_{rd}(\boldsymbol{x}-\boldsymbol{x}',\hat{\boldsymbol{v}}-\hat{\boldsymbol{v}}',t-t')J_{r\to t}(\boldsymbol{x}',\hat{\boldsymbol{v}}',t') \\ +\psi'_\gamma(t)\int \sin\theta'\,d\theta' d\phi' d^3x\, g_{rd}(\boldsymbol{x}-\boldsymbol{x}',\hat{\boldsymbol{v}}-\hat{\boldsymbol{v}}',t)p_{t0}(\boldsymbol{x}',\hat{\boldsymbol{v}}')\,, \tag{3.77}$$

where g_{rs} and g_{rt} are the propagators corresponding to the differential equations

$$\frac{\partial g_{rs}}{\partial t} = D\nabla^2 g_{rs} - D_r\hat{L^2} g_{rs} - \boldsymbol{v}\cdot\nabla g_{rs}\,, \tag{3.78}$$

$$\frac{\partial g_{rd}}{\partial t} = D\nabla^2 g_{rd} - D_{ra}\hat{L^2} g_{rd}\,, \tag{3.79}$$

with the initial condition $g_{rs}(\boldsymbol{x},\hat{\boldsymbol{v}}-\hat{\boldsymbol{v}}',0) = g_{rd}(\boldsymbol{x},\hat{\boldsymbol{v}}-\hat{\boldsymbol{v}}',0) = \delta(\boldsymbol{x})\delta(\hat{\boldsymbol{v}}-\hat{\boldsymbol{v}}')$. And

$$\psi(t) = e^{-\lambda t}\,, \qquad \psi'(t) = e^{-t/\delta} \tag{3.80}$$

are the probabilities for the cell in each state (running and tumbling correspondingly) that nothing happens in the time t after the last state jump. We also define

$$\psi_\alpha(t) = e^{-\lambda t}\,, \qquad \psi'_\alpha(t) = e^{-t/\delta}\,, \tag{3.81}$$

the probabilities for the cell in each state (running and tumbling correspondingly) that nothing happens in the time t after the initial time.

$$\psi_\beta(t) = \lambda e^{-\lambda t}\,, \qquad \psi'_\beta(t) = \frac{1}{\delta}e^{-t/\delta} \tag{3.82}$$

are the probabilities for a state jump ($r \to t$ and $t \to r$ correspondingly) happens at the time t after the last state jump.

$$\psi_\gamma(t) = \lambda e^{-\lambda t}\,, \qquad \psi'_\gamma(t) = \frac{1}{\delta}e^{-t/\delta} \tag{3.83}$$

are the probabilities for a state jump ($r \to t$ and $t \to r$ correspondingly) happens at the time t after the initial time.

To make the expression looks easier, we first notice that we can formally write the solution of the equation

$$\frac{\partial f}{\partial t} = \hat{\mathcal{L}} f \tag{3.84}$$

with the initial condition $f(\boldsymbol{x}, \hat{\boldsymbol{v}}, t = t_0)$ as

$$e^{(t-t')\hat{\mathcal{L}}} f = \int \sin\theta' \, d\theta' d\phi' d^3x' \, g(\boldsymbol{x} - \boldsymbol{x}', \hat{\boldsymbol{v}} - \hat{\boldsymbol{v}}', t - t_0) f(\boldsymbol{x}', \hat{\boldsymbol{v}}', t_0) \, . \tag{3.85}$$

Inserting into the constructed solution and Laplace transforming, noticing $e^{(t-t')\hat{\mathcal{L}}}$ just causes a shift in the Laplace transformed time s, we have

$$\begin{aligned} p_r &= \psi(s - \hat{\mathcal{L}}_{rs}) J_{t\to r} + \psi_\alpha(s - \hat{\mathcal{L}}_{rs}) p_{r0} \, , & (3.86) \\ p_t &= \psi'(s - \hat{\mathcal{L}}_{rd}) J_{r\to t} + \psi'_\alpha(s - \hat{\mathcal{L}}_{rd}) p_{t0} \, , & (3.87) \\ J_{r\to t} &= \psi_\beta(s - \hat{\mathcal{L}}_{rs}) J_{t\to r} + \psi_\gamma(s - \hat{\mathcal{L}}_{rs}) p_{r0} \, , & (3.88) \\ J_{t\to r} &= \psi'_\beta(s - \hat{\mathcal{L}}_{rd}) J_{r\to t} + \psi'_\gamma(s - \hat{\mathcal{L}}_{rd}) p_{t0} \, . & (3.89) \end{aligned}$$

Here the operators $\hat{\mathcal{L}}$ are the corresponding differential operators in the right hand side of the equations (3.79) and (3.78). Noticing $\psi_\beta = \psi_\gamma = \lambda\psi_\alpha = \lambda\psi$, $\psi'_\beta = \psi'_\gamma = \delta^{-1}\psi'_\alpha = \delta^{-1}\psi'$, we denote the operator $\hat{\psi} = \psi(s - \hat{\mathcal{L}}_{rs})$, $\hat{\psi}' = \psi'(s - \hat{\mathcal{L}}_{rd})$, and solve the equations formally as

$$\begin{aligned} J_{r\to t} &= \left(1 - \frac{\lambda}{\delta}\hat{\psi}\hat{\psi}'\right)^{-1} \left(\frac{\lambda}{\delta}\hat{\psi}\hat{\psi}' p_{t0} + \lambda\hat{\psi} p_{r0}\right) \, , & (3.90) \\ J_{t\to r} &= \left(1 - \frac{\lambda}{\delta}\hat{\psi}'\hat{\psi}\right)^{-1} \left(\frac{\lambda}{\delta}\hat{\psi}'\hat{\psi} p_{r0} + \frac{1}{\delta}\hat{\psi} p_{t0}\right) \, , & (3.91) \\ p_r &= \left(1 - \frac{\lambda}{\delta}\hat{\psi}\hat{\psi}'\right)^{-1} (\hat{\psi} p_{r0} + \delta^{-1}\hat{\psi}\hat{\psi}' p_{t0}) \, , & (3.92) \\ p_t &= \left(1 - \frac{\lambda}{\delta}\hat{\psi}'\hat{\psi}\right)^{-1} (\hat{\psi}' p_{t0} + \lambda\hat{\psi}'\hat{\psi} p_{r0}) \, . & (3.93) \end{aligned}$$

The later two equations, when expressed in the Fourier space and expand in the spherical harmonics series, is the same with (3.70) and (3.71).

Chapter 4

Run-and-tumble motion with obstacles

Now we consider a simple but common case of the interaction between particles and environment: the particles may be blocked by the obstacles when swimming. In practice, bacteria moving in semi-solid agar gel can be approximated as this model. The proteins in the semi-solid agar form poriferous network of fibres [44], and the cells moving in it may be trapped by the fibres.

This bloking interaction has effects in the experimental observation. Researchers found that the cell migration rate is increasing with higher tumbling rate [45], and the experiments in our lab showed a bell-shape curve of the diffusion coefficient to the tumbling rate [22], which contradicts with the well-known effective diffusion coefficients (3.15) for the run-and-tumble motion in liquid. It indicates the significant difference between the motion in gel and the motion in liquid, which deserves careful investigation.

4.1 Run-and-tumble particles on lattice

We can approach this problem using a lattice model. The calculation of diffusion coefficient reduces to calculate the mean square displacement $\langle \Delta x^2 \rangle \sim 2Dt$. Considering non-interacting run-and-tumble particles moving on lattice, they can hop into the neighbour sites with the hopping rate h, and change their heading with the tumbling rate λ. The definition of tumbling rate is the same as before, but the hopping rate is related with the swimming speed v by $v = h\delta l$, where δl is the lattice spacing. Firstly, we need to know in what condition the on-lattice model resembles off-lattice model. Hence we need to investigate the model without obstacles.

Denoting the position of a lattice site to be the vector $\boldsymbol{x}$ in the d-dimensional lattice, we consider the probability distribution $P(\boldsymbol{x}, \boldsymbol{e}_i, t)$ of finding a particle at the site $\boldsymbol{x}$ heading to the direction $\boldsymbol{e}_i$ at the time t. Assuming the particle is at the origin with a random heading direction at time $t = 0$, the initial condition is $P(\boldsymbol{x}, \boldsymbol{e}_i, 0) = \delta_{\boldsymbol{x},0}/2d$. The master equation can be written as

$$\begin{aligned}\frac{\partial P(\boldsymbol{x}, \boldsymbol{e}_i, t)}{\partial t} =& hP(\boldsymbol{x} - \boldsymbol{e}_i, \boldsymbol{e}_i, t) - hP(\boldsymbol{x}, \boldsymbol{e}_i, t) \\ & - \lambda P(\boldsymbol{x}, \boldsymbol{e}_i, t) + \frac{\lambda}{2d} \sum_i P(\boldsymbol{x}, \boldsymbol{e}_i, t)\,. \end{aligned} \tag{4.1}$$

Direction $\boldsymbol{e}_i$, where $i = \pm 1, \pm 2, \cdots, \pm d$, is a vector that the only non-zero component is the $|i|$-th component, which is ± 1 with the sign same as the sign of i.

To calculate $\langle \Delta x^2 \rangle$, without loss of generality, we introduced the one-variable generating function

$$F_i(z, t) = \sum_{\boldsymbol{x}=-\infty}^{\infty} z^{x_1} P(\boldsymbol{x}, \boldsymbol{e}_i, t)\,, \tag{4.2}$$

$$F_0(z, t) = \sum_{i \neq \pm 1} F_i(z, t)\,, \tag{4.3}$$

$$F(z, t) = \sum_i F_i(z, t)\,. \tag{4.4}$$

Then

$$\langle \Delta x_1^2 \rangle = \langle x_1^2 \rangle = \left. \frac{\partial^2 F(z, t)}{\partial z^2} \right|_{z=1}\,, \tag{4.5}$$

as from the symmetry of the system, $\langle x_1 \rangle = 0$ obviously.

Now we have

$$\frac{\partial F_1(z, t)}{\partial t} = h(z - 1) F_1(z, t) - \lambda F_1(z, t) + \frac{\lambda}{2d} F(z, t)\,, \tag{4.6}$$

$$\frac{\partial F_{-1}(z, t)}{\partial t} = h\left(\frac{1}{z} - 1\right) F_{-1}(z, t) - \lambda F_{-1}(z, t) + \frac{\lambda}{2d} F(z, t)\,, \tag{4.7}$$

$$\frac{\partial F_0(z, t)}{\partial t} = - \lambda F_0(z, t) + \frac{\lambda(2d - 2)}{2d} F(z, t)\,. \tag{4.8}$$

Differentiating (4.6)-(4.8) with z, taking $z = 1$, and noticing $F_{\pm 1}(1, t) = 1/2d$,

$\partial F/\partial z|_{z=1} = 0$, we have

$$\frac{\partial}{\partial t}\left.\frac{\partial F_1(z,t)}{\partial z}\right|_{z=1} = \frac{h}{2d} - \lambda \left.\frac{\partial F_1(z,t)}{\partial z}\right|_{z=1}, \tag{4.9}$$

$$\frac{\partial}{\partial t}\left.\frac{\partial F_{-1}(z,t)}{\partial z}\right|_{z=1} = -\frac{h}{2d} - \lambda \left.\frac{\partial F_{-1}(z,t)}{\partial z}\right|_{z=1}, \tag{4.10}$$

$$\frac{\partial}{\partial t}\left.\frac{\partial F_0(z,t)}{\partial z}\right|_{z=1} = -\lambda \left.\frac{\partial F_0(z,t)}{\partial z}\right|_{z=1}. \tag{4.11}$$

With the initial conditions $\partial F_i/\partial z|_{z=1,t=0} = 0$, we have

$$\left.\frac{\partial F_1(z,t)}{\partial z}\right|_{z=1} = -\left.\frac{\partial F_{-1}(z,t)}{\partial z}\right|_{z=1} = \frac{h}{2d\lambda}(1 - e^{-\lambda t}), \tag{4.12}$$

$$\left.\frac{\partial F_0(z,t)}{\partial z}\right|_{z=1} = 0. \tag{4.13}$$

Differentiating (4.6)-(4.8) twice with z, we have

$$\begin{aligned}\frac{\partial}{\partial t}\left.\frac{\partial^2 F(z,t)}{\partial z^2}\right|_{z=1} &= 2h\left.\frac{\partial F_1(z,t)}{\partial z}\right|_{z=1} - 2h\left.\frac{\partial F_{-1}(z,t)}{\partial z}\right|_{z=1} + \frac{h}{d}\\ &= \frac{2h^2}{d\lambda}(1 - e^{-\lambda t}) + \frac{h}{d}.\end{aligned} \tag{4.14}$$

With the initial condition $\partial^2 F(z,0)/\partial z^2 = 0$, we can get the mean square displacement

$$\langle \Delta x_1^2 \rangle = \left.\frac{\partial^2 F(z,t)}{\partial z^2}\right|_{z=1} = \left(\frac{2h^2}{d\lambda} + \frac{h}{d}\right) t + \frac{2h^2}{d\lambda^2}(e^{-\lambda t} - 1). \tag{4.15}$$

In the short time scales, $\langle \Delta x_1^2 \rangle \propto t^2$ and the particle shows ballistic motion. In the large time scales where $t \gg 1/\lambda$, the later term is a constant and of little importance. Comparing with (2.36) in the continuous space, we found an artificial term ht/d due to the discretization of space. If we recover the unit of length, this term should be $vt\,\delta l/d$ and tends to 0 for $\delta l \to 0$.

Assuming the artificial term to be small, we get a condition for the on-lattice model to be same with off-lattice model

$$h \gg \frac{\lambda}{2}. \tag{4.16}$$

By the hopping rate to be much larger than the tumbling rate, we are taking the lattice spacing δl to be small enough compared to the mean run length v/λ of the particle. In such condition the discretization of space doesn't have much effects on the physics of the run-and-tumble motion, and we can safely generalize our conclusion into the continuous space.

4.2 Lattice model with obstacles

In this section I will formulate the model with the obstacles. Let's denote the set of the blocking sites to be $\mathcal{B}$, and define the field $\eta(\boldsymbol{x})$ such that

$$\eta(\boldsymbol{x}) = \begin{cases} 0, & \boldsymbol{x} \in \mathcal{B} \\ 1, & \boldsymbol{x} \notin \mathcal{B} \end{cases} . \tag{4.17}$$

When a particle runs into an obstacle, it should stop moving until it tumbles into another direction without any obstacle in front of it. Then the master equation needs to be modified as

$$\begin{aligned} \frac{\partial P(\boldsymbol{x}, \boldsymbol{e}_i, t, [\eta])}{\partial t} =& h\eta(\boldsymbol{x}) P(\boldsymbol{x} - \boldsymbol{e}_i, \boldsymbol{e}_i, t, [\eta]) - h\eta(\boldsymbol{x} + \boldsymbol{e}_i) P(\boldsymbol{x}, \boldsymbol{e}_i, t, [\eta]) \\ & - \lambda P(\boldsymbol{x}, \boldsymbol{e}_i, t, [\eta]) + \frac{\lambda}{2d} \sum_i P(\boldsymbol{x}, \boldsymbol{e}_i, t, [\eta]) , \end{aligned} \tag{4.18}$$

with $P(\boldsymbol{x}, \boldsymbol{e}_i, t, [\eta]) = 0$ for $\boldsymbol{x} \in \mathcal{B}$. We should notice now the probability distribution is related with the specific configuration of the obstacles. When we need to sum over all the configurations of the obstacles, we make an additional restriction that the probability of finding an obstacle at each site is constant $\mathcal{P}\{\eta(\boldsymbol{x} = 0)\} = c$.

4.3 Mean field approximation

We cannot deal with (4.18) without any approximation. Here I will introduce a continuous model. This model cannot be obtained by directly taking the lattice spacing to tend to zero, as it will also take the scattering section of the obstacles to be zero. So we construct this model from the underlining physics in the continuous space. As a result, this model serves as a mean field approximation of (4.18), which means the obstacles are considered to be homogeneous with the concentration c. It should be valid in the large scales and for the low density of the obstacles. To be comparable with the on-lattice model, I will use the hopping rate h instead of the speed v.

The continuous version of the master equation for the running particles p_r is similar with the simple model in Chapter 2. Tumbles change the heading of the particles, but the particle after tumbling have the probability c to meet an obstacle, causing a flow from the running particles p_r to the blocked particles p_b. The running particles now have a certain probability to meet an obstacle and stop moving. As the average number of the obstacles that a particle meets in unit time is hc, we assume the cells have the probability $\gamma_d(c)hc$ to be blocked. Here I introduce a factor $\gamma_d(c)$, which seems to be a function of the obstacle density and the dimension only according to the simulations. $\gamma_d(c) > 1$ and should arise from preserving the finite scattering section of the obstacles when passing from the on-lattice model to the off-lattice model, and causing the particles to feel an effective density $\gamma_d(c)c$ of obstacles in the continuous limitation. It has the unit of inverse of length if we are not using the dimensionless length. For now I don't have a theory to calculate $\gamma_d(c)$, but in the next section I will show that we can obtain the $\gamma_d(c)$ from the simulation data. Thus the master equations for the running particles p_r and the blocked particles p_b are

$$\frac{\partial p_r(\boldsymbol{x},\hat{\boldsymbol{v}},t)}{\partial t} = -\nabla\cdot(\hat{\boldsymbol{v}}hp_r) - \lambda p_r - \gamma_d(c)hcp_r + \frac{\lambda(1-c)}{\Omega}\rho_M\,, \tag{4.19}$$

$$\frac{\partial p_b(\boldsymbol{x},\hat{\boldsymbol{v}},t)}{\partial t} = -\lambda p_b + \frac{\lambda c}{\Omega}\rho_M + \gamma_d(c)hcp_r\,, \tag{4.20}$$

with

$$\rho_M = \int (p_r + p_b)\,d\Omega\,. \tag{4.21}$$

The coarse graining of this continuous model is straightforward with the moment expansion introduced in Chapter 2. I will omit the lengthy deduction here and only present the effective diffusion coefficient

$$D_{\text{eff}} = \frac{h^2\lambda(1-c)}{d[hc\gamma_d(c)+\lambda]^2}\,. \tag{4.22}$$

If $c = 0$, it recovers the diffusion coefficient (2.19) of the free run-and-tumble particles. If $c = 1$, obviously $D_{\text{eff}} = 0$ and the particle will not be moving any more. We can expect this approximation to be invalid for the large c. If c is large enough that all empty sites forms only isolated regions, the particles cannot move to infinity after infinity time and $D_{\text{eff}} = 0$ unlike (4.22).

The effective diffusion coefficient (4.22) is a monotonic increasing function of v, but is a bell-shape function of λ. By taking $\partial D_{\text{eff}}/\partial\lambda = 0$, we have the maximum $D_{\text{eff}} = (1-c)h/4dc\gamma_d(c)$ at $\lambda = hc\gamma_d(c)$. For $\lambda < hc\gamma_d(c)$, tumbles help the particles avoid the obstacles, and increase the diffusion coefficient. It can explain the experimental observation of an increasing migration rate with the higher tumbling rate [45].

In addition, if the running particles can undergo Brownian motion, we need to add $D\nabla^2 p_r$ in the master equation (4.19), and the effective diffusion coefficient is

$$D_{\text{eff}} = \frac{h^2\lambda(1-c)}{d[hc\gamma_d(c)+\lambda]^2} + \frac{\lambda(1-c)}{\lambda + hc\gamma_d(c)}D\,. \tag{4.23}$$

It means the obstacles can also slow down the Brownian motion of the run-and-tumble particles.

4.4 Simulation

We can apply the Gillespie simulation (for details, please refer to Section 7.1.1) to investigate the system defined by the master equations (4.18). Consider a square lattice with the periodic boundary condition. I put the obstacle with the probability c on each site, and put the non-interacting particles randomly on the vacant sites. The simulation was run in the continuous time. I sampled the happening time of every events, and calculated the average displacement of all particles after the hopping events. Then a linear fitting is done to the later part of $\langle \Delta x^2(t) \rangle$ to get the effective diffusion coefficient in the large time scales.

Firstly, the influence of the obstacles can be checked, as shown in Figure 4.1. The diffusion coefficient was decreasing with more obstacles, both in 2D and 3D. The continuous approximation captured the tendency assuming $\gamma_d(c) = 1$, but didn't fit the simulation quantitatively. Here we should mention the discrepancy of the simulation and the theory at $c = 0$ is mainly due to the artificial diffusivity $h/2d$ of discretization discussed in Section 4.1.

As the continuous approximation, the diffusivity increased monotonously with the increasing hopping rate, regardless of c, as in Figure 4.2. But unlike the situation without the obstacles in which $D \propto v^2$, $\lim_{h \to \infty} D$ is finite. Higher hopping rate cannot make the particles move faster, as they will eventually hit on the obstacles, and only tumbles can help them escape.

For the low c, the 2D system has larger diffusivity than the 3D system, as the diffusivity of the system with no obstacles should decrease with dimension. We noticed that for the high c, the diffusivity in the 3D system is larger than the 2D system. It may because the higher dimension makes the particles to have more choice to escape from the obstacles.

If we plot D with respect to λ, as in Figure 4.3, we can found a bell shape curve as we expected in last section. Without tumbles, the particles will finally hit on the obstacles and stop moving. But the high tumbling rate also makes the cell not able to move. Thus it must have a local maximum in between. With proper choice of $\gamma_d(c)$, the continuous approximation can fit the simulation pretty well for the small λ. The simulations with the large λ are close to violate the condition $h \gg \lambda/2$, which may explain the inconsistency.

Interestingly, it seems $\gamma_d(c)$, which relates with the effective concentration of the obstacles, increases with c but decreases with d. We can't assert anything for now. More data are needed to confirm this observation.

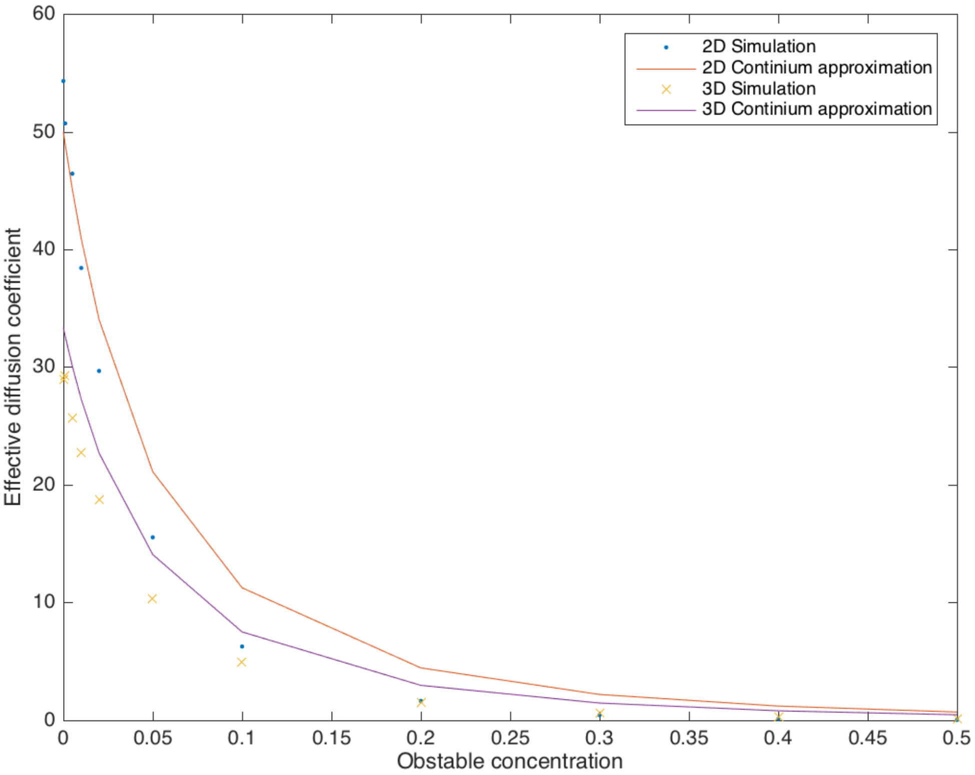


Figure 4.1: Diffusion coefficient with respect to obstacle concentration. The parameters were taken to be $h = 10$ and $\lambda = 1$. The lattice size was 1024×1024, and 10000 particles were simulated. The total time was taken to be 100, and the data of $100 > t > 70$ were used in fitting the diffusion coefficient. In the theoretical prediction, $\gamma_d(c)$ was taken to be 1.

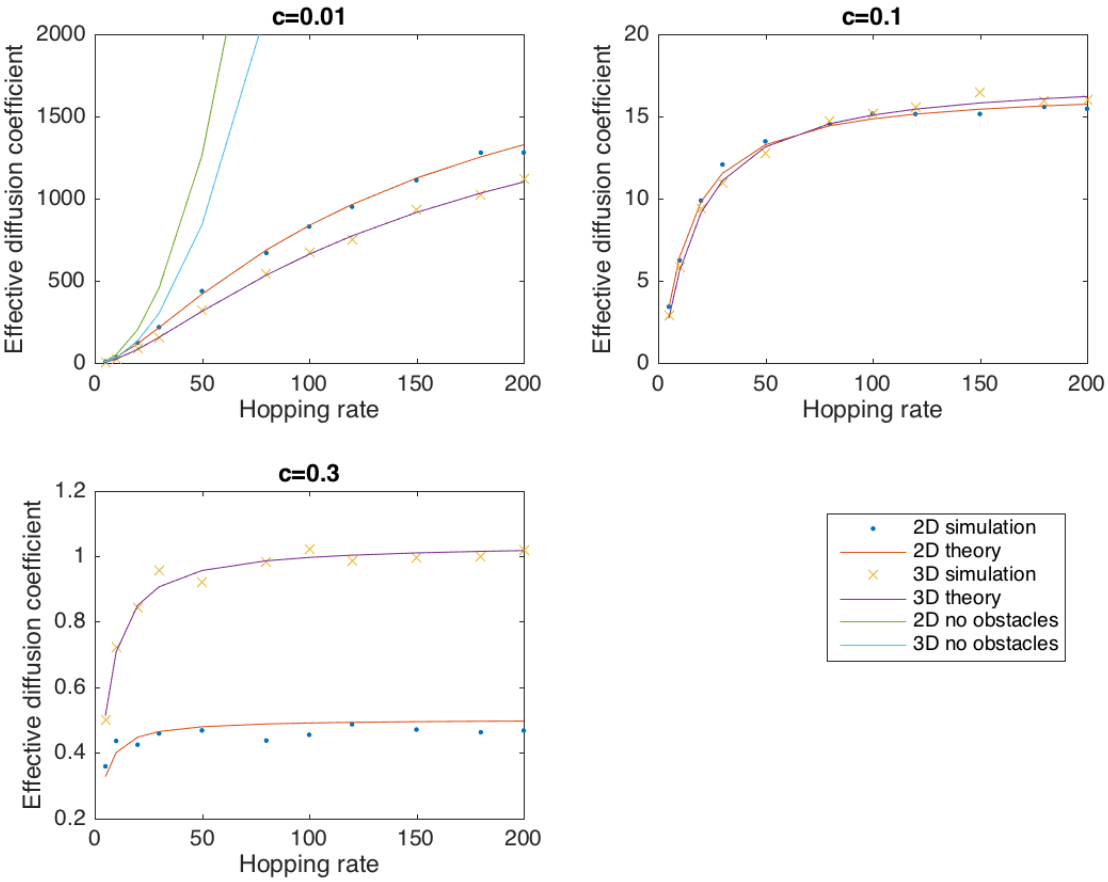


Figure 4.2: Diffusion coefficient with respect to hopping rate, with $\lambda = 1$. The lattice size was 1024×1024, and 10000 particles were simulated. The total time was taken to be 100, and the data of $100 > t > 70$ were used in fitting the diffusion coefficient. In the theoretical prediction, I took $\gamma_2(0.01) = 1.43$, $\gamma_3(0.01) = 1.23$, $\gamma_2(0.1) = 1.64$, $\gamma_3(0.1) = 1.31$, $\gamma_2(0.3) = 2.78$, $\gamma_3(0.3) = 1.58$.

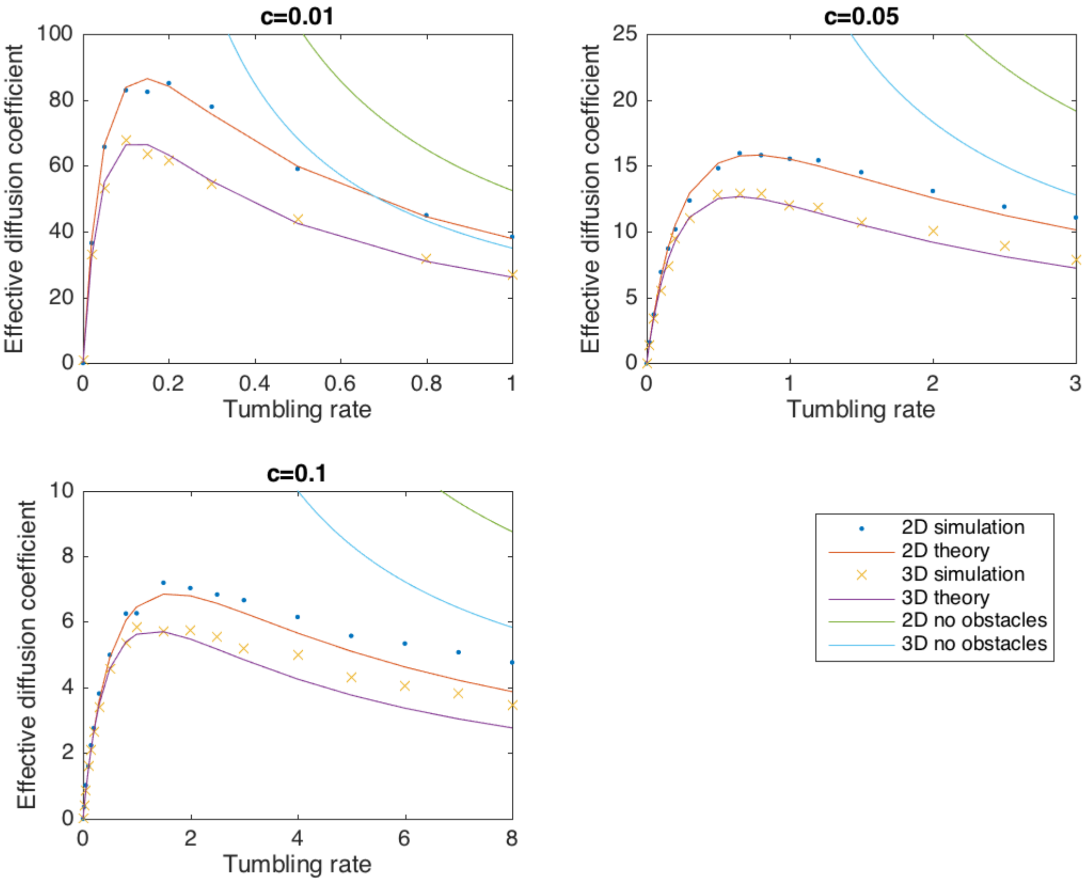


Figure 4.3: Diffusion coefficient with respect to tumbling rate, with $v = 10$. The lattice size was 1024×1024, and 10000 particles were simulated. The total time was taken to be 100, and the data of $100 > t > 70$ were used in fitting the diffusion coefficient. In the theoretical prediction, I took $\gamma_2(0.01) = 1.43$, $\gamma_3(0.01) = 1.23$, $\gamma_2(0.05) = 1.5$, $\gamma_3(0.05) = 1.25$, $\gamma_2(0.1) = 1.64$, $\gamma_3(0.1) = 1.31$.

Chapter 5

Interaction of two species

Now we consider the system with interactions between two populations of particles, based on the density dependent mobility regulations. Such systems can have phase separation and form spatial structures, and can provide a possible mechanism for segregation and co-migration. In the biological context, such system can correspond to two species of cells which can sense the density of each other. Such so-called quorum sensing can be achieved via some short-lifetime signal molecules. This is still a on-going project. In this chapter, I will present some preliminary theoretical result originally from our collaborators and experimental data.

5.1 Density dependent mobility regulation

First let's investigate how the density dependent mobility will affect the development of the particle density field. As the experimental time scales ($\sim$ 10h) for observing bacteria colony development is much longer than $1/\lambda \sim$ 1s, we can make use of the diffusion limit (3.14) introduced in Section 3.1.

$$\frac{\partial \rho_a}{\partial t} = \nabla \left[\frac{v_a}{d\lambda_a} \nabla \left(\frac{v_a}{1 + \lambda_a \delta_a} \rho_a \right) \right] . \tag{5.1}$$

Here the Brownian diffusion is neglected, $\rho_a(\boldsymbol{x}, t)$ is the density field of particle a, and the parameters may depend on the density field $\rho_b(\boldsymbol{x}, t)$ of particle b.

If we assume $v_a = v_a(\rho_b)$, we have

$$\frac{\partial \rho_a}{\partial t} = \nabla \left[\frac{v_a^2(\rho_b)}{d\lambda_a(1 + \lambda_a \delta_a)} \nabla \rho_a + \frac{v_a(\rho_b) v_a'(\rho_b) \rho_a}{d\lambda_a(1 + \lambda_a \delta_a)} \nabla \rho_b \right] . \tag{5.2}$$

To do the linear analysis, we assume the density field to fluctuate around an invariant homogeneous field, and write the perturbation in Fourier transformation,

$$\rho(\boldsymbol{x}, t) = \rho_0 + \int \delta\rho(\boldsymbol{q}, t) e^{i\boldsymbol{q}\cdot\boldsymbol{x}}\, d^d x\,. \tag{5.3}$$

Substituting into (5.2), and leaving only the linear terms, we have

$$\frac{\partial \delta\rho_a}{\partial t} = -q^2 \left[\frac{v_a^2(\rho_{b0})}{d\lambda_a(1+\lambda_a\delta_a)} \delta\rho_a + \frac{v_a(\rho_{b0}) v_a'(\rho_{b0})\rho_{a0}}{d\lambda_a(1+\lambda_a\delta_a)} \delta\rho_b \right]. \tag{5.4}$$

The variations of ρ_a are damped by the diffusion term. Hence the first term predict an exponential decay of a perturbation, and of little importance in the large time scales. The later term means the dynamic of ρ_a will response to the fluctuation of ρ_b. We should notice the response depends on the sign of $v_a'(\rho_{b0})$. If $v_a'(\rho_{b0}) < 0$, which means the velocity of particle a decreases where particle b is dense, $\delta\rho_a$ will tend to have the same shape with $\delta\rho_b$ and the two species tend to co-migrating. Otherwise if $v_a'(\rho_{b0}) > 0$, $\partial\delta\rho_a/\partial t \propto -\delta\rho_b$ and the two species begin to segregate.

The similar conclusions can be made for λ and δ with the same method. The co-migration happens when $v_a'(\rho_{b0}) < 0$, $\lambda_a'(\rho_{b0}) > 0$, or $\delta_a'(\rho_{b0}) > 0$. The segregation happens when $v_a'(\rho_{b0}) > 0$, $\lambda_a'(\rho_{b0}) < 0$, or $\delta_a'(\rho_{b0}) < 0$. In summary, if the mobility of one particle is inhibited by the density of the other one, the two populations tend to co-migrate, otherwise they tend to segregate. It can be easily understood, as particles will be stuck and accumulate where their mobility is low.

5.2 Mutual interactions

If we take into account of the mutual interactions, we can have phase separation and spatial inhomogeneous structure. The following discussion is originally made by our collaborators Agnese Curatolo and Julien Tailleur in their unpublished note for the case of density-dependent velocity. Here, I would like to generalize it to density-dependent tumbling rate and duration, as from the biochemical pathways the regulation may be on the tumbling rate or duration.

Let's consider two populations of particles ρ_a and ρ_b, and assume the mutual interaction is regulating the tumbling rate. So $\lambda_a = \lambda_a(\rho_b)$, $\lambda_b = \lambda_b(\rho_a)$. From (3.14) we have

$$\frac{\partial \rho_a}{\partial t} = \nabla \left[\frac{v_a^2}{d\lambda_a(\rho_b)(1+\lambda_a(\rho_b)\delta_a)} \nabla\rho_a - \frac{v_a^2 \delta_a \lambda_a'(\rho_b)\rho_a}{d\lambda_a(\rho_b)(1+\lambda_a(\rho_b)\delta_a)^2} \nabla\rho_b \right], \tag{5.5}$$

$$\frac{\partial \rho_b}{\partial t} = \nabla \left[\frac{v_b^2}{d\lambda_b(\rho_a)(1+\lambda_b(\rho_a)\delta_b)} \nabla\rho_b - \frac{v_b^2 \delta_b \lambda_b'(\rho_a)\rho_b}{d\lambda_b(\rho_a)(1+\lambda_b(\rho_a)\delta_b)^2} \nabla\rho_a \right]. \tag{5.6}$$

To show the occurrence of instability, we perform the linear stability analysis. We insert

$$\rho(\boldsymbol{x},t) = \rho_0 + \int \delta\rho(\boldsymbol{q},t) e^{i\boldsymbol{q}\cdot\boldsymbol{x}}\, d^d x \tag{5.7}$$

into (5.5) and (5.6). Keeping only the linear terms, we have

$$\frac{\partial \delta\rho_a}{\partial t} = -\, q^2 \left[\frac{v_a^2}{d\lambda_a(\rho_{b0})(1+\lambda_a(\rho_{b0})\delta_a)} \delta\rho_a - \frac{v_a^2 \delta_a \lambda_a'(\rho_{b0})\rho_{a0}}{d\lambda_a(\rho_{b0})(1+\lambda_a(\rho_{b0})\delta_a)^2} \delta\rho_b \right] , \tag{5.8}$$

$$\frac{\partial \delta\rho_b}{\partial t} = -\, q^2 \left[\frac{v_b^2}{d\lambda_b(\rho_{a0})(1+\lambda_b(\rho_{a0})\delta_b)} \delta\rho_b - \frac{v_b^2 \delta_b \lambda_b'(\rho_{a0})\rho_{b0}}{d\lambda_b(\rho_{a0})(1+\lambda_b(\rho_{a0})\delta_b)^2} \delta\rho_a \right] . \tag{5.9}$$

If the eigenvalues of the coefficient matrix are all negative, the perturbation $\delta\rho_{a,b}$ will decay with time, and the homogeneous states will be stable states. However, if an eigenvalue become positive, the homogeneous state loses its stability. After simple calculation, the bifurcation condition for the larger eigenvalue η to be positive is

$$\lambda_a'(\rho_{b0})\lambda_b'(\rho_{a0}) > \frac{(1+\lambda_a(\rho_{a0})\delta_a)(1+\lambda_b(\rho_{b0})\delta_b)}{\delta_a\delta_b\rho_{a0}\rho_{b0}} , \tag{5.10}$$

and the corresponding eigenvector satisfies

$$\left(\frac{v_a^2}{d\lambda_a(\rho_{b0})(1+\lambda_a(\rho_{b0})\delta_a)} + \eta \right) \delta\rho_a = \frac{v_a^2 \delta_a \lambda_a'(\rho_{b0})\rho_{a0}}{d\lambda_a(\rho_{b0})(1+\lambda_a(\rho_{b0})\delta_a)^2} \delta\rho_b \, . \tag{5.11}$$

As all the parameters are positive, (5.10) means $\lambda_a'(\rho_{b0})$ and $\lambda_b'(\rho_{a0})$ should have the same sign to have spatial inhomogeneous stable state. $\lambda_a'(\rho_{b0}) > 0$, $\lambda_b'(\rho_{a0}) > 0$ corresponds to the mutual inhibition of motility, and $\lambda_a'(\rho_{b0}) < 0$, $\lambda_b'(\rho_{a0}) < 0$ corresponds to the mutual activation. For the mutual inhibition, (5.11) indicates that $\delta\rho_a$ and $\delta\rho_b$ have the same sign, which means two populations are in the same phase when instability emerges. For the mutual activation, the two populations are out of phase as $\delta\rho_a \propto -\delta\rho_b$.

With only the mobility terms, the bifurcation is not the Turing bifurcation, as the eigenvalues are independent on q. To have a spatial pattern, we need some factor to have the coarsening effect to select a specific length scale. Growth can be one of the candidates. We can include a competing logistic growth term, and write (5.5) and (5.6) as

$$\frac{\partial \rho_a}{\partial t} = \nabla \left[D_a(\rho_b)\nabla\rho_a + \frac{D_a'(\rho_b)\lambda_a(\rho_b)\delta_a\rho_a}{(1+2\lambda_a(\rho_b)\delta_a)} \nabla\rho_b \right] + k_a\rho_a \left(1 - \frac{\rho_a+\rho_b}{M} \right) , \tag{5.12}$$

$$\frac{\partial \rho_b}{\partial t} = \nabla \left[D_b(\rho_a)\nabla\rho_b + \frac{D_b'(\rho_a)\lambda_b(\rho_a)\delta_b\rho_b}{(1+2\lambda_b(\rho_a)\delta_b)} \nabla\rho_a \right] + k_b\rho_b \left(1 - \frac{\rho_a+\rho_b}{M} \right) , \tag{5.13}$$

where $D = v^2/d\lambda(1+\lambda\delta)$ is the diffusion coefficient we have met before, and M is the environment capacity. k_a and k_b are the corresponding growth rate. The appearance of growth term largely complicate the analysis. The calculation is lengthy and is omitted here. For the mutual activation situation, we can find a certain region in parameter space that the largest eigenvalue is positive in a finite range of q, as shown in Figure 5.1. Here an asymmetry in parameters is necessary for patterning. For the mutual inhibition case, it is more complex. The growth term only is not enough for patterning. In M.E. Cates, 2010 [21], a $-\kappa\nabla^4\rho$ term, which is a phenomenological surface tension term, is introduced in the single-species system for patterning, and it is also true for the two-species system. A thorough discussion is beyond the scope of this thesis.

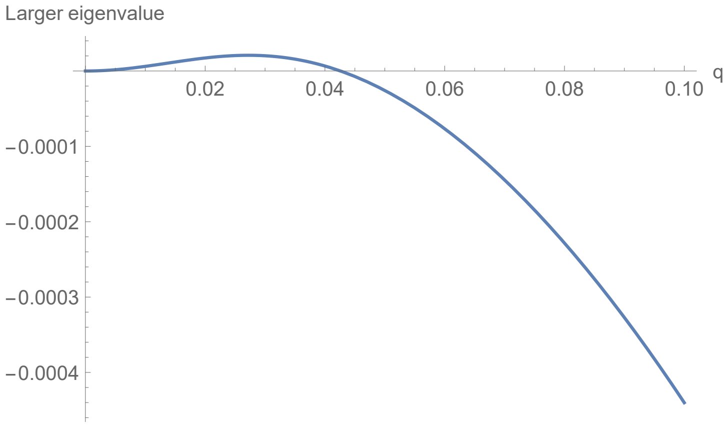


Figure 5.1: The largest eigenvalue of mutual activation system with growth term. Here we take $\delta_a = \delta_b = 0.1$, $v_a = v_b = 10$, $\rho_{a0} = \rho_{b0} = 1$, $k_a = k_b = 0.1$, $\lambda_a(\rho_{b0}) = \lambda_b(\rho_{a0}) = 1$, $\lambda'_a(\rho_{b0}) = -12.05$, $\lambda'_b(\rho_{a0}) = -10$. The eigenvalue is positive only in a finite range of q.

The same argument applies for δ and v. If $\delta_a = \delta_a(\rho_b)$, $\delta_b = \delta_b(\rho_a)$, the bifurcation condition without growth is

$$\delta'_a(\rho_{b0})\delta'_b(\rho_{a0}) > \frac{(1+\lambda_a\delta_a(\rho_{a0}))(1+\lambda_b\delta_b(\rho_{b0}))}{\lambda_a\lambda_b\rho_{a0}\rho_{b0}}\,. \tag{5.14}$$

And for v, according to the unpublished note of Agnese Curatolo, the condition without growth is

$$v'_a(\rho_{b0})v'_b(\rho_{a0}) > \frac{v_a(\rho_{b0})v_b(\rho_{a0})}{\rho_{a0}\rho_{b0}}\,. \tag{5.15}$$

After considering the growth of cells, they all show the same phenomenon: If the two populations have the mutual inhibition of mobility ($\lambda'_{a,b}(\rho_{b0,a0}) > 0$, $\delta'_{a,b}(\rho_{b0,a0}) > 0$, or $v'_{a,b}(\rho_{b0,a0}) < 0$), the system shows co-migrating pattern. If the two populations have the mutual activation of mobility ($\lambda'_{a,b}(\rho_{b0,a0}) < 0$, $\delta'_{a,b}(\rho_{b0,a0}) < 0$, or $v'_{a,b}(\rho_{b0,a0}) > 0$), the system shows segregating pattern.

The instability of homogeneous state can be understood intuitively. If the density of particles a increases a little bit from the homogeneous state, particles b will move faster (or slower). Particles b then accumulate at where they move slower, which is where a are sparser (or denser). Sparser (or denser) b make a move slower, causing a to be more denser. Thus the fluctuation is enhanced and the homogeneous state loses its stability.

5.3 Experimental observation

The pattern suggested in the last section can be realized in the experiments. We can engineer the bacteria such as *E. coli* to have such quorum sensing system. The construct of such system was done by Zhou Nan in our lab. Besides inserting the quorum sensing circuits, he made the two populations to express two different fluorescence proteins constantly, in order to distinguish them in imaging.

Then we need to build the imaging setup for the large scales fluorescence imaging. The difficulty in this setup is to trade off between the height limitation and the problem of large incident angle. For the fluorescence imaging, we need to illuminate the sample with monochromatic light, and separate the weak fluorescence signal with the illumination light. The common filters we can access is the interference filters, which require perpendicular incidence of light. For microscopes, it is easily satisfied as the scale of the image is small enough compared to the filter size and the working distance. But for the large scale imaging, the sample is of $\sim$ 10cm diameter while the common filters are only of $\sim$ 2cm diameter. So we need to increase the working distance of the setup to prevent the problem of large incident angle.

The design of the setup is shown in Figure 5.2. We used a rotational stage in this setup to image multiple samples simultaneously. Putting the whole setup in the 37°C degree warm room, we can do the time-lapse live imaging. The diffusor is to make the illumination homogeneously for quantitative imaging. We should notice that the diffusor may increase the incident angle of the excitation light. Thus for GFP, the band of the excitation filter and the emission filter should not be too close. If necessary, we can tilt the lighting system to avoid the light spot of the light source in the image, and use Fresnel lens to re-focus the background light to avoid the camera.

Then we can do the pattern formation experiments with the engineered *E. coli* strains. We mixed the two strains of *E. coli* in liquid LB, and planted a drop of the mixture in the middle of the semi-solid agar plates with half LB. The plates were kept in a 37° degree warm room for 10 hours for patterns to grow. The samples were prepared by Zhou Nan in our lab.

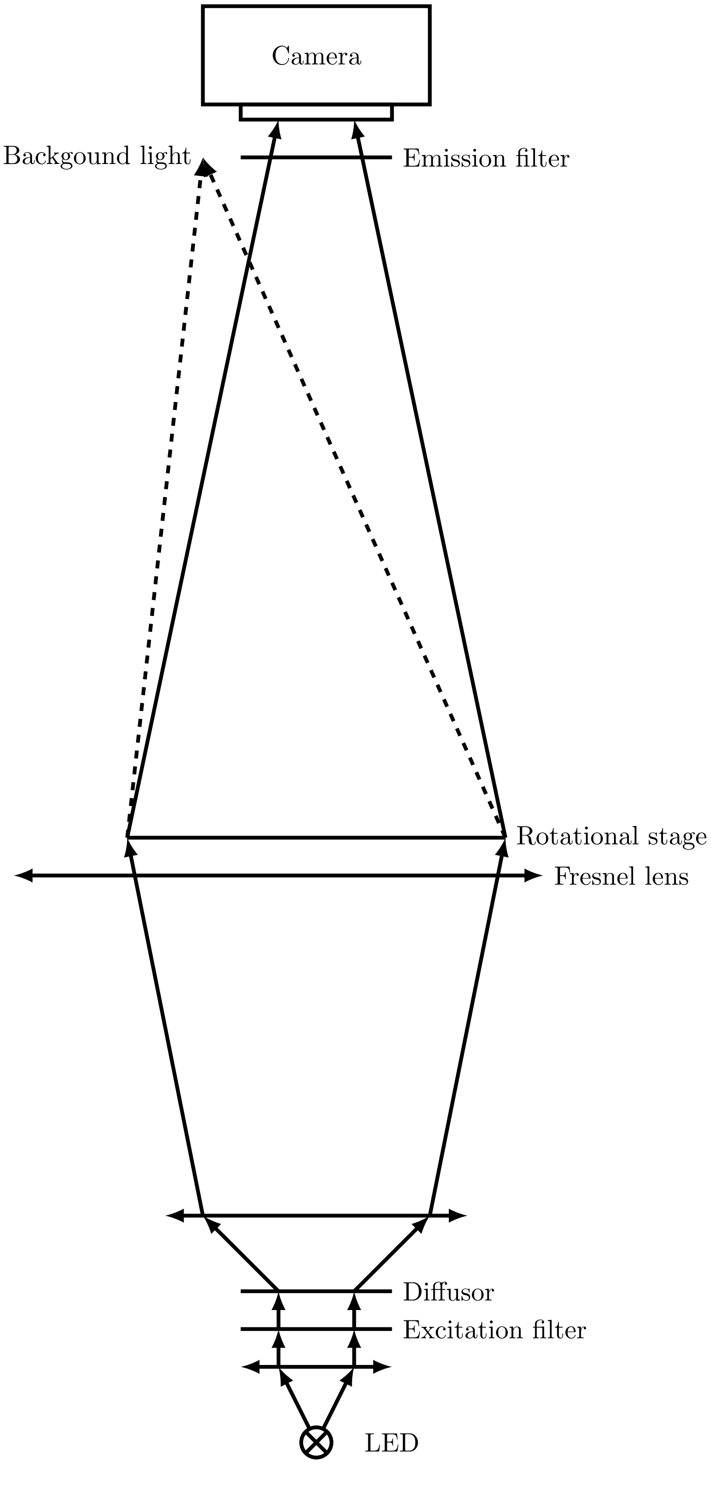


Figure 5.2: Design of large scale fluorescence imaging setup.

The preliminary data is shown in Figure 5.3. We can see the experimental result can support our prediction from the theory. The mutual activation strains show a segregating ring pattern, and the mutual inhibition strains show a co-migrating ring pattern. The inhomogeneous background may due to the inhomogeneous illumination of our old setup. The further work will be discussed in Chapter 9.

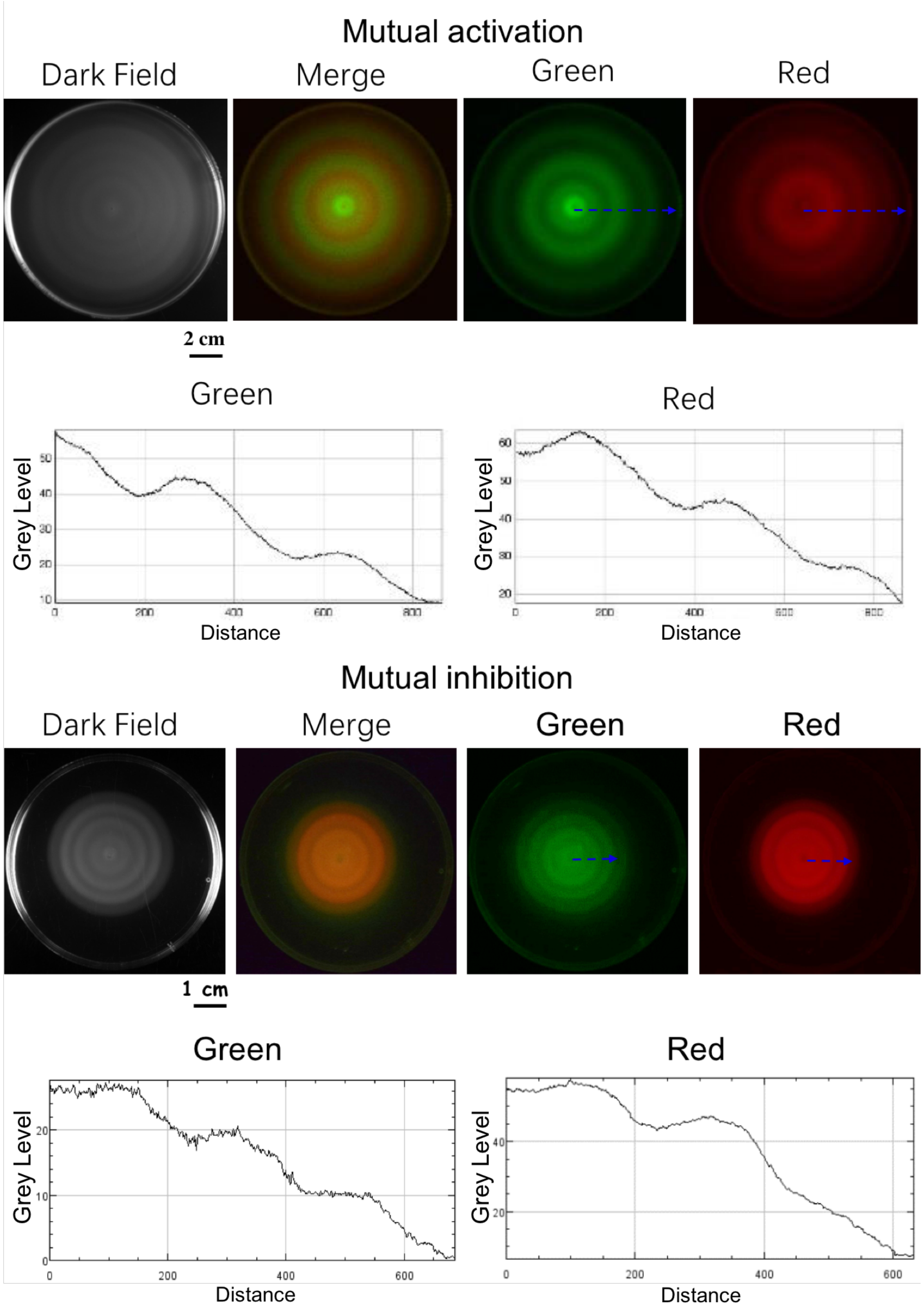


Figure 5.3: Patterns formed by two strains of *E. coli* with mutual activating and mutual inhibiting motility. The two strains were labelled with GFP and mCherry respectively. The fluorescence intensity profiles on the blue dashed line in the photos are shown below.

Chapter 6

Differential dynamic microscopy (DDM)

Now we introduce a novel method, based on the formulism used in Chapter 2 and 3, for measuring the microscopic parameters of the run-and-tumble particles, such as the swimming speed and the tumbling rate.

6.1 Intermediate scattering function (ISF)

Considering point particles, the microscopic particle density can be written as

$$\rho(\boldsymbol{x}, t) = \sum_{k=1}^{N} \delta(\boldsymbol{x} - \boldsymbol{x}_k(t)) \,, \tag{6.1}$$

where the $\boldsymbol{x}_i(t)$ is the trajectory of the i-th particle, N is the number of the particles, and $\delta(x)$ is the Dirac delta function. The instantaneous microscopic particle density is related to the smooth macroscopic density by an ensemble average

$$\rho_M(\boldsymbol{x}, t) = \langle \rho(\boldsymbol{x}, t) \rangle_e \,. \tag{6.2}$$

The Fourier transform of the microscopic density is

$$\rho(\boldsymbol{q}, t) = \sum_{k=1}^{N} e^{-i\boldsymbol{q}\cdot\boldsymbol{x}_k(t)} \,. \tag{6.3}$$

We define the **intermediate scattering function (ISF)** as the normalized autocorrelation of the particle density fluctuation

$$f(\boldsymbol{q},\tau)=\frac{\langle\Delta\rho(\boldsymbol{q},\tau)\Delta\rho^*(\boldsymbol{q},0)\rangle_e}{\langle\Delta\rho(\boldsymbol{q},0)\Delta\rho^*(\boldsymbol{q},0)\rangle_e}\,, \tag{6.4}$$

where $\Delta\rho(\boldsymbol{q},0)=\rho(\boldsymbol{q},0)-\langle\rho\rangle_e$ is the fluctuation of the density around its expectation. The ensemble is taken with the same system with the same population of the particles. The initial position of each particle should obey the uniform probability distribution.

If the system is ergodic (as diffusion and tumble make the particle be able to reach every position in the configuration space), we can exchange all the ensemble average with the time average

$$\langle A(0)\rangle_e=\langle A(t)\rangle_t\,. \tag{6.5}$$

Here the time average is defined as

$$\langle f(t)\rangle_t=\lim_{T\to\infty}\frac{1}{T}\int_{-T/2}^{T/2}f(t)\,dt\,, \tag{6.6}$$

and is invariant under time translation.

By direct substitution, it is easily shown that

$$\langle\Delta\rho(\boldsymbol{q},t+\tau)\Delta\rho^*(\boldsymbol{q},t)\rangle_t=\langle\rho(\boldsymbol{q},t+\tau)\rho^*(\boldsymbol{q},t)\rangle_t-|\langle\rho(\boldsymbol{q},t)\rangle_t|^2\,, \tag{6.7}$$

and

$$\langle\Delta\rho(\boldsymbol{q},t)\Delta\rho^*(\boldsymbol{q},t)\rangle_t=\langle|\rho(\boldsymbol{q},t)|^2\rangle_t-|\langle\rho(\boldsymbol{q},t)\rangle_t|^2\,, \tag{6.8}$$

$\rho(\boldsymbol{q}=\boldsymbol{0},t)=N$ is just the total number of particles, which is assumed to be constant and of no interest. So we only focus on $\boldsymbol{q}\neq\boldsymbol{0}$. Considering the non-interacting point particles, we have

$$\begin{aligned}\langle\rho(\boldsymbol{q},t+\tau)\rho^*(\boldsymbol{q},t)\rangle_t&=\langle\sum_{j,k=1}^{N}e^{-i\boldsymbol{q}\cdot(\boldsymbol{x}_k(t+\tau)-\boldsymbol{x}_j(t))}\rangle_t\\&=\sum_{k=1}^{N}\langle e^{-i\boldsymbol{q}\cdot(\boldsymbol{x}_k(t+\tau)-\boldsymbol{x}_k(t))}\rangle_t+\sum_{k\neq j}\langle e^{-i\boldsymbol{q}\cdot(\boldsymbol{x}_k(t+\tau)-\boldsymbol{x}_j(t))}\rangle_t\,.\end{aligned} \tag{6.9}$$

We assume the motion of different particles is uncorrelated statistically, the latter term becomes

$$\sum_{k\neq j}\langle e^{-i\boldsymbol{q}\cdot(\boldsymbol{x}_k(t+\tau)-\boldsymbol{x}_j(t))}\rangle_t=\sum_{k\neq j}\langle e^{-i\boldsymbol{q}\cdot\boldsymbol{x}_k(t)}\rangle_t\langle e^{i\boldsymbol{q}\cdot\boldsymbol{x}_j(t)}\rangle_t\,. \tag{6.10}$$

Similarly,

$$\begin{aligned}\langle \rho(\boldsymbol{q},t)\rho^*(\boldsymbol{q},t)\rangle_t &= \sum_{k=1}^{N}\langle e^{-i\boldsymbol{q}\cdot(\boldsymbol{x}_k(t)-\boldsymbol{x}_k(t))}\rangle_t + \sum_{k\neq j}\langle e^{-i\boldsymbol{q}\cdot(\boldsymbol{x}_k(t)-\boldsymbol{x}_j(t))}\rangle_t \\ &= N + \sum_{k\neq j}\langle e^{-i\boldsymbol{q}\cdot\boldsymbol{x}_k(t)}\rangle_t \langle e^{i\boldsymbol{q}\cdot\boldsymbol{x}_j(t)}\rangle_t\,. \end{aligned} \tag{6.11}$$

Then the ISF becomes

$$f(\boldsymbol{q},\tau) = \frac{\sum_{k=1}^{N}\langle e^{-i\boldsymbol{q}\cdot(\boldsymbol{x}_k(t+\tau)-\boldsymbol{x}_k(t))}\rangle_t - \sum_{k=1}^{N}|\langle e^{-i\boldsymbol{q}\cdot\boldsymbol{x}_k(t)}\rangle_t|^2}{N-\sum_{k=1}^{N}|\langle e^{-i\boldsymbol{q}\cdot\boldsymbol{x}_k(t)}\rangle_t|^2}\,. \tag{6.12}$$

Now exchanging the time average with the ensemble average, we have

$$\langle e^{-i\boldsymbol{q}\cdot\boldsymbol{x}_k(t)}\rangle_t = \langle e^{-i\boldsymbol{q}\cdot\boldsymbol{x}_k(0)}\rangle_e = \frac{1}{V}\int_V e^{-i\boldsymbol{q}\cdot\boldsymbol{x}}\,d^3x\,. \tag{6.13}$$

As we take a large enough system that the boundary effect can be neglected, we take the limitation that $V\to\infty$, the integration will tend to $\delta(\boldsymbol{q})$. Thus we have

$$\sum_{k=1}^{N}|\langle e^{-i\boldsymbol{q}\cdot\boldsymbol{x}_k(t)}\rangle_t|^2 = 0\,,\text{for } q\neq 0\,, \tag{6.14}$$

and the ISF becomes

$$f(\boldsymbol{q},\tau) = \frac{1}{N}\sum_{k=1}^{N}\langle e^{-i\boldsymbol{q}\cdot(\boldsymbol{x}_k(\tau)-\boldsymbol{x}_k(0))}\rangle_e = \frac{1}{N}\sum_{k=1}^{N}\int e^{-i\boldsymbol{q}\cdot\boldsymbol{x}}\langle\delta(\boldsymbol{x}-\Delta\boldsymbol{x}_k(\tau))\rangle_e\,d^3x\,, \tag{6.15}$$

where $\Delta\boldsymbol{x}_k(\tau)=\boldsymbol{x}_k(\tau)-\boldsymbol{x}_k(0)$. If we construct a new ensemble from the real one, with the systems in which we translate the initial position of every particle to be at the origin, $\boldsymbol{x}_k(0)=0$. Then the summation

$$\sum_{k=1}^{N}\delta(\boldsymbol{x}-\Delta\boldsymbol{x}_k(\tau)) = \rho(\boldsymbol{x},\tau) \tag{6.16}$$

is just the microscopic density of the new system, and its ensemble average is just the macroscopic density, so

$$f(\boldsymbol{q},\tau) = \frac{1}{N}\int e^{-i\boldsymbol{q}\cdot\boldsymbol{x}}\rho_M(\boldsymbol{x},\tau)\,d^3x = \frac{1}{N}\rho_M(\boldsymbol{q},\tau) = p(\boldsymbol{q},\tau)\,. \tag{6.17}$$

Here $p(\boldsymbol{q},\tau)$ is the Fourier transform of the spatial probability distribution given that the initial distribution to be $\delta(\boldsymbol{x})$, and can be solved from the corresponding master equation. We have already solved it for various models.

Special caution should be taken for the straight swimmers and (6.12). Although the straight swimmers without diffusion is not ergodic, the conclusion above still holds. As the average $\langle e^{-i\boldsymbol{q}\cdot\boldsymbol{x}_k(0)}\rangle_e = \delta(\boldsymbol{q}) = 0$ already for $\boldsymbol{q} \neq 0$, the ensemble average is easily done to get (6.15). And we don't need to bridge the microscopic and the macroscopic densities, since without Brownian diffusion, the system is purely deterministic without dispersion.

It should be noted that (6.17) indicates the ISF is additive due to the additivity of probability distribution. If the system can be divided into two subsystems which obey the probability distributions p_1 and p_2, the ISF of the whole system should be $\alpha p_1 + (1-\alpha)p_2$, where α is the fraction of the population of one subsystem. We will use this property in the treatment of the models we consider.

6.2 Differential dynamic microscopy (DDM)

The two expressions of ISF in the last section give the foundation of differential dynamic microscopy.

$$f(\boldsymbol{q},\tau) = \frac{\langle \Delta\rho(\boldsymbol{q},t+\tau)\Delta\rho^*(\boldsymbol{q},t)\rangle_e}{\langle \Delta\rho(\boldsymbol{q},t)\Delta\rho^*(\boldsymbol{q},t)\rangle_e} = p(\boldsymbol{q},\tau)\,. \tag{6.18}$$

From the first expression with exchanging the ensemble average with the time average, we can measure experimentally the ISF of a given system, while the last expression enable us to theoretically calculate the ISF from its microscopic mechanism. If we compare the theoretical calculation of ISF with experimental measurement, we are able to know the detail of its microscopic mechanism.

To measure experimentally the auto-correlation of the particle density fluctuation, we follow the steps of Wilson and Martinez, et al [33]. We assume that the fluctuation of the image intensity is proportional to the fluctuation of the density,

$$\Delta I(\boldsymbol{x},t) = \kappa\Delta\rho(\boldsymbol{x},t)\,, \tag{6.19}$$

we can Fourier transform the image intensity, calculate the module square of the image difference, and do the time average

$$g(\boldsymbol{q},\tau) = \langle |I(\boldsymbol{q},t+\tau) - I(\boldsymbol{q},t)|^2\rangle_t\,. \tag{6.20}$$

Since $I(\boldsymbol{q},t) = \langle I\rangle_t + \Delta I(\boldsymbol{q},t)$ where $\langle I\rangle_t$ is independent of t, we have

$$\begin{aligned} g(\boldsymbol{q},\tau) &= \langle |I(\boldsymbol{q},t+\tau) - I(\boldsymbol{q},t)|^2\rangle_t = \langle |\Delta I(\boldsymbol{q},t+\tau) - \Delta I(\boldsymbol{q},t)|^2\rangle_t \\ &= \langle |\Delta I(\boldsymbol{q},t+\tau)|^2\rangle_t + \langle |\Delta I(\boldsymbol{q},t)|^2\rangle_t - 2\mathrm{Re}\,\langle \Delta I(\boldsymbol{q},t+\tau)\Delta I^*(\boldsymbol{q},t)\rangle_t\,. \end{aligned} \tag{6.21}$$

However, for the isotropic systems, $p(\boldsymbol{x},\boldsymbol{v},t)$ should be an even function of $\boldsymbol{x}$ as $p(\boldsymbol{x},\boldsymbol{v},0) \propto \delta(\boldsymbol{x})$ is an even function, and $p(\boldsymbol{q},\boldsymbol{v},t)$ should be real. Thus as inferred by (6.19),

$$\langle \Delta I(\boldsymbol{q},t+\tau)\Delta I^*(\boldsymbol{q},t)\rangle_t = \kappa^2\langle \Delta\rho(\boldsymbol{q},t+\tau)\Delta\rho^*(\boldsymbol{q},t)\rangle_t \tag{6.22}$$

should be real already. Then

$$\begin{aligned} g(\boldsymbol{q},\tau) &= 2\langle|\Delta I(\boldsymbol{q},t)|^2\rangle_t - 2\langle \Delta I(\boldsymbol{q},t+\tau)\Delta I^*(\boldsymbol{q},t)\rangle_t \\ &= 2\langle|\Delta I(\boldsymbol{q},t)|^2\rangle_t \left(1 - \frac{\langle\Delta\rho(\boldsymbol{q},t+\tau)\Delta\rho^*(\boldsymbol{q},t)\rangle_t}{\langle\Delta\rho(\boldsymbol{q},t)\Delta\rho^*(\boldsymbol{q},t)\rangle_t}\right) \\ &= A(\boldsymbol{q})(1 - f(\boldsymbol{q},\tau))\,, \end{aligned} \tag{6.23}$$

where $A(\boldsymbol{q}) = 2\langle|\Delta I(\boldsymbol{q},t)|^2\rangle_t$. To capture the background noise, we further add an additional term $B(\boldsymbol{q})$ to the last equation, so we have

$$g(\boldsymbol{q},\tau) = A(\boldsymbol{q})(1 - f(\boldsymbol{q},\tau)) + B(\boldsymbol{q})\,. \tag{6.24}$$

As for the isotropic systems, ρ and p should be independent of the direction of $\boldsymbol{q}$, so we can average the q direction in the experimental data and drop the direction dependence to get

$$g(q,\tau) = A(q)(1 - f(q,\tau)) + B(q)\,, \tag{6.25}$$

and this is the equation we are going to fit.

In practice, we cannot capture the vertical motion by an usual microscope, and measure $\Delta I(\boldsymbol{x},t)$ in three dimensions. The measured intensity fluctuation is actually proportional to

$$\int_{-d/2}^{d/2} \Delta\rho(x,y,z)c(z)\,dz \tag{6.26}$$

where $c(z)$ is a contrast function in the vertical direction, and d is the depth of view of the imaging system. It will not be a problem if $d \to \infty$ and $c(z) = 1$, since by means of integration, we are looking at $q_z = 0$ plane in $\boldsymbol{q}$ space without losing any information, so the two-dimensional measurement can resolve the three-dimensional motion. However, in practice, d is always finite and $c(z)$ is like [34]

$$c(z) = C_0\left(1 - \frac{4z^2}{d^2}\right)\,. \tag{6.27}$$

It will mix the contribution of the components with different q_z. Martinez, et al [34] did some simulations to investigate this effect, and found that simulated $g(q,\tau)$ was distorted for $q \lesssim 2\pi/d$, but stayed fine for $q \gtrsim 2\pi/d$.

6.3 Intermediate scattering function for different models

6.3.1 The run-and-diffuse model (RD model)

For the straight swimmers, we can use the solution we got in Section 2.5.2. In practice, we need to include the contribution of the dead cells, which are not swimming but undergoing Brownian motion. Making use of the additivity mentioned at the end of Section 6.1, the ISF for the run-and-diffuse model is

$$f(q,\tau) = (1-\alpha)e^{-Dq^2\tau} + \alpha e^{-Dq^2\tau}\frac{\sin(vq\tau)}{vq\tau}\,, \tag{6.28}$$

and α is the fraction of the swimming cells.

But this ISF is of little use in practice, so we often consider the ISF described in the following section.

6.3.2 The run-and-diffuse model with $P(v)$ (RDP model)

Usually the cells don't have an unique speed, but obey some distribution $P(v)$ instead. To implement this feature in the model, we can simply use the additivity of the probability density, and the new ISF will be

$$f(q,\tau) = (1-\alpha)e^{-Dq^2\tau} + \alpha e^{-Dq^2\tau}\int_0^\infty \frac{\sin(vq\tau)}{vq\tau}P(v)\,dv\,. \tag{6.29}$$

It's easily shown that the previous model is just a special case for $P(v) = \delta(v-\bar{v})$.

In practice, we use the Schultz distribution to approximate the real distribution, same with Wilson and Martinez, et al [33, 34].

$$P(v) = \frac{v^Z}{\Gamma(Z+1)}\left(\frac{Z+1}{\bar{v}}\right)^{Z+1} e^{-(Z+1)v/\bar{v}}\,, \tag{6.30}$$

where $\bar{v}$ is the average of v, the standard deviation is $\sigma = \bar{v}/\sqrt{Z+1}$, and $\Gamma(x)$ is the gamma function. This distribution is a bell-shape curve of v, with the maximum at $\bar{v}$. If $Z \to \infty$, it will reduce to the delta distribution $P(v) = \delta(v-\bar{v})$. Using this distribution, the integration can be evaluated analytically to get

$$f(q,\tau) = (1-\alpha)e^{-Dq^2\tau} + \alpha e^{-Dq^2\tau}\left(\frac{Z+1}{Zq\bar{v}\tau}\right)\frac{\sin(Z\arctan\xi)}{(1+\xi^2)^{Z/2}}\,, \tag{6.31}$$

where $\xi = q\bar{v}\tau/(Z+1)$.

To get an intuition of this intermediate scattering function, we can plot $f(q,\tau)$ with respect to τ with fixed q. As shown in Figure 6.1. As a normalized autocorrelation function, $f(q,0) = 1$ and $\lim_{\tau\to\infty} f(q,\tau) = 0$. Clearly the curve of $f(q,\tau)$ is divided into two parts. The separation of the two parts is due to the different time scales of diffusion and ballistic motion. The fast decay in the low $\tau < \tau_{\text{bal}} \sim 1/vq$ part is due to the ballistic motion of particles, which is regulated by $\bar{v}$. The second part with $\tau > \tau_{\text{diff}} \sim 1/Dq^2$ is a plateau of diffusion, which is regulated by D. The variation in q shifts the connection of the two parts, and the variation in α changes the relative height of the two parts as in Figure 6.2. Changing $\bar{v}$ will shift the first part but leave the second part unmoved, while change D will change the plateau of the second part but leave the first part aligned. The σ (or Z) affects mainly the shape of the junction of the two parts. The lower σ will help exhibit the oscillation due to the sinc function in ISF.

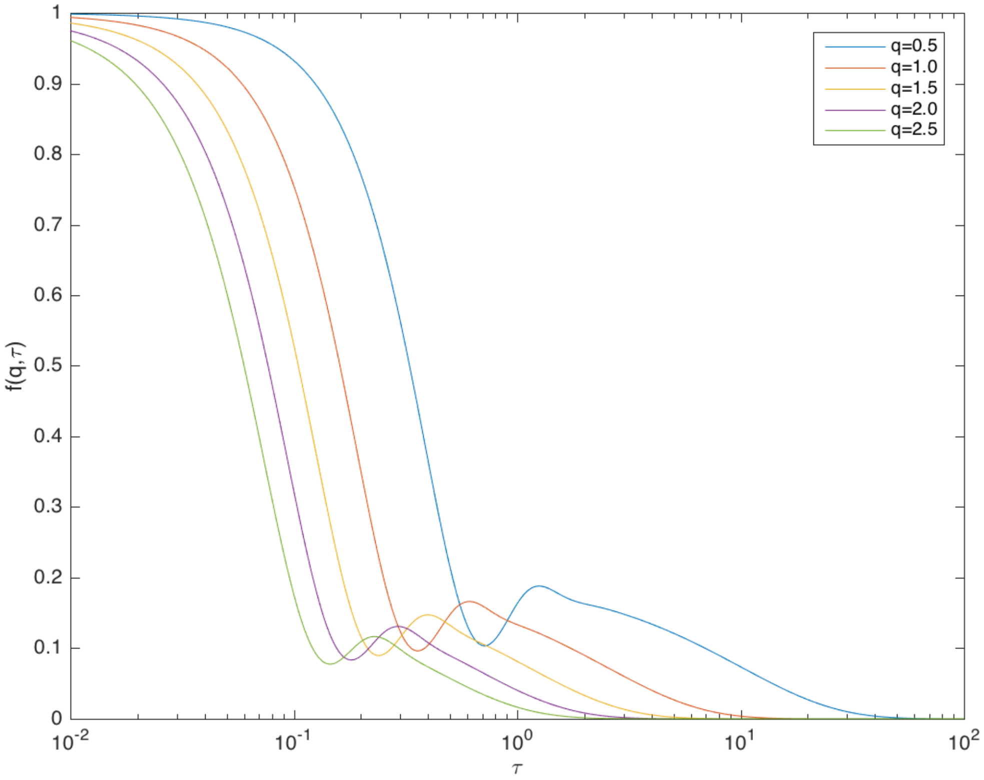


Figure 6.1: RDP ISF $f(q, \tau)$ with respect to τ with different q. The other parameters were taken to be $\alpha = 0.8$, $D = 0.4$, $\bar{v} = 13$, $Z = 15$.

6.3.3 The run-and-tumble-and-diffuse model (RTD model)

As in Section 6.3.1, we include the contribution of the non-moving cells. Recalling the result in Section 2.4, the ISF for the run-and-tumble-and-diffuse model

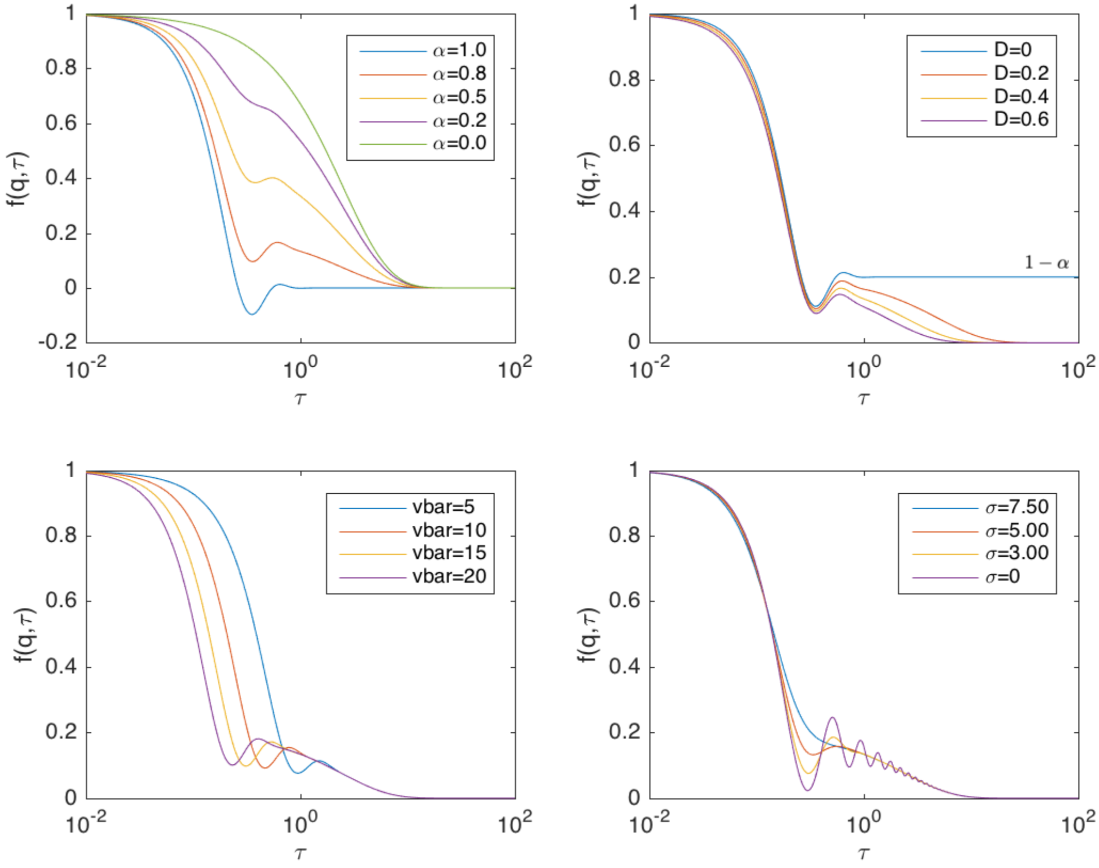


Figure 6.2: RDP ISF $f(q, \tau)$ with respect to τ with different parameters. $q = 1$ was taken in all figures. In the upper-left figure, $\bar{v} = 13$, $D = 0.4$, $Z = 15$. In the upper-right figure, $\alpha = 0.8$, $\bar{v} = 13$, $Z = 15$. In the lower-left figure, $\alpha = 0.8$, $D = 0.4$, $Z = 15$. In the lower-right figure, $\alpha = 0.8$, $\bar{v} = 15$, $D = 0.4$.

is

$$f(q,\tau) = (1-\alpha)e^{-Dq^2\tau} + \alpha p(q,\tau)\,, \tag{6.32}$$

where $p(q,\tau)$ is the inverse Laplace transform of (2.32),

$$p(q,s) = \frac{\arctan(qv/(s+Dq^2+\lambda))}{qv - \lambda \arctan(qv/(s+Dq^2+\lambda))}\,, \tag{6.33}$$

and α is the fraction of the swimming cells.

However, this ISF is also of little use in practice. The reason is the same with ISF of RD model.

6.3.4 The run-and-tumble-and-diffuse model with $P(v)$ (RTDP model)

The reason that the speed of the cells should obey some distribution $P(v)$ is identical with Section 6.3.2. Same modifications as (6.29), the ISF should be

$$f(q,\tau) = (1-\alpha)e^{-Dq^2\tau} + \alpha \int_0^\infty p(q,t)P(v)\,dv\,, \tag{6.34}$$

where $p(q,t)$ is the inverse Laplace transform of (6.33).

As in Section 6.3.2, we use the Schultz distribution to approximate the real distribution.

$$P(v) = \frac{v^Z}{\Gamma(Z+1)}\left(\frac{Z+1}{\bar{v}}\right)^{Z+1} e^{-(Z+1)v/\bar{v}}\,, \tag{6.35}$$

where $\bar{v}$ is the average of v, the standard deviation is $\sigma = \bar{v}/\sqrt{Z+1}$, and $\Gamma(x)$ is the gamma function.

To make sense of the effect of the new parameter λ, we tune λ to plot the $f(q,\tau)$, as shown in Figure 6.3. The overall features remain unchanged. The effect of λ is also on the junction of the two parts of the curve, and is quite similar to the effect of σ, except for λ has a little influence to the first decay part of the curve. The effect of λ is more significant for the low q. It suggests in the experiments we need to focus on the low q data to fit λ, and fix $\bar{v}$ and σ precisely to fit λ, otherwise the algorithm will tune $\bar{v}$ and σ instead of λ to finish the fitting. The details can be found in Section 7.5.

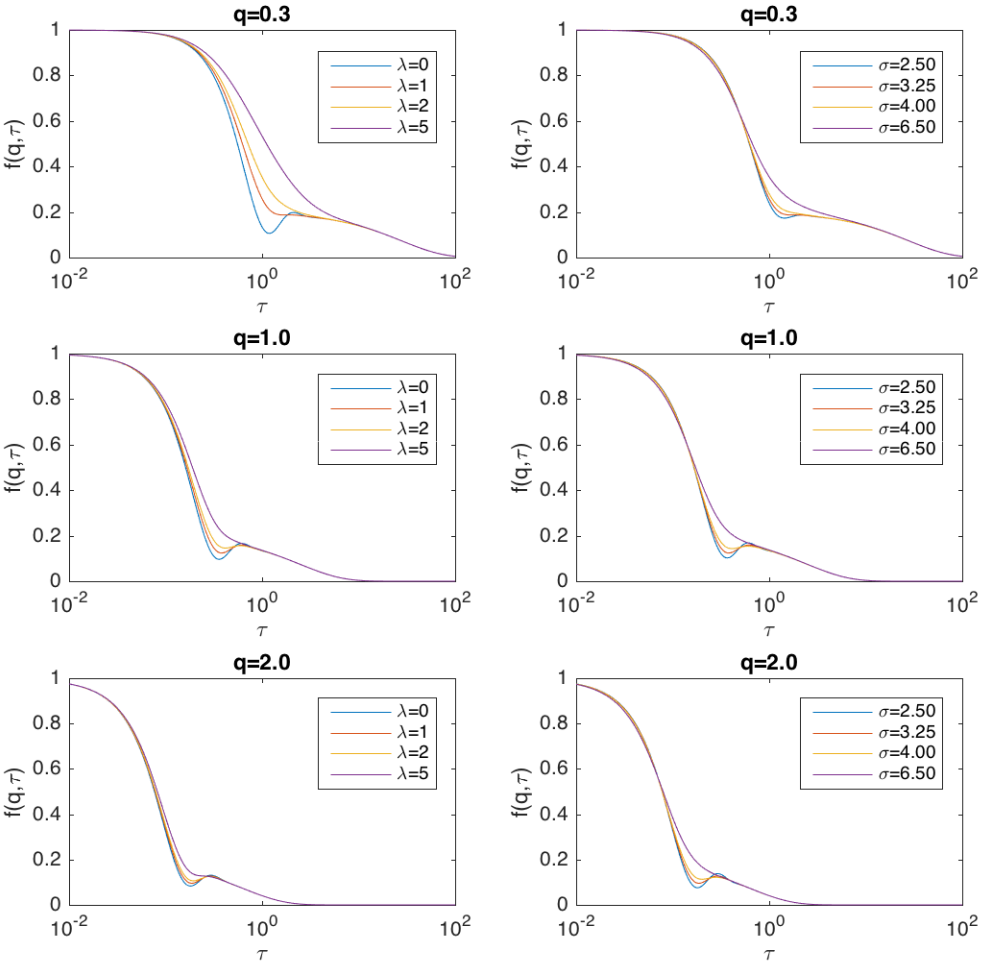


Figure 6.3: RTDP ISF $f(q,\tau)$ with respect to τ with different λ and σ. $q = 1$ was taken in both figures. In the left figure, $\alpha = 0.8$, $\bar{v} = 13$, $D = 0.4$, $Z = 15$. In the right figure, $\alpha = 0.8$, $\bar{v} = 13$, $D = 0.4$, $\lambda = 1$.

6.3.5 The run-and-tumble-and-diffuse model with $P(v)$ and tumbling time (RTDPTT model)

Same modifications with (6.29), the ISF is

$$f(q,\tau) = (1-\alpha)e^{-Dq^2\tau} + \alpha\int_0^\infty p(q,\tau)P(v)\,dv\,, \tag{6.36}$$

where $p(q,t)$ is the inverse Laplace transform of (3.29)

$$p(q,s) = \frac{qv\delta^2\lambda + [1+\delta(Dq^2+2\lambda+s)]\arctan(qv/(Dq^2+\lambda+s))}{(1+\delta\lambda)[qv(1+\delta Dq^2+\delta s) - \lambda\arctan(qv/(Dq^2+\lambda+s))]}\,, \tag{6.37}$$

and we choose the Schultz distribution again.

$$P(v) = \frac{v^Z}{\Gamma(Z+1)}\left(\frac{Z+1}{\bar{v}}\right)^{Z+1} e^{-(Z+1)v/\bar{v}}\,. \tag{6.38}$$

To make sense of the effect of the new parameter δ, we tune δ to plot the $f(q,\tau)$, as shown in Figure 6.4. The overall features remain unchanged. The effect of δ is also mainly on the junction of the two parts, and the effect is significant for a large q range. Now we have three parameters in total that work on the junction of two decay parts of ISF: σ, λ, and δ. The mixing in the time scales of the three parameters will bring us a lot of trouble in fitting. The details can be found in Section 7.7.

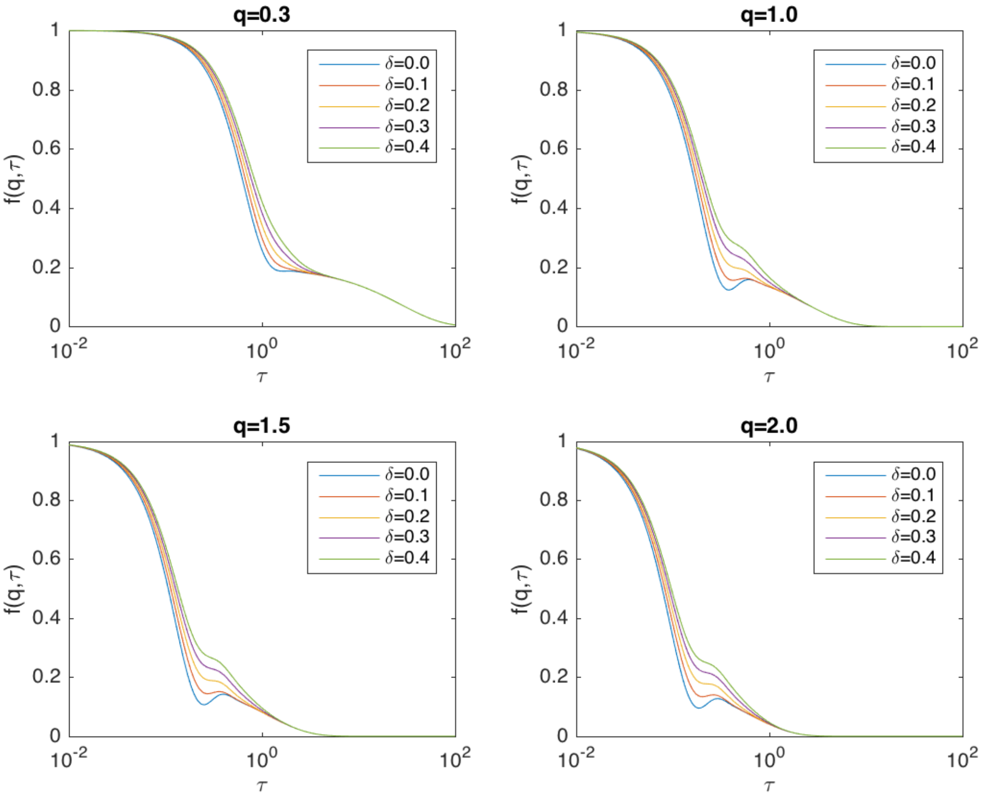


Figure 6.4: RTDPTT ISF $f(q, \tau)$ with respect to τ with different δ. The other parameters were set to be $q = 1$, $\alpha = 0.8$, $\bar{v} = 13$, $D = 0.4$, $Z = 15$ (or $\sigma = 3.25$), $\lambda = 1$.

Chapter 7

DDM Simulation

To test the algorithm for DDM, it is instructive to do the agent based simulations to generate data comparable to the experiments, and then process these simulation data to see if DDM can find the parameters we set in the simulations.

7.1 Agent based model

We used agent based simulations to simulate the motion of every particles in the system. The simulations are done in a continuous time scheme, using the traditional Gillespie stochastic simulation algorithm to sample each "event" of the system. The event may refer to tumbles or recovery from tumbling. Between two successive events, the displacement of a particle is calculated by

$$\Delta \boldsymbol{x}_i = \boldsymbol{v}_i \Delta t + \Delta \boldsymbol{x}_{di} \, , \tag{7.1}$$

where Δt is the time difference between two events, and $\Delta \boldsymbol{x}_{di} \sim N(\mu = 0, \sigma^2 = 2D\Delta t)$ is the contribution of Brownian diffusion. Here D is the Brownian diffusion constant. Then we can get the position and the velocity of every particles at every time for which we construct the intermediate scattering function $f(q, \tau)$.

7.1.1 Gillespie stochastic simulation algorithm

We sampled all the events of the system at the beginning of the simulations. As we were simulating non-interacting particles, we can sample the events of every particles separately, and then sorting all the events by time. In some simple cases we can also run the sampling with N "reactions", to get a list of the events of the whole system directly, where N is the number of the particles.

For each particle, we assume an event (tumbling or recovery) has the probability $\nu\, dt$ to happen in the time interval dt. If $P_0(\tau)$ is the probability that no event happens in range $(t, t+\tau)$, we have

$$P_0(\tau + d\tau) = P_0(\tau)\,(1 - \nu\, d\tau)\ , \tag{7.2}$$

which gives an ordinary differential equation

$$\frac{dP_0}{d\tau} = -\nu P_0\,. \tag{7.3}$$

With the initial condition $P_0(0) = 1$, we have

$$P_0(\tau) = e^{-\nu\tau} \tag{7.4}$$

and the probability that an event happens in $(t+\tau, t+\tau+d\tau)$ is

$$P_0(\tau)\nu\, d\tau = \nu e^{-\nu\tau}\, d\tau\,. \tag{7.5}$$

Then we need to sample τ according to the distribution $\nu e^{-\nu\tau}$ to generate a list of time of the events.

If we have N reactions, the probability distribution $P_i(\tau)$ that the i-th reaction is the first reaction happened in $(t+\tau, t+\tau+d\tau)$ needs a minor generalization

$$P_i(\tau)\, d\tau = P_0(\tau)\nu_i\, d\tau = \nu_i e^{-\sum_{i=1}^{N}\nu_i\tau}\, d\tau\,. \tag{7.6}$$

And in this case we need to sample τ and i successively. The single particle simulation is just the case $N = 1$.

To sample τ and i according to the above distribution, we can sample r_1 and r_2 in uniform distribution in $(0, 1)$. Then we have

$$\tau = -\frac{\ln r_1}{\sum_{j=1}^{N}\nu_j}\ , \tag{7.7}$$

and i is just the minimal integer such that

$$\sum_{j=1}^{i}\nu_j > r_2\sum_{j=1}^{N}\nu_j\,. \tag{7.8}$$

7.1.2 Rotational diffusion

To simulate the rotational diffusion of a particle, we need to sample it's direction after a period of time. Denoting the probability distribution of the direction of the particle as $P(\theta, \phi, t)$, given the initial condition $P(\theta, \phi, 0) = \delta(\theta - \theta_0)\delta(\phi - \phi_0)/\sin\theta$, it satisfies

$$\frac{\partial P}{\partial t} = -D_r\hat{L}^2 P\,, \tag{7.9}$$

where

$$\hat{L}^2 = -\frac{1}{\sin\theta}\frac{\partial}{\partial\theta}\left(\sin\theta\frac{\partial}{\partial\theta}\right) - \frac{1}{\sin^2\theta}\frac{\partial^2}{\partial\phi^2} \tag{7.10}$$

is the total angular momentum operator. Expanding $P(\theta,\phi,t)$ in the spherical harmonics

$$P(\theta,\phi,t) = \sum_{l=0}^{\infty}\sum_{m=-l}^{l} a_{lm}(t)Y_l^m(\theta,\phi) \tag{7.11}$$

where

$$Y_{lm}(\theta,\phi) = (-1)^m\sqrt{\frac{(2l+1)(l-m)!}{4\pi(l+m)!}}P_l^m(\cos\theta)e^{im\phi} \tag{7.12}$$

is the spherical harmonics, and

$$P_l^m(x) = (-1)^{l+m}\frac{(l+m)!}{(l-m)!}\frac{1}{2^l l!(1-x^2)^{m/2}}\frac{d^{l-m}}{dx^{l-m}}[(1-x^2)^l] \tag{7.13}$$

is the associated Legendre polynomial, we have

$$\frac{\partial a_{lm}}{\partial t} = -D_r l(l+1)a_{lm}\,, \tag{7.14}$$

$$a_{lm}(t) = a_{lm}(0)e^{-D_r l(l+1)t}\,. \tag{7.15}$$

As

$$a_{lm}(0) = \int_0^{2\pi} d\phi \int_0^{\pi} d\theta\, \delta(\theta-\theta_0)\delta(\phi-\phi_0)Y_{lm}^*(\theta,\phi) = Y_{lm}^*(\theta_0,\phi_0)\,, \tag{7.16}$$

we have

$$a_{lm}(t) = Y_{lm}^*(\theta_0,\phi_0)e^{-D_r l(l+1)t}\,. \tag{7.17}$$

Substituting it into (7.11) and making use of the addition theorem of the spherical harmonics

$$\sum_{m=-l}^{l} Y_{lm}^*(\theta_0,\phi_0)Y_{lm}(\theta,\phi) = \frac{2l+1}{4\pi}P_l(\cos\gamma)\,, \tag{7.18}$$

we have

$$P(\theta,\phi,t) = \sum_{l=0}^{\infty}\frac{2l+1}{4\pi}P_l(\cos\gamma)e^{-D_r l(l+1)t}\,. \tag{7.19}$$

Here $P_l(x)$ is the Legendre polynomial of degree l (P_l^0), and γ is the angle between the directions (θ,ϕ) and (θ_0,ϕ_0), satisfying

$$\cos\gamma = \sin\theta_0\sin\theta\cos(\phi-\phi_0) + \cos\theta_0\cos\theta\,. \tag{7.20}$$

Having (7.19), we can implement the rejection method to sample θ and ϕ for an arbitrary time interval t. As $P_l(x) < 1$, the probability distribution is bounded by

$$P_{\max} = \sum_{l=0}^{\infty} \frac{2l+1}{4\pi} e^{-D_r l(l+1)t} \,. \tag{7.21}$$

Then we sample $\cos\theta$ and ϕ uniformly on $(-1,1)$ and $(0,2\pi)$, and calculate $P(\theta,\phi,t)$. If another uniform random variable $\nu \in (0,1)$ satisfies $P_{\max}\nu < P(\theta,\phi,t)$, we accept the pair (θ,ϕ) as a sample.

However, for the small $\sqrt{2D_r t}$, the probability distribution will be too narrow that the algorithm has terrible efficiency. Here we used a trick to sample only a small patch around the original velocity direction. If $\gamma_T = 5\sqrt{2D_r t} < 0.4$rad, $P_l(\cos\gamma_T)$ is technically 0, we assumed the distribution is constrained in $\gamma < \gamma_T$. For simplicity, we first sampled $(\cos\theta,\phi)$ uniformly in a circular patch around the north pole $(1-\cos\gamma_T, 1) \times (0,2\pi)$, and then rotated the patch to be around the original velocity direction. As the original velocity direction is along (θ_0,ϕ_0), we first rotated around the x-axis about the angle $-\theta_0$, then around the z-axis about the angle $\phi_0 - \pi/2$. The rotation matrix was

$$\begin{aligned} R &= \begin{pmatrix} \cos\left(\phi_0 - \frac{\pi}{2}\right) & -\sin\left(\phi_0 - \frac{\pi}{2}\right) & 0 \\ \sin\left(\phi_0 - \frac{\pi}{2}\right) & \cos\left(\phi_0 - \frac{\pi}{2}\right) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(-\theta_0) & -\sin(-\theta_0) \\ 0 & \sin(-\theta_0) & \cos(-\theta_0) \end{pmatrix} \\ &= \begin{pmatrix} \sin\phi_0 & \cos\theta_0\cos\phi_0 & \sin\theta_0\cos\phi_0 \\ -\cos\phi_0 & \cos\theta_0\sin\phi_0 & \sin\theta_0\sin\phi_0 \\ 0 & -\sin\theta_0 & \cos\theta_0 \end{pmatrix} . \end{aligned} \tag{7.22}$$

Then we used the rejection method on the rotated region. This algorithm worked efficiently.

7.2 Constructed ISF

Sometimes it's hard to get an analytical expression of ISF, but we can construct it from simulations into arbitrary precision. It will help us test the model deduction and the numerical evaluation of the theoretical ISF. We can do the construction by using the first equation of (6.15)

$$f(\boldsymbol{q},\tau) = \frac{1}{N} \sum_{k=1}^{N} \langle e^{-i\boldsymbol{q}\cdot\Delta\boldsymbol{x}_k(\tau)} \rangle_e \,. \tag{7.23}$$

As the particles are independent, we can simply increase the number of particles N to improve the ensemble average. As ISF for an isotropic system is independent of the direction of $\boldsymbol{q}$, we can take an average on the directions of $\boldsymbol{q}$

to get

$$f(q,\tau) = \frac{1}{4\pi N}\sum_{k=1}^{N}\int_0^{2\pi} d\phi \int_0^{\pi} e^{-q\Delta x_k(\tau)\cos\theta} \sin\theta \, d\theta = \frac{1}{N}\sum_{k=1}^{N}\frac{\sin(q\Delta x_k)}{q\Delta x_k}\,. \tag{7.24}$$

Then we can simulate N particles, with random directions of velocity and initial positions at the origin. At each time τ, the displacement of the particles is just $\Delta x_k(\tau) = x_k(\tau) - x_k(0) = x_k(\tau)$. For each choice of q, we simply do the summation in (7.24).

Using constructed ISF, we can also find the number of particles we need to get a good statistics. As an example shown in Figure 7.1. For $N \lesssim 10^3$, the constructed ISF is quite noisy compared with the numerical evaluation of analytical expression. But for $N \gtrsim 10^4$, the statistics will be good enough to give an ISF with little noise. This means in experiments we should make the cell number in the view to be of the order higher than 10^3. In this study, we often took $N = 10^5$.

7.3 Constructed ISF for rotational diffusion

With the constructed ISF, we can investigate the effect of biased tumbling, which means the change of the particle direction is not isotropic. We simulated tumbling events as active rotational diffusion. A reference for the rotational diffusion constant is $D_{ra} = 3.5\text{s}^{-1}$, and the mean tumbling duration $\delta = 0.14\text{s}$. Then the mean tumbling angle is $\sqrt{2D_r\delta} \approx 57°$. As reported by [25], $D_{ra} = 3.5\text{s}^{-1}$ and $\delta = 0.14\text{s}$ best fitted the experiments.

The constructed ISF $f(q,\tau)$ with respect to τ for different q is shown in Figure 7.2. We noticed the effect of a finite D_{ra} was limited to the small q, and just on the junction of the two parts of the curve. So we can expect the fitting result of the data of particles with active rotational diffusion tumbling should be similar to the particles with isotropic tumbling for the large q.

Then we add a weak rotational diffusion effect during particle running to the model with active rotational diffusion, with $D_{ra} = 3.5\text{s}^{-1}$ and $\delta = 0.14\text{s}$. The constructed ISF is shown in Figure 7.3. A non-zero D_r has only a little effect on the low q region, and can be neglected for the parameter range we often meet in *E. coli* systems.

Thus we can expect the fitting of simulations with an weak rotational diffusion effect will not change much to the result of the particles without rotational diffusion. Our simulation data confirmed the expectation, but are not shown here.

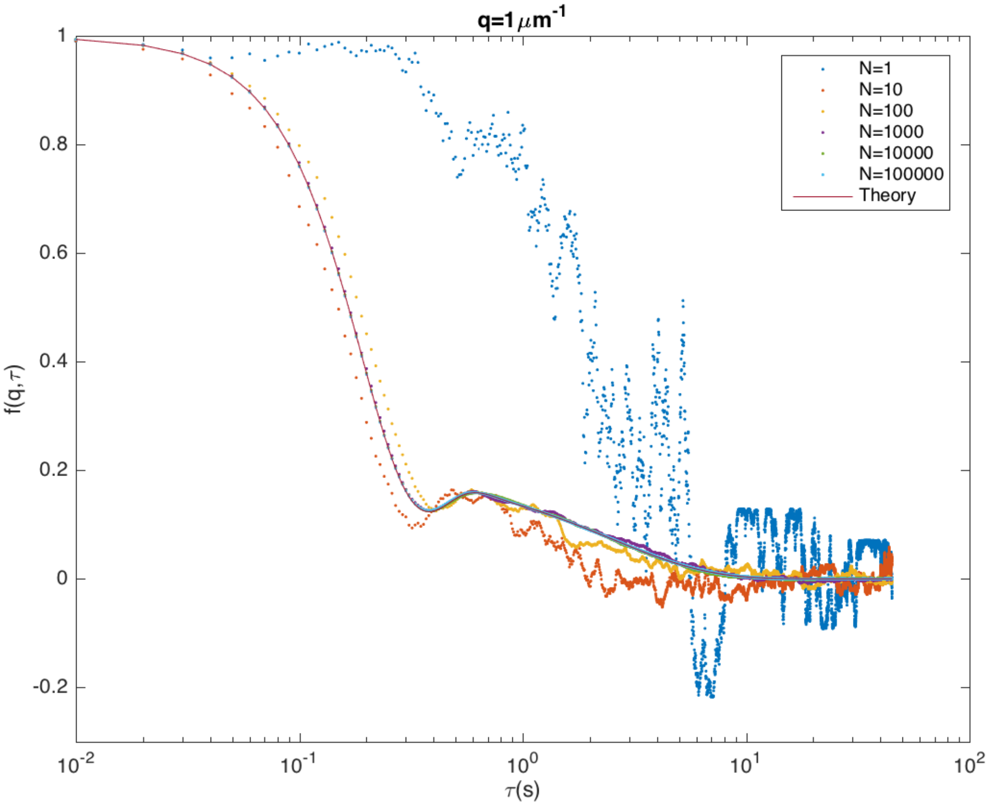


Figure 7.1: Constructed RTDP ISF for different number of particles N. The solid line is from the numerical inverse Laplace transform of analytical expression. The parameters are $\alpha = 0.8,\ v = 13\mu\text{m/s},\ Z = 15,\ \lambda = 1\text{s}^{-1},\ D = 0.4\mu\text{m}^2/\text{s}$.

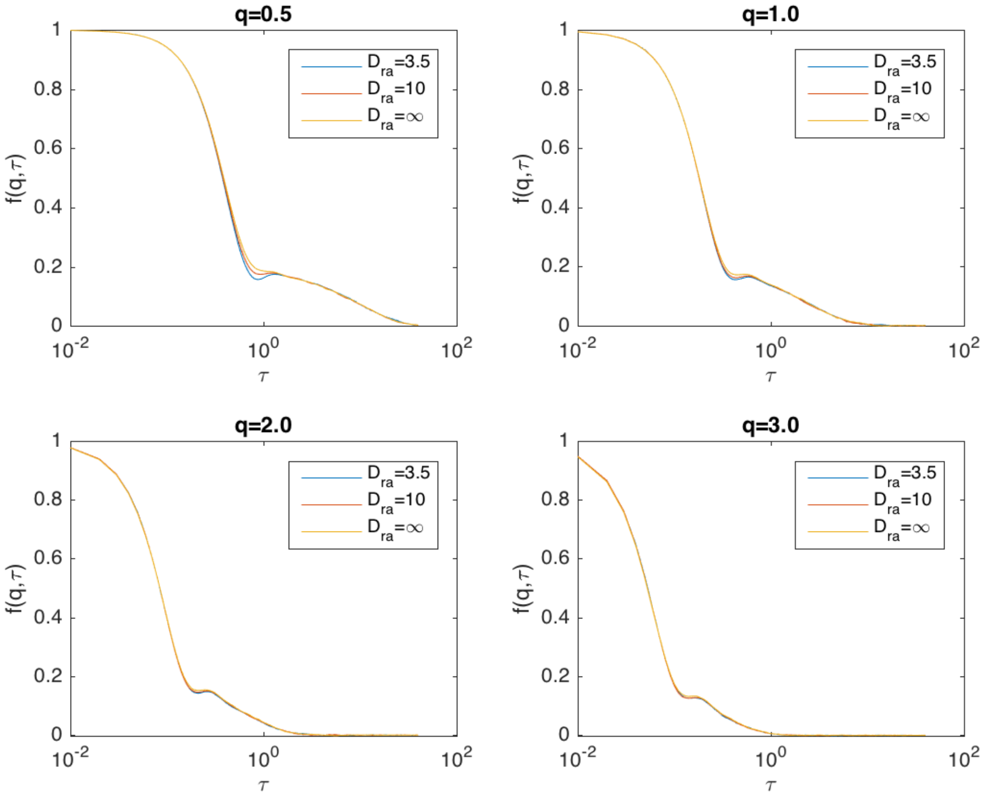


Figure 7.2: ISF $f(q, \tau)$ of RTDPTT particles with active rotational diffusion with respect to τ with different D_{ra}. The ISF was constructed from simulations of 10^5 particles, with the other parameters set to be $\alpha = 0.8$, $\bar{v} = 13$, $D = 0.4$, $Z = 15$ (or $\sigma = 3.25$), $\lambda = 1$, $\delta = 0.14$.

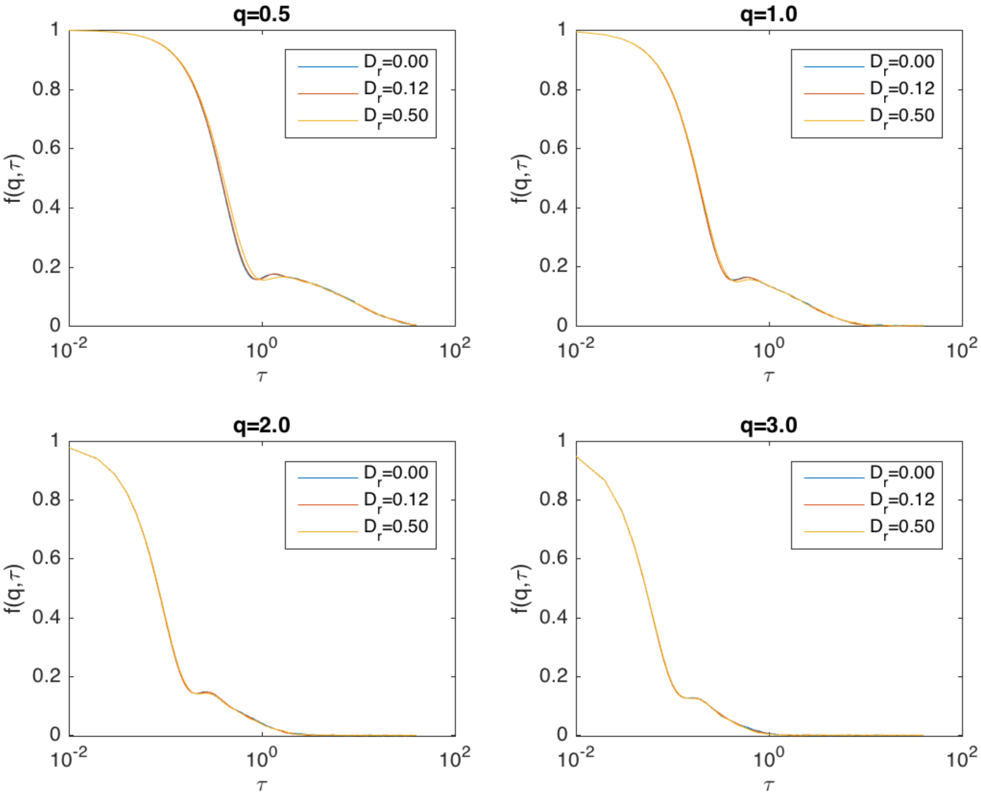


Figure 7.3: ISF $f(q,\tau)$ of RTDPTT particles with rotational diffusion and active rotational diffusion with respect to τ with different D_r. The ISF was constructed from simulations of 10^5 particles, with the other parameters set to be $\alpha = 0.8$, $\bar{v} = 13$, $D = 0.4$, $Z = 15$ (or $\sigma = 3.25$), $\lambda = 1$, $\delta = 0.14$, $D_{ra} = 3.5$.

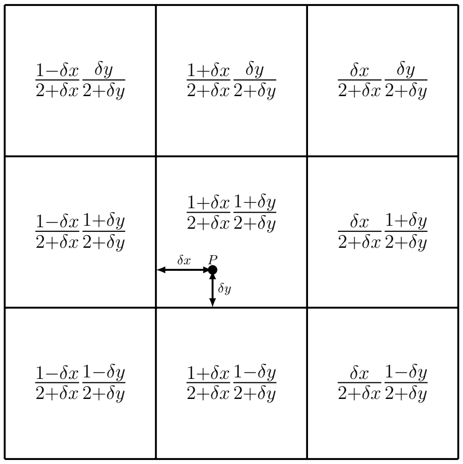
7.4 DDM simulation

We refer to the simulations that resemble the generation of experimental data as **DDM simulations**. In DDM simulations, a given number of particles move in a cuboid container with the specific size $L \times L \times D$. The boundary is taken to be periodic to keep the constant density, which means when a particle move across the boundary, it will appear from the opposite boundary.

The initial positions of the particles is uniformly sampled in the container. To generate a snapshot, at each time point, we take a region $-l/2 < x, y < l/2$, $-d/2 < z < d/2$ in the container, where $l < L$ is the size of the image, and $d < D$ is the depth of field. We project all the particles in the region to the x-y plane with a weight (contrast) function

$$C(z) = 1 - \frac{4z^2}{d^2} \,. \tag{7.25}$$

Then we smear the shadow of each particle into the pixel containing this particle and the eight neighbour pixels. The grey level value of the pixel is determined by the position of the particle inside the pixel. Denoting the shortest distance (divided by the pixel size) between the particle position and the grids to be $(\delta x, \delta y)$ (Figure 7.4), where without loss of generality we assume $\delta x, \delta y < 0.5$, the weight of the grey level value of each pixel is shown in Figure 7.4 [46]. Then we add the contribution of all the particles linearly to get a snapshot, as Figure 7.5. We can do DDM on these snapshots to test the algorithm.



$\frac{1-\delta x}{2+\delta x}\frac{\delta y}{2+\delta y}$	$\frac{1+\delta x}{2+\delta x}\frac{\delta y}{2+\delta y}$	$\frac{\delta x}{2+\delta x}\frac{\delta y}{2+\delta y}$
$\frac{1-\delta x}{2+\delta x}\frac{1+\delta y}{2+\delta y}$	$\frac{1+\delta x}{2+\delta x}\frac{1+\delta y}{2+\delta y}$	$\frac{\delta x}{2+\delta x}\frac{1+\delta y}{2+\delta y}$
$\frac{1-\delta x}{2+\delta x}\frac{1-\delta y}{2+\delta y}$	$\frac{1+\delta x}{2+\delta x}\frac{1-\delta y}{2+\delta y}$	$\frac{\delta x}{2+\delta x}\frac{1-\delta y}{2+\delta y}$

Figure 7.4: Stretch of how we smear a particle into 9 pixels. The expression in each pixel denotes the weight of the grey level value. The weights sum up to 1.

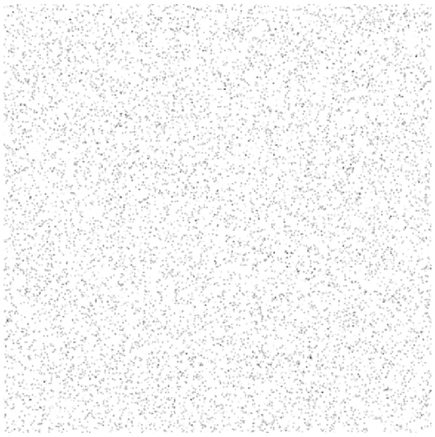


Figure 7.5: An example of snapshots of DDM simulation.

7.5 Run-and-tumble-and-diffuse particles with $P(v)$ (RTDP particles)

We first fitted the constructed ISF of run-and-tumble-and-diffuse particles with $P(v)$, giving the good initial guesses of α and $\bar{v}$ (if not, the fitting iterations may be trapped by another local minimum). The noise in the constructed ISF is low enough that the fitting gave good estimations for the data of $q \lesssim 1.2\mu\text{m}^{-1}$ (Figure 7.6). For the large q, the ISF becomes insensitive to λ, and the fitting began to deviate from the exact value.

Fitting the data from DDM simulation of RTDP particles with the RTDP model also requires good estimations of α and $\bar{v}$. We can do a fitting with the RDP model firstly. The result is shown in Figure 7.7. The fitting can give good estimations of α, $\bar{v}$ and D, but a bad estimation of Z. As indicated in Section 6.3.4, it is because the tumbles don't affect the ballistic part and the diffusive part of ISF. Then we can use these result to be the initial guesses of RTDP fitting.

Directly fitting the DDM simulation data with the RTDP model can not give all the parameters. As shown in Figure 7.8, the fitting cannot find a precise λ due to its insensitivity and the mixing of effects with σ. However, we are able to get good estimations of α, $\bar{v}$, σ (or Z), and D in a reasonable range of q. So we can do a second fitting, with all of these parameters fixed, to get a good estimation of λ. As shown in Figure 7.9. Figure A.1 shows the goodness of the fitting.

In the simulations, we noted that the goodness of the fitted λ will largely depend on the error in the estimation of $\bar{v}$, where a small change in $\bar{v}$ can give rise to a

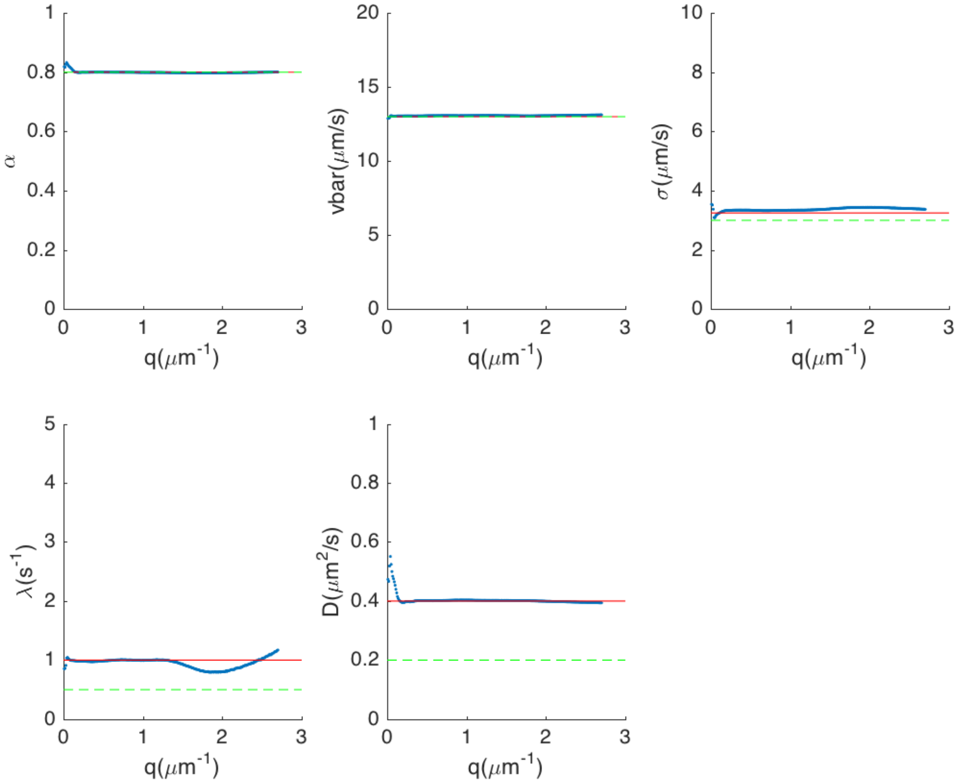


Figure 7.6: Fitting result of constructed ISF of RTDP particles with RTDP model, without fixing any parameters. The initial estimations of α and $\bar{v}$ were chosen to be their exact values. The red line denotes the settings in the simulation, which are $\alpha = 0.8,\ \bar{v} = 13\mu\text{m/s},\ Z = 15,\ \lambda = 1\text{s}^{-1},\ D = 0.4\mu\text{m}^2/\text{s}$. The green dashed lines denotes the initial guesses in fitting. The ISF data were constructed from 10^5 particles.

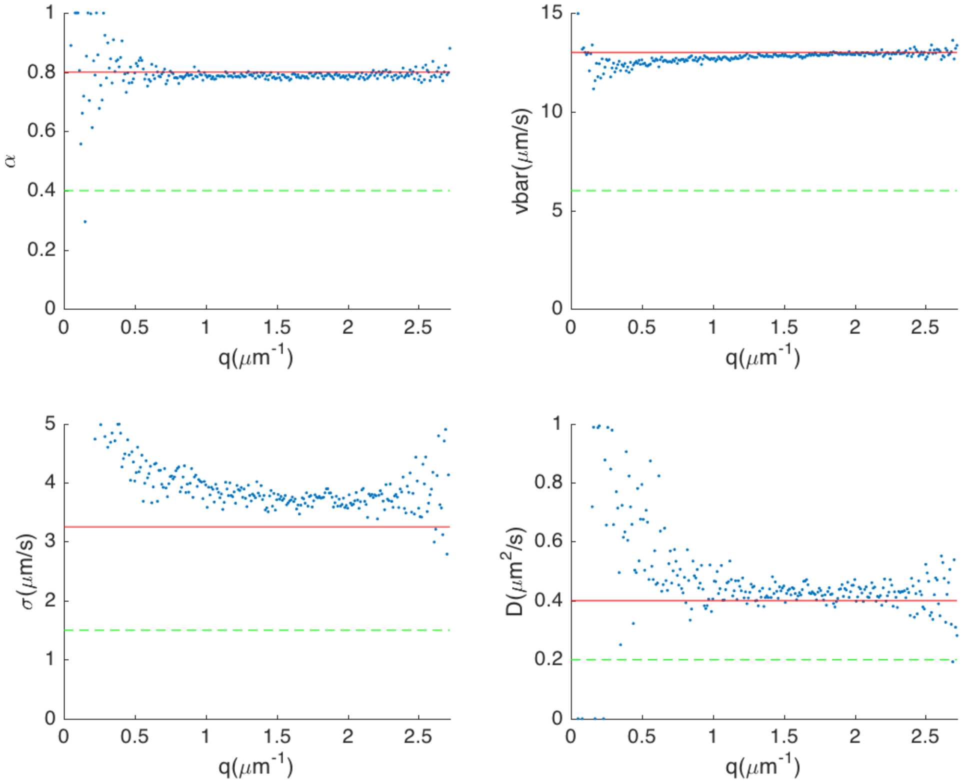


Figure 7.7: Fitting result of DDM simulation of RTDP particles with RDP model. The red line denotes the settings in the simulation, which are $\alpha = 0.8,\ \bar{v} = 13\mu\text{m/s},\ Z = 15,\ D = 0.4\mu\text{m}^2/\text{s}$. The green dashed lines denotes the initial guesses in fitting. The simulation was done with the particle density $5 \times 10^{-4}\mu\text{m}^{-3}$, the pixel size of 1.625μm, the image size of 512×512 pixels, and the depth of field of 40μm. Averaging the data in $q < 1.934\mu\text{m}^{-1}$ (with the extraordinary points excluded), we can get $\alpha = 0.787 \pm 0.007$, $\bar{v} = 12.7 \pm 0.2\mu\text{m/s}$, $Z = 9 \pm 3$, $D = 0.44 \pm 0.05\mu\text{m}^2/\text{s}$.

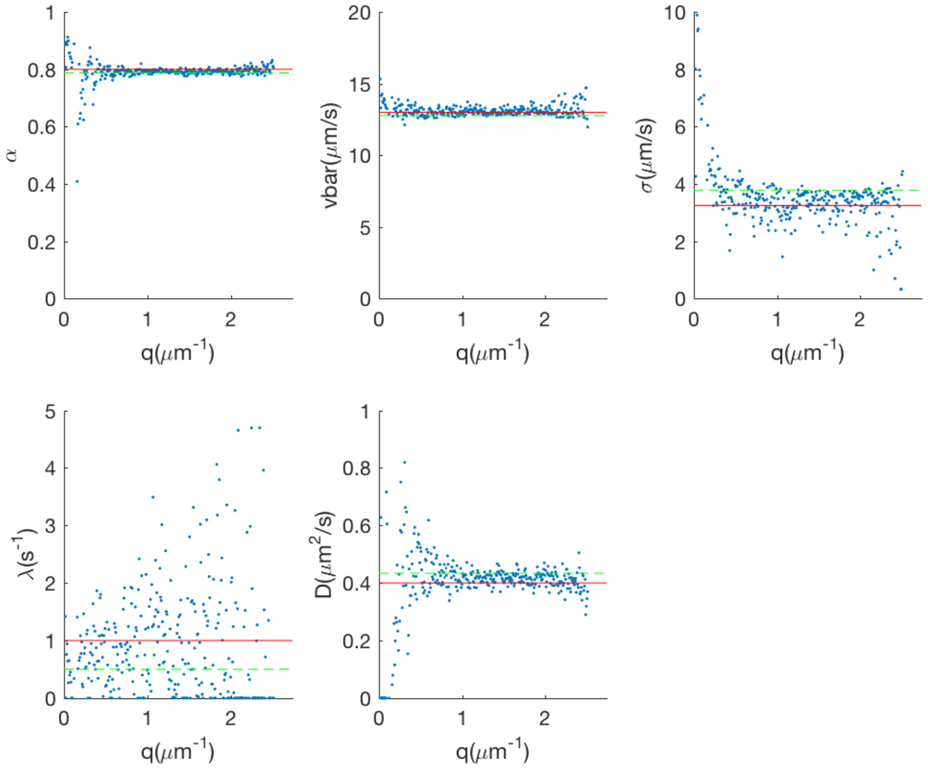


Figure 7.8: Fitting result of DDM simulation (same with Figure 7.7) of RTDP particles with RTDP model, without fixing any parameters. The red line denotes the settings in the simulation, which are $\alpha = 0.8$, $\bar{v} = 13\mu\text{m/s}$, $Z = 15$, $\lambda = 1\text{s}^{-1}$, $D = 0.4\mu\text{m}^2/\text{s}$. The green dashed lines denotes the initial guesses in fitting. The simulation was done with the particle density of $5 \times 10^{-4}\mu\text{m}^{-3}$, the pixel size of 1.625μm, the image size of 512 × 512 pixels, and the depth of field of 40μm. Averaging the data in $q < 1.934\mu\text{m}^{-1}$ (with the extraordinary points excluded), we can get $\alpha = 0.792 \pm 0.006$, $\bar{v} = 13.1 \pm 0.2\mu\text{m/s}$, $\sigma = 3.3 \pm 0.4\mu\text{m/s}$, $\lambda = 0.9 \pm 0.6\text{s}^{-1}$, $D = 0.42 \pm 0.02\mu\text{m}^2/\text{s}$.

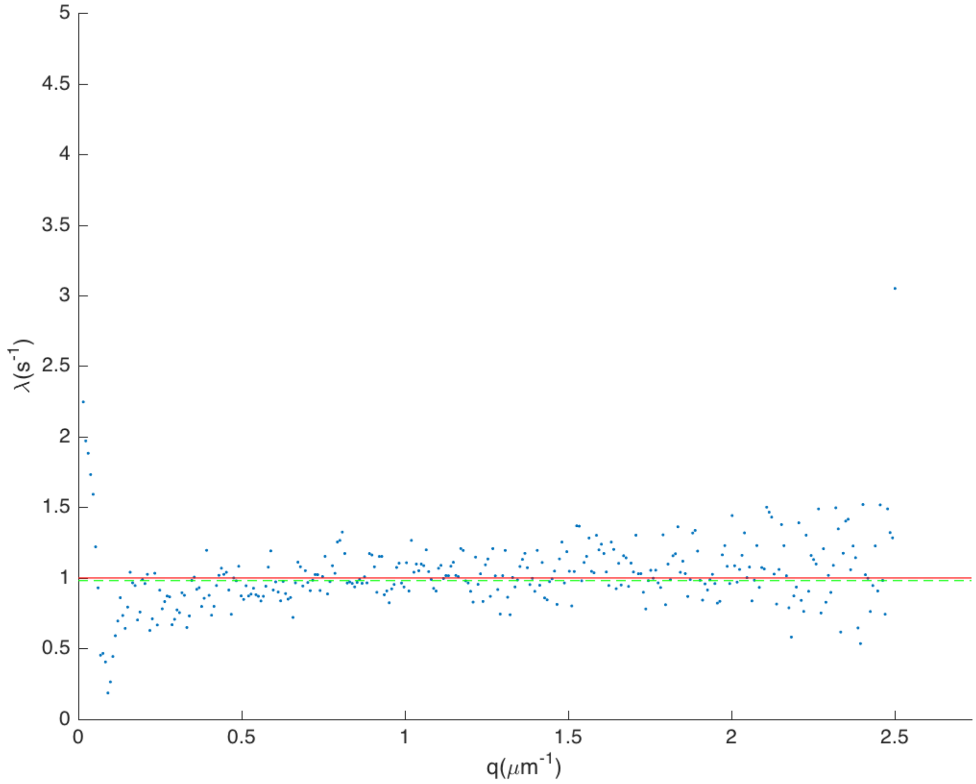


Figure 7.9: Fitting result of same DDM simulation with Figure 7.8 of RTDP particles with RTDP model, with fixed $\alpha = 0.792$, $\bar{v} = 13.1\mu\text{m/s}$, $\sigma = 3.3\mu\text{m/s}$, $D = 0.42\mu\text{m}^2/\text{s}$. The red line denotes the settings in the simulation, which is $\lambda = 1\text{s}^{-1}$. The green dashed lines denotes the initial guesses in fitting. Averaging the data in $0.5\mu\text{m}^{-1} < q < 1.5\mu\text{m}^{-1}$ (with the extraordinary points excluded), we can get $\lambda = 1.1 \pm 0.1\text{s}^{-1}$.

$\lambda(q)$ dependency. But the flatness of $\lambda(q)$ dependence also enable us to correct the small error in the estimation of $\bar{v}$, as shown in Figure 7.10.

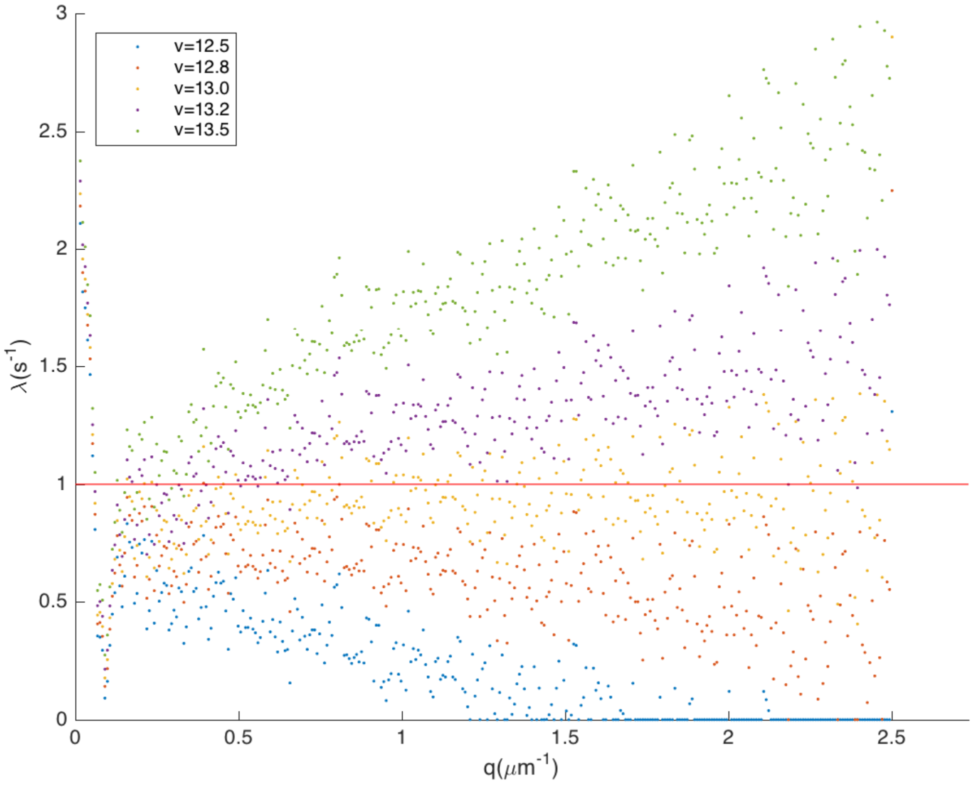


Figure 7.10: Fitting result of same DDM simulation with Figure 7.7 of RTDP particles with RTDP model, with fixed $\alpha = 0.792$, $\sigma = 3.29$µm/s, $D = 0.42$µm^2/s. We tuned $\bar{v}$ (in the unit of µm/s) around its precise value, to show the influence of the error in v to the fitting of λ. The red line denotes the λ in the simulation, which is 1s^{-1}.

The fitting results of the DDM simulations with different λ suggested the algorithm is robust. We also found that a larger pixel size may give a better estimation of v and thus λ, as Figure A.2, A.3, A.4, A.5.

7.6 Run-and-tumble-and-diffuse particles with $P(v)$ and $P(\lambda)$

Then instead of setting the same tumbling rate λ for all the particles, we simulated a group of particles whose λ satisfy a Schultz distribution

$$P(\lambda;\bar{\lambda},Z_\lambda)=\frac{\lambda^{Z_\lambda}}{\Gamma(Z_\lambda+1)}\left(\frac{Z_\lambda+1}{\bar{\lambda}}\right)^{Z_\lambda+1}e^{-(Z_\lambda+1)\lambda/\bar{\lambda}}\,, \tag{7.26}$$

where $\bar{\lambda}$ is the average of λ, the standard deviation is $\sigma_\lambda=\bar{\lambda}/\sqrt{Z_\lambda+1}$, and $\Gamma(x)$ is the gamma function. As shown in Figure 7.11, with the goodness of the fitting shown in A.6, we found that the distribution has little influence to ISF, making the fitting find the mean value of λ. We also tested the situation that λ satisfies a more complicated distribution, like a double-peak Schultz distribution $0.5P(\lambda;1,8)+0.5P(\lambda;5,15)$, but the conclusion is the same, as Figure A.7 A.8.

This may due to the insensitivity of the ISF to λ. If we expand $f(q,s;\lambda)=p_0(q,s)+p_1(q,s)\lambda+p_2(q,s)\lambda^2+O(\lambda^3)$ in terms of λ, usually for the ordinary values of parameters, we have $p_0\gg p_1\gg p_2$, where ”$\gg$” often means two orders of magnitude larger. This not only means the ISF is not sensitive to λ, but also indicates that the linear approximation of $f(q,s;\lambda)$ is already a good one, making the shape of $P(\lambda)$ has little contribution to the ISF except for its mean value.

7.7 Run-and-tumble-and-diffuse particles with $P(v)$ and tumbling duration (RTDPTT particles)

Firstly, we constructed the ISF of run-and-tumble-and-diffuse particles with $P(v)$ and tumbling duration from a simulation, and fitted it with the RTDPTT model, as shown in Figure 7.12 and 7.13. Comparing the two fittings with different initial guesses, we should notice the fitting now depends on the initial guesses of the parameters. Later we will see, besides the initial guesses of α and $\bar{v}$, the fitting result is largely dependent on λ.

Then we can do the same protocol with the RTDP particles. The RDP fitting result is shown in Figure 7.14. Compared with Figure 7.7, as an effect of the non-zero tumbling duration, $\bar{v}$ gets underestimated, and a q-dependence of α and $\bar{v}$ becomes obvious. The discrepancy of fitted $\bar{v}$ and the exact value in the low q limit is roughly of a factor $1/\sqrt{1+\delta\lambda}$, which corresponds to the discussion of (3.32) in Section 3.1.2. If we do the RTDP fitting, we will find a strong $\lambda(q)$ dependence in the second fitting, as well as some weak q-dependence of α, $\bar{v}$, and σ. The $\lambda(q)$ dependence prevents us from finding a good estimation of λ. If

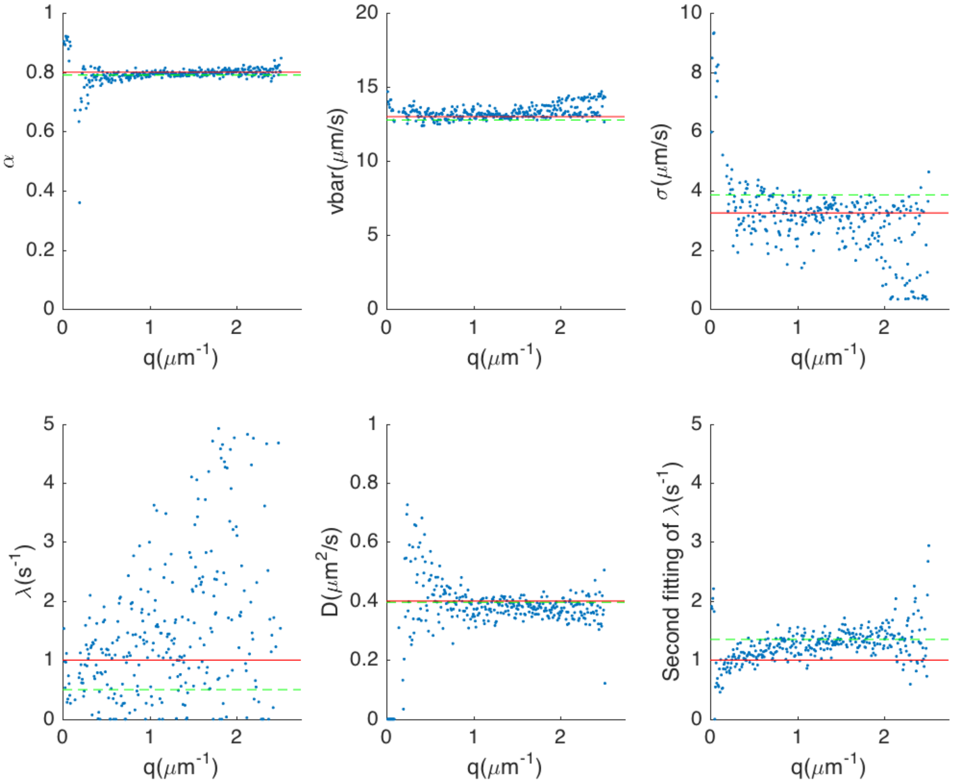


Figure 7.11: Fitting result of DDM simulation of RTDP (with $P(\lambda)$) particles with RTDP model, without fixing any parameters. The red line denotes the settings in the simulation, which are $\alpha = 0.8,\ \bar{v} = 13\mu\text{m/s},\ Z = 15,\ D = 0.4\mu\text{m}^2/\text{s}$. The green dashed lines denotes the initial guesses in fitting. The simulation was done with $\bar{\lambda} = 1\text{s}^{-1},\ Z_\lambda = 8$, the particle density of $5 \times 10^{-4}\mu\text{m}^{-3}$, the pixel size of 1.625μm, the image size of 512×512 pixels, and the depth of field of 40μm. Averaging the data in $q < 1.934\mu\text{m}^{-1}$ (with the extraordinary points excluded), we can get $\alpha = 0.797 \pm 0.008$, $\bar{v} = 13.2 \pm 0.3\mu\text{m/s}$, $\sigma = 3.2 \pm 0.6\mu\text{m/s}$, $\lambda = 1.2 \pm 0.9\text{s}^{-1}$ $D = 0.38 \pm 0.02\mu\text{m}^2/\text{s}$. Then a second fitting was done with fixed $\alpha = 0.807$, $\bar{v} = 13.1\mu\text{m/s}$, $\sigma = 3.6\mu\text{m/s}$, $D = 0.45\mu\text{m}^2/\text{s}$. Averaging the data in $0.5\mu\text{m}^{-1} < q < 1.5\mu\text{m}^{-1}$ (with the extraordinary points excluded), we can get $\lambda = 1.2 \pm 0.2\text{s}^{-1}$.

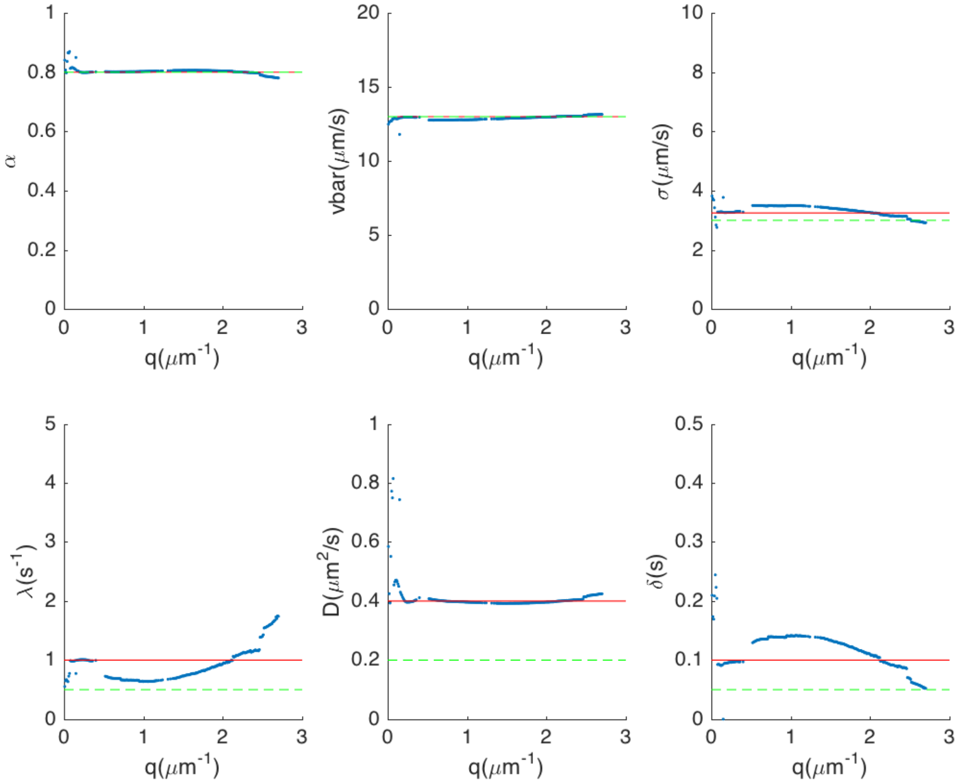


Figure 7.12: Fitting result of constructed ISF of RTDPTT particles with RTDPTT model, without fixing any parameters. The initial estimations of α and $\bar{v}$ were chosen to be their exact values. The red line denotes the settings in the simulation, which are $\alpha = 0.8,\ \bar{v} = 13\mu\text{m/s},\ Z = 15,\ \lambda = 1\text{s}^{-1},\ D = 0.4\mu\text{m}^2/\text{s},\ \delta = 0.1\text{s}$. The green dashed lines denotes the initial guesses in fitting. The ISF data were constructed from 10^5 particles.

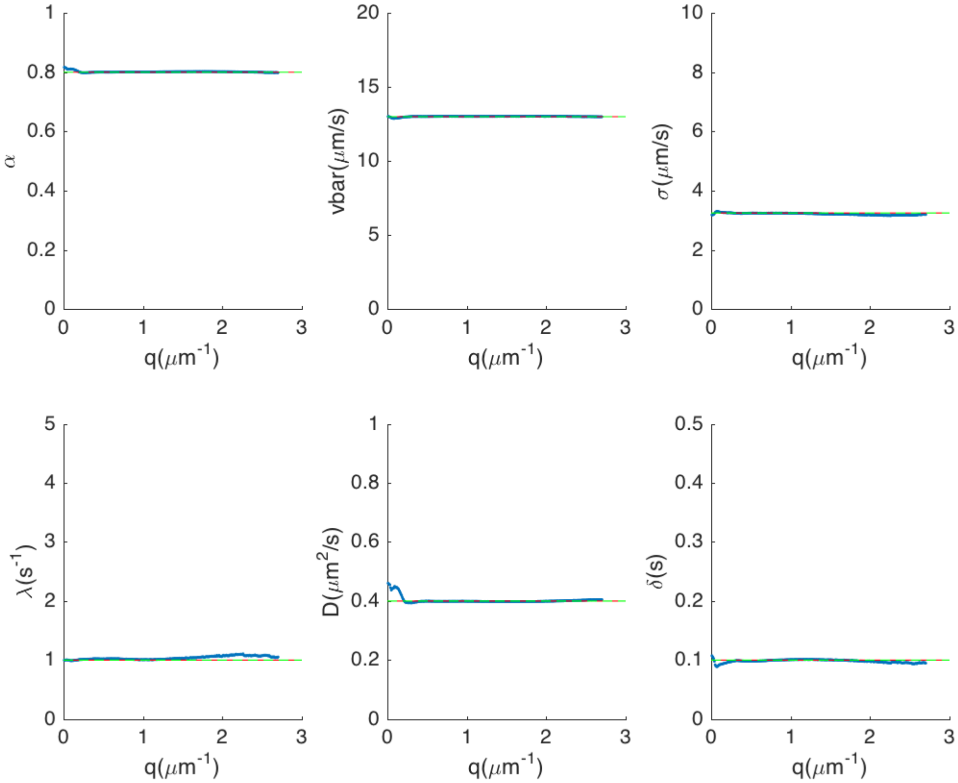


Figure 7.13: Fitting result of constructed ISF of RTDP particles with RTDP model, without fixing any parameters. The initial estimations of all paramters were chosen to be their exact values. The red line denotes the settings in the simulation, which are $\alpha = 0.8,\ \bar{v} = 13\mu\text{m/s},\ Z = 15,\ \lambda = 1\text{s}^{-1},\ D = 0.4\mu\text{m}^2/\text{s},\ \delta = 0.1\text{s}$. The green dashed lines denotes the initial guesses in fitting. The ISF data were constructed from 10^5 particles.

we do a image with a larger size (Figure A.9), the result doesn't change much. But we noticed $\lim_{q\to 0}\lambda(q)$ of the fitted $\lambda(q)$ seems to be close to the real value.

Then we can try the fitting with the RTDPTT model as shown in Figure 7.16 and 7.17. Now the q-dependence of α, $\bar{v}$, and σ disappears. In the first fitting the result depends on the initial guesses, and we cannot find definite λ and δ. In the second fitting, the result still depends on initial guesses. But if we have known the exact λ, we can find the δ from the second fitting. Noticing the difference between the slopes of $\delta(q)$ dependence with different initial guesses, we may use it to find a better estimation of λ.

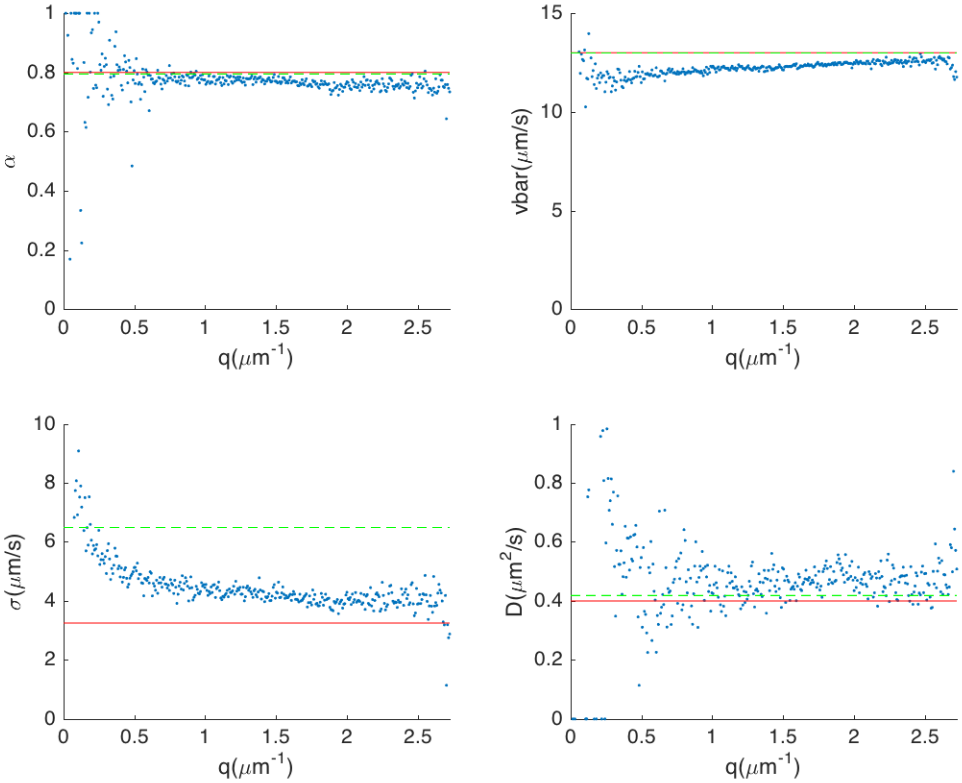


Figure 7.14: Fitting result of DDM simulation of RTDPTT particles with RDP model. The red line denotes the settings in the simulation, which are $\alpha = 0.8$, $\bar{v} = 13\mu\text{m/s}$, $\sigma = 3.25\mu\text{m/s}$, $\lambda = 1\text{s}^{-1}$, $D = 0.4\mu\text{m}^2/\text{s}$, $\delta = 0.1\text{s}$. The green dashed lines denotes the initial guesses in fitting. The simulation was done with the particle density of $5 \times 10^{-4}\mu\text{m}^{-3}$, the pixel size of $1.625\mu\text{m}$, the image size of 512×512 pixels, and the depth of field of $40\mu\text{m}$. Averaging the data in $q < 1.934\mu\text{m}^{-1}$ (with the extraordinary points excluded), we can get $\alpha = 0.77 \pm 0.02$, $\bar{v} = 12.1 \pm 0.3\mu\text{m/s}$, $\sigma = 6 \pm 2\mu\text{m/s}$, $D = 0.46 \pm 0.04\mu\text{m}^2/\text{s}$.

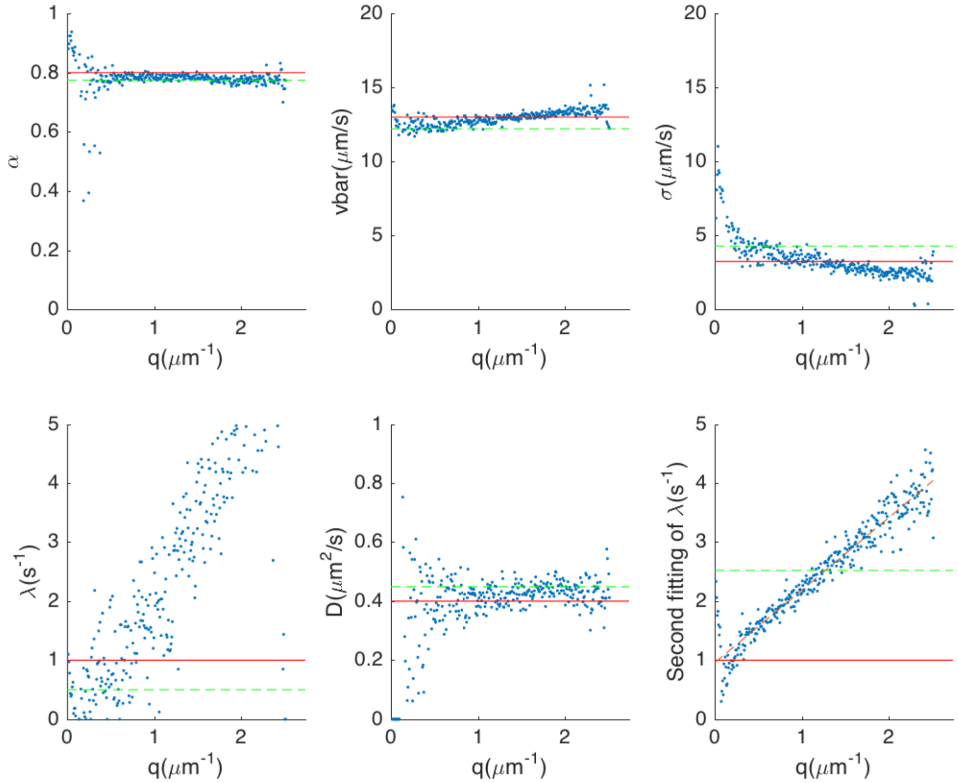


Figure 7.15: Fitting result of DDM simulation of RTDPTT particles with RTDP model. The red line denotes the settings in the simulation, which are $\alpha = 0.8$, $\bar{v} = 13\mu m/s$, $\sigma = 3.25\mu m/s$, $\lambda = 1s^{-1}$, $D = 0.4\mu m^2/s$, $\delta = 0.1s$. The green dashed lines denotes the initial guesses in fitting. The simulation was done with the particle density of $5 \times 10^{-4}\mu m^{-3}$, the pixel size of 1.625μm, the image size of 512×512 pixels, and the depth of field of 40μm. Averaging the data in $q < 1.934\mu m^{-1}$ (with the extraordinary points excluded), we can get $\alpha = 0.78 \pm 0.01$, $\bar{v} = 12.8 \pm 0.4\mu m/s$, $\sigma = 3.4 \pm 0.7\mu m/s$, $\lambda = 2.1 \pm 1.5s^{-1}$, $D = 0.43 \pm 0.03\mu m^2/s$. Then a second fitting was done with fixed $\alpha = 0.785$, $\bar{v} = 12.86\mu m/s$, $\sigma = 3.2\mu m/s$, $D = 0.42\mu m^2/s$. Now a q-dependence occurred. The red dash line shows a linear fitting which is $\lambda = 0.9507 + 1.231q$.

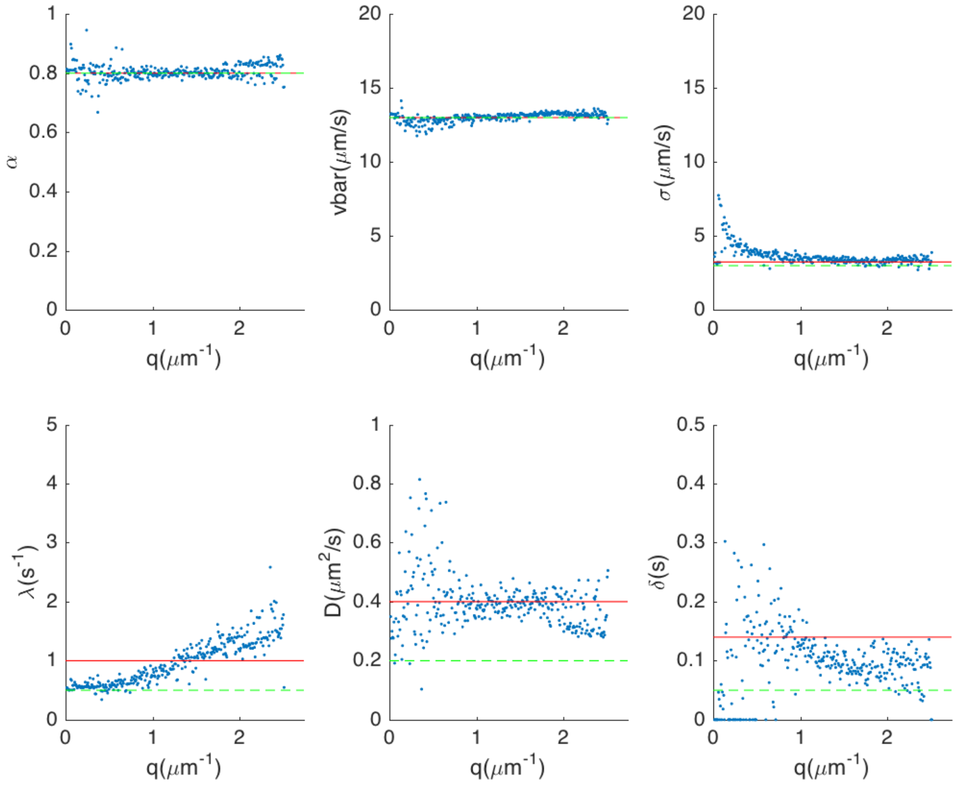


Figure 7.16: Fitting result of DDM simulation of RTDPTT particles with RTDPTT model, without fixing any parameters. The initial guesses were given only good α and $\bar{v}$. The red line denotes the settings in the simulation, which are $\alpha = 0.8,\ \bar{v} = 13\mu\text{m/s},\ \sigma = 3.25\mu\text{m/s},\ \lambda = 1\text{s}^{-1},\ D = 0.4\mu\text{m}^2/\text{s},\ \delta = 0.1\text{s}$. The green dashed lines denotes the initial guesses in fitting. The simulation was done with the particle density of $5 \times 10^{-4}\mu\text{m}^{-3}$, the pixel size of 1.625μm, the image size of 512×512 pixels, and the depth of field of 40μm. Averaging the data in $q < 1.934\mu\text{m}^{-1}$ (with the extraordinary points excluded), we can get $\alpha = 0.80 \pm 0.01$, $\bar{v} = 13.1 \pm 0.2\mu\text{m/s}$, $\sigma = 3.4 \pm 0.2\mu\text{m/s}$, $\lambda = 0.8 \pm 0.3\text{s}^{-1}$, $D = 0.39 \pm 0.03\mu\text{m}^2/\text{s}$, $\delta = 0.10 \pm 0.02\text{s}$.

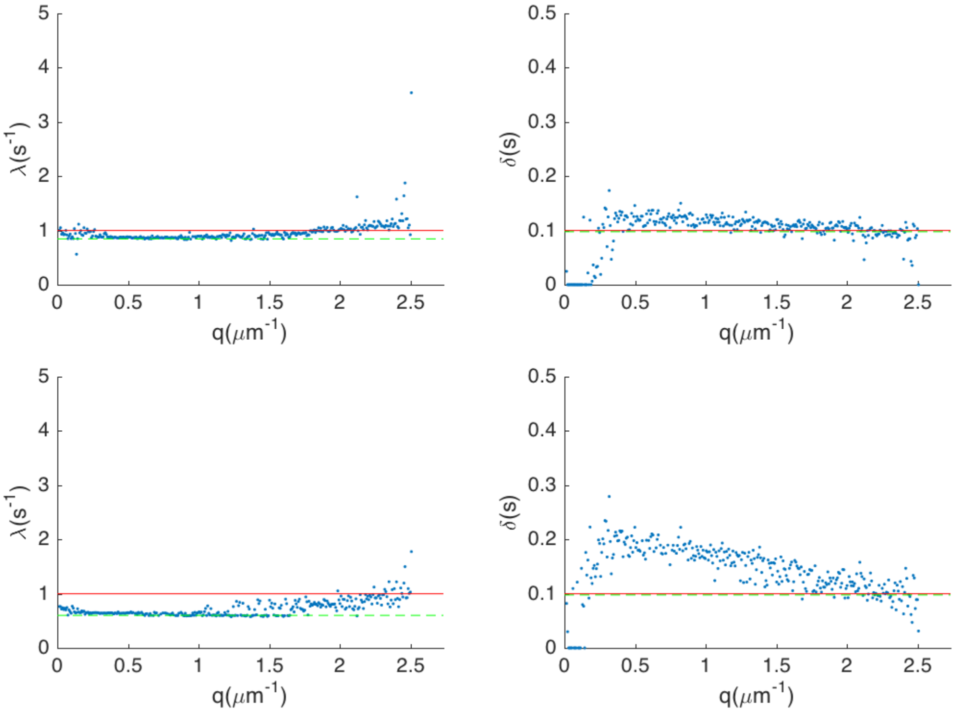


Figure 7.17: Two fitting results of same DDM simulation with Figure 7.16 of RTDPTT particles with RTDPTT model, both with fixed $\alpha = 0.80$, $\bar{v} = 13.1\mu\text{m/s}$, $\sigma = 3.4\mu\text{m/s}$, $D = 0.39\mu\text{m}^2/\text{s}$. The only difference for the upper two figures and lower two figures is that we took a different initial guess of λ in fitting. The red line denotes the settings in the simulation, which is $\lambda = 1\text{s}^{-1}$, $\delta = 0.1\text{s}$. The green dashed lines denotes the initial guesses in fitting. For the one with good initial guesses, averaging the data in $q < 1.934\mu\text{m}^{-1}$ (with the extraordinary points excluded), we can get $\lambda = 0.92 \pm 0.05\text{s}^{-1}$, $\delta = 0.11 \pm 0.01\text{s}$.

Chapter 8

DDM Experiment

8.1 Experimental requirements

The DDM requires a set of time-lapse images with the frame rate of the order of 10^2. From previous works and our experiments, 100fps is considered to be a suitable choice. From simulation results, we can still do the DDM even for 25fps with a little more exceptional points in the fitting results (data not shown). Lower frame rate may give sparser points in the first and the most important decay of ISF, which may be problematic. But a larger frame rate requires a better imaging system, as well as larger memory in pre-processing the data. For a reference, processing 4500 pieces of 512×512 16-bit images requires roughly 9GB memory.

From the simulation result, we should focus on the $q < 2\mu\text{m}^{-1}$ region in the Fourier space. Theoretically the upper limit of q satisfies

$$q_{\text{max}} = \frac{\sqrt{2}\pi}{\delta x}\,, \tag{8.1}$$

where δx is the pixel size of the image. The data near q_{max} will always have a bad statistics, so I would recommend choosing the pixel size to be

$$\delta x = \pi / q_{\text{max}} \sim 2\mu\text{m}\,. \tag{8.2}$$

This can be accomplished by a $14\mu\text{m} \times 14\mu\text{m}$ pixel size camera with $10\times$ or $4\times$ magnification, or $6.5\mu\text{m} \times 6.5\mu\text{m}$ pixel size with $2\times$ or $4\times$ magnification.

To reduce the influence of water flow and vibration of the equipment, I recommend to use a sealed chamber to present the sample, with at least $400\mu\text{m}$ depth to ensure the 3D motion of *E. coli*. For *E. coli*, if the chamber is airtight, the sample should be imaged within 30 minutes to avoid hypoxia.

8.2 AB1157 Wild type *E. coli*

This data is from Vincent Martinez for his published paper [34]. We tried to fit this sample with the RDP, RTDP, and RTDPTT models. The RDP fitting results are shown in Fig. 8.1. As we expected, the fitting gave a $\bar{v}(q)$ dependence. But we can still have an estimation for fitting with the RTDP model.

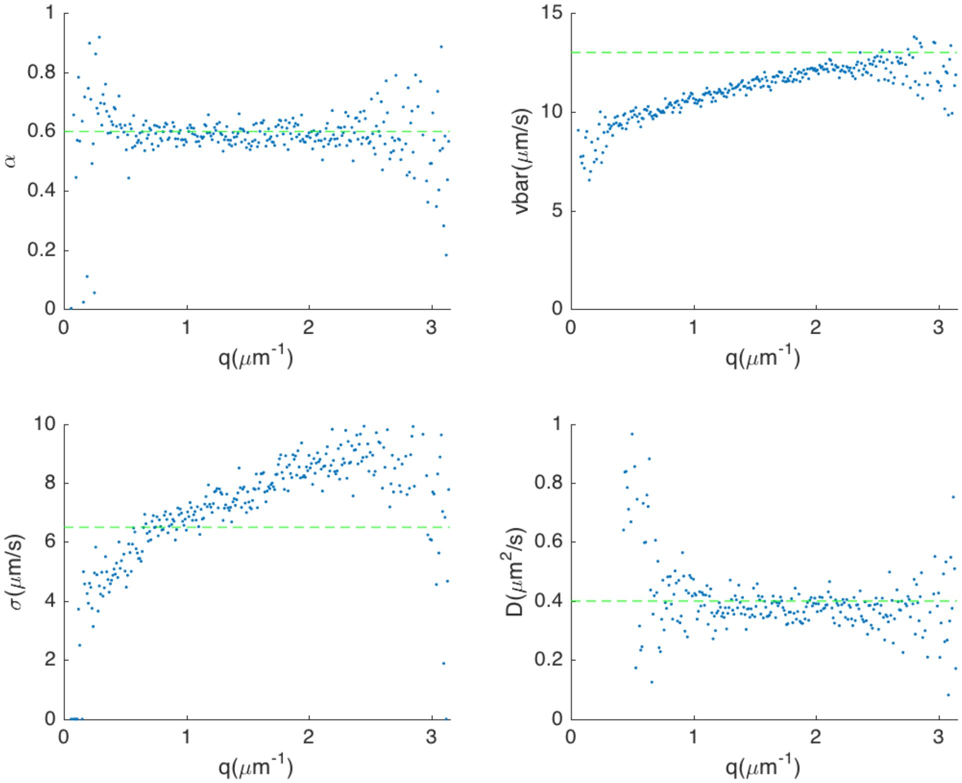


Figure 8.1: Fitting result of wild type *E. coli* movie with RDP model. The green dash line denotes the initial guesses in the fitting. The images were taken with the pixel size of 1.40449μm, the image size of 512×512 pixels. Averaging the data in $q < 2.0\mu\text{m}^{-1}$ (with the extraordinary points excluded), we can get $\alpha = 0.59 \pm 0.03$, $\bar{v} = 11.0 \pm 0.7\mu\text{m/s}$, $Z = 1.3 \pm 0.4$, $D = 0.38 \pm 0.06\mu\text{m}^2/\text{s}$.

The fitting results with the RTDP model is shown in Figure 8.2, with the goodness of fitting shown in Figure A.10. It is noticed that in RTDP fitting, a non-linear $\lambda(q)$ dependence appears, but the shape is different from the prediction from the simulation. Similar with the simulation data, the RDP model and the RTDP model gave a similar α and a little higher $\bar{v}$.

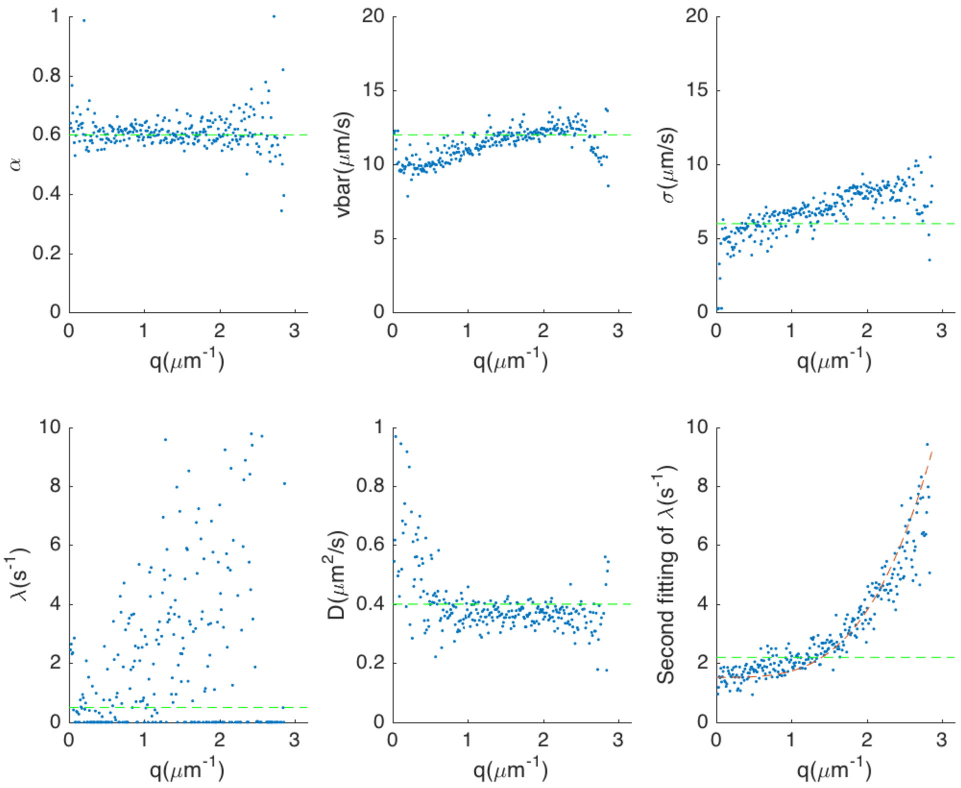


Figure 8.2: Fitting result of the same data with Fig. 8.1 of wild type *E. coli* movie with RTDP model. The green dash line denotes the initial guesses in the fitting. The images were taken with the pixel size of 1.40449μm, the image size of 512×512 pixels. Averaging the data in $q < 2.0\mu\text{m}^{-1}$ (with the extraordinary points excluded), we can get $\alpha = 0.61 \pm 0.03$, $\bar{v} = 11.4 \pm 0.8\mu$m/s, $\sigma = 6.8 \pm 1.0\mu$m/s, $\lambda = 2 \pm 2\text{s}^{-1}$, $D = 0.37 \pm 0.03\mu\text{m}^2/\text{s}$. Then a second fitting was done with fixed $\alpha = 0.61$, $\bar{v} = 11.4\mu$m/s, $\sigma = 6.8\mu$m/s, $D = 0.37\mu\text{m}^2/\text{s}$. $\lambda(q)$ dependence appeared, and we fitted this with a power function $\lambda = p_0 + p_1 q^b$. The red line indicates the fitting result, and the corresponding parameters are $p_0 = 1.5224$, $p_1 = 0.21501$, $b = 2.2974$.

The fitting results with the RTDPTT model are shown in Figure 8.3 and 8.4, with the goodness of fitting in Figure A.11. $\bar{v}$ was higher than fitting results with the RTDP model, and didn't exhibit a q-dependence as expected. We are not sure if the low q limit in the second fitting indicates the real tumbling rate, but it is unexpected that this value was less than that from the RTDPTT model. Unfortunately we can't find a definite tumbling duration.

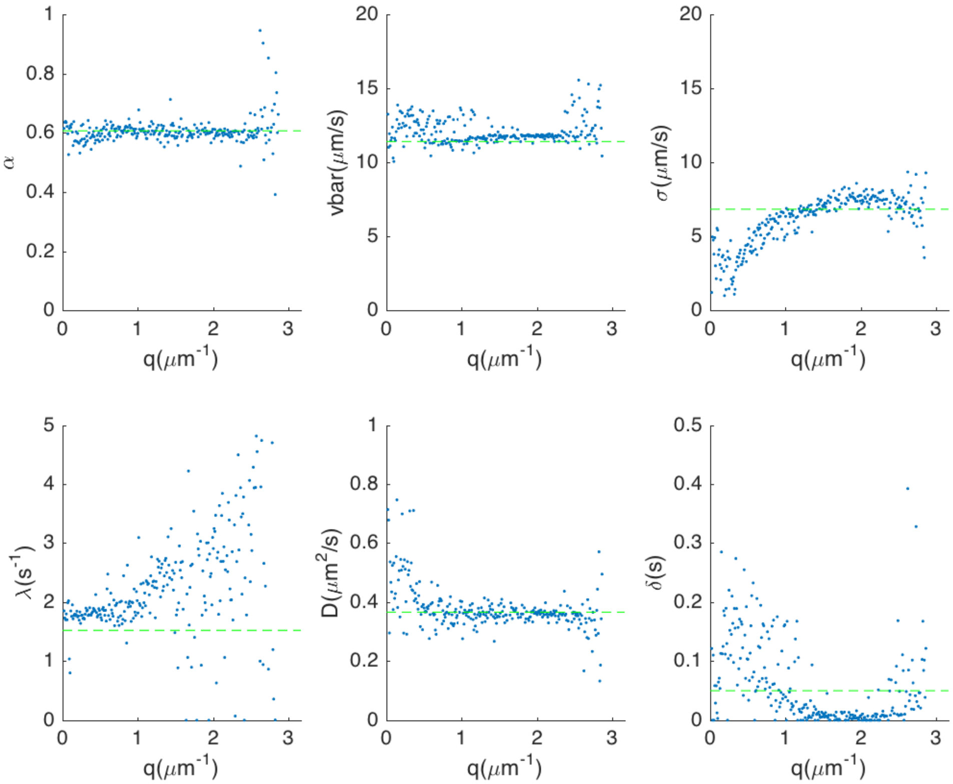


Figure 8.3: Fitting result of the same data with Fig. 8.1 of wild type *E. coli* movie with RTDPTT model, without fixing any parameters. The green dash line denotes the initial guesses in the fitting. The images were taken with the pixel size of 1.40449μm, the image size of 512 × 512 pixels. Averaging the data in $q < 2.0\mu\text{m}^{-1}$ (with the extraordinary points excluded), we can get $\alpha = 0.60 \pm 0.02$, $\bar{v} = 11.8 \pm 0.6\mu\text{m/s}$, $\sigma = 6 \pm 1\mu\text{m/s}$, $\lambda = 2.1 \pm 0.5\text{s}^{-1}$, $D = 0.36 \pm 0.02\mu\text{m}^2/\text{s}$, $\delta = 0.009 \pm 0.009\text{s}$.

8.3 AB1157 ΔCheY *E. coli*

This data is from Vincent Martinez for his published paper [34]. We tried to fit this sample with the RDP, RTDP, and RTDPTT models. The RDP fitting results are shown in Fig. 8.5. Same with the wild-type *E. coli*, the fitting gave a

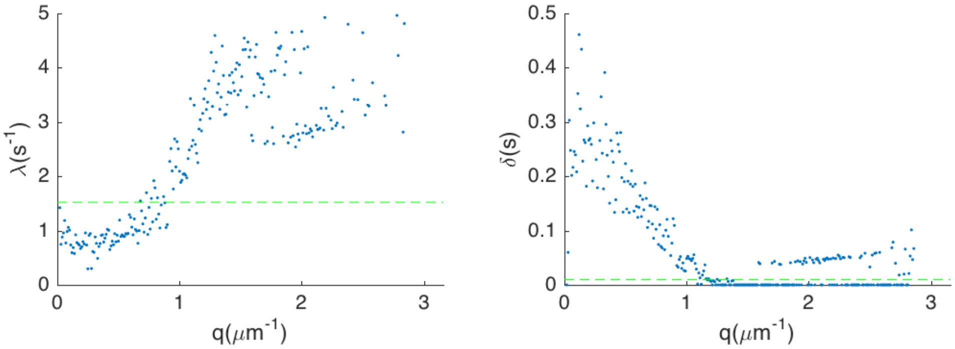


Figure 8.4: Dots are fitting result of the same data with Fig. 8.1 of wild type *E. coli* movie with RTDPTT model, with fixed $\alpha = 0.60$, $\bar{v} = 11.8\mu$m/s, $\sigma = 6.4\mu$m/s, $D = 0.36\mu$m^2/s. The green dash line denotes the initial guesses in the fitting. Averaging the data in $q < 2.0\mu$m^{-1} (witht the extraordinary points excluded), we get $\lambda = 2.3 \pm 1.4$s^{-1}, $\delta = 0.012 \pm 0.018$s.

$\bar{v}(q)$ dependence. But we can still have an estimation for the fitting with RTDP model.

The fitting results with the RTDP model is shown in Figure 8.6, with the goodness of fitting shown in Figure A.12. It is noticed that in the RTDP fitting, a non-linear $\lambda(q)$ dependence appears. The RDP model and the RTDP model gives similar α and $\bar{v}$ as expected. But the q-dependence of $\bar{v}$ was different from the simulation. It may due to the hydrodynamic flow in the sample, which is problematic for DDM.

The fitting results with the RTDPTT model are shown in Figure 8.7 and 8.8, with the goodness of fitting in Figure A.13. The $\bar{v}(q)$ dependence still appeared for some unknown reason. The tumbling rate from fitting seems to be zero, which is expected. As little tumbles happened, we can't expect to find a definite tumbling duration.

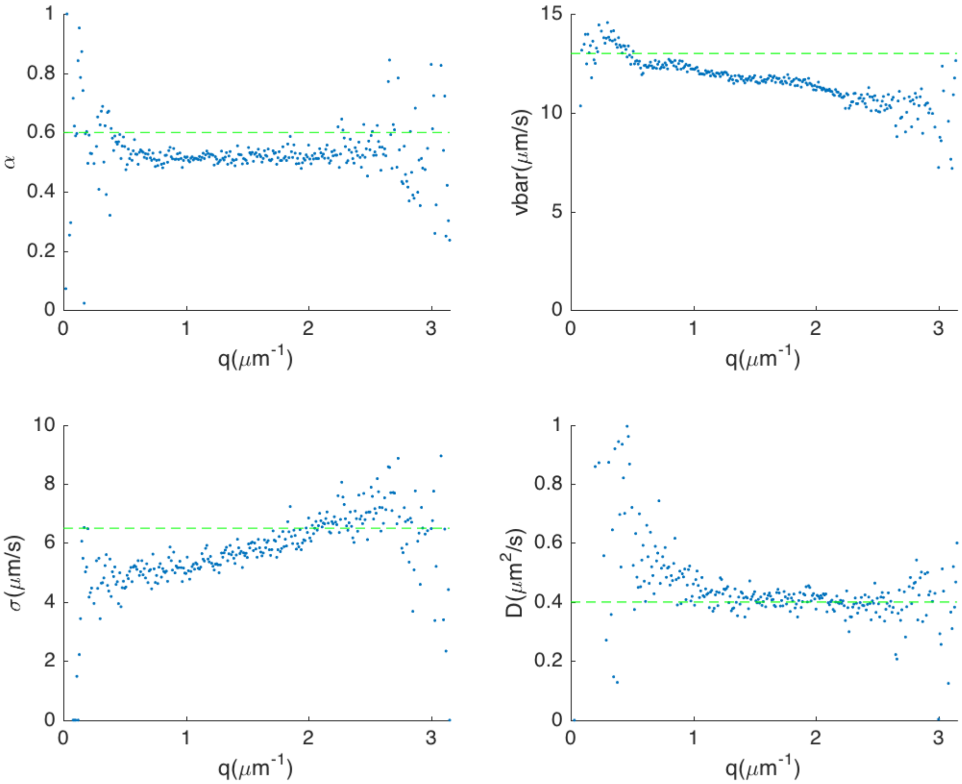


Figure 8.5: Fitting result of ΔCheY *E. coli* movie with RDP model. The green dash line denotes the initial guesses in the fitting. The images were taken with the pixel size of 1.40449μm, the image size of 512×512 pixels. Averaging the data in $q < 2.0\mu\text{m}^{-1}$ (with the extraordinary points excluded), we can get $\alpha = 0.51 \pm 0.02$, $\bar{v} = 12.0 \pm 0.4\mu\text{m/s}$, $Z = 3.9 \pm 1.1$, $D = 0.44 \pm 0.05\mu\text{m}^2/\text{s}$.

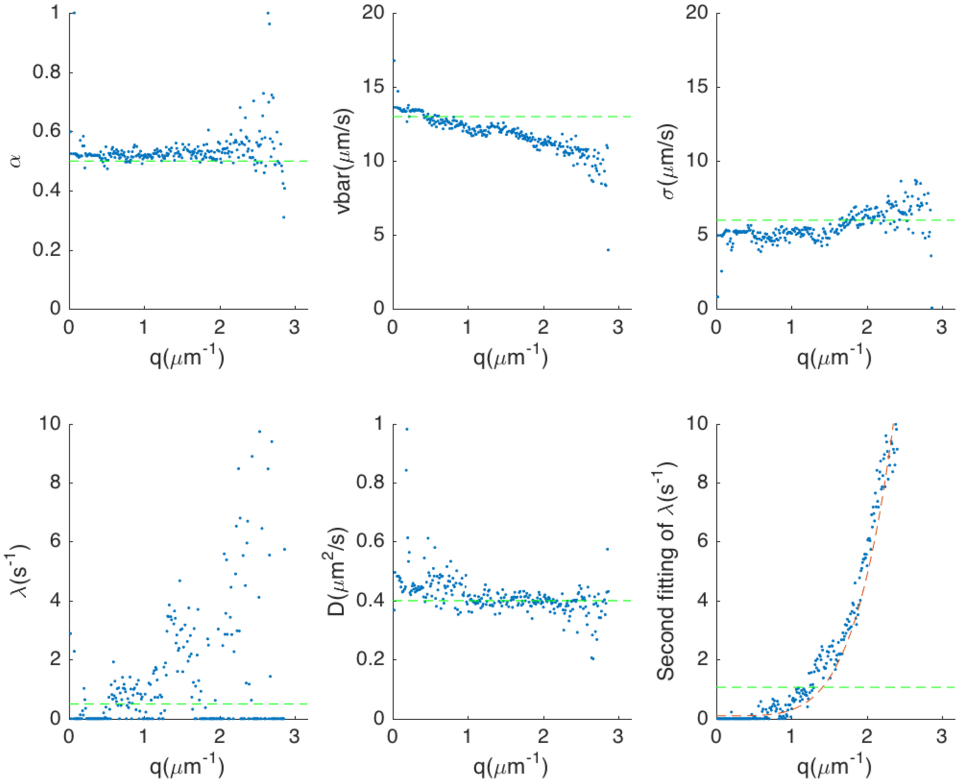


Figure 8.6: Fitting result of the same data with Fig. 8.5 of ΔCheY *E. coli* movie with RTDP model. The green dash line denotes the initial guesses in the fitting. The images were taken with the pixel size of 1.40449μm, the image size of 512×512 pixels. Averaging the data in $q < 2.0\mu\text{m}^{-1}$ (with the extraordinary points excluded), we can get $\alpha = 0.52 \pm 0.01$, $\bar{v} = 12.1 \pm 0.5\mu$m/s, $\sigma = 5.3 \pm 0.6\mu$m/s, $\lambda = 1 \pm 1\text{s}^{-1}$, $D = 0.40 \pm 0.02\mu\text{m}^2$/s. Then a second fitting was done with fixed $\alpha = 0.52$, $\bar{v} = 12.1\mu$m/s, $\sigma = 5.3\mu$m/s, $D = 0.40\mu\text{m}^2$/s. $\lambda(q)$ dependence appeared, and we fitted it with a power function $\lambda = p_0 + p_1 q^b$. The red line indicates the fitting result, and the corresponding parameters are $p_0 = 0.10172$, $p_1 = 0.20274$, $b = 4.5532$.

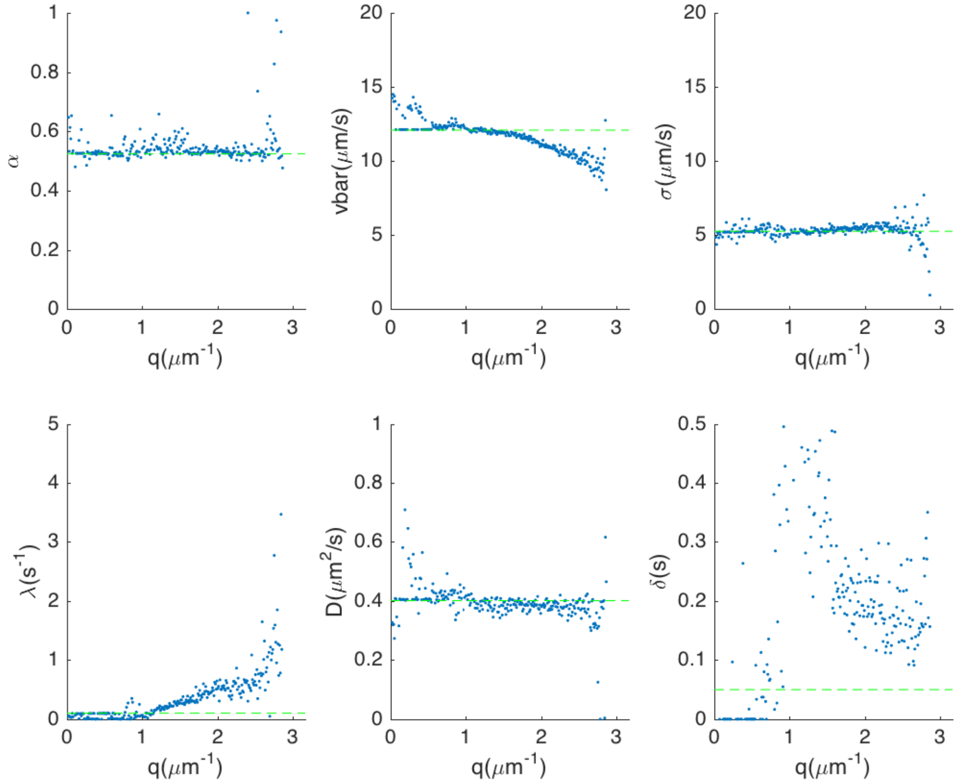


Figure 8.7: Fitting result of the same data with Fig. 8.5 of ΔCheY *E. coli* movie with RTDPTT model, without fixing any parameters. The green dash line denotes the initial guesses in the fitting. The images were taken with the pixel size of 1.40449μm, the image size of 512×512 pixels. Averaging the data in $q < 2.0\mu\text{m}^{-1}$ (with the extraordinary points excluded), we can get $\alpha = 0.54 \pm 0.02$, $\bar{v} = 12.0 \pm 0.4\mu$m/s, $\sigma = 5.3 \pm 0.2\mu$m/s, $\lambda = 0.2 \pm 0.2\text{s}^{-1}$, $D = 0.39 \pm 0.02\mu\text{m}^2$/s, $\delta = 0.3 \pm 0.2$s.

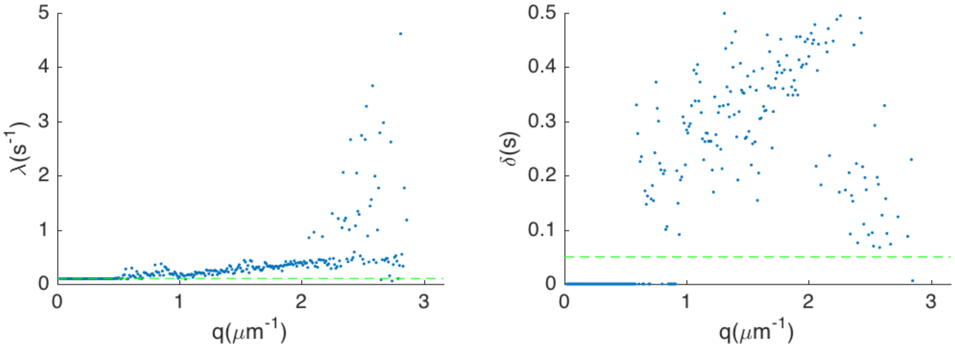


Figure 8.8: Dots are fitting result of the same data with Fig. 8.5 of ΔCheY *E. coli* movie with RTDPTT model, with fixed $\alpha = 0.54$, $\bar{v} = 12.0\mu\text{m/s}$, $\sigma = 5.3\mu\text{m/s}$, $D = 0.39\mu\text{m}^2/\text{s}$. The green dash line denotes the initial guesses in the fitting. The fitting result was not better than that in Fig. 8.7. Averaging the data in $q < 2.0\mu\text{m}^{-1}$ (with the extraordinary points excluded), we get $\lambda = 0.2 \pm 0.1\text{s}^{-1}$, $\delta = 0.34 \pm 0.08\text{s}$.

Chapter 9

Discussion

9.1 Summery and discussion

In Chapter 2 and 3, I showed how to establish a mathematical model from detailed microscopic behaviour of the run-and-tumble particles, based on the master equation. And I showed some conclusions of the model, such as the diffusivity and the intermediate scattering function. In particular, I added a finite tumbling duration into the simple model, and get the intermediate scattering function which can be used for the differential dynamic microscopy.

Then I presented a lattice model for the bacteria moving in the gel, with a continuous approximation. The model and the simulations give us the hint of why and under what condition we can observe the enhanced mobility with the increasing tumbling rate in experiments. The intermediate scattering function is not hard to obtain for the continuous approximation, but with an undefined function $\gamma_d(c)$, whether it can be used for the differential dynamic microscopy is still an open question. And the direct analysis of the original master equation is also left as an open question.

With the diffusive limitation of the microscopic model, we can investigate the interaction of two species with the density dependent motility regulation, which is the question we are most interested. The current results suggest a general principle: up-regulation of density-dependent motility causes the particles to segregate from a given signal, and inhibition of density-dependent motility makes the particles to co-migrate with the given signal. The general principle may be important for understanding some phenomena of collective motion in morphogenesis and ecology, as well as the pattern formation experiments we have already had. The patterns formed by mutual interactions also followed this principle, and the mechanism is distinctly different from the Turing pattern.

Another part of this thesis is on the differential dynamic microscopy. On the basis of previous work [33] and [34], I implemented the numerical inverse Laplace transform in the algorithm to measure the tumbling rate and the tumbling duration of the run-and-tumble particles. For the simulated run-and-tumble particles, the algorithm can find their tumbling rate, given the particle speed from a fitting with the RDP model. For the simulated run-and-tumble particles with tumbling duration or active rotational diffusion tumbling, a $\lambda(q)$ dependence will appear in the fitting with the RTDP model. A model with tumbling duration may helps find the tumbling duration of the simulated particles, only if the tumbling rate is already known.

The problem that we need a relatively precise guess of speed in the fitting with the RTDP model and a guess of tumbling rate in the fitting with the RTDPTT model may be due to the redundant information we put in the model, which means changing some parameters won't have much effect on the model. As suggested by [41], the appearance of an insensitive (sloppy) parameter may be universal. In the temporal-spatial scales we were investigating, tumbling rate and tumbling duration are relatively insensitive parameters, as shown in Section 2.5.4 and 3.1.3.

It is a good strategy to measure the different parameters in the different scales. As we already did, measuring the diffusion coefficient requires the small spatial scales which corresponds to the large q. To have a good estimation of tumbling rate, we should focus on the large spatial scales. However, resolving the large spatial scales needs more pixels or larger magnification, and a larger depth of field. The longer time in imaging is also needed for catching the diffusion plateau of the intermediate scattering function. This may bring difficulties in experiments: more pixels, a larger depth of field, and longer time-lapse imaging require a better imaging system. The larger magnification may bring difficulty in finding the focus, as the cells are too small to be seen by human eyes. And more memory will be needed in processing the larger and more images.

A larger depth of fields d is also necessary for measuring the tumbling rate of the particles. According to the condition in Section 6.2, the data of $q \lesssim 2\pi/d$ will be distorted, and the low q data is exactly what we need to find the tumbling rate. So a larger depth of fields can reduce the noise in the data we require, and may also enable us to use the approximation (2.46) to simplify the fitting.

The problem of the sloppy parameters also prevents us from including more ingredients into the measurement, even if we have got the intermediate scattering function from a more detailed model. If the information from six parameters is already redundant, we cannot expected any improvement with more parameters such as rotational diffusion coefficient, and the new parameters introduced in the model with obstacles.

Besides the difficulty in parameter insensitivity, for the particles with tumbling duration or active rotational diffusion, how to resolve the real tumbling rate from $\lambda(q)$ curve is also a question. For the particles with tumbling duration, $\lim_{q\to 0}\lambda(q)$ in the fitting with the RTDP model seems to be a good estimation for λ, especially for the larger images. The data from experiments also support this guess. But for the particles with active rotational diffusion, it is not that simple. We still need more efforts to find a way.

The $\lambda(q)$ dependence for the simulated particles seems to be different from that of the cells in the experiments. The simulated particles always showed a linear $\lambda(q)$ dependence, while the experiments always showed a nonlinear dependence. One possible reason is the noise from imaging. The noise mainly influences the high q range, in which the tumbling rate is very insensitive. Another possibility is that there's still some ingredients we haven't found and included in the model, such power law distribution instead of exponential distribution of the mean run length. We may find the clues by more simulation. But as the discussion on the insensitive parameters, we cannot assure that we can measure it from the $\lambda(q)$ dependence.

9.2 Future work

Fot the future work of DDM, one major problem is how to analyse and get the intermediate scattering function from the models with rotational diffusion. If we are to do DDM to measure the rotational diffusion coefficient, the computation of ISF up to arbitrary accuracy is the least requirement. We already have the formal Fourier-Laplace transformed solution. Whether truncation can give us a good approximation is still needed for further investigation.

To finalize the protocol for DDM measurement, we need more efforts in both the simulations and the experiments. We need to find a better temporal-spatial scale from simulations to measure the tumbling rate and the tumbling duration. We also need to explore more possibility of improve the imaging system for large depth of fields, and finalize an experimental protocol for the sample preparation and the imaging.

For the model of the cells moving in the semi-solid agar gel, we only have a preliminary model based on the mean field approximation, with an unknown factor $\gamma_d(c)$. We haven't gotten a way to analyse the stochastic model to have a better approximation. As for the simulations, we haven't implemented the cell-cell interactions and the cell proliferation. These interactions will give us a lattice model with the mobility regulations, with which we can test if our hypothesis for the mobility regulations still holds in the systems with the obstacles. As the experimental systems are in semi-solid agar, we need these work to ensure our theoretical prediction remains unchanged.

With the techenics we have developed, then we can turn to the work of two species interactions. For the theory, we haven't fully understood the effects of cell proliferation. The cell proliferation helps the system to select a wavelength in the mobility induced phase separation of single species [21]. But for the two species systems with mutual interactions, the calculation is complicated and the analysis is difficult. We also haven't got the amplitude equation for the pattern selection problem.

For the experiments, firstly we need to have time-lapse images to study the dynamics of pattern formation. More importantly, the engineered *E. coli* systems need to be characterized. Now we have some indirect evidences indicating that the designed systems were working as we expected. But with the DDM method, we can directly measure the mobility of *E. coli* to provide direct and quantitative evidences. These data would be useful for tuning the gene circuits to have the desired functions as well.

Appendix A

Supplementary figures

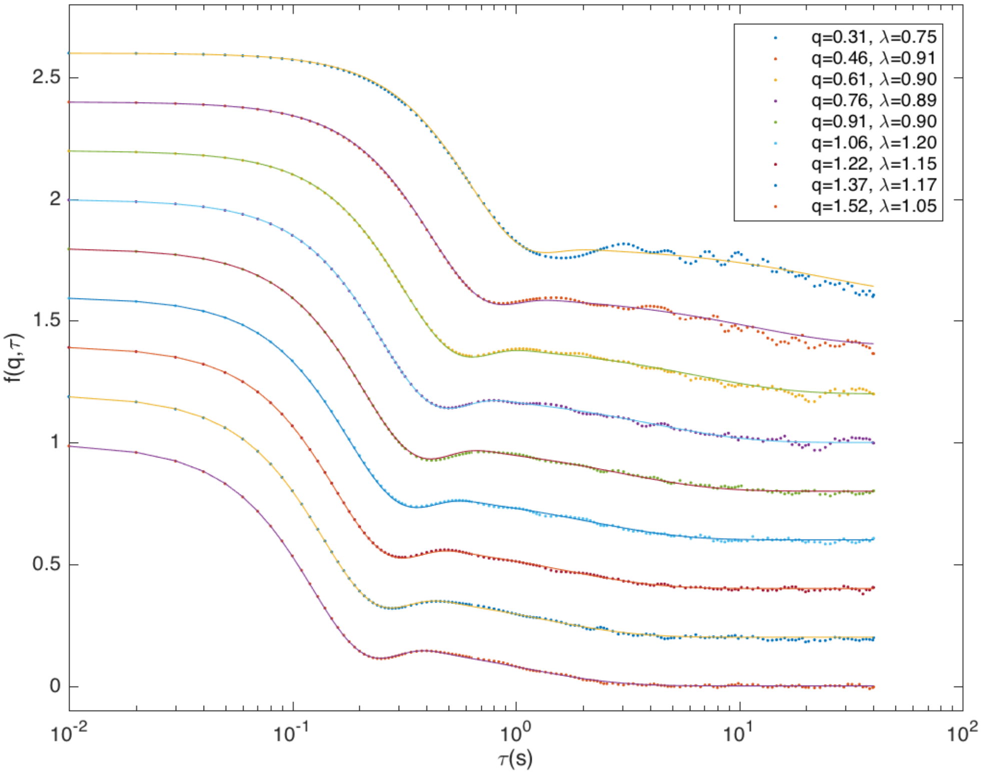


Figure A.1: The goodness of fitting in Figure 7.9. Dots are data from the DDM simulation, and solid lines are from the numerical Laplace inverse transform of the analytical expression of $f(q,s)$. The other parameters are $\alpha = 0.792$, $\bar{v} = 13.1$µm/s, $\sigma = 3.3$µm/s, $D = 0.2$µm^2/s. The curves are translated to make them clearer.

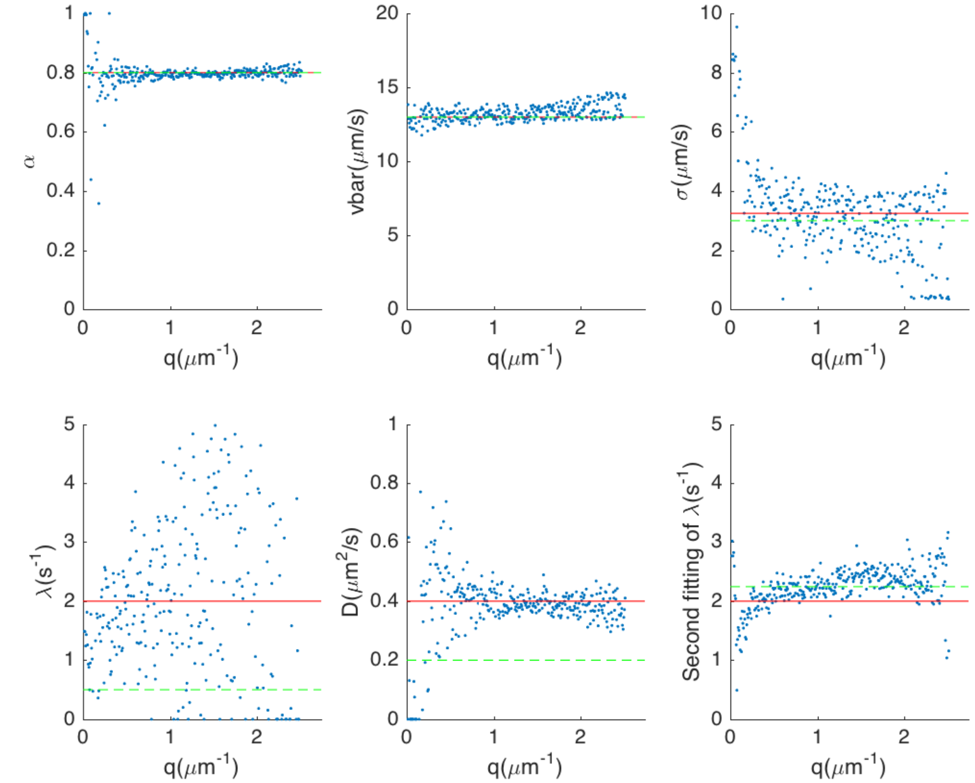


Figure A.2: Fitting result of DDM simulation of RTDP particles with RTDP model, without fixing any parameters. The red line denotes the settings in the simulation, which are $\alpha = 0.8$, $\bar{v} = 13\mathrm{\mu m/s}$, $Z = 15$, $\lambda = 2\mathrm{s}^{-1}$, $D = 0.4\mathrm{\mu m}^2/\mathrm{s}$. The green dashed lines denotes the initial guesses in the fitting. The simulation was done with the particle density of $5 \times 10^{-4}\mathrm{\mu m}^{-3}$, the pixel size of 1.625μm, the image size of 512 × 512 pixels, and the depth of field of 40μm. Averaging the data in $q < 1.934\mathrm{\mu m}^{-1}$ (with the extraordinary points excluded), we can get $\alpha = 0.796 \pm 0.008$, $\bar{v} = 13.1 \pm 0.4\mathrm{\mu m/s}$, $\sigma = 3.2 \pm 0.8\mathrm{\mu m/s}$, $\lambda = 2 \pm 1\mathrm{s}^{-1}$ $D = 0.39 \pm 0.03\mathrm{\mu m}^2/\mathrm{s}$. Then a second fitting was done with fixed $\alpha = 0.796$, $\bar{v} = 13.2\mathrm{\mu m/s}$, $\sigma = 3.2\mathrm{\mu m/s}$, $D = 0.40\mathrm{\mu m}^2/\mathrm{s}$. Averaging the data in $0.5\mathrm{\mu m}^{-1} < q < 1.5\mathrm{\mu m}^{-1}$ (with the extraordinary points excluded), we can get $\lambda = 2.3 \pm 0.2\mathrm{s}^{-1}$.

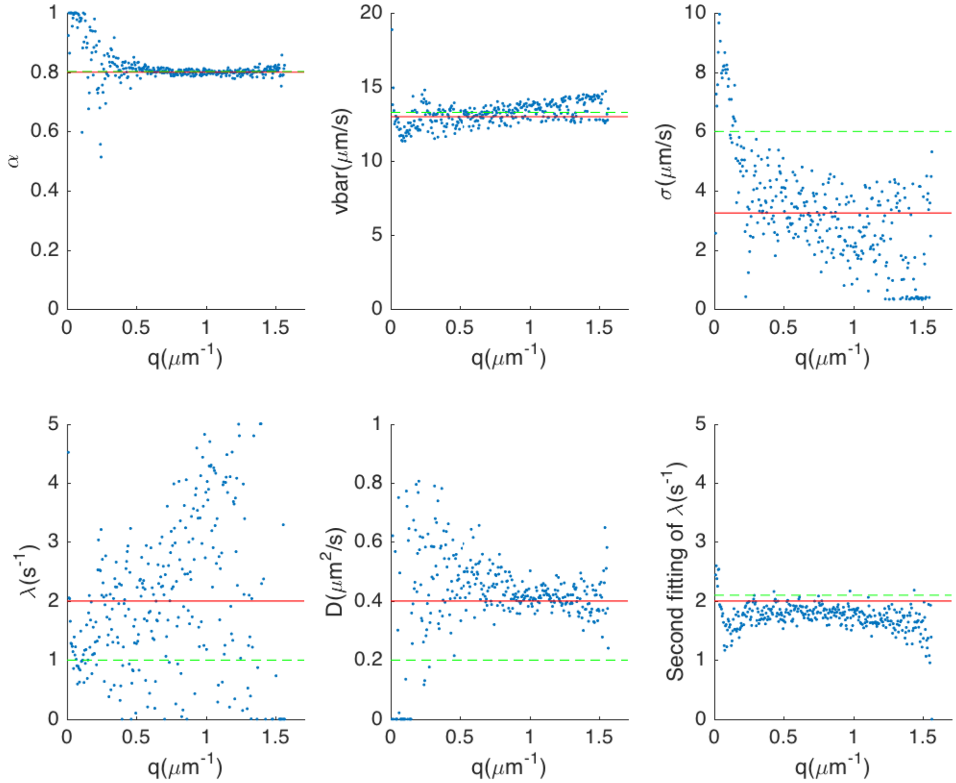


Figure A.3: Fitting result of DDM simulation of RTDP particles with RTDP model, without fixing any parameters. The red line denotes the settings in the simulation, which are $\alpha = 0.8$, $\bar{v} = 13\text{µm/s}$, $Z = 15$, $\lambda = 2\text{s}^{-1}$, $D = 0.4\text{µm}^2/\text{s}$. The green dashed lines denotes the initial guesses in fitting. The simulation was done with the particle density of $5 \times 10^{-4}\text{µm}^{-3}$, the pixel size of 2.6µm, the image size of 512×512 pixels, and the depth of field of 40µm. Averaging the data in $q < 1.209\text{µm}^{-1}$ (with the extraordinary points excluded), we can get $\alpha = 0.801 \pm 0.009$, $\bar{v} = 13.1 \pm 0.7\text{µm/s}$, $\sigma = 3.6 \pm 1.6\text{µm/s}$, $\lambda = 2 \pm 1\text{s}^{-1}$ $D = 0.44 \pm 0.05\text{µm}^2/\text{s}$. Then a second fitting was done with fixed $\alpha = 0.801$, $\bar{v} = 13.1\text{µm/s}$, $\sigma = 3.6\text{µm/s}$, $D = 0.44\text{µm}^2/\text{s}$. Averaging the data in $0.5\text{µm}^{-1} < q < 1.209\text{µm}^{-1}$ (with the extraordinary points excluded), we can get $\lambda = 1.8 \pm 0.1\text{s}^{-1}$.

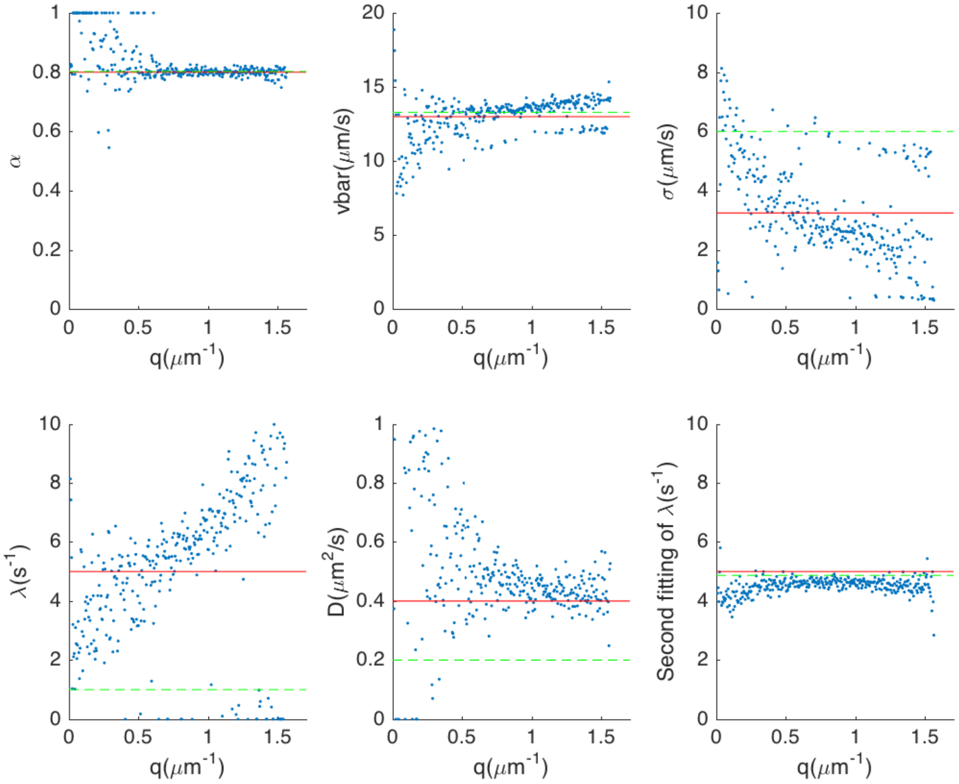


Figure A.4: Fitting result of DDM simulation of RTDP particles with RTDP model, without fixing any parameters. The red line denotes the settings in the simulation, which are $\alpha = 0.8,\ \bar{v} = 13\text{µm/s},\ Z = 15,\ \lambda = 5\text{s}^{-1},\ D = 0.4\text{µm}^2/\text{s}$. The green dashed lines denotes the initial guesses in fitting. The simulation was done with the particle density of $5\times10^{-4}\text{µm}^{-3}$, the pixel size of 2.6µm, the image size of 512×512 pixels, and the depth of field of 40µm. Averaging the data in $q < 1.209\text{µm}^{-1}$ (with the extraordinary points excluded), we can get $\alpha = 0.80\pm0.01$, $\bar{v} = 13.0\pm1.0\text{µm/s}$, $\sigma = 3.5\pm1.5\text{µm/s}$, $\lambda = 6\pm2\text{s}^{-1}$ $D = 0.46\pm0.07\text{µm}^2/\text{s}$. Then a second fitting was done with fixed $\alpha = 0.80$, $\bar{v} = 13.0\text{µm/s}$, $\sigma = 3.55\text{µm/s}$, $D = 0.46\text{µm}^2/\text{s}$. Averaging the data in $0.5\text{µm}^{-1} < q < 1.209\text{µm}^{-1}$ (with the extraordinary points excluded), we can get $\lambda = 4.6 \pm 0.1\text{s}^{-1}$.

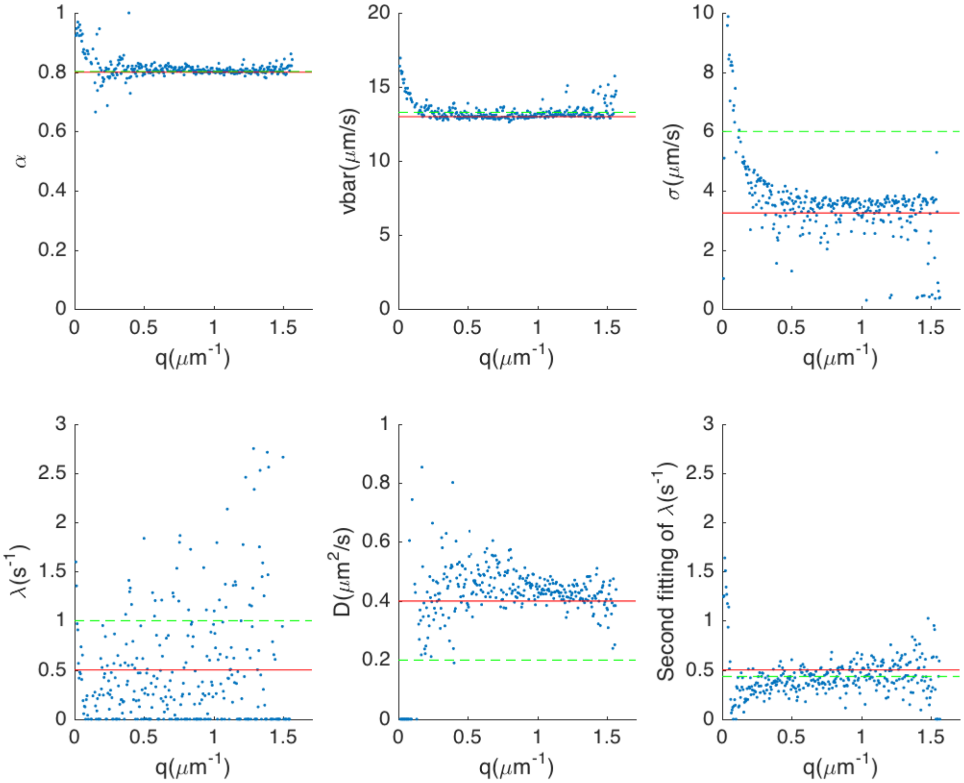


Figure A.5: Fitting result of DDM simulation of RTDP particles with RTDP model, without fixing any parameters. The red line denotes the settings in the simulation, which are $\alpha = 0.8$, $\bar{v} = 13\mu\text{m/s}$, $Z = 15$, $\lambda = 0.5\text{s}^{-1}$, $D = 0.4\mu\text{m}^2/\text{s}$. The green dashed lines denotes the initial guesses in fitting. The simulation was done with the particle density of $5 \times 10^{-4}\mu\text{m}^{-3}$, the pixel size of 2.6μm, the image size 512 × 512 of pixels, and the depth of field of 40μm. Averaging the data in $q < 1.209\mu\text{m}^{-1}$ (with the extraordinary points excluded), we can get $\alpha = 0.807 \pm 0.009$, $\bar{v} = 13.1 \pm 0.2\mu\text{m/s}$, $\sigma = 3.6 \pm 0.5\mu\text{m/s}$, $\lambda = 0.5 \pm 0.5\text{s}^{-1}$ $D = 0.45 \pm 0.05\mu\text{m}^2/\text{s}$. Then a second fitting was done with fixed $\alpha = 0.807$, $\bar{v} = 13.1\mu\text{m/s}$, $\sigma = 3.6\mu\text{m/s}$, $D = 0.45\mu\text{m}^2/\text{s}$. Averaging the data in $0.5\mu\text{m}^{-1} < q < 1.209\mu\text{m}^{-1}$ (with the extraordinary points excluded), we can get $\lambda = 0.4 \pm 0.1\text{s}^{-1}$.

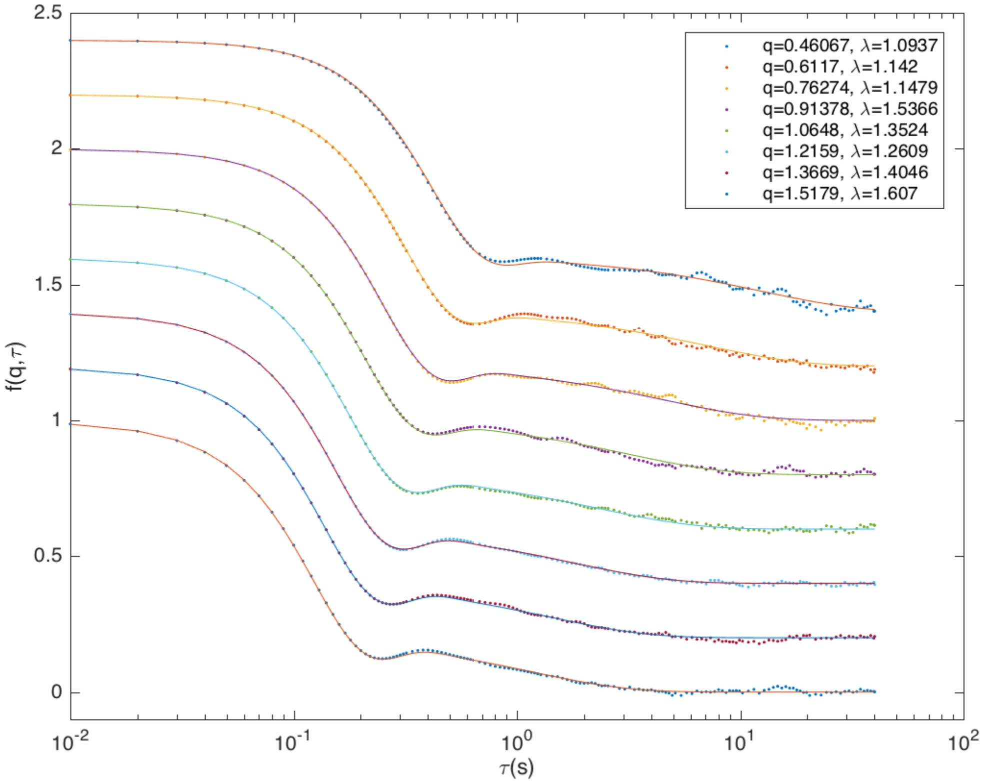


Figure A.6: The goodness of the second fitting in Figure 7.11. Dots are data from the DDM simulation, and solid lines are from the numerical Laplace inverse transform of the analytical expression of $f(q,s)$. The other parameters are indicated in the figure. The curves are translated to make them clearer.

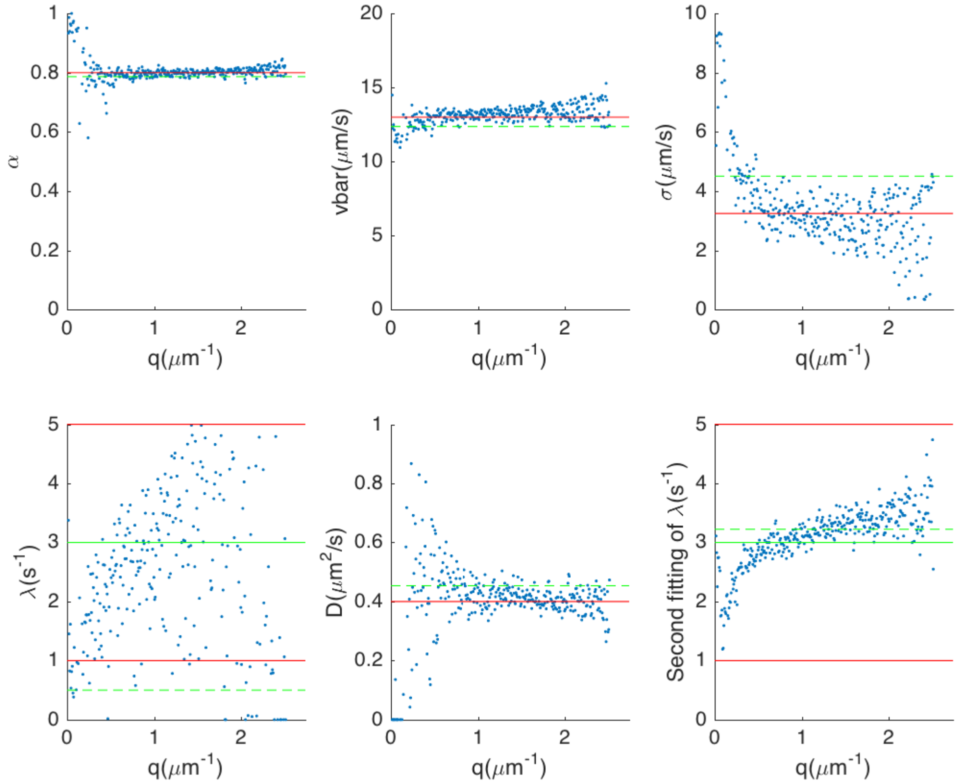


Figure A.7: Fitting result of DDM simulation of RTDP (with $P(\lambda) = 0.5P(\lambda; 1, 8) + 0.5P(\lambda; 5, 15)$) particles with RTDP model, without fixing any parameters. In q-λ figures, the red line denotes the two peaks ($\lambda = 1\mathrm{s}^{-1}$ and $\lambda = 5\mathrm{s}^{-1}$) of $P(\lambda)$, and green line denotes the average value of λ ($\lambda = 3\mathrm{s}^{-1}$). The green dashed lines denotes the initial guesses in fitting. In other figures, the red line denotes the settings in the simulation, which are $\alpha = 0.8$, $\bar{v} = 13\mathrm{\mu m/s}$, $Z = 15$, $D = 0.4\mathrm{\mu m}^2/\mathrm{s}$. The simulation was done with the particle density of $5 \times 10^{-4}\mathrm{\mu m}^{-3}$, the pixel size of 1.625μm, the image size of 512×512 pixels, and the depth of field of 40μm. Averaging the data in $q < 1.934\mathrm{\mu m}^{-1}$ (with the extraordinary points excluded), we can get $\alpha = 0.801 \pm 0.009$, $\bar{v} = 13.1 \pm 0.4\mathrm{\mu m/s}$, $\sigma = 3.3 \pm 0.8\mathrm{\mu m/s}$, $\lambda = 3 \pm 1\mathrm{s}^{-1}$ $D = 0.41 \pm 0.03\mathrm{\mu m}^2/\mathrm{s}$. Then a second fitting was done with fixed $\alpha = 0.799$, $\bar{v} = 13.2\mathrm{\mu m/s}$, $\sigma = 3.2\mathrm{\mu m/s}$, $D = 0.42\mathrm{\mu m}^2/\mathrm{s}$. Averaging the data in $0.5\mathrm{\mu m}^{-1} < q < 1.5\mathrm{\mu m}^{-1}$ (with the extraordinary points excluded), we can get $\lambda = 3.1 \pm 0.2\mathrm{s}^{-1}$.

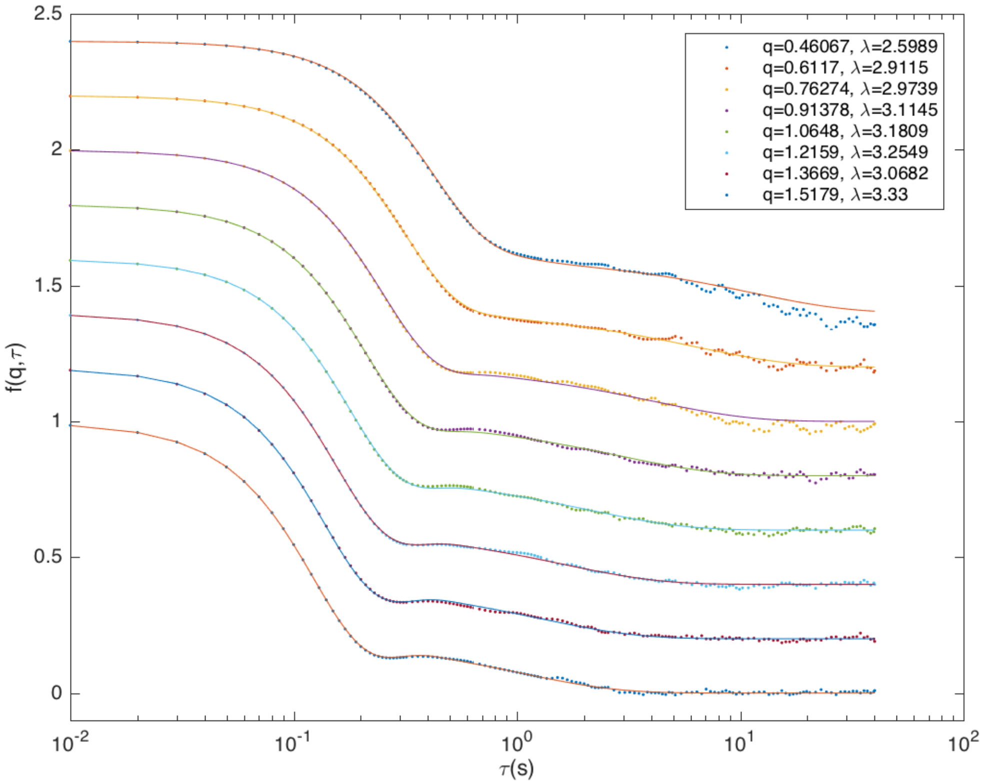


Figure A.8: The goodness of the second fitting in Figure A.7. Dots are data from the DDM simulation, and solid lines are from the numerical Laplace inverse transform of the analytical expression of $f(q, s)$. The other parameters are indicated in the figure. The curves are translated to make them clearer.

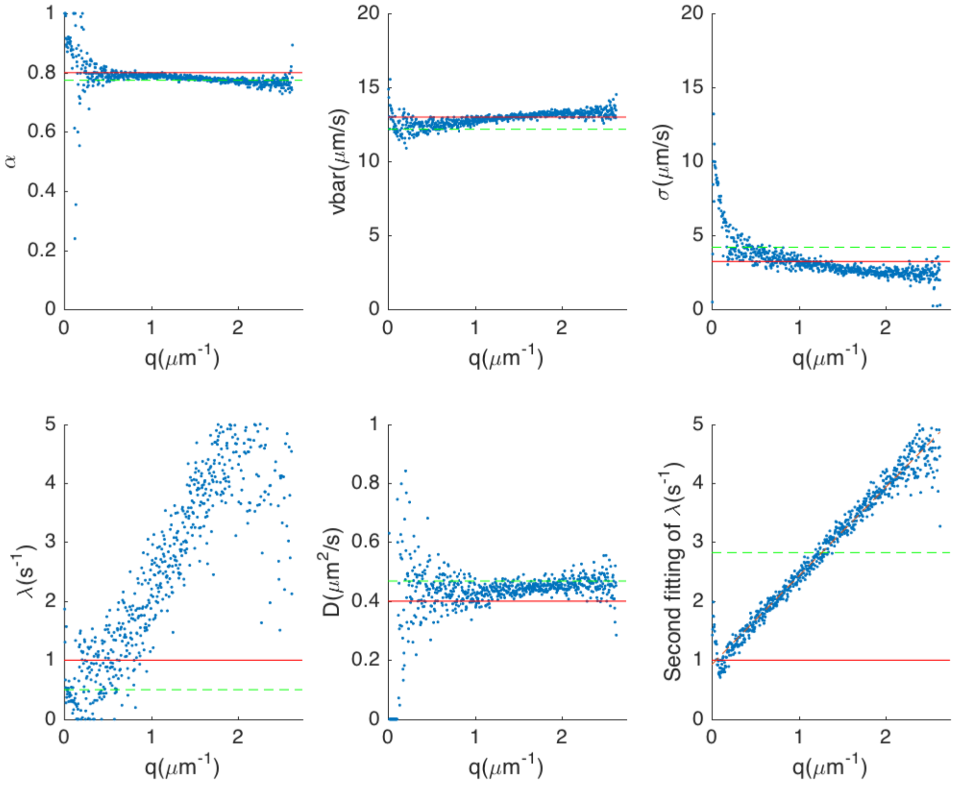


Figure A.9: Fitting result of DDM simulation of RTDPTT particles with RTDP model. The red line denotes the settings in the simulation, which are $\alpha = 0.8$, $\bar{v} = 13$µm/s, $\sigma = 3.25$µm/s, $\lambda = 1\text{s}^{-1}$, $D = 0.4\text{µm}^2/\text{s}$, $\delta = 0.1$s. The green dashed lines denotes the initial guesses in fitting. The simulation was done with the particle density of $5 \times 10^{-4}\text{µm}^{-3}$, the pixel size of 1.625µm, the image size of 1024×1024 pixels, and the depth of field of 40µm. Averaging the data in $q < 1.934\text{µm}^{-1}$ (with the extraordinary points excluded), we can get $\alpha = 0.79 \pm 0.01$, $\bar{v} = 12.8 \pm 0.4$µm/s, $\sigma = 3.2 \pm 0.6$µm/s, $\lambda = 2.2 \pm 1.4\text{s}^{-1}$, $D = 0.44 \pm 0.02\text{µm}^2/\text{s}$. Then a second fitting was done with fixed $\alpha = 0.786$, $\bar{v} = 12.91$µm/s, $\sigma = 2.96$µm/s, $D = 0.43\text{µm}^2/\text{s}$. Now a q dependence occurred. The red dash line shows a linear fitting which is $\lambda = 0.9247 + 1.5034q$.

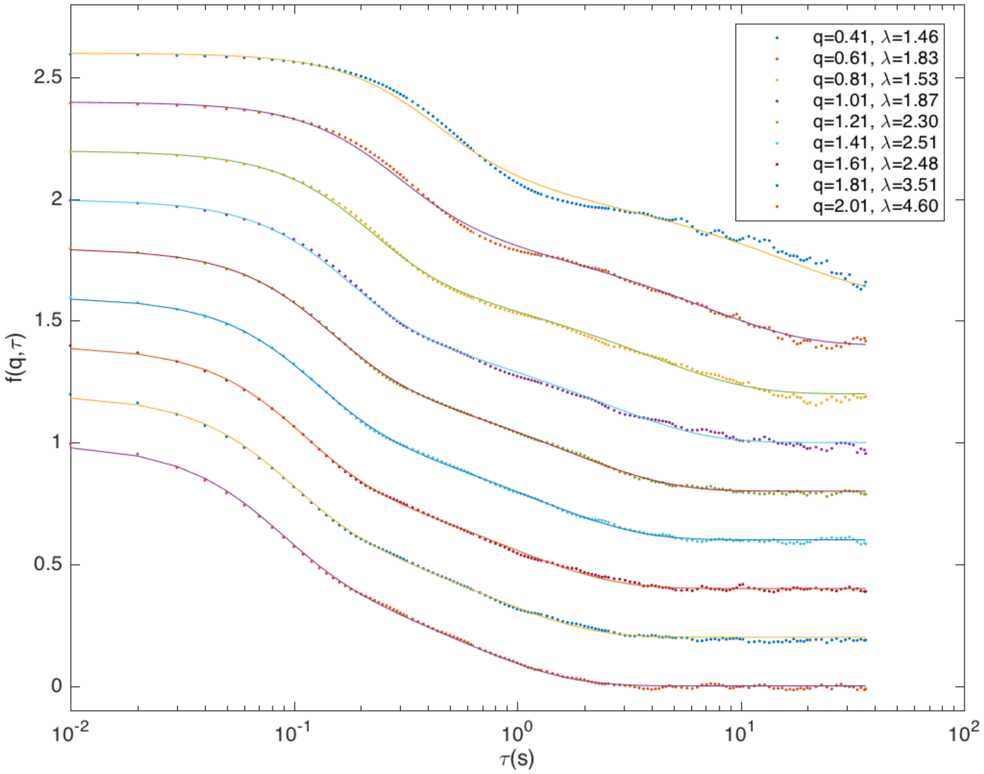


Figure A.10: The goodness of fitting in Fig. 8.2. Dots are data from the experiment, and solid lines are from the numerical Laplace inverse transform of the analytical expression of $f(q,s)$. The other parameters are $\alpha = 0.61$, $\bar{v} = 11.4\mu$m/s, $\sigma = 6.8\mu$m/s, $D = 0.37\mu\text{m}^2$/s. The curves are translated to make them clearer.

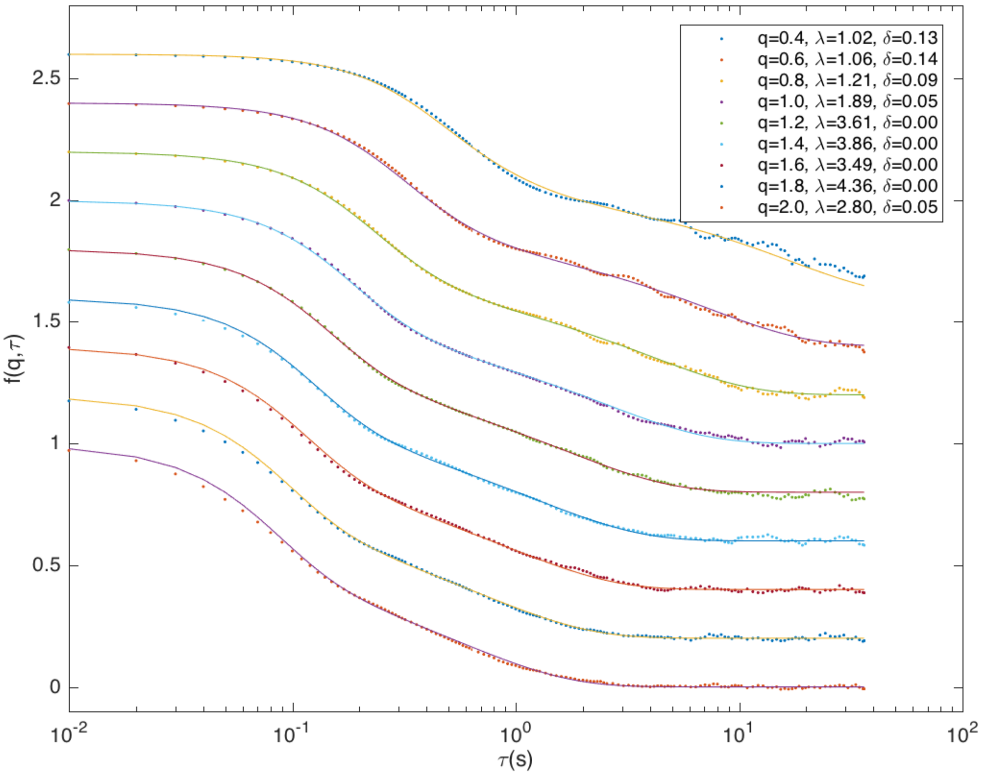


Figure A.11: The goodness of fitting in Fig. 8.4. Dots are data from the experiment, and solid lines are from the numerical Laplace inverse transform of the analytical expression of $f(q, s)$. The other parameters are $\alpha = 0.60$, $\bar{v} = 11.8\mu\text{m/s}$, $\sigma = 6.4\mu\text{m/s}$, $D = 0.36\mu\text{m}^2/\text{s}$. The curves are translated to make them clearer.

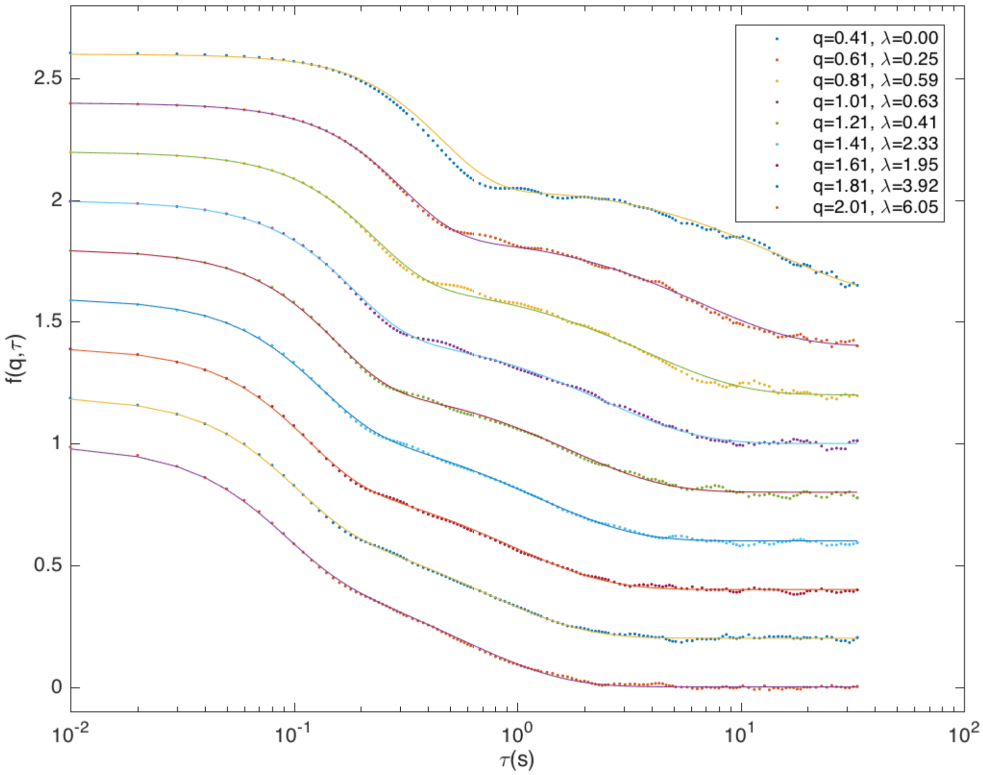


Figure A.12: The goodness of fitting in Fig. 8.6. Dots are data from the experiment, and solid lines are from the numerical Laplace inverse transform of the analytical expression of $f(q,s)$. The other parameters are $\alpha = 0.52$, $\bar{v} = 12.1\mu$m/s, $\sigma = 5.3\mu$m/s, $D = 0.40\mu\text{m}^2$/s. The curves are translated to make them clearer.

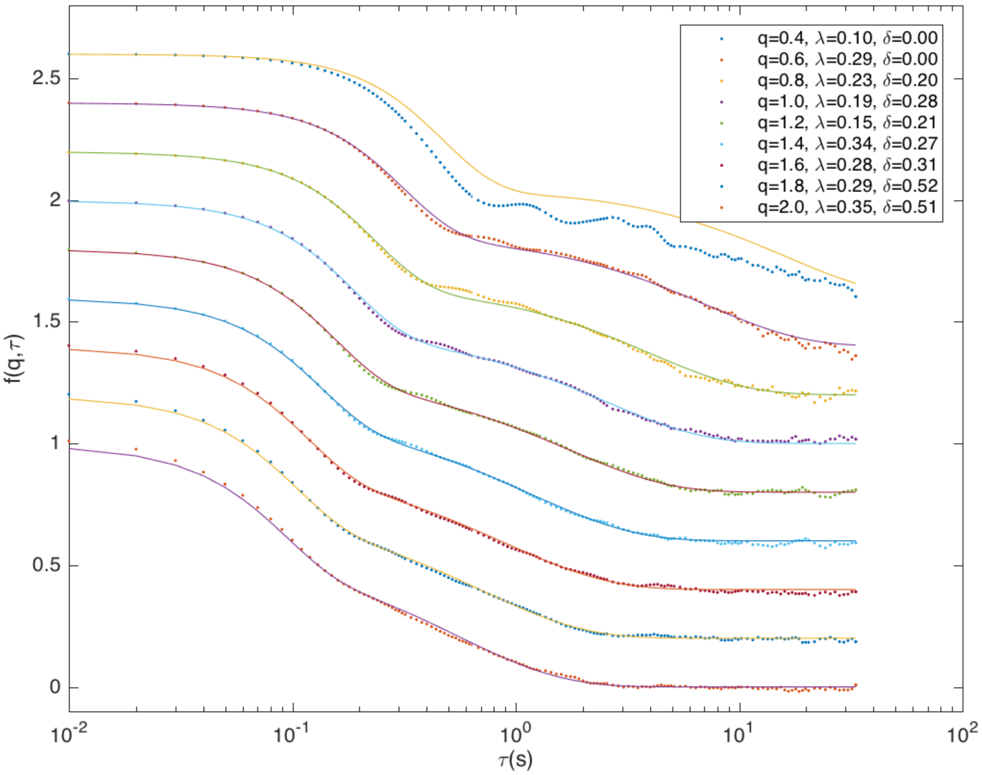


Figure A.13: The goodness of fitting in Fig. 8.8. Dots are data from the experiment, and solid lines are from the numerical Laplace inverse transform of the analytical expression of $f(q,s)$. The other parameters are $\alpha = 0.54$, $\bar{v} = 12.0\mu$m/s, $\sigma = 5.3\mu$m/s, $D = 0.39\mu\text{m}^2$/s. The curves are translated to make them clearer.

Appendix B

Algorithm reference

B.1 Procedure

The detailed steps of the algorithm are listed below:

1. Load all the images, and do FFT on them. The data will be stored in the vector `imageSeqk`.

2. Calculate $g(\boldsymbol{q}, l\Delta t) = \sum_k |I(\boldsymbol{q}, t_k + l\Delta t) - I(\boldsymbol{q}, t_k)|^2$ to approximate the time average, where k is from 0 to $\texttt{numOfSeq} - \texttt{numOfDiff} - 1$, and l is sampled in the log scale. (For now, l is of the form of $p2^n$, where $p \leq 31$ is a prime number, and $n \geq 0$ is an integer.) The data will be stored in the vector `imagekDiff`.

3. Clean up the vector `imageSeqk`.

4. Average on the the direction of $\boldsymbol{q}$ by integrating the bilinear interpolation function. The detail can be found in Section B.2. The result will be stored in the matrix `datag`.

5. Clean up the vector `imagekDiff`.

6. Print the unfitted $g(q, \tau)$ data, the q array, and the τ array into files.

7. Call `lmsder` routine to fit $g(q, \tau)$ with the theoretical ISF for each q value or multiple q values at the same time. `lmsder` is a robust and efficient rotine of the Levenberg-Marquardt algorithm. The details of the algorithm please refer to the official manual of GSL [47]. If using the model that fits the tumbling rate, the inverse Laplace transform will be called in the

calculation of $f(q,\tau)$. See Section B.3.

8. Print the fitting results into files.

B.2 Average of direction of q

The average of the direction of q is done by integrating the bilinear interpolation function. After Fourier transform, we have the $g(\boldsymbol{q},\tau)$ on the lattice (q_x, q_y). Thus in a small lattice cell, we have the bilinear interpolation function (for convenience, the τ dependence is omitted, and $g_{lm} = g(q_{xl}, q_{ym})$ is the value at the lattice site (l,m).)

$$\begin{aligned} g(q_x,q_y) =& g_{11} + \frac{q_y - q_{y1}}{q_{y2}-q_{y1}}(g_{12}-g_{11}) + \frac{q_x - q_{x1}}{q_{x2}-q_{x1}}(g_{21}-g_{11}) \\ &+ \frac{q_x - q_{x1}}{q_{x2}-q_{x1}}\frac{q_y - q_{y1}}{q_{y2}-q_{y1}}(g_{22}-g_{21}-g_{12}+g_{11})\,. \end{aligned} \tag{B.1}$$

We write it in the polar coordinates $g(q,\theta)$, and then integrate it, to get

$$\begin{aligned} g_I(q,\theta) &= \int g(q,\theta)\,d\theta \\ &= g_{11}\theta - \frac{q\cos\theta + q_{y1}\theta}{q_{y2}-q_{y1}}(g_{12}-g_{11}) + \frac{q\sin\theta - q_{x1}\theta}{q_{x2}-q_{x1}}(g_{21}-g_{11}) \\ &\quad + \frac{q_{x1}q\cos\theta + q_{x1}q_{y1}\theta - q_{y1}q\sin\theta - \frac{1}{4}q^2\cos 2\theta}{(q_{x2}-q_{x1})(q_{y2}-q_{y1})}(g_{22}-g_{21}-g_{12}+g_{11})\,. \end{aligned} \tag{B.2}$$

The average will be

$$g(q) = \frac{\sum_k (g_I(q,\theta_{k+1}) - g_I(q,\theta_k))}{\sum_k(\theta_{k+1}-\theta_k)}\,, \tag{B.3}$$

where θ_k is the angle of the intersection points between the circle of constant q and the q-grid.

In the code, after we determine the q list that is to be averaged, we loop over all the lattice cells. In each lattice cell, we determine all the arcs of constant q in this cell, calculate the end points of each arc, and do the integration in (B.2). After the loop, we add up all the integrations on arcs, and do the average according to (B.3).

B.3 Numerical inverse Laplace transform

B.3.1 Weeks' method

The inverse numerical Laplace transform is always regarded as a difficult problem in numerical analysis. Due to the oscillation nature of the integration kernel in the inverse Laplace transform, we can't find a stable and universal algorithm for all kinds of problems. It is lucky that due to some property of the run-and-tumble ISF, we can implement the classical Weeks' method [48] in DDM.

Denoting the function to be inverse transformed as $F(s)$, Weeks' method first require $s^p F(s)$ to be analytic at the infinity, which means $F(s)$ should behave like $1/s^p$ at the infinity. Then we can make use of the generalized Laguerre polynomials $L_n^{p-1}(x)$ to expand the inverse transformed function $f(t)$ [49]. In our case, $sF(s)$ is already analytical, so we only use the Laguerre polynomials $L_n^0(x) = L_n(x)$.

Firstly we need a Möbius transformation from s to z,

$$s = \sigma - b\frac{z+1}{z-1}\,, \quad z = \frac{s-\sigma-b}{s-\sigma+b} \tag{B.4}$$

where $b > 0$ and σ are the free parameters of the algorithm. As the Möbius transformation maps the region Re $s > \sigma$ into the unit circle, and we don't want any singularities inside because we are going to use the Cauchy's integral formula later, σ should be larger than the largest real part of all the singularities of $F(s)$. The performance of the algorithm will critically depend on the choice of σ and b, and we will discuss it in section B.3.2.

Then we expand the function

$$(s-\sigma+b)F(s) = \frac{2b}{1-z}F\left(\sigma - b\frac{z+1}{z-1}\right) = \sum_{n=0}^{\infty} a_n z^n \tag{B.5}$$

into Taylor series. It means to expand $F(s)$ into

$$F(s) = \sum_{n=0}^{\infty} a_n \frac{(s-\sigma-b)^n}{(s-\sigma+b)^{n+1}}\,. \tag{B.6}$$

As it is easily shown that

$$\mathcal{L}[L_n(2bt)] = \frac{(s-2b)^n}{s^{n+1}}\,, \tag{B.7}$$

we actually get an expansion of the inverse transformed function as

$$\mathcal{L}^{-1}[F(s)] = f(t) = \sum_{n=0}^{\infty} a_n e^{(\sigma-b)t} L_n(2bt)\,. \tag{B.8}$$

The convergence of this series is determined by the radius of convergence R of Taylor series (B.5). As the asymptotic behaviour of $L_n(x)$ at large n is

$$L_n(t) = \frac{e^{t/2}}{\sqrt[4]{\pi^2 nt}} \cos\left(2\sqrt{nt} - \frac{\pi}{4}\right) + O(n^{-3/4})\,, \tag{B.9}$$

$e^{(\sigma-b)t}L_n(2bt)$ is uniformly bounded. And $|a_n| < \frac{K(R)}{R^n}$, where $K(R)$ is a constant related to R [49]. As long as $R > 1$, the series converges. And the analyticity of $sF(s)$ at the infinity guarantees the $z = 1$ is not a singularity of (B.5), so $R > 1$. As the series converges like R^{-n}, we can truncate the series as an approximation of $f(t)$.

The calculation of coefficients a_n can be done by the Cauchy's integral formula

$$\begin{aligned} a_n &= \frac{1}{n!}\frac{d^n}{dz^n}\left[\frac{2b}{1-z}F\left(\sigma - b\frac{z+1}{z-1}\right)\right]\Bigg|_{z=0} \\ &= \frac{1}{2\pi i}\oint_C \frac{1}{z^{n+1}}\frac{2b}{1-z}F\left(\sigma - b\frac{z+1}{z-1}\right)dz\,. \end{aligned} \tag{B.10}$$

We take the contour to be the unit circle $z = e^{i\theta}$, the integration becomes

$$\begin{aligned} a_n &= \frac{1}{2\pi}\int_0^{2\pi} e^{-in\theta}\frac{2b}{1-e^{i\theta}}F\left(\sigma - b\frac{e^{i\theta}+1}{e^{i\theta}-1}\right)d\theta \\ &\approx \frac{e^{n\pi i/M}}{M}\sum_{m=0}^{M-1} e^{-2nm\pi i/M}\frac{2b}{1-e^{(2m+1)\pi i/M}}F\left(\sigma - b\frac{e^{(2m+1)\pi i/M}+1}{e^{(2m+1)\pi i/M}}\right)\,. \end{aligned} \tag{B.11}$$

In the later equation, we take M points $\theta_m = 2m\pi/M$ and use the mid-point rule to approximate the quadrature. If we keep the first M terms in the expansion (B.5), the summation in (B.11) is a discrete Fourier transformation and can be done by fast Fourier transform (FFT) algorithm, which is considered to be fast and stable numerically.

B.3.2 Selection of σ and b

As we mentioned in the last section, the choice of σ and b is critical. We should try to make the radius of convergence R larger in the expansion (B.5), which is determined by σ, to improve the convergence. The large σ and b can make the Möbius transformed singularities of $F(s)$ away from the unit circle, but too large σ or b may make the numerical evaluation of the series (B.8) fails at the large t. There is a trade-off in the choice of σ and b.

Generally, one can find the optimal choice by minimizing the error estimation

[49]

$$E(\sigma, b) = \max_t \left\{ e^{\sigma t} \left(\sum_{n=M/2}^{M-1} |a_n| + \epsilon \sum_{n=0}^{M/2-1} |a_n| \right) \right\} , \tag{B.12}$$

where ϵ is the machine precision. The first term is the estimation of the truncation error, and the second one is of the round-off error. However, the two dimensional optimization problem is often costly, especially if we need to estimate σ and b at every iteration after we change the parameters in ISF. Another problem is when `nan` is returned, it becomes even harder to do the searching.

Wiedman did both analytical and numerical analysis in 1999 for the optimal choice of σ and b [50]. He found that with the knowledge of singularities of $F(s)$, it is possible to find an equation relating the optimal σ and b, and the search will be more efficient.

The idea is to find the optimal circle that enclose all the singularities of $F(s)$, then left us the adjustment of the radius of the mapped circle after Möbius transformation. Generally there will be two situations.

1. If the circle is determined by one pair of singularities $s = \alpha \pm i\beta$, b and σ should satisfy

$$b^2 - (\sigma - \alpha)^2 - \beta^2 = 0\,. \tag{B.13}$$

2. If the circle is determined by two pairs of singularities $s_{1,2} = \alpha_{1,2} \pm i\beta_{1,2}$, b and σ should satisfy

$$b^2 - \sigma^2 + \frac{|s_2|^2(\sigma - \alpha_1) - |s_1|^2(\sigma - \alpha_2)}{\alpha_2 - \alpha_1} = 0\,. \tag{B.14}$$

Then we only need to choose a suitable σ. Wiedman suggested that we can do one dimensional optimization to minimize (B.12), but due to the cost of the search, we simply choose σ to be not far away from the rightmost singularity, and the convergence of inverse transform is already fast enough.

Here we list the singularities of the intermediate scattering functions we have considered.

RTD model

The function is

$$f(q, s) = \frac{\arctan(qv/(s + Dq^2 + \lambda))}{qv - \lambda \arctan(qv/(s + Dq^2 + \lambda))}\,. \tag{B.15}$$

Firstly, if $qv/\lambda < \pi/2$, it is possible for the denominator of (6.33) to be zero, which provide a pole

$$s_1 = \frac{qv}{\tan(qv/\lambda)} - Dq^2 - \lambda\,. \tag{B.16}$$

As

$$\arctan\frac{qv}{s + Dq^2 + \lambda} = \frac{i}{2}\ln\left(\frac{s + Dq^2 + \lambda - iqv}{s + Dq^2 + \lambda + iqv}\right)\,, \tag{B.17}$$

$s_{2,3} = -Dq^2 - \lambda \pm iqv$ will be two branch points, and the line in between will be the branch cut except for $s_4 = -Dq^2 - \lambda$, which is also a branch point. Then if $qv/\lambda < \pi/2$, s_1 should be the rightmost singularity, and the circle in the Weidman's method will be determined by two pairs of singularities s_1 (a degenerated pair) and $s_{2,3}$. If $qv/\lambda \geq \pi/2$, $s_{2,3,4}$ should be the rightmost singularity, and the circle in the Weidman's method will be determined by one pair of singularities $s_{2,3}$.

RTDP model

To find the singularities of

$$f(q,s) = \int_0^\infty \frac{\arctan(qv/(s + Dq^2 + \lambda))}{qv - \lambda\arctan(qv/(s + Dq^2 + \lambda))} P(v)\,dv\,, \tag{B.18}$$

we need to note it may not be a meromorphic function. We could focus on how the integration behaves when passing through a singularity of the ISF of the RTD model, as the integration will always exist for s that doesn't pass any singularity when integrating. As if $qv/\lambda < \pi/2$, $qv/\tan(qv/\lambda) < qv/(qv/\lambda) = \lambda$, $s_1 < -Dq^2$ for the ISF of the RTD model. Thus for $-Dq^2 \geq s > -Dq^2 - \lambda$, there exists $v \geq 0$ that satisfies

$$\frac{qv}{\lambda} = \arctan\frac{qv}{s + Dq^2 + \lambda} \tag{B.19}$$

and the integration will meet a pole and diverge. For $s > -Dq^2$ or $qv/\lambda \geq \pi/2$, (B.19) cannot be satisfied, and the integration cannot meet more poles. So the points on the line segment $-Dq^2 \geq s > -Dq^2 - \lambda$ are singularities of the ISF of the RTDP model. On the other hand, for $s = -Dq^2 - \lambda + iy$, $y \neq 0$, the integration pass through a branch point and can still be determined. So the singularities of the ISF of the RTDP model are just the line segment $-Dq^2 \geq s \geq -Dq^2 - \lambda$, and the circle in the Weidman's method is determined by two (degenerated) pairs of singularities $s_1' = -Dq^2$ and $s_2' = -Dq^2 - \lambda$.

RTDPTT model

The ISF is

$$p(q,s) = \frac{qv\delta^2\lambda + [1 + \delta(Dq^2 + 2\lambda + s)]\arctan(qv/(Dq^2 + \lambda + s))}{(1 + \delta\lambda)[qv(1 + \delta Dq^2 + \delta s) - \lambda\arctan(qv/(Dq^2 + \lambda + s))]}\,, \tag{B.20}$$

with

$$P(v) = \frac{v^Z}{\Gamma(Z+1)} \left(\frac{Z+1}{\bar{v}} \right)^{Z+1} e^{-(Z+1)v/\bar{v}} \,. \tag{B.21}$$

At a glance, $s = -Dq^2, -Dq^2 - \lambda$ are singularities of $f(q,s)$. Then, for each value of v, if there exists an s satisfying

$$\frac{qv(\delta(s+Dq^2)+1)}{\lambda} \leq \frac{\pi}{2} \tag{B.22}$$

and

$$\tan \frac{qv(\delta(s+Dq^2)+1)}{\lambda} = \frac{qv}{s+\lambda+Dq^2} \,, \tag{B.23}$$

we have a singularity. When the first condition holds, we have

$$\frac{qv}{s+\lambda+Dq^2} = \tan \frac{qv(\delta(s+Dq^2)+1)}{\lambda} \geq \frac{qv(\delta(s+Dq^2)+1)}{\lambda} \,, \tag{B.24}$$

giving us an upper and a lower limit of s

$$(s+\lambda+Dq^2)(\delta(s+Dq^2)+1) \leq \lambda \,, \tag{B.25}$$

$$-Dq^2 - \lambda - \frac{1}{\delta} \leq s \leq -Dq^2 \,. \tag{B.26}$$

All the singularities of $f(q,s)$ should be on real axis and in this range. This can be used in the numerical inverse Laplace transform.

B.3.3 Clenshaw summation

Usually it's not a good idea to evaluate a series summation such that

$$f(x) = \sum_{n=0}^{N} a_n L_n(x) \tag{B.27}$$

directly, as it's time-consuming to evaluate special functions like Laguerre polynomials, and may have some numerical instability problems. We implement the Clenshaw summation [49] to evaluate $f(x)$ recursively.

If we have some recursion relation

$$L_n(x) = c_{1n}(x)L_{n-1}(x) + c_{2n}(x)L_{n-2}(x) + \cdots + c_{mn}(x)L_{n-m}(x) \tag{B.28}$$

for all $n \geq 1$ (the terms with a negative label are assigned to be zero). We consider a series y_n such that

$$\begin{aligned} y_{N+m-1} &= y_{N+m-2} = \cdots = y_{N+1} = 0 \,, \\ y_N &= a_N \,, \\ y_n &= a_n + c_{1,n+1}y_{n+1} + c_{2,n+2}y_{n+2} + \cdots + c_{m,n+m}y_{n+m} \,. \end{aligned} \tag{B.29}$$

Then $y_0(x)L_0(x) = f(x)$. Usually, the 0th order special function $L_0(x)$ is simple. Then we finished the summation.

To show it's true, we first solve

$$a_n = y_n - c_{1,n+1}y_{n+1} - c_{2,n+2}y_{n+2} - \cdots - c_{m,n+m}y_{n+m} \tag{B.30}$$

and substitute into (B.27), we have

$$\begin{aligned} f(x) &= \sum_{i=0}^{N}(y_i - c_{1,i+1}y_{i+1} - \cdots - c_{m,i+m}y_{i+m})L_i \\ &= \sum_{i=m}^{N}(L_i - c_{1i}L_{i-1} - \cdots - c_{mi}L_{i-m})y_i \\ &\quad + (L_{m-1} - c_{1i}L_{i-1} - \cdots - c_{m-1,m-1}L_0)y_{m-1} \\ &\quad + \cdots \\ &\quad + (L_1 - c_{11}L_0)y_1 \\ &\quad + L_0y_0\,. \end{aligned} \tag{B.31}$$

By the recursion relation (B.28), all terms except L_0y_0 is zero, then we finish the proof.

For the Laguerre polynomials we use, the recursion relation reads

$$L_n(t) = \frac{2n-1-t}{n}L_{n-1}(t) - \frac{n-1}{n}L_{n-2}(t)\,, \tag{B.32}$$

and the series y_n should be like

$$\begin{aligned} y_{N+1} &= 0\,, \\ y_N &= a_N\,, \\ y_n &= a_n + \frac{2n+1-t}{n+1}y_{n+1} - \frac{n+1}{n+2}y_{n+2}\,, \end{aligned} \tag{B.33}$$

and $\sum_{n=0}^{N} a_nL_n(t) = y_0$. Then we don't need to evaluate any Laguerre polynomials.

In practice, we also truncate a_n at N_0 where $a_{N_0} < \epsilon$ which is a small number close to the machine precision, to make it more stable numerically. However, in numerical inverse Laplace transform, the summation can still fail at the extremely large t. Use the `long double` data type longer than 64-bit can help increase the largest t that can be calculated.

Appendix C

Quick start of DDM code

C.1 Provided models

The process of DDM data need to specify a model priorly (for details, see Chapter 6. In this code, we provided 5 models and 2 variations.

- Run-and-diffuse (RD) model, corresponding to straight swimmer with Brownian motion. The variables to be fitted will be `alpha` (α), `D` (D), `vbar` (v).

- Run-and-diffuse with $P(v)$ (RDP) model, corresponding to straight swimmer with Brownian motion and a distribution in v (approximated by Schultz distribution). The variables to be fitted will be `alpha` (α), `D` (D), `vbar` ($\bar{v}$), `Z` (Z).

- Run-and-tumble-and-diffuse (RTD) model, corresponding to run-and-tumble particles with Brownian motion. The variables to be fitted will be `alpha` (α), `vbar` (v), `lambda` (λ), `D` (D).

- Run-and-tumble-and-diffuse with $P(v)$ (RTDP) model, corresponding to run-and-tumble particles with Brownian motion and a distribution in v (approximated by Schultz distribution). The variables to be fitted will be `alpha` (α), `vbar` ($\bar{v}$), `sigma` (σ), `lambda` (λ), `D` (D).

- Run-and-tumble-and-diffuse with $P(v)$ model, but fixed everything except tumbling rate (RTDPfix). The variables to be fitted will be `lambda` (λ).

- Run-and-tumble-and-diffuse with $P(v)$ and tumbling duration (RTDPTT) model, corresponding to run-and-tumble particles with Brownian motion and a distribution in v (approximated by Schultz distribution) and a finite tumbling duration. The variables to be fitted will be `alpha` (α), `vbar` ($\bar{v}$), `sigma` (σ), `lambda` (λ), `D` (D), `delta` (δ).

- Run-and-tumble-and-diffuse with $P(v)$ and tumbling duration model, but fixed everything except tumbling rate and tumbling duration (RTDPT-Tfix). The variables to be fitted will be `lambda` (λ), `delta` (δ).

C.2 System requirement

The main limit of this routine is the memory, as we load all the images when processing the data. Processing 4500 pieces of 512×512 16-bit images requires roughly 9GB memory.

The inverse Laplace transform routine will take a lot of time. To measure the tumbling rate of *E. coli*, 700-1500 min CPU time is required per sample in 3.33GHz Intel Xeon X5680.

As Visual Studio doesn't provide `long double` data type longer than 64-bit, we don't recommend to use Windows. However the `parameters.h` still keeps the macro switch `WINDOWS` to import the required external statistic libraries and the proper header. Unix (including Mac OS X) and Linux should be the perfect platforms to run this code.

C.3 Get the source code

The source code is stored in the git repository in github.com. To get the source code, make a new folder and run the following command in bash

```
git clone https://github.com/zhaoyongfeng1990/DDM.git
```

C.4 Library dependence

The code is mainly written in C++. Before compiling the code, the following libraries should be available in user's system

- GNU Scientific Library (GSL)
- FFTW

- OpenMP

Users should make sure the version of GSL is new enough to provide the following features

- Matrix data types
- `lngamma` function
- `psi` function
- `lmsder` nonlinear curve fitting routine
- `cquad` numerical integration routine

The parallel computing is implemented using the OpenMP library, so the multi-thread FFTW is not required. But the long double FFTW library is required and is critical in numerical inverse Laplace transform.

C.5 Compilation

Before compilation, user should examine the `CXX` variable in `makefile`, to make sure it's the desired compiler.

To compile the source code, we provided several `sh` scripts. `compileAll.sh` can automatically generate the executable binary files for different models, named as `ddm$ModelName$`, where `$ModelName$` can be RD, RDP, RTD, RTDP, RTDPfix, RTDPTT, RTDPTTfix. To use the script, just type

```
./compileAll.sh
```

or

```
sh ./compileAll.sh
```

in the commend line. The binary files can be found in `./bin/` . The code will generate 2 binary files for RTDP model. `ddmRTDP` will perform fitting on all the parameters, while `ddmRTDPfix` will perform fitting by fixing all the parameters except the tumbling rate λ. Details can be found in Section C.10.

To compile an individual model, user can use `compile.sh` as following

```
./compile.sh $ModelName$
```

or

```
sh ./compile.sh $ModelName$
```

where `$ModelName$` can be RD, RDP, RTD, RTDP, RTDPfix, RTDPTT, RTDPTTfix. The generated file can be found in `./bin/` .

Actually, we are using macro to switch between different model, which is written in the header `config.h`. The `sh` scripts are to detect and generate `config.h` automatically, then call on `make`. Users can also (but not recommended) manually edit this file and then use

```
make
```

or

```
make ddm
```

The output file `ddm` can be found in the current folder. To clean up all the `.o` files, user can use

```
make clean
```

C.6 Debug

If users would like to dubug or try something, they can write the code in `debugmain.cpp`, and then call

```
make debug
```

The binary file will be named as `debug` and will be run automatically.

C.7 Configuration

Before processing the data, several parameters need to be given in prior. Users can copy `./bin/config.sh` to data folder, and edit this file. It contains following parameters

- `omp_num`: Number of threads to be used by OpenMP library.
- `nqcurve`: Number of curves that to be fitted simultaneously. If the model is not RTDP or RTDPfix, this should be set to 1.
- `qincre`: If `nqcurve` = 1, this should be left as blank. If `nqcurve` > 1, user should specify the `nqcurve` − 1 increments of index of q between two

successive curves. Different increments should be separated by space.

- `dimx`: Image width in pixel.
- `dimy`: Image height in pixel.
- `numOfSeq`: Number of images in the whole time-lapse images.
- `numOfDiff`: Maximum available number of τ. `numOfSeq` $-$ `numOfDiff` images will be used in calculating average.
- `dx`: Pixel size in μm.
- `qmin`: Minimal q value, better to be set to be the twice of `qstep`, in μm^{-1}.
- `qstep`: Increment of q, better to be set as 2π/`dx`/`dimx`.
- `dt`: Time difference between two successive images, in second.
- `timeWindow`: Threshold of time, in second. Only data for $\tau <$ `timeWindow` will be used in fitting.
- `maxIter`: Maximum number of iterations in nonlinear curve fitting solver. (Actually, `maxIter` $+ 1$.)
- `$VariableName$Guess`: The initial guess of corresponding variable. Different model will make use of different variables.

After specifying these parameters, call

```
./config.sh
```

or

```
sh ./config.sh
```

to configure the solver. The script will generate `parameters.txt` in the current folder. `ddm` file should be in the same folder with `parameters.txt`, it will read `parameters.txt` at the of when running.

C.8 How to use

To begin data processing, users need to provide the prefix of the file name

```
./ddm$ModelName$ $FilePrefix$
```

The image files should be individual files named as from `$FilePrefix$1.tif` to, for instance, `$FilePrefix$4500.tif`.

Due to the incomplete support of the TIFF standard, only limited formats of TIFF file are supported in this program. If the file format is not supported, try use ImageJ to converge the file into text image. The program can read text image in ASCII code using

```
./ddm$ModelName$ simulationTXT
```

The files should be named as from `0.txt` to, for instance, `4499.txt` .

Reading ASCII text file is extremely slow. If possible, we recommend to use binary files, especially for the simulation data. The binary file should be named as `0` to, say, `4499` , then call

```
./ddm$ModelName$ simulation
```

Please notice the data type in binary files should be 64-bit double.

To save time, the program can run the fitting provided with the $g(q, \tau)$ data without processing the original files again. To do this, please keep the files `datag.txt`, `q.txt`, and `tau.txt`, then call

```
./ddm$ModelName$ recover
```

If doing so, the parameters from `dimx` to `dt` (refer to the list in Section C.7) in `config.sh` and `parameters.txt` will not be involved in the calculation.

C.9 Output files

After running, the following files will be generated.

- `datag.txt`: The $g(q, \tau)$ function. Each row denotes the $g(q, \tau)$ for a given q.
- `q.txt`: The list of q value.
- `tau.txt`: The list of τ value.
- `fitparafile.txt`: The result of the fitting. The order is same with the description in Section C.1, then followed with pairs of $A(q)$ and $B(q)$. The number of $A(q)$ and $B(q)$ depends on `nqcurve` parameter.
- `fiterrfile.txt`: The estimated error of the fitting, using the covariant matrix, but not alway precise. The order is same as `fitparafile.txt` .

- `status.txt`: The exiting status of each fitting.
- `statusq.txt`: Same with `status.txt`, but using only one integer number to denote the status, make it easy for other routine to read. 0 denotes the fitting is successful, while 11 means the solver reached the maximum number of iterations.

C.10 Protocol for *E. coli* measurement

The motion of *E. coli* has been summarized as run-and-tumble. We are able to measure the fraction of live *E. coli* α, mean velocity $\bar{v}$, standard deviation of velocity distribution σ, tumbling rate λ, and diffusion coefficient of Brownian motion D using the following protocol.

1. Do RDP fitting, to get an estimation of α and $\bar{v}$.

   This is necessary because RTDP fitting is not so stable, it requires a good guesses of α and $\bar{v}$. RDP fitting would be super fast (in a few seconds).

2. Do RTDP fitting, using α and $\bar{v}$ from step 1 as initial guess.

   This is necessary because RDP fitting always underestimates sigma. This would take usually 600 min to 1200 min CPU time in 3.33GHz Intel Xeon X5680 (depends on the noise level of the data).

3. Do RTDP fitting with fixed α, $\bar{v}$, σ, and D, to fit lambda only.

   This would take usuallty 100 min to 400 min CPU time in 3.33GHz Intel Xeon X5680 (depends on the noise level of the data).

4. Do RTDPTT fitting, using all results from step 2 and 3 as initial guess.

   This is necessary because RDP fitting always underestimates sigma. This would take usually 600 min to 1200 min CPU time in 3.33GHz Intel Xeon X5680 (depends on the noise level of the data).

5. Do RTDPTT fitting with fixed α, $\bar{v}$, σ, and D, to fit λ and δ only.

   This would take usuallty 100 min to 400 min CPU time in 3.33GHz Intel Xeon X5680 (depends on the noise level of the data).

In estimating the parameters, we suggest the following algorithm.

1. Calculate the mean m and the standard deviation s in the selected range of q (say, $[1.0\mu\mathrm{m}^{-1}, 1.5\mu\mathrm{m}^{-1}]$).

2. Exclude all the points that are out of the range $(m - 3s, m + 3s)$.
3. Return to step 1 until no point is excluded in step 2. Then the mean value is taken to be the estimation of the corresponding variables.

We included a Python script `protocol.py` to automatically do this protocol and plot the result. The parameters are the same with `config.sh` in Section C.7. Users just need to put all the `ddm` binaries in the same folder, and then call any one of the following

```
python ./protocol.py $FilePrefix$
python ./protocol.py simulation
python ./protocol.py simulationTXT
python ./protocol.py recover
```

The script was tested with Python 2.7 and 3.4, with `numpy`, `matplotlib`, and `subprocess` libraries.

Appendix D

Code reference

The structure of the code mainly falls into two classes, `ddm` and `NILT`. `ddm` class is the main class that encapsulates all the data and the functions used in ddm, and `NILT` class is to perform the numerical inverse Laplace transform. The class `NILT` is designed to be as independent as possible, but still have some coupling with the class `ddm`.

In this chapter we will list all the functions and structures in the code. As the code is controlled by several macro switch, I will also mention how they will change the code.

D.1 Macro switches

All the macro switches should be enabled or disabled in `parameters.h`.

- `ISFRD`
  `ISFRDP`
  `ISFRTD`
  `ISFRTDP`
  `ISFRTDPfix`
  `ISFRTDPTT`
  `ISFRTDPTTfix`

  Switch between different models.

- `NeedNumericalInverseLaplaceTransformation`

  Switch the numerical inverse Laplace transform routine. Automatically enabled in `RTD`, `RTDP`, and `RTDPfix` model.

- `IfComplexIntegration`

  Switch on this macro to enable the integration with v in numerical inverse Laplace transform routine. Automatically enabled in `RTDP` and `RTDPfix` model.

- `WINDOWS`

  To enable some platform specific code for Windows system.

D.2 Global constants

- `const long double pi`

  High precision of π.

- `const int numOfPara`

  Number of model parameters to be fitted, excluding $A(q)$ and $B(q)$.

- `const int M`

  Valid if using numerical inverse Laplace transform.

  The number of points in evaluating numerical integration in Weeks method (Chapter B.3). This also gives the number of terms in series expansion. But not all the terms in expansion is used in evaluating $f(q, \tau)$.

- `const double epsabs`

  `const double epsrel`

  `const int workspaceSize`

  Valid if integration with v is performed in numerical inverse Laplace transform.

  They are error tolerance and size of workspace (memory usage) in numerical integration. The GSL integration routine will give the integration result I if it satisfies

$$|\text{Error Estimation}| < \max\{\texttt{epsabs}, \texttt{epsrel}|I|\}\,. \tag{D.1}$$

D.3 Supporting structures

- `dataStruct`

  This struct is to pass parameters and data to GSL nonlinear fitting routine.

  The common members for all models are

  ```
  double* data
  double* tau
  double* q
  int num_fit
  int num_qCurve
  ```

  `data` is to store the data to be fitted. `tau` and `q` is the list of τ and q values. `num_fit` is the number of points to be fitted in each curve with specific q, and `num_qCurve` is the number of curves with different q.

  If the numerical inverse Laplace transform is to be performed, the structure may also have

  ```
  NILT* ISFILT
  NILT* dvbarISFILT
  NILT* dsigmaISFILT
  NILT* dDISFILT
  NILT* dlambdaISFILT
  NILT* dTTISFILT
  ```

  They are objectives for corresponding numerical inverse.

  If the model is `RTDPfix` or `RTDPTTsfix`, we include the following constants to speed up

  ```
  long double alpha
  long double D
  long double vbar
  long double sigma

  long double vbsigma2
  long double logfactor
  long double vb2sigma2
  long double cpsiz1
  long double vb2sigma3
  ```

- `typedef complex<long double> cpx`

  A rename of complex container.

- `warper`

  Valid if integration with v is performed in the numerical inverse Laplace transform.

  To bridge the GSL real function integration routine and the complex function integration, we use this structure to pass the complex function into interfaces `Re` and `Im` functions, which is then passed to GSL routine.

  `cpx z`

  The complex number to be evaluated.

  `cpx (*fun)(cpx z, long double* para, long double x)`

  The function pointer of complex function. The function should be in the form of $\texttt{fun}(z, x; \texttt{para})$ to perform the integration $\int \texttt{fun}(z, x; \texttt{para})dx$. All laplace transformed ISF should meet this interface.

  `long double* parameters`

  The parameters needed in evaluating ISF.

D.4 Functions

Some functions are defined outside the classes, because GSL will use their pointers, but member function of classes can't be treated as pointers.

- `int ISFfun(const gsl_vector* para, void* sdata, gsl_vector* y)`

  `int dISFfun(const gsl_vector* para, void* sdata, gsl_matrix* J)`

  The `ISFfun` and `dISFfun` calculated $w_i(f(q, \tau_i) - y_i)$ and the Jacobian of $f(q, \tau)$ with respect to parameters. The interface is arranged to follow the requirement of GSL routine. `para` is the list of parameters. `sdata` is actually a pointer to `dataStruct` to pass the data into the function. `y` and `J` is the corresponding output result.

- `double Re(double x, void* params)`

  `double Im(double x, void* params)`

  Valid if integration with v is performed in numerical inverse Laplace transform.

  They are to split the real and image part of integrating a complex function, and the interface is arranged to meet the requirement of GSL routine.

- `void readTiff(const string tifName, gsl_matrix* data, const int dimx, const int dimy)`

  `void readSim(const string fileName, gsl_matrix* data, const int dimx, const int dimy)`

  `void readSimTXT(const string fileName, gsl_matrix* data, const int dimx, const int dimy)`

  Reading data from TIFF files, binary files, or ASCII files. The size of the image need to be specified. The result will be stored in `data`. The support of TIFF files is not complete.

D.5 `ddm` class

D.5.1 Functions

- `ddm()`
  `~ddm()`

  Construction and deconstruction functions. Used for initialization and clean up.

- `void recover()`

  Reading $g(q, \tau)$, q, and τ data from corresponding files. the data will be stored in `datag`, `qabs`, and `tau`. `qsize` and `num_fit` will also be set correspondingly.

- `void readAndFFT(const string filePrefix)`

  Reading data from images or simulation result (binary or text), and do FFT on it. `filePrefix` is the prefix of the file name. Result will be stored in `imageSeqk`. `openMP` is used when loading and doing FFT.

- `void averSqrModTau()`

  Calculating $g(\boldsymbol{q}, l\Delta t) = \sum_k |I(\boldsymbol{q}, t_k + l\Delta t) - I(\boldsymbol{q}, t_k)|^2$. Results will be stored in `imagekDiff`. `tau` and `num_fit` will be set up correspondingly. `openMP` is used in the loop.

- `void aveQBilinear()`

Average the directions of q, by integrating bilinear interpolation function discussed in Section B.2. Result will be stored in `datag`. `qabs` and `qsize` will be set up correspondingly. `openMP` is used in the loop.

- `void fitting()`

  Fitting the data with ISF. The initialization of fitting solver will be different for different models. The fitted parameters will be stored in `fittedPara`. An estimation of fitting error is stored in `fittedErr`. `openMP` is used in the loop.

- `int norm0_rel_test(const gsl_vector * dx, const gsl_vector * x, double tol, double tole)`

  Test fitting result using 0-norm (maximum absolute value) of the relative step size dx/x. `tol` is the relative tolerance of error, `tole` is the absolute tolerance of error.

- `void printG()`

  Print unfitted data to files.

- `void printFit()`

  Print fitting result to files.

- `void printdebugM(gsl_matrix* m, const string filename)`

  Print arbitrary `gsl_matrix` object for debugging.

- `void cleanSeqk()`

  Free the memory used by `imageSeqk`. `openMP` is used in the loop.

- `void cleankDiff()`

  Free the memory used by `imagekDiff`. `openMP` is used in the loop.

D.5.2 Data members

- `vector<gsl_matrix_complex*> imageSeqk`

  Sequence of images.

- `vector<gsl_matrix*> imagekDiff`

  Sequence for storing image after FFT.

- `vector<double> qabs`

  Absolute value of q.

- `vector<double> tau`

  List of time points.

- `gsl_matrix* datag`

  $g(q, \tau)$ data matrix.

- `gsl_matrix* fittedPara`

  To store the fitting result.

- `gsl_matrix* fitErr`

  To store the fitting error.

- `int* status`

  Record of the status of fitting.

- `int* qIncreList`

  The increment of q for multiple $g(q, \tau)$ curves with different q.

- `double inipara[numOfPara]`

  Initial parameters.

- `double dx`

  Pixel size.

- `double dqx`

  `double dqy`

  q step after FFT.

- `double qmin`

  Minimal q value.

- `double qmax`

  Maximum possible value of q.

- `double qstep`

  The width of cirque when averaging the direction of q.

- `double dt`

  Time step.

- `double maxIter`

  Maximum iteration number in fitting.

- `double alphaGuess`
  `double DGuess`
  `double vbarGuess`
  `double lambdaGuess`
  `double ZGuess`
  `double sigmaGuess`

  The initial values to be passed from outside.

- `int qsize`

  Size of q array.

- `int OMP_NUM_THREADS`

  Number of threads for `openMP`.

- `int dimy`

  `int dimx`

  Size of the image.

- `int dimkx`

  `int dimky`

  Number of wavenumbers. (Size of q lattice.)

- `int numOfSeq`

  Number of total time points in experiment.

- `int numOfDiff`

  Size of τ array.

- `int numOfk`

  Number of points in q lattice.

- `int num_fit`

  Number of data points used in fitting.

- `int num_qCurve`

  Number of $g(q, \tau)$ curves with different q

- `double timeWindow`

  Only data for $\tau <$ `timeWindow` will be used in fitting.

D.6 `NILT` class

D.6.1 Functions

- `NILT(int omp_num)`
  `~NILT()`

  Construction and deconstruction functions. Used for initialization and clean up.

- `void NiLT_weeks(cpx (*fun)(cpx, const long double*), const long double* para)`

  Calculation of the coefficients in Laguerre polynomial expansion of the inverse Laplace transform of `fun`.

- `double clenshaw(long double t)`

  Clenshaw summation for function evaluation.

- `void weideman(long double alpha1, long double beta1, long double incre)`

  Estimate parameter b by Weideman's method, if the function is dominated by one pair of singularities $\alpha_1 \pm i\beta_1$. The σ is set to be close to the singularity ($\mathtt{sigma} = \mathtt{alpha1} + \mathtt{incre}$).

- `void weideman(long double alpha1, long double beta1, long double alpha2, long double beta2, long double incre)`

Estimate parameter b by Weideman's method, if the function is dominated by two pair of singularities $\alpha_{1,2} \pm i\beta_{1,2}$. The sigma is set to be close to the singularity with largest real part, which should set to be `alpha2` (`sigma` $=$ `alpha2` $+$ `incre`).

- `void NiLT_weeks(long double* para)`

  Valid if integration with v is performed in numerical inverse Laplace transform.

  Calculation of the coefficients in Laguerre polynomial expansion of inverse transform of `invfun`, with integration with v.

- `cpx invfun(cpx x, long double* para)`

  Valid if integration with v is performed in numerical inverse Laplace transform.

  Numerical evaluation of function to be inverse transformed. Including integration of a complex function in v. The complex function to be integrated is the function pointer in `cfun` which is a pointer of `warper` type.

D.6.2 Data members

The class is usually defined outside the parallel part of the code, to avoid allocate memory at every iteration. But this will causs memory conflict in shared memory model like `openMP`. So everything should be kept as a list with number of elements equals to number of threads, and different thread uses different element.

- `int OMP_NUM_THREADS`

  Number of threads.

- `fftwl_plan* integration`
  `fftwl_complex** fftwIn`
  `fftwl_complex** fftwOut`

  FFTW stuff used in calculating numerical integration. Input and output matrix is stored in `fftwIn` and `fftwOut`.

- `vector<long double>* CoeA`

  Coefficients in Laguerre polynomial expansion.

- `long double* b`

  `long double* sigma`
  `long double* b2`
  `long double* sigmab`

  Important paramters in solver, and some constant for convenient. `b2` $= 2b$, `sigmab` $= \sigma - b$.

- `gsl_function* pRe`
  `gsl_function* pIm`

  Valid if integration with v is performed in numerical inverse Laplace transform.

  Pointers of `Re` and `Im` function.

- `gsl_integration_cquad_workspace** workspace`

  Valid if integration with v is performed in numerical inverse Laplace transform.

  The workspace used by GSL `cquad` routine.

- `warper* cfun`

  Valid if integration with v is performed in numerical inverse Laplace transform.

  To make the iterface between real itegration and complex evaluation of functions, we use this structure to present the function that is to be integrated.

Bibliography

[1] Francis HC Crick. On protein synthesis. *Symposia of the Society for Experimental Biology*, 1958.

[2] Bruce Alberts, Alexander Juhnson, Julian Lewis, Martin Raff, Keith Roberts, and Peter Walter. *Molecular Biology of the Cell*. Garland Science, 5th edition, 2007.

[3] *Human Protein Reference Database*.

[4] *Pathway Commons*.

[5] *Reactome*.

[6] Uri Alon. *An Introduction to Systems Biology: Design Principles of Biological Circuits*. Chapman and Hall/CRC, 1st edition, 2006.

[7] Piers J Ingram, Michael Stumpf, and Jaroslav Stark. Network motifs: structure does not determine function. *BMC Genomics*, 7(1):108, 2006.

[8] Wenzhe Ma, Ala Trusina, El-Samad, Hana, Wendell A Lim, and Chao Tang. Defining network topologies that can achieve biochemical adaptation. *Cell*, 138(4):760–773, 2009.

[9] Angela H Chau, Jessica M Walter, Jaline Gerardin, Chao Tang, and Wendell A Lim. Designing synthetic regulatory networks capable of self-organizing cell polarization. *Cell*, 151(2):320–332, 2012.

[10] Wendell A Lim, Connie M Lee, and Chao Tang. Design principles of regulatory networks: searching for the molecular algorithms of the cell. *Molecular Cell*, 49(2):202–212, 2013.

[11] S Kim, H Li, ER Dougherty, N Cao, and Y Chen. Can markov chain models mimic biological regulation? *Journal of Biological*, 10(4):337–357, 2002.

[12] Arup K Chakraborty and Andrej Komrlj. Statistical mechanical concepts

in immunology. *Annual Review of Physical Chemistry*, 61:283–303, 2010.

[13] Guangwei Si, Tailin Wu, Qi Ouyang, and Yuhai Tu. Pathway-Based Mean-Field model for escherichia coli chemotaxis. *Physical Review Letters*, 109(4), 2012.

[14] D. Ewen Cameron, Caleb J. Bashor, and James J. Collins. A brief history of synthetic biology. *Nature Reviews Microbiology*, 12(5):381–390, 5 2014.

[15] George M. Church, Michael B. Elowitz, Christina D. Smolke, Christopher A. Voigt, and Ron Weiss. Realizing the potential of synthetic biology. *Nature Reviews Molecular Cell Biology*, 15(4):289–294, 4 2014.

[16] Nikolce Gjorevski and Celeste M Nelson. Integrated morphodynamic signalling of the mammary gland. *Nature Reviews Molecular Cell Biology*, 12(9):581–593, 2011.

[17] Celeste M Nelson and Jason P Gleghorn. Sculpting organs: mechanical regulation of tissue development. *Annual Review of Biomedical Engineering*, 14:129–154, 2012.

[18] M Cross and P Hohenberg. Pattern formation outside of equilibrium. *Reviews of Modern Physics*, 65, 1993.

[19] AM Turing. The chemical basis of morphogenesis. *Philosophical Transactions of the Royal Society B*, 237, 1952.

[20] Rushikesh Sheth, Luciano Marcon, Félix M Bastida, Marisa Junco, Laura Quintana, Randall Dahn, Marie Kmita, James Sharpe, and Maria A Ros. Hox genes regulate digit patterning by controlling the wavelength of a Turing-Type mechanism. *Science*, 338(6113):1476–1480, 2012.

[21] M E Cates, D Marenduzzo, I Pagonabarraga, and J Tailleur. Arrested phase separation in reproducing bacteria creates a generic route to pattern formation. *Proceedings of the National Academy of Sciences of the United States of America*, 107(26):11715–11720, 2010.

[22] Chenli Liu, Xiongfei Fu, Lizhong Liu, Xiaojing Ren, Carlos K Chau, Sihong Li, Lu Xiang, Hualing Zeng, Guanhua Chen, Lei-Han H Tang, Peter Lenz, Xiaodong Cui, Wei Huang, Terence Hwa, and Jian-Dong D Huang. Sequential establishment of stripe patterns in an expanding cell population. *Science*, 334(6053):238–241, 2011.

[23] Hiroaki Yamanaka and Shigeru Kondo. In vitro analysis suggests that difference in cell movement during direct interaction can generate various pigment patterns in vivo. *Proceedings of the National Academy of Sciences of the United States of America*, 111(5):1867–1872, 2014.

[24] HC Berg and DA Brown. Chemotaxis in escherichia coli analysed by three-dimensional tracking. *Nature*, 1972.

[25] Jonathan Saragosti, Pascal Silberzan, and Axel Buguin. Modeling e. coli tumbles by rotational diffusion. implications for chemotaxis. *PLoS ONE*, 7(4), 2012.

[26] Gerald L Hazelbauer. Bacterial chemotaxis: the early years of molecular studies. *Annual Review of Microbiology*, 66:285–303, 2012.

[27] Howard C Berg. The rotary motor of bacterial flagella. *Annual Review of Biochemistry*, 72:19–54, 2003.

[28] George H Wadhams and Judith P Armitage. Making sense of it all: bacterial chemotaxis. *Nature Reviews Molecular Cell Biology*, 2004.

[29] Steven L Porter, George H Wadhams, and Judith P Armitage. Signal processing in complex chemotaxis pathways. *Nature Reviews. Microbiology*, 9(3):153–165, 2011.

[30] Yuhai Tu. Quantitative modeling of bacterial chemotaxis: Signal amplification and accurate adaptation. *Biophysics*, 42(1):337–359, 2013.

[31] Khuloud Jaqaman, Dinah Loerke, Marcel Mettlen, Hirotaka Kuwata, Sergio Grinstein, Sandra L Schmid, and Gaudenz Danuser. Robust single-particle tracking in live-cell time-lapse sequences. *Nature Methods*, 5(8):695–702, 2008.

[32] Roberto Cerbino and Veronique Trappe. Differential dynamic microscopy: probing wave vector dependent dynamics with a microscope. *Physical Review Letters*, 100(18):188102, 2008.

[33] Laurence G Wilson, Vincent A Martinez, Jana Schwarz-Linek, and Julien Tailleur. Differential dynamic microscopy of bacterial motility. *Physical Review Letters*, 2011.

[34] Vincent A Martinez, Rut Besseling, Ottavio A Croze, Julien Tailleur, Mathias Reufer, Jana Schwarz-Linek, Laurence G Wilson, Martin A Bees, and Wilson CK Poon. Differential dynamic microscopy: a high-throughput method for characterizing the motility of microorganisms. *Biophysical Journal*, 103(8):1637–1647, 2012.

[35] ME Cates and J Tailleur. When are active brownian particles and run-and-tumble particles equivalent? consequences for motility-induced phase separation. *EPL (Europhysics Letters)*, 101(2), 2013.

[36] AP Solon, ME Cates, and J Tailleur. Active brownian particles and run-

and-tumble particles: A comparative study. *The European Physical Journal Special Topics*, 224(7):1231–1262, 2015.

[37] MJ Schnitzer. Theory of continuum random walks and application to chemotaxis. *Physical Review. E, Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics*, 48(4):2553–2568, 1993.

[38] K Martens, L Angelani, R Di Leonardo, and L Bocquet. Probability distributions for the run-and-tumble bacterial dynamics: an analogy to the lorentz model. *The European Physical Journal. E, Soft Matter*, 35(9):84, 2012.

[39] Kevin S Brown and James P Sethna. Statistical mechanical approaches to models with many poorly known parameters. *Physical Review. E, Statistical, Nonlinear, and Soft Matter Physics*, 68(2 Pt 1):021904, 2003.

[40] K S Brown, C C Hill, G A Calero, C R Myers, K H Lee, J P Sethna, and R A Cerione. The statistical mechanics of complex signaling networks: nerve growth factor signaling. *Physical Biology*, 1(3):184, 2004.

[41] Joshua J Waterfall, Fergal P Casey, Ryan N Gutenkunst, Kevin S Brown, Christopher R Myers, Piet W Brouwer, Veit Elser, and James P Sethna. Sloppy-Model universality class and the vandermonde matrix. *Physical Review Letters*, 97(15), 2006.

[42] Ryan N Gutenkunst, Joshua J Waterfall, Fergal P Casey, Kevin S Brown, Christopher R Myers, and James P Sethna. Universally sloppy parameter sensitivities in systems biology models. *PLoS Computational Biology*, 3(10), 2007.

[43] F Detcheverry. Non-Poissonian run-and-turn motions. *EPL (Europhysics Letters)*, 111(6):60002, 2015.

[44] Jean Guenet and Cyrille Rochas. Agarose sols and gels revisited. *Wiley Online Library*, 242(1):65–70, 2006.

[45] AJ Wolfe and HC Berg. Migration of bacteria in semisolid agar. *Proceedings of the National Academy of Sciences of the United States of America*, 86:6973–6977, 1989.

[46] This idea is from Dr. Julien Tailleur via private communication.

[47] *GNU Scientific Library Reference Manual.*

[48] William T Weeks. Numerical inversion of laplace transforms using laguerre functions. *Journal of the ACM (JACM)*, 13(3):419–429, 1966.

[49] Brian Davies. *Intrgral Transforms and Their Applications.* Springer, 3rd

edition, 2002.

[50] J. A. C. Weideman. Algorithms for parameter selection in the weeks method for inverting the laplace transform. *SIAM Journal on Scientific Computing*, 21(1):111–128, 1999.