Recent modelling frameworks for systems of interacting particles

Benjamin Franz
St Anne's College
University of Oxford

A thesis submitted for the degree of
Doctor of Philosophy
Hilary 2014

To C.

Acknowledgements

First and foremost I would like to express my gratitude to my supervisor Radek Erban, who was always great at guiding my research as well as discussing opportunities for conferences, collaborations and interesting projects with me. I have been extremely lucky to have been given a great number of opportunities throughout my DPhil, including getting experience as a co-supervisor myself and developing my own project ideas, especially including swarm robotic experiments. One of the greatest lessons I have learned from Radek is never to stop and be happy with your results, but always to ask what is the next step.

I would also like to thank my collaborators Jon Chapman, Mark Flegg, Chuan Xue, Kevin Painter, Jan Haskovec, Kit Yates and Jake Taylor-King for an excellent collaboration throughout and for interesting discussions leading to our joint papers. A particular thanks goes to Jake, who performed the first batch of robotic experiments and who gave me the chance to be his supervisor. Additionally, I have had countless interesting discussions at conferences or with visitors in Oxford, made possible by the vibrant research environment in and around OCCAM.

The research leading to this thesis has received funding from the European Research Council under the *European Community's* Seventh Framework Programme *(FP7/2007-2013)* / ERC *grant agreement* No. 239870. Additionally, this work was partly supported by Award No KUK-C1-013-04, made by King Abdullah University of Science and Technology (KAUST). The robot experiments were funded by a Royal Society equipment grant.

On a more personal note, I would like to thank my colleagues in OCCAM and the WCMB who were always ready to help. A special thanks goes to Jean Charles for sharing his excellent knowledge and some of his code for numerical solutions of PDEs. I would like to thank my parents for their continued support. Finally, I would like to thank Chloé for showing me that there is more to life than living in the Oxford bubble.

Abstract

In this thesis we study three different modelling frameworks for biological systems of dispersal and combinations thereof. The three frameworks involved are individual-based models, group-level models in the form of partial differential equations (PDEs) and robot swarms. In the first two chapters of the thesis, we present ways of coupling individual based models with PDEs in so-called hybrid models, with the aim of achieving improved performance of simulations. Two classes of such hybrid models are discussed that allow an efficient simulation of multi-species systems of dispersal with reactions, but involve individual resolution for certain species and in certain parts of a computational domain if desired. We generally consider two types of example systems: bacterial chemotaxis and reaction-diffusion systems, and present results in the respective application area as well as general methods. The third chapter of this thesis introduces swarm robotic experiments as an additional tool to study systems of dispersal. In general, those experiments can be used to mimic animal behaviour and to study the impact of local interactions on the group-level dynamics. We concentrate on a target finding problem for groups of robots. We present how PDE descriptions can be adjusted to incorporate the finite turning times observed in the robotic system and that the adjusted models match well with experimental data. In the fourth and last chapter, we consider interactions between robots in the form of hard-sphere collisions and again derive adjusted PDE descriptions. We show that collisions have a significant impact on the speed with which the group spreads across a domain. Throughout these two chapters, we apply a combination of experiments, individual-based simulations and PDE descriptions to improve our understanding of interactions in systems of dispersal.

Statement of originality

Chapter 1 contains a review of the literature on modelling of systems of dispersal as well as the introduction of well established random walks. The latter parts present three modelling frameworks for systems of dispersal. This chapter is based on the invited book chapter [66] that is co-authored with my supervisor Dr. Radek Erban and of which I am the first author.

Chapter 2 is mainly based on two pieces of literature of which I am the first author in both cases: the invited book chapter [66] and the publication [69]. For the book chapter I performed all the simulations myself and developed the mathematics in collaboration with my supervisor. For the publication [69] I performed all simulations of hybrid models, whilst our collaborator Dr. Kevin Painter performed numerical solutions of the PDE models. The travelling wave analysis was performed in close collaboration with Dr. Chuan Xue and Dr. Radek Erban.

Chapter 3 is strongly based on the published article [67] of which I am the first author. The ideas leading to the algorithm were suggested by myself and refined in discussions with Prof. Jon Chapman, Dr. Radek Erban and Dr. Mark Flegg. I wrote the code for the simulations and performed those simulations myself.

Chapter 4 presents different results regarding swarm robotics experiments. Sections 4.1 and 4.2 are solely my own work. Section 4.3 is based on the submitted article [157] of which I am the second author. The first author, Jake Taylor-King, was an MSc student that I co-supervised together with Dr. Radek Erban and Dr. Christian Yates. He performed the experiments without signals and we developed the introduction of turning delays in collaboration and parts of it were also presented in his MSc dissertation.

Chapter 5 regarding the effect of hard-sphere collisions is my own work.

Throughout the thesis wherever results or methods from the literature have been used, they have been identified and attributed.

Contents

Chapter 1

Introduction

1.1 Motivation

From small packs of wolves [20] to swarms of birds [123] or populations of large numbers of bacteria [95], the range of biological systems displaying dispersal is vast. In this thesis, we examine various systems with the principal aim of studying and improving different modelling frameworks.

The literature knows two classical modelling approaches for biological systems of dispersal – agent-based models and continuum models. In agent-based models, the behaviour of individuals is prescribed by an evolution update rule that can be given in the form of an algorithm or differential equations for the individual behaviour. Examples in the literature include the modelling of humans in panic situations [135], collective nest choice in ants [138] and swarming behaviour in fish schools [38, 37].

On the other hand, continuum models do not consider the behaviour of the individual, but use group-level statistics as their indicator. Most commonly this type of model uses partial differential equations (PDEs) in order to model the evolution of concentrations of particles. Countless examples of applications for continuum techniques exist, to model e.g. bacterial chemotaxis [95], the spread of advantageous genes [60] or pattern formation in the brain [61, 122].

In this thesis, we investigate two approaches to model biological systems of dispersal that have recently gained importance in the literature – hybrid models and swarm robotic experiments. Hybrid models seek to combine advantages of PDEs and agent-based models through the exploitation of inhomogeneities inherent in the studied system. Such models have already been used successfully in studies of carcinogenesis [6, 75], the analysis of forest health [104] and to model the spread of epidemics [15]. Simply speaking, the aim of hybrid models is to reduce the computational complexity of an agent-based model by approximating parts of the system

through PDEs, thereby allowing the use of stochastic models on systems that are intractable for purely agent-based models.

The second approach presented in this thesis are swarm robotic experiments. The main idea is to implement an individual-based algorithm on each of the robots and to analyse the emerging behaviour of the swarm. The major advantage of this approach is that physical uncertainties and noises are inherently present in the system. Additionally, by implementing a proposed model onto robots, one can verify that no unrealistic assumptions about communications between individuals and knowledge of other individuals' internal dynamics have been made during the modelling process. One can also imagine that the imitation of biological group behaviour can motivate and improve the use of robot swarms in various application areas, including the detection of mines [100] and the confrontation of oil spills [92].

Throughout this thesis we introduce the approaches with a certain target system in mind, but we keep explanations purposely general, because the concepts are more important than the specific application. All the modelling approaches presented can easily be adjusted to a wide range of systems of dispersal and different approaches can be combined.

1.2 Outline of the thesis

In the remainder of this chapter, we present the mathematical background that is extensively used throughout this thesis and introduce mathematical frameworks for agent-based, continuum and hybrid models as well as swarm robotic systems. The description of these frameworks mainly stems from the invited book chapter [66] that appeared in the book *Dispersal, individual movement and spatial ecology: A mathematical perspective* published by Springer in 2013.

Chapter 2 presents the application of the hybrid modelling approach to the biological example of bacterial chemotaxis. We start by describing the biological background of the system, before going into more detail about the conception of hybrid models for this application. We show that the hybrid models lead to non-trivial results in comparison to the respective continuum models. In the next step, we introduce the possibility of death and proliferation into the hybrid model and show how this influences the comparison to PDE models. This chapter is based on results shown in [66] as well as the article [69] that has been published in the *Bulletin of Mathematical Biology*.

Chapter 3 discusses a second type of hybrid model that can be used to efficiently simulate stochastic reaction-diffusion systems by exploiting spatial inhomogeneities. We present a *PDE assisted Brownian dynamics* algorithm that combines Brownian dynamics simulations with reaction-diffusion PDEs. We show that the algorithm provides correct mean values and variances of concentrations inside the particle regime and apply it to a number of biological examples. This chapter is based on the article [67] that has been published in *SIAM Journal on Applied Mathematics*.

Chapter 4 investigates the possibility of using swarm robotics experiments to gain insight into systems of dispersal. We study a target finding problem using a combination of experiments, individual-based simulations and continuum approximations. The experiments begin with a classical velocity jump model, but we soon show that the classical velocity jump process is not sufficient to describe the behaviour of the group as effects like finite turning times have an influence on the outcome. By repeating the experiments on a surface with a grey-black colour gradient, we introduce an external signal that can be detected by the robots. We mimic the behaviour of bacterial chemotaxis studied in Chapter 2 in order to improve the target finding capacity of the robots. Parts of this chapter have been accepted for publication in *IMA Journal of Applied Mathematics* [157].

Chapter 5 analyses the effect caused by hard-sphere interactions in velocity jump processes. In particular, we study the influence of a certain type of collision on the diffusivity of the group of particles using a mix of simulations and analytic results. We derive corrections to the classical velocity jump models and show that those match well with simulation results. The results in this chapter have been accepted for publication in *Physical Review E* [68].

Chapter 6 finally discusses the presented results in combination with the literature and draws a bigger picture about the modelling of systems of dispersal. Based on this, we discuss areas for future research and application areas of the various approaches presented.

1.3 Mathematical background

This section gives an overview over two classical approaches for the modelling of random dispersal or diffusion – Brownian dynamics (BD) and velocity jump (VJ) processes.

1.3.1 Brownian dynamics simulations

In this section we review the standard definitions leading to Brownian dynamics simulations that are used extensively throughout the thesis. Those definitions can be found in more detail in various text books, e.g. [9, 70, 78, 161]. Let us start by introducing the definition of a *Wiener process*.

Definition 1.1. *The random process W_t is called a* Wiener process *if the following four conditions are satisfied*

1. $W_0 = 0$.
2. *For $0 \leq t < s$, the random increment $W_s - W_t$ is Gaussian with zero mean and variance $t - s$, i.e. $W_s - W_t \sim \mathcal{N}(0; t - s)$.*
3. *For given times $0 \leq t_0 < t_1 < \cdots < t_n$, the increments $W_{t_1} - W_{t_0}, W_{t_2} - W_{t_1}, \ldots, W_{t_n} - W_{t_{n-1}}$ are independent random variables.*
4. *The function $t \mapsto W_t$ is almost surely everywhere continuous.*

Let us now define an Itô stochastic integral.

Definition 1.2. *Let $f : [0, t] \mapsto \mathbb{R}$ be an arbitrary function. Then, we call the random variable X_t the* Itô integral *of function f with respect to the Wiener process W_t denoted through*

$$X_t = \int_0^t f(\tau)\, \mathrm{d}W_\tau \,,$$

if the equation

$$X_t = \lim_{\Delta t \to 0} \sum_{j=1}^{n} f(t_{j-1})(W_{t_j} - W_{t_{j-1}}) \,,$$

holds for all series of time steps $0 = t_0 < t_1 < t_2 < \cdots < t_n = t$ with

$$\Delta t = \sup\{t_1 - t_0, t_2 - t_1, \ldots, t_n - t_{n-1}\} \,.$$

In practice we often write X_t in the form of the Itô differential equation

$$\mathrm{d}X_t = f(t)\, \mathrm{d}W_t \,.$$

Defining the *drift term* $F : [0, t] \times \mathbb{R} \mapsto \mathbb{R}$ and the *diffusion term* $D : [0, t] \times \mathbb{R} \mapsto \mathbb{R}$, we can formulate the general stochastic differential equation (SDE) that we will also call the *Brownian walk* as

$$\mathrm{d}X_t = F(t, X_t)\, \mathrm{d}t + \sqrt{2D(t, X_t)}\, \mathrm{d}W_t \,. \tag{1.1}$$

We can formally write the solution X_t of (1.1) as

$$X_t = \int_0^t F(\tau, X_\tau)\,\mathrm{d}\tau + \int_0^t \sqrt{2D(\tau, X_\tau)}\,\mathrm{d}W_\tau + X_0\,,$$

where X_0 is the initial value of the process X_t. If $D(t,x) \equiv D$ is a constant and $F(t,x) \equiv F(t)$ is independent of X_t, this simplifies to

$$X_t = \int_0^t F(\tau)\,\mathrm{d}\tau + \sqrt{2D}\,W_\tau \sim \mathcal{N}\left(\int_0^t F(\tau)\,\mathrm{d}\tau + X_0, 2Dt\right)\,.$$

In order to numerically solve the SDE (1.1), we introduce the Euler-Maruyama approximation defined through [99]

$$Y_{t+\Delta t} = Y_t + F(t, Y_t)\Delta t + \sqrt{2D(t, Y_t)\Delta t}\,\xi\,, \tag{1.2}$$

where $\xi \sim \mathcal{N}(0,1)$, i.e. ξ is a normally distributed random variable. Note, that if $F(t,x) \equiv F$ and $D(t,x) \equiv D$ are constants and $Y_t = X_t$, then $Y_{t+\Delta t}$ is an exact sample from the random variable X_t for arbitrary $\Delta t > 0$. Additionally, the Euler-Maruyama method becomes exact in the limit $\Delta t \to 0$. Throughout the remainder of this thesis, we will generally interpret the random variable X_t as (one component of) the position of a given particle inside a domain $\Omega \subset \mathbb{R}^d$, $d = 1, 2, 3$ and we will repeatedly study different copies of this process that can influence each others drift and diffusion terms. Let us now see how the SDE (1.1) can be analysed using the probability density function of the random variable X_t and its development over time.

Fokker-Planck equation

The following derivation is adapted from the lecture notes [45]. Assuming that $t > t_0$, we define by $p(t, x|t_0, x_0)\,\mathrm{d}x$ the probability[1] that $X_t \in [x, x+\mathrm{d}x)$ given that $X_{t_0} = x_0$. If we take an intermediate time $s \in (t_0, t)$, we can formulate the so-called *Chapman-Kolmogorov* equation

$$p(t, x|t_0, x_0) = \int_{\mathbb{R}} p(s, y|t_0, x_0)\,p(t, x|s, y)\,\mathrm{d}y\,,$$

where we use the fact that the process X_t is a Markov process, i.e. that it is memoryless [161]. Using $t - s = \Delta t$, this can be rewritten as

$$p(s + \Delta t, x|t_0, x_0) = \int_{\mathbb{R}} p(s, y|t_0, x_0)\,p(s + \Delta t, x|s, y)\,\mathrm{d}y\,.$$

[1] Being a probability distribution, for every $t > t_0$, we have $\int_{\mathbb{R}} p(t, x|t_0, x_0)\,\mathrm{d}x = 1$.

Multiplying both sides with a smooth test function $\varphi : \mathbb{R} \mapsto \mathbb{R}$ and integrating with respect to $x \in \mathbb{R}$, we obtain

$$\int_{\mathbb{R}} p(s+\Delta t, x)\, \varphi(x)\, \mathrm{d}x = \int_{\mathbb{R}} \int_{\mathbb{R}} p(s+\Delta t, x|s, y)\, \varphi(x)\, \mathrm{d}x\, p(s, y)\, \mathrm{d}y\,,$$

where we drop the conditional t_0, x_0 if appropriate to improve readability. We change the name of the integration variable on the left-hand side and Taylor expand the smooth function $\varphi(x)$ in y to get

$$\begin{aligned}\int_{\mathbb{R}} p(s+\Delta t, y)\, \varphi(y)\, \mathrm{d}y = &\int_{\mathbb{R}} \int_{\mathbb{R}} p(s+\Delta t, x|s, y)\, \mathrm{d}x\, \varphi(y)\, p(s, y)\, \mathrm{d}y \\ &+ \int_{\mathbb{R}} \int_{\mathbb{R}} p(s+\Delta t, x|s, y)(x-y)\, \mathrm{d}x\, \varphi'(y)\, p(s, y)\, \mathrm{d}y \\ &+ \frac{1}{2} \int_{\mathbb{R}} \int_{\mathbb{R}} p(s+\Delta t, x|s, y)(x-y)^2\, \mathrm{d}x\, \varphi''(y)\, p(s, y)\, \mathrm{d}y + \dots\,.\end{aligned}$$

We can evaluate the integrals individually as follows

$$\begin{aligned}\int_{\mathbb{R}} p(s+\Delta t, x|s, y)\, \mathrm{d}x &= 1\,, \\ \int_{\mathbb{R}} p(s+\Delta t, x|s, y)(x-y)\, \mathrm{d}x &= \mathbb{E}(X_{t+\Delta t} - X_t) \qquad = \Delta t F(s, x)\,, \\ \int_{\mathbb{R}} p(s+\Delta t, x|s, y)(x-y)^2\, \mathrm{d}x &= \mathbb{E}\left((X_{t+\Delta t} - X_t)^2\right) = 2\Delta t D(s, x) + \mathcal{O}\left((\Delta t)^2\right)\,,\end{aligned}$$

where the first integral is part of the definition of a probability density function and the other two follow directly from the Euler-Maruyama method as seen in (1.2) and the fact that $Y_t = X_t$ for $\Delta t \to 0$. Note that all other terms from the Taylor expansion result in terms of $\mathcal{O}\left((\Delta t)^2\right)$. Applying integration by parts to the resulting terms, dividing by Δt and considering the limit $\Delta t \to 0$, we obtain the *Fokker-Planck* equation for the SDE (1.1) as

$$\frac{\partial p(t, x|t_0, x_0)}{\partial t} = -\frac{\partial}{\partial x}\big(F(t, x)\, p(t, x|t_0, x_0)\big) + \frac{\partial^2}{\partial x^2}\big(D(t, x)\, p(t, x|t_0, x_0)\big)\,. \tag{1.3}$$

The Fokker-Planck equation provides a way of exactly deriving the probability distribution of a Brownian process. On the other hand, one can also interpret $Np(t, x)$ as an approximate particle density if one considers a system of $N \gg 1$ independent identical Brownian processes of the form (1.1). It is also possible to add boundary conditions to the Fokker-Planck equation (1.3). Consider for example that the process X_t as defined in (1.1) is confined to the interval $[a, b]$ with an adsorbing boundary at $x = a$ and a reflective boundary at $x = b$, the boundary conditions for (1.3) take the form

$$p(t, a) = 0\,, \qquad \frac{\partial}{\partial x}\big(D(t, b)\, p(t, b)\big) = F(t, b)\,,$$

for $t > 0$. Similarly, if X_{t_0} is sampled from the initial distribution $p_0(x)$, the initial condition for (1.3) is

$$p(t_0, x) = p_0(x)\,.$$

Backwards Kolmogorov equation

Using a similar approach to the derivation of the Fokker-Planck equation one can also derive the backwards Kolmogorov equation that takes the form [45]

$$-\frac{\partial p(t, x|t_0, x_0)}{\partial t_0} = F(t_0, x_0)\frac{\partial p(t, x|t_0, x_0)}{\partial x_0} + D(t_0, x_0)\frac{\partial^2 p(t, x|t_0, x_0)}{\partial x_0^2}\,. \tag{1.4}$$

In the remainder of this derivation, we assume that the functions F and D are independent of time. We also maintain the example of a particle moving inside the domain $[a, b]$ with an adsorbing boundary condition at $x = a$ and a reflective boundary at $x = b$. The derivation is again transcribed from [45]. Let us now define the probability that a particle starting at $x_0 \in (a, b]$ at time 0 is still in the domain at time t as

$$h(t, x_0) = \int_a^b p(t, x|0, x_0)\,\mathrm{d}x = \int_a^b p(0, x|-t, x_0)\,\mathrm{d}x\,,$$

where we use the time independence of the probability density p. Switching the sign of integration on the left-hand side of (1.4) in order to evaluate it for $(0, x|-t, x_0)$ we obtain

$$\frac{\partial p(0, x|-t, x_0)}{\partial t} = F(x_0)\frac{\partial p(0, x|-t, x_0)}{\partial x_0} + D(x_0)\frac{\partial^2 p(0, x|-t, x_0)}{\partial x_0^2}\,.$$

We can now integrate this equation with respect to $x \in [a, b]$ to obtain an equation for $h(t, x_0)$

$$\frac{\partial h(t, x_0)}{\partial t} = F(x_0)\frac{\partial h(t, x_0)}{\partial x_0} + D(x_0)\frac{\partial^2 h(t, x_0)}{\partial x_0^2}\,. \tag{1.5}$$

Seeing that $h(t, x_0)$ represents the probability of a particle that started at position x_0 to still be inside the domain $(a, b]$ at time t, we can define the mean time of exit $\tau(x_0)$ from the domain[2] through

$$\tau(x_0) = \int_0^\infty t \lim_{\Delta t \to 0} \frac{h(t, x_0) - h(t + \Delta t, x_0)}{\Delta t}\,\mathrm{d}t = -\int_0^\infty t\frac{\partial h(t, x_0)}{\partial t}\,\mathrm{d}t = \int_0^\infty h(x_0, t)\,\mathrm{d}t\,,$$

where we use the definition of the derivative in the first step and integration by parts in the second[3]. Integrating (1.5) with respect to $t \in [0, \infty)$ and using the fact that

[2] We will repeatedly use the terms *mean exit time* and *mean first passage time* for this.

[3] Assuming $\lim_{t\to\infty} h(t, x_0) = 0$, i.e. the probability of eventually leaving the domain is 1.

$h(0, x_0) = 1$, we obtain the mean first passage time equation

$$-1 = F(x_0)\frac{\partial \tau(x_0)}{\partial x_0} + D(x_0)\frac{\partial^2 \tau(x_0)}{\partial x_0^2}\,.$$

This equation is valid for all $x_0 \in (a, b)$. The boundary conditions are

$$\tau(a) = 0\,, \qquad \frac{\partial}{\partial x_0}\tau(b) = 0\,.$$

Brownian dynamics and discrete random walks

Both, the Fokker-Planck equation (1.3) and the backwards Kolmogorov equation (1.4) are often derived from discrete random walk models [35]. Let us therefore assume a one-dimensional random walk on lattice points $r_l = l\,\Delta x$, $l \in \mathbb{Z}$. We denote the number of particles at lattice point l and time t by $P_l(t) \in \mathbb{N}$. In each time step Δt, every particle can perform one of three possible moves: (a) move to the neighbouring lattice point on the right with probability $\sigma/2$ (b) move to the neighbouring lattice point on the left with probability $\sigma/2$ (c) stay at its current lattice point with probability $1 - \sigma$, where $\sigma = \Delta t\, D/(\Delta x)^2$. The update equation for P_l then takes the form:

$$P_l(t + \Delta t) = (1 - \sigma)P_l(t) + \frac{\sigma}{2}\left(P_{l+1}(t) + P_{l-1}(t)\right)\,.$$

Substituting $\sigma = \Delta t\, D/(\Delta x)^2$, this equation can be written as

$$\frac{P_l(t + \Delta t) - P_l(t)}{\Delta t} = D\,\frac{P_{l+1}(t) + P_{l-1}(t) - 2P_l(t)}{2\,(\Delta x)^2}\,.$$

Taking the limit $\Delta t \to 0$, we obtain

$$\frac{\mathrm{d}P_l(t)}{\mathrm{d}t} = D\,\frac{P_{l+1}(t) + P_{l-1}(t) - 2P_l(t)}{2\,(\Delta x)^2}\,.$$

This equation represents a discretised version of the diffusion equation

$$\frac{\partial P(t, x)}{\partial t} = D\,\frac{\partial P(t, x)}{\partial x^2}\,,$$

showing the correspondence between a random walk that is discrete in time and space and the space-time continuous Brownian walk.

Brownian dynamics in higher dimensions

Biological systems of dispersal are typically observed in a two or three dimensional domain, whereas we have thus far introduced Brownian dynamics only in $d = 1$ dimensions. Let us therefore denote a Wiener process $\mathbf{W}_t$ in $d = 2, 3$ dimensions as the vector of d independent one-dimensional Wiener processes. The SDE (1.1) in d dimensions then takes the form

$$\mathrm{d}\mathbf{X}_t = \mathbf{F}(t, \mathbf{X}_t)\,\mathrm{d}t + \hat{\mathbf{D}}(t, \mathbf{X}_t)\,\mathrm{d}\mathbf{W}_t\,,$$

where $\mathbf{F} \in \mathbb{R}^d$ and $\hat{\mathbf{D}} \in \mathbb{R}^{d\times d}$ represent the forcing and diffusive terms, respectively. The Fokker-Planck equation in higher dimensions then takes the form

$$\frac{\partial p(t,\mathbf{x})}{\partial t} = -\nabla\cdot(\mathbf{F}(t,\mathbf{x})\,p(t,\mathbf{x})) + \nabla\cdot(\nabla\cdot(\mathbf{D}(t,\mathbf{x})\,p(t,\mathbf{x})))\,, \quad \mathbf{D} = \frac{1}{2}\hat{\mathbf{D}}\,\hat{\mathbf{D}}^T\,.$$

This equation is valid inside the domain $\Omega \subset \mathbb{R}^d$ of the process $\mathbf{X}_t$. Similarly to the one-dimensional case, boundary conditions for the adsorbing and reflective boundaries can be described through Dirichlet (adsorbing) and Robin (reflective) boundaries, respectively. Similarly to the Fokker-Planck equation, one can extend the derivation of the backwards equation and the mean first passage time problem into higher dimensions.

1.3.2 Velocity jump processes

A second random process that we will use repeatedly throughout the thesis is the so-called *velocity jump process* [34]. The name stems from the comparison to the discretised Brownian motion (1.2) that is often called a position jump process[4]. In a velocity jump process the position $\mathbf{X}_t \in \Omega \subset \mathbb{R}^d$, $d = 1, 2, 3$, of a particle changes according to its current velocity $\mathbf{V}_t \in V \subset \mathbb{R}^d$, where Ω is the domain in which the particle moves and V is the velocity space. In the interval $[t, t + \mathrm{d}t)$ the velocity changes with probability $\lambda\,\mathrm{d}t$, where the *turning frequency* $\lambda \in \mathbb{R}$ can depend on various internal and external factors. If a change of velocity does occur, the new velocity is drawn from the so-called *turning kernel* $T(\mathbf{v}_*, \mathbf{v}) : V \times V \mapsto \mathbb{R}$ that denotes the probability[5] of choosing $\mathbf{v}_* \in V$ as the new velocity, given that the old velocity was $\mathbf{v} \in V$. We derive a PDE for the mesoscopic probability distribution $p(t, \mathbf{x}, \mathbf{v})$ that formally denotes the probability that the random walk is at time $t \geq 0$

[4]Note though that a Brownian motion is by definition almost surely continuous everywhere, which means that technically the position does not undergo a jump.

[5]Being a probability distribution, T has to satisfy $\int_V T(\mathbf{v}_*, \mathbf{v})\,\mathrm{d}\mathbf{v}_* = 1$.

positioned at $\mathbf{x} \in \Omega$ with a velocity $\mathbf{v} \in V$. For the derivation, we assume that we know the probability distribution at time t and derive the probability distribution at time $t + \mathrm{d}t$ by

$$\begin{aligned} p(t + \mathrm{d}t, \mathbf{x}, \mathbf{v}) =& (1 - \lambda \, \mathrm{d}t) \, p(t, \mathbf{x} - \mathrm{d}t \, \mathbf{v}, \mathbf{v}) \\ &+ \lambda \, \mathrm{d}t \int_V T(\mathbf{v}, \mathbf{v}_*) \, p(t, \mathbf{x} - \mathrm{d}t \, \mathbf{v}_*, \mathbf{v}_*) \, \mathrm{d}\mathbf{v}_* + \mathcal{O}\left((\mathrm{d}t)^2\right) , \end{aligned}$$

where the first term on the right-hand side represents the case that the particle did not turn in the interval $[t, t + \mathrm{d}t)$. The second term corresponds to the case that a particle has turned from its previous velocity $\mathbf{v}_*$ into $\mathbf{v}$. Using a Taylor expansion in position $\mathbf{x}$, we can write

$$p(t, \mathbf{x} - \mathrm{d}t \, \mathbf{v}, \mathbf{w}) = p(t, \mathbf{x}, \mathbf{w}) - \mathrm{d}t \, \mathbf{v} \cdot \nabla_{\mathbf{x}} p(t, \mathbf{x}, \mathbf{w}) \; + \mathcal{O}\left((\mathrm{d}t)^2\right) ,$$

for $\mathbf{w} \in \{\mathbf{v}, \mathbf{v}_*\}$. Plugging this into the above equation, dividing by $\mathrm{d}t$ and considering the limit $\mathrm{d}t \to 0$, we obtain the classical velocity jump equation [128]

$$\frac{\partial p(t, \mathbf{x}, \mathbf{v})}{\partial t} + \mathbf{v} \cdot \nabla_{\mathbf{x}} p(t, \mathbf{x}, \mathbf{v}) = -\lambda p(t, \mathbf{x}, \mathbf{v}) + \lambda \int_V T(\mathbf{v}, \mathbf{v}_*) \, p(t, \mathbf{x}, \mathbf{v}_*) \, \mathrm{d}\mathbf{v}_* \, . \tag{1.6}$$

Whilst a great part of literature regarding the velocity jump processes deals with the properties and effects of the turning kernel [56, 171], most models considered in this thesis will use an unbiased turning kernel of the form $T(\mathbf{v}, \mathbf{v}_*) = 1/|V|$ and a constant speed $s \in \mathbb{R}$ such that

$$V = s \cdot \mathbb{S}^{d-1} = \left\{ \mathbf{v} \in \mathbb{R}^d \; : \; \|\mathbf{v}\| = s \right\} .$$

The Telegrapher's equations

We now investigate such an unbiased velocity jump process with constant speed in one space dimension. In this case the velocity space V reduces to $V = \{s, -s\}$ and we can define the two probability density functions $p^{\pm}(t, x) = p(t, x, \pm s)$. The velocity jump equation takes the form of the so-called *Telegrapher's equations* [128]

$$\begin{aligned} \frac{\partial p^+}{\partial t} + s \frac{\partial p^+}{\partial x} &= \frac{\lambda}{2} \left(p^- - p^+ \right) , \\ \frac{\partial p^-}{\partial t} - s \frac{\partial p^-}{\partial x} &= \frac{\lambda}{2} \left(p^+ - p^- \right) . \end{aligned} \tag{1.7}$$

Let us define the macroscopic density $n(t, x)$ and the flux $j(t, x)$ through

$$n(t, x) = p^+(t, x) + p^-(t, x) \, , \qquad j(t, x) = s p^+(t, x) - s p^-(t, x) \, . \tag{1.8}$$

Adding the two telegrapher's equations (1.7), we obtain an equation for the evolution of $n(t,x)$

$$\frac{\partial n}{\partial t} + \frac{\partial j}{\partial x} = 0\,. \tag{1.9}$$

Similarly, by multiplying (1.7) by $\pm s$ and adding the two equations, we obtain

$$\frac{\partial j}{\partial t} + s^2 \frac{\partial n}{\partial x} = -\lambda j\,. \tag{1.10}$$

Differentiating (1.9) with respect to t and (1.10) with respect to x and subtracting one from the other, we obtain the damped wave equation [93]

$$\frac{1}{\lambda}\frac{\partial^2 n}{\partial t^2} + \frac{\partial n}{\partial t} = \frac{s^2}{\lambda}\frac{\partial^2 n}{\partial x^2}\,. \tag{1.11}$$

One could also view (1.11) as a *hyperbolic diffusion equation*, because taking adequate scaling limits as described in [55], the classical diffusion equation

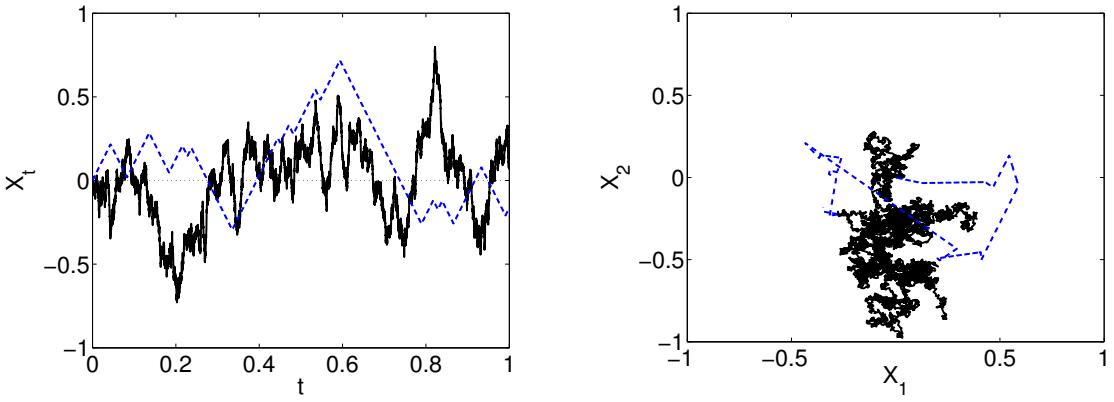
$$\frac{\partial n}{\partial t} = D_0 \frac{\partial^2 n}{\partial x^2}\,, \qquad D_0 = \frac{s^2}{\lambda_0}\,,$$

can be obtained. Taking the limit $\lambda, s \to \infty$ whilst keeping $D_0 = s^2/\lambda$ constant, this relation is exact and the velocity jump process described above becomes a Brownian path with the diffusion constant D_0. In Figure 1.1(a) we show comparisons of realisations of each process in one dimension, where we choose $s = 5$, $\lambda = 25$ for the velocity jump process and $F \equiv 0$, $D = 1$ for the Brownian dynamics simulation. One can clearly see the different nature of the two processes, but it is easy to imagine that passing to the limit $\lambda \to \infty$ while keeping $D_0 = s^2/\lambda$ constant could result in a Brownian path.

The Cattaneo approximation

In higher space dimensions the derivation above does not generally hold. Instead, one can use the so-called *Cattaneo approximation* to derive an effective drift diffusion equation related to a given velocity jump process [86]. The general idea is similar to the one-dimensional case above. Hence, let us define the velocity moments of the mesoscopic probability density function $p(t,\mathbf{x},\mathbf{v})$ as follows:

$$\begin{aligned}
m^{(0)}(t,\mathbf{x}) &= \int_V p(t,\mathbf{x},\mathbf{v})\,\mathrm{d}\mathbf{v}\,,\\
m_i^{(1)}(t,\mathbf{x}) &= \int_V v_i\, p(t,\mathbf{x},\mathbf{v})\,\mathrm{d}\mathbf{v}\,, \qquad\qquad i = 1,\dots,d\,,\\
m_{i,j}^{(2)}(t,\mathbf{x}) &= \int_V v_i\, v_j\, p(t,\mathbf{x},\mathbf{v})\,\mathrm{d}\mathbf{v}\,, \qquad i,j = 1,\dots,d\,,
\end{aligned}$$



(a) (b)

Figure 1.1: *Comparison between Brownian walks and velocity jump processes for* $d = 1, 2$ *achieved with the following parameters:*
(a) *One dimension* ($d = 1$)*:* $s = 5$, $\lambda = 25$, $D = 1$, $\Delta t = 10^{-4}$.
(b) *Two dimensions* ($d = 2$)*:* $s = 5$, $\lambda = 25$, $D = 0.5$, $\Delta t = 10^{-4}$.
In both panels the solid (black) line represents the Brownian walk and the dashed (blue) line represents the velocity jump process.

where $\mathbf{v} = [v_1, \ldots, v_d]^T$. Throughout this thesis, we will also use the shorter notation

$$\mathbf{m}^{(1)} = \left[m_1^{(1)}, \ldots, m_d^{(1)}\right]^T \in \mathbb{R}^d\,,$$
$$\mathbf{M}^{(2)} = \left(m_{i,j}^{(2)}\right)_{i,j} \in \mathbb{R}^{d\times d}\,.$$

Integrating the classical velocity jump equation (1.6) with respect to $\mathbf{v} \in V$, we obtain

$$\frac{\partial m^{(0)}}{\partial t} + \sum_{i=1}^{d} \int_V v_i \frac{\partial p}{\partial x_i}\,\mathrm{d}\mathbf{v} = -\lambda m_0 + \int_V \int_V T(\mathbf{v}, \mathbf{v}_*)\, p(t, \mathbf{x}, \mathbf{v}_*)\,\mathrm{d}\mathbf{v}_*\,\mathrm{d}\mathbf{v}\,.$$

Changing the order of integration and differentiation on the left-hand side and changing the order of integration on the right-hand side using $\int_V T(\mathbf{v}, \mathbf{v}_*)\,\mathrm{d}\mathbf{v} = 1$, we obtain the equation for the zeroth moment

$$\frac{\partial m^{(0)}}{\partial t} + \nabla \cdot \mathbf{m}^{(1)} = 0\,.$$

In order to derive equations for the first moments $m_i^{(1)}$, $i = 1, \ldots, d$, we multiply (1.6) by v_i and integrate again with respect to $\mathbf{v} \in V$. We obtain after changing the order of integration / differentiation

$$\frac{\partial m_i^{(1)}}{\partial t} + \sum_{j=1}^{d} \frac{\partial}{\partial x_j}\left(m_{i,j}^{(2)}\right) = -\lambda m_i^{(1)} + \int_V \int_V v_i\, T(\mathbf{v}, \mathbf{v}_*)\,\mathrm{d}\mathbf{v}\, p(t, \mathbf{x}, \mathbf{v}_*)\,\mathrm{d}\mathbf{v}_*\,.$$

In the case of an unbiased process, we have $\int_V \mathbf{v}\, T(\mathbf{v}, \mathbf{v}_*)\, \mathrm{d}\mathbf{v} = \mathbf{0}$ and we can write the first moment equation in the form

$$\frac{\partial \mathbf{m}^{(1)}}{\partial t} + \nabla \cdot \mathbf{M}^{(2)} = -\lambda \mathbf{m}^{(1)} .$$

In a similar manner, one can derive second moment equations for $m^{(2)}_{i,j}$ that will depend on the third moments and so on. In order to close this system Hillen [86] proposes an L^2 moment closure by looking for a function $u(t, \mathbf{x}, \mathbf{v})$ that has the zeroth moment $m^{(0)}(t, \mathbf{x})$, the first moments $m^{(1)}_i(t, \mathbf{x})$, $i = 1, \ldots, d$, and a minimal L^2 integral. To close this system, Hillen [86] then assumes that $\mathbf{M}^{(2)}$ is equal to the second moment of $u(t, \mathbf{x}, \mathbf{v})$, which in the case of constant speed $s \in \mathbb{R}$ results in

$$\mathbf{M}^{(2)} = \frac{s^2}{d} m^{(0)}\, I\,, \tag{1.12}$$

with $I \in \mathbb{R}^{d\times d}$ being the identity matrix. The complete Cattaneo approximation can then be written as

$$\begin{aligned} \frac{\partial m^{(0)}}{\partial t} + \nabla \cdot \mathbf{m}^{(1)} &= 0\,, \\ \frac{\partial \mathbf{m}^{(1)}}{\partial t} + \frac{s^2}{d} \nabla m^{(0)} &= -\lambda \mathbf{m}^{(1)} . \end{aligned}$$

Using similar techniques as in the one-dimensional case, we can derive a hyperbolic diffusion equation similar to (1.11):

$$\frac{1}{\lambda} \frac{\partial^2 m^{(0)}}{\partial t^2} + \frac{\partial m^{(0)}}{\partial t} = D_0 \Delta m^{(0)}\,, \quad D_0 = \frac{s^2}{d\lambda}\,.$$

Again, if one considers times $t \ll 1/\lambda$ this equation behaves like a d-dimensional diffusion equation with diffusion constant D_0. We will return to the technique of the Cattaneo approximation at different points throughout this thesis in order to adjust it for different situations. We show a comparison between a Brownian walk and a velocity jump process in two dimensions in Figure 1.1(b).

1.4 Modelling frameworks for systems of dispersal

In this section we introduce mathematical frameworks for agent-based, continuum and hybrid models as well as swarm robotics systems and show their applications in various examples.

1.4.1 Agent-based models

The main characteristic of agent-based models is that they directly describe the behaviour of individuals depending on their environment [140]. This can include ongoing internal processes as well as the interaction between individuals through direct contact or other means of communication. The main advantage of these models is therefore that a very specific description of the individual behaviour and interactions can be considered; these details are often lost in continuum descriptions. On the downside, individual-based models require significant computational effort in the case of high numbers of individuals and complex communication patterns. Due to this, agent-based models can only be used for systems that contain a tractable number of particles. One common application area is the modelling of collective animal behaviour; e.g. Couzin et al. [38] present a model that mimics the motion of bird flocks and fish schools. Another popular application area for agent-based modelling is the behaviour of human crowds in panic situations [82, 135].

An informal description

While continuum models have a well-developed mathematical theory, descriptions for agent-based models are often given in an informal way or even written as computer routines. This makes it difficult to create a unified framework for agents and to theoretically analyse their behaviour. The literature also fails to agree on a general definition of an agent. In this thesis, we use a definition which is adapted from [170] and used in [65].

Definition 1.3. *An agent is a system that uses a fixed set of rules based on communication with other agents and information about the environment in order to change its internal state and fulfil its design objective.*

Mathematical framework

We now want to shape this informal definition into a mathematical framework that describes the general behaviour of an agent. Following from Definition 1.3 this framework needs to incorporate behavioural rules for each agent as well as the possibility of communication between the agents. Let us therefore consider N agents numbered $i = 1, \dots, N$. In practice this number N can change during the course of a simulation through death or reproduction of agents, hence $N \equiv N(t) \in \mathbb{N}$. Each of these agents has n_i internal state variables $y_j^{(i)}$, $j = 1, \dots, n_i$, that form the internal state vector

$\mathbf{y}^{(i)} \in \mathbb{R}^{n_i}$ of agent $i = 1, \ldots, N$. These internal states can describe the position, velocity or properties like hunger or sleep deprivation of an agent [154]. The behavioural rules of an agent are incorporated in the update rules for these internal states $\mathbf{y}^{(i)}$. We additionally assume that one agent does not have access to all internal state variables of another agent. Instead every agent has a set of observable state variables $\mathbf{w}^{(i)} \in \mathbb{R}^{m_i}$, $i = 1, \ldots, N$, that can be perceived by other agents. These observable states depend directly on the internal states and in most cases form a subset of these. The multi-agent system can then be written in the form

$$\mathbf{y}^{(i)}(t + \Delta t) = \mathbf{f}^{(i)}\left(t, \mathbf{y}^{(i)}(t), \Delta t, \mathcal{X}\right), \quad i = 1, \ldots, N, \tag{1.13}$$

$$\mathbf{w}^{(i)}(t) = \mathbf{g}^{(i)}\left(\mathbf{y}^{(i)}(t)\right), \quad i = 1, \ldots, N, \tag{1.14}$$

$$\mathcal{X} = \left\{\mathbf{w}^{(1)}, \ldots, \mathbf{w}^{(N)}\right\}, \tag{1.15}$$

where we defined the set $\mathcal{X}$ as the set of the external states of all agents to simplify the notation. We can see that the evolution of $\mathbf{y}^{(i)}$ is given by the function $\mathbf{f}^{(i)}$, which notably depends on the time step Δt. This general description can entail discretised versions of ordinary differential equations (ODEs) as well as stochastic differential equations (SDEs). For example the SDE (1.1) can be written in the form (1.13)–(1.15) using the Euler-Maruyama method as shown in (1.2). Additionally, cellular automaton models like the ones shown in [141] can be formulated in the form (1.13)–(1.15).

We understand the external states of an agent merely as an observable representation of the internal states, which is why $\mathbf{w}^{(i)}(t)$ directly depends on $\mathbf{y}^{(i)}(t)$ through the function $\mathbf{g}^{(i)}$. The distinction between observable and non-observable states is often used to represent internal memories that cannot be perceived by other agents [65]. In many practical cases, agents are limited to perceiving the observable states of other agents close by [38], which in the presented framework can be incorporated into the update function $\mathbf{f}^{(i)}$.

As mentioned before, the advantage of this type of model over a continuum description is the possibility of accurately defining the individual behaviour as well as the communication between agents. This allows one to consider individuals with complex behavioural rules and systems with very low copy numbers, where stochastic effects play a major role. On the other hand agent-based models contain a number of variables that is of $\mathcal{O}(N)$, i.e. proportional to the number of agents in the system. Pairwise interactions incorporated into (1.13) through the set $\mathcal{X}$ lead to a computational complexity of $\mathcal{O}(N^2)$, which can present a significant computational effort in the case of high copy numbers in the system. Similarly this high number of variables

and interaction terms and the sometimes rather loose definition of the update process, mean that analytic results for agent-based models are only very rarely achievable and one often has to be content with numerical results.

1.4.2 Continuum models

For systems with very high copy numbers, like reaction-diffusion PDEs [49], the computational complexity of purely agent-based models becomes intractable and one often reverts to continuum models. Rather than individual properties, these models consider all individuals of a given species through group-level quantities. The continuum models we are presenting here use the spatio-temporal concentrations as the main indicator function of a species. The update rules can be acquired by phenomenological approaches or by formal mathematical limiting processes, through so-called mean field equations. In Section 1.3 we have already seen two of those mean field limits that lead to the equations (1.3) and (1.6) that can be interpreted as equations for spatio-temporal concentrations of a species.

Mathematical framework

We consider a system of M biological or chemical species inside a computational domain $\Omega \subset \mathbb{R}^d$, $d = 1, 2, 3$, and describe these species through their respective concentrations $c_i(t, \mathbf{x}) : [0, \infty) \times \Omega \mapsto \mathbb{R}_0^+$, $i = 1, \dots, M$, at a given point $\mathbf{x} \in \Omega$ and at time $t \geq 0$. To simplify notation we introduce the vector $\mathbf{c} = (c_1, \dots, c_M)^T \in \mathbb{R}^M$. The update relations for these concentrations are in general given through an evolution equation of the form

$$\frac{\partial \mathbf{c}}{\partial t}(t, \mathbf{x}) = \mathcal{L}(t, \mathbf{x}, \mathbf{c}) \,, \qquad \mathbf{x} \in \Omega \,, \qquad t \geq 0 \,, \tag{1.16}$$

where $\mathcal{L}$ is an operator that acts on the concentrations $\mathbf{c}$ as a function of $\mathbf{x} \in \Omega$ and $t \geq 0$. The operator $\mathcal{L}$ most commonly takes the form of a differential operator, e.g. in the Fokker-Planck equation (1.3), or an integro-differential operator, e.g. in the velocity jump equation (1.6). In these cases the update rule (1.16) takes the form of a PDE or an integro-differential equation (IDE). Additionally, one could also imagine the addition of stochastic terms to obtain stochastic partial differential equations (SPDEs) as seen in [29]. In addition to the update relation (1.16), initial data for the concentrations is given through

$$\mathbf{c}(0, \mathbf{x}) = \mathbf{c}_0(\mathbf{x}) \,, \qquad \mathbf{x} \in \Omega \,. \tag{1.17}$$

In order to enforce certain behaviour at the boundary $\partial\Omega$ of the domain Ω, boundary conditions are given in the form

$$\mathcal{G}(t, \mathbf{x}, \mathbf{c}) = 0\,, \qquad \mathbf{x} \in \partial\Omega\,, \qquad t > 0\,, \tag{1.18}$$

where $\mathcal{G}$ is an operator on the concentrations $\mathbf{c}$. In most practical examples these boundary conditions describe the values (Dirichlet conditions) or the normal derivatives (Neumann conditions) of $\mathbf{c}$ on the boundary of the domain. Equations (1.16)–(1.18) form the full continuum model that describes the behaviour of the concentrations of the M species. Note, that one can extend this framework to allow different species to be defined in different domains $\Omega_i \subset \Omega$, if for example certain species are confined by the presence of a membrane [73].

Numerical and analytic solutions

In the case where $\mathcal{L}$ is a differential operator and (1.16) takes the form of a system of PDEs, one can make use of the existing numerical and theoretical results that can be found in the vast literature about systems of PDEs. Analytic solutions of systems of PDEs are for example studied in [125] whilst an introduction to their numerical solution is given in [41]. In this subsection we want to discuss aspects of the numerical solution of continuum systems (1.16)–(1.18) that might be of interest for the introduction of hybrid models in Section 1.4.3.

Let us assume that we have been given L points $\mathbf{r}^{(1)}, \ldots, \mathbf{r}^{(L)} \in \Omega$ that form an adequate representation of Ω. We will call these points *gridpoints* even though they do not have to form a grid in Ω. The numerical solver tries to approximate the values of $\mathbf{c}\left(\mathbf{r}^{(l)}\right)$, $l = 1, \ldots, L$, at given times $t_k = k\Delta t$, $k = 1, 2, \ldots$. To simplify notation we define the matrices

$$\mathbf{C}_k = \left[\mathbf{c}\left(t_k, \mathbf{r}^{(1)}\right), \ldots, \mathbf{c}\left(t_k, \mathbf{r}^{(L)}\right)\right] \in \mathbb{R}^{M \times L}\,. \tag{1.19}$$

The update rule for the matrices $\mathbf{C}_k$ can now be formulated through

$$\{\mathbf{C}_0, \ldots, \mathbf{C}_k\} \xmapsto{\mathcal{L}_D} \mathbf{C}_{k+1}\,, \tag{1.20}$$

where $\mathcal{L}_D$ represents a discretised version of the operator $\mathcal{L}$ defined in (1.16). This discretised version represents the choice of solver used to numerically analyse the system of PDEs (1.16). Common solvers include finite difference approaches as well as more complex finite element methods [41].

1.4.3 Hybrid models

So far we have reviewed the two classical modelling approaches for systems of dispersal: agent-based models that directly incorporate the individual behaviour and interactions between individuals and continuum models that describe the group of individuals through group-level quantities. As a rule of thumb one mostly uses continuum approaches if the number of individuals in a system is large, if these individuals have relatively simplistic behaviour and interactions, and are small compared to the size of the considered domain. For systems with a smaller number of particles or a more complex behavioural pattern it is often advantageous to use agent-based models as the additional freedom in the description can help the modelling process.

In certain situations, it might, however, be desirable to combine the two frameworks into one model in order to exploit inhomogeneities in the system. For example, one might wish to have the choice between agent-based models and continuum descriptions for each of the species in the system independently. This has the advantage that one does not have to find a compromise between varying properties of different species. On the other hand certain spatial characteristics of the system, e.g. caused by membranes that are only permeable for certain molecules [73] or special behaviour close to a reaction surface [48], might render it advantageous to be able to switch between the two modelling types depending on the spatial regime.

The option of using different modelling approaches for different species or in different spatial regimes helps to accommodate differences in properties like size, complexity and number. For example, Dallon and Othmer [39] developed a hybrid model for chemotaxis of slime mould *Dictyostelium discoideum* in which the cells are treated as individuals in a continuum field of the chemoattractant which evolves according to a reaction-diffusion PDE. A similar hybrid modelling framework has also been applied to chemotaxis of bacteria [44, 174] and leukocytes [79]. The use of a hybrid approach for these problems allows for faster simulations than the purely agent-based model which would treat the small and abundant extracellular molecules as another set of agents and would lead to an intractable computational complexity.

The use of hybrid models is becoming more widespread especially with the growing computational power that allows one to consider more complex systems in this manner. Examples include the modelling of tumour growth [134] and forest dynamics [104]. In cancer biology, several hybrid cellular automaton models have been proposed in the literature [141, 149]. For example, Smallbone et al. [149] coupled a two-dimensional cellular automaton model (describing cells) with continuum (PDE-based models) of glucose, H^+ and oxygen concentrations, building on the previous

work of Patel et al. [134] and Alarcón et al. [3]. A similar hybrid approach has been used in a number of other studies in cancer biology [6, 75, 127]. A hybrid forest model with trees modelled as agents and a continuum approach used for oxygen and other atmospheric gases is presented in [104]. In economical research, hybrid models are used to estimate prices in the petrol market [83] and in general markets with a non-uniform spatial demand of products [84, 85]. In these models the demand is described as a continuous function of space whereas the retailers are considered as agents.

Mathematical framework

Because of their hybrid nature the general framework for these models necessarily combines the two frameworks presented in Sections 1.4.1 and 1.4.2. We therefore define M_C concentrations $c_i(t, \mathbf{x}) : [0, \infty) \times \Omega \mapsto \mathbb{R}_0^+$, $i = 1, \ldots, M_C$, on the domain $\Omega \subset \mathbb{R}^d$, $d = 1, 2, 3$. Similarly to the update rule (1.16) for the continuum case, the evolution of $\mathbf{c} = [c_1, \ldots, c_{M_C}]^T$ is governed by an operator $\mathcal{L}$. For the hybrid model, however, this operator $\mathcal{L}$ also depends on the current states of the agents. The $N \equiv N(t) \in \mathbb{N}$ agents are represented by their internal state variables $\mathbf{y}^{(i)}(t)$ and their set of observable states $\mathbf{w}^{(i)}(t)$ as defined in (1.14). To allow for interactions between the agents and the continuous variables $\mathbf{c}$, the set of observable states $\mathcal{X}$ as defined in (1.15) is used. The update rules for the system are formally

$$\frac{\partial \mathbf{c}}{\partial t} = \mathcal{L}\left(t, \mathbf{x}, \mathbf{c}, \mathcal{X}\right) , \qquad \mathbf{x} \in \Omega , \tag{1.21}$$

$$\mathbf{y}^{(i)}(t + \Delta t) = \mathbf{f}^{(i)}\left(t, \mathbf{y}^{(i)}(t), \Delta t, \mathcal{X}, \mathbf{c}\right) , \quad i = 1, \ldots, N , \tag{1.22}$$

where $\mathcal{X}$ is given in (1.15). In (1.21) we see that the agents can influence the continuous variables $\mathbf{c}$ through the set of observable states $\mathcal{X}$. Similarly, the behaviour of the agents can be altered by the continuous variables, as the operator $\mathbf{f}^{(i)}$ now also depends on $\mathbf{c}$. Figure 1.2 shows a graphical representation of the hybrid model. It contains the N agents represented by the internal states $\mathbf{y}^{(i)}$ on the left. Through the function $\mathbf{g}^{(i)}$ the observable states $\mathbf{w}^{(i)}$ are generated. These in turn influence the update of the continuous variables $\mathbf{c}$ as well as the agents' behaviour.

Example 1.1. ***Hybrid cellular automaton model for carcinogenesis***
In [149] *Smallbone et al. present a hybrid cellular automaton model for the formation of cancer. This model uses reaction-diffusion equations to calculate the concentration of oxygen, glucose and hydrogen ions in the environment of the cells. The concentrations of these chemicals therefore constitute the continuous variables* $\mathbf{c}$*. Each cell*

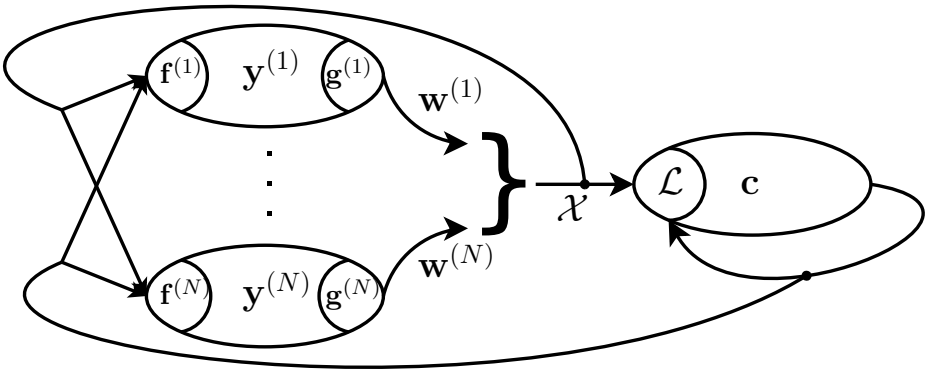


Figure 1.2: *Concept of a hybrid model. Arrows symbolise direction of influence.*

of the cellular automaton is represented by an agent with the internal state $y^{(i)} \in \mathbb{N}$ *defining which of the finite number of possible phenotypes the cell at this position has (including the "phenotype" empty). As these phenotypes are observable by neighbouring cells, we have* $w^{(i)} = y^{(i)}$*. This cellular automaton model has a generation-based update rule, which means that the states* $y^{(i)}$ *are only updated once every time step. The rules of the model then represent the probabilistic functions* $f^{(i)}$ *in equation* (1.13)*, where the change depends on the current phenotype, the neighbouring cells and the concentrations of the considered chemicals at the cell position.*

Example 1.2. ***Hybrid model for chemotaxis of*** **Dictyostelium discoideum** *Dallon and Othmer developed a hybrid model for the chemotaxis of* Dictyostelium discoideum [39] *that combines individual cell movement with a continuous extracellular concentration of cAMP modelled by a PDE. The internal states of the agents are the position of the individual* $\mathbf{x}^{(i)} \in \mathbb{R}^2$*, as well as the variables representing the intracellular processes. Only the position and one of the intracellular variables influence the external field and therefore form the observable states* $\mathbf{w}^{(i)} \in \mathbb{R}^3$*. The update rules* $\mathbf{f}^{(i)}$ *are given through ODEs for the internal dynamics and rules of motion for the position.*

A position-based hybrid model

So far, we have defined a general framework for hybrid models that couples agent-based models and continuum models with a great freedom. Let us now refine this framework more specifically for the systems presented in this thesis.

We start by introducing the concept of species into the agent-based parts of the models. Let us therefore assume that the N agents are divided onto M_A different species, similar to the M_C species of the continuum components. Each species is

represented by $N_j(t)$, $j = 1, \ldots, M_A$, agents[6]. Additionally, we now require every agent to have a position $\mathbf{x}^{(i,j)} \in \Omega$, $i = 1, \ldots, N_j(t)$, $j = 1, \ldots, M_A$, inside the domain Ω that is part of its internal variables $\mathbf{y}^{(i,j)}$ and is always observable, i.e. it is also part of $\mathbf{w}^{(i,j)}$. We will additionally require that all agents of the same species have consistent internal and observable state variables as well as identical update functions $\mathbf{f}^{(i,j)}$ and observing functions $\mathbf{g}^{(i,j)}$. Additionally to the position of an agent, we introduce the mass $m^{(i,j)} \in \mathbb{R}$ of an agent that is always part of the observable state vector. In most cases considered in this thesis, we use constant values for this mass. Using this, we can define density functions $\varrho_\delta^{(j)}$ for each of the species $j = 1, \ldots, M_A$ through

$$\varrho_\delta^{(j)}(t, \mathbf{x}) = \sum_{i=1}^{N_j} m^{(i,j)}(t)\, \delta\left(\mathbf{x} - \mathbf{x}^{(i,j)}(t)\right) , \qquad \mathbf{x} \in \Omega ,$$

where δ is the Dirac δ function. When discussing numerical simulations of hybrid models of chemotaxis in Chapter 2, we will see that this definition of $\varrho_\delta^{(j)}$ provides a first step towards obtaining a continuous density function for the agents. Assuming that $\mathbf{w}^{(i,j)} = [\mathbf{x}^{(i,j)}, m^{(i,j)}]$ for all species, we can redefine the operator $\mathcal{L}$, which governs the behaviour of the continuous variables $\mathbf{c}$ to directly incorporate $\varrho_\delta^{(j)}$ as follows

$$\frac{\partial \mathbf{c}}{\partial t} = \mathcal{L}\left(t, \mathbf{x}, \mathbf{c}, \varrho_\delta^{(1)}, \ldots, \varrho^{(M_A)}\right) .$$

This concept is still very flexible in that a large number of currently used hybrid models can be formulated this way. The aim of Chapters 2 and 3 of this thesis is the development of specific ways of implementing this concept for different types of systems with different inhomogeneities to be exploited. We will see how the hybrid concept can strongly reduce computational effort whilst maintaining individual-based resolutions where necessary or desired.

1.4.4 Swarm robotic experiments

Whilst swarm robotic experiments cannot be seen as a mathematical modelling framework as such, they can still provide further insight into biological systems of dispersal, and have become widespread in the literature [18]. Most commonly swarm robotic systems are used to mimic biological behaviour in order to improve understanding of specific systems and to eventually alter the behaviour of the target system [72]. In other application areas the mimicking of real systems can be used directly to enhance the operation of an existing robotic system [32]. Additionally, swarm robots are often

[6]Evidently, we require $\sum_{j=1}^{M_A} N_j(t) = N(t)$.

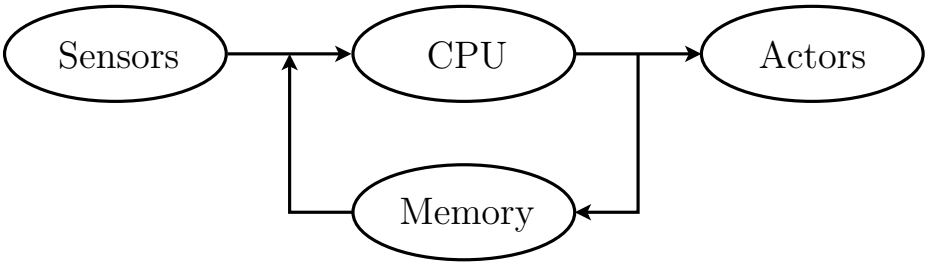


Figure 1.3: *Concept of a robotics system. Arrows symbolise data flow.*

used for hazardous tasks where a group of affordable robots is advantageous compared to one large scale operator, e.g. in oil spill confrontation [92].

One can generally split the components of an individual robot into four parts: sensors, processing unit, internal memory and actors. Sensors are used to observe the environment and typically give noisy and therefore imperfect data. Common types of sensors for swarm robots are proximity sensors, microphones and cameras. The sensor data is typically transmitted to the central processing unit (CPU), where the important information, e.g. positions of obstacles, gets extracted and is then used to control the behaviour of the robot. The CPU runs a set of rules that have been programmed beforehand and that is often given in the form of an endlessly repeating iteration of a number of steps. The evaluation of the program depends on data read from the internal memory and can save or manipulate data in this memory. The resulting control outputs are then transferred to the actors that influence the physical behaviour of the robot. Actors can appear in a number of forms, but are most commonly motors, status indicators or speakers. Other robots and external observers can only perceive changes administered by those actors, e.g. through change of position or lighting of a status LED. This concept is illustrated in Figure 1.3.

It is clear that swarm robotics systems have a number of things in common with agent-based models. In particular, the internal memory can be interpreted as a set of internal variables $\mathbf{y}^{(i)}$ and the observable states $\mathbf{w}^{(i)}$ of a robot are controlled by the actors that take a role similar to the observer function $\mathbf{g}^{(i)}$ in agent-based models. The main program along with the evaluation of sensor readings is described through the update function $\mathbf{f}^{(i)}$. The main difference between robots and agent-based models is that the environment and thereby the observable states of other robots can only be seen through the sensor data. This sensor data is often noisy and only provides an imperfect approximation of those states.

In Chapter 4 we study a swarm robotic system on the application of target finding. We discuss the differences between agent-based systems and aim to improve classical

models to account for those differences. We then study collisions between robots in more detail in Chapter 5. Using the results from those two chapters one can build reasonably accurate agent-based and continuum models that are able to describe the behaviour of simple robotic systems.

Chapter 2

Hybrid modelling of bacterial chemotaxis

In this chapter, we pick up from the mathematical framework of hybrid models presented in Section 1.4.3 and apply it to the process of bacterial chemotaxis. The focus thereby lies both on the application area and on the implementation of more general concepts in hybrid models of a certain type. We start the chapter by explaining the biological background of bacterial chemotaxis and follow up with the discussion of the classical continuum model for this system and some of its successors. In Section 2.3 we extend the hybrid modelling concept for this application area, thereby discussing certain aspects that play an important role in the implementation of hybrid models in practice. These concepts will be applied to obtain a hybrid version of the classical chemotaxis model in Section 2.4. The main focus of this section is to compare hybrid models to corresponding continuum models in order to understand what advantages and difficulties these models present. In the next step, we introduce growth processes into the chemotaxis system, thereby addressing another important conceptual question of hybrid models. All results will be summed up and discussed in Section 2.6.

Most results presented in Section 2.4 as well as concepts presented in the earlier sections were published in the invited book chapter [66]. Whilst the majority of the models presented in those sections are taken from the literature, the unifying modelling framework and the methodical study of coupling problems in hybrid models is novel. The results presented in Section 2.5 were published in the *Bulletin of Mathematical Biology* paper [69] and are novel.

2.1 Biological background

Definition 2.1. Bacterial chemotaxis *is a phenomenon where bacteria adjust their direction of motion depending on the concentrations of certain chemicals in their environment.*

This phenomenon plays an important role for bacteria as it allows them to direct their motion towards food sources (e.g. glucose) and away from harmful substances (e.g. phenol) [14]. We will call those chemicals that attract bacteria *chemoattractants* and those that repel them *chemorepellents*. This driven form of motion plays a fundamental role in many biological processes, including development [102], wound healing [168] and tumour invasion [74].

The process of bacterial chemotaxis was first described in the late 19th century by Engelmann [43] and Pfeffer [136]. It was not until the late 1960s, however, that the pioneering experiments of Adler were able to demonstrate and quantify the process [1, 2]. In his experiments Adler injected a population of *Escherichia coli* (*E. coli*) bacteria into one end of a closed tube that was filled with oxygen and an energy rich substance (e.g. galactose). Adler reports the formation of two bands of bacteria that are visible to the naked eye. He deduces that the bacteria consume the two substances (oxygen and galactose) and create a concentration gradient that they follow through chemotactic movement. He attributes the formation of two bands to the excess of one of the substances over the other, which causes the first band to leave a residual that can be used to form the second band. Interestingly, Adler also observed that the two bands move with a close to constant speed over a long period of time [1].

Further studies of *E. coli* bacteria have revealed that their movement is controlled by four to six rotating flagella that are spread across the cell surface [11]. Two different types of motion can be identified and are directly linked to the rotation of those flagella. If the flagella rotate counter-clockwise, they coalesce into a bundle resulting in a straight movement called a *run*. Conversely, clockwise flagella rotation does not lead to bundle formation and results in a rotation on the spot of the bacterium that is called a *tumble* [14]. Further experiments have revealed that the average run duration is $\sim 1\,\text{sec}$ with a tumble duration of $\sim 0.1\,\text{sec}$ [12]. Additionally, it was shown that the reorientation of the tumble is almost random with a small bias towards the previous direction and that *E. coli* bacteria have a complex internal reaction network that leads to the chemotactic behaviour observed in experiments through prolongation of a run phase if the direction is advantageous and shortening if it is disadvantageous. The reason why *E. coli* is often used as a model system in the mathematical and biological

literature is that its internal reaction network is extremely well understood, see e.g. [131] for a review of the literature in this research area.

2.2 The Keller-Segel model

One of the earliest mathematical models of bacterial chemotaxis was developed by Keller and Segel [95]. The model, which is now known as the *Keller-Segel model*, considers both the bacteria and an extracellular chemical, in a continuum limit. The concentration of bacteria is denoted by $n(t,x)$ and the concentration of the extracellular signal by $S(t,x)$. The update equation (1.16) for their model takes the form [95]

$$\frac{\partial n}{\partial t} = D_n \frac{\partial^2 n}{\partial x^2} - \frac{\partial}{\partial x}\left(n\chi(S)\frac{\partial S}{\partial x}\right), \tag{2.1}$$

$$\frac{\partial S}{\partial t} = D_S \frac{\partial^2 S}{\partial x^2} - k(S)n\,, \tag{2.2}$$

where $\chi(S)$ is the chemotactic sensitivity function and $k(S)$ governs the consumption of extracellular signal by the bacteria. We can see that the behaviour of the bacteria takes the form of a general advection-diffusion equation that can be interpreted as a Fokker-Planck equation as seen in (1.3) in Section 1.3.1. The diffusion of the bacteria occurs with the diffusion constant D_n, while the advection is governed by the chemotactic sensitivity $\chi(S)$. Depending on the sign of $\chi(S)$ the chemical S can be a chemoattractant ($\chi > 0$) or a chemorepellent ($\chi < 0$). The extracellular signal diffuses with the diffusion constant D_S and is consumed by the bacteria with a consumption rate $k(S)$ that depends on the concentration of S.

One can mimic the experiments performed by Adler by considering system (2.1)–(2.2) on the domain $\Omega = [0, L]$ using the initial conditions

$$n(0,x) = \delta(x) \qquad \text{and} \qquad S(0,x) = S_\infty\,, \qquad x \in [0,L]\,, \tag{2.3}$$

where $\delta(x)$ denotes the Dirac δ function and $S_\infty \in \mathbb{R}^+$ is the constant concentration of extracellular signal at the beginning of the experiment. We apply no-flow boundary conditions at both ends of the domain, which corresponds to the closed tube. Those conditions take the form

$$D_n \left.\frac{\partial n}{\partial x}\right|_{x=0,L} = \left. n\chi(S)\frac{\partial S}{\partial x}\right|_{x=0,L}, \qquad D_S \left.\frac{\partial S}{\partial x}\right|_{x=0,L} = 0\,. \tag{2.4}$$

These conditions (2.3)–(2.4) now complete the continuum model defined by equations (1.16)–(1.18).

In a direct follow-up to the original paper, Keller and Segel investigated the existence of travelling wave solutions of the system (2.1)–(2.2) [96]. The derivations in the remainder of this section are transcribed from [96]. For the study of travelling wave solutions it is necessary to consider the Keller-Segel system on the infinite domain $\Omega = \mathbb{R}$ with boundary conditions given through

$$\begin{aligned} \lim_{x\to\pm\infty} n(t,x) &= 0\,, \quad \lim_{x\to\pm\infty} \frac{\partial n(t,x)}{\partial x} = 0\,, \qquad t>0\,, \\ \lim_{x\to-\infty} S(t,x) &= 0\,, \quad \lim_{x\to\infty} S(t,x) \quad = S_\infty\,, \qquad t>0\,. \end{aligned} \tag{2.5}$$

To simplify the derivation they neglected the (passive) diffusion of the extracellular signal S, because it is slower than the (active) motion of the bacteria [95]. Additionally, they assumed $k(S) = k_0 \in \mathbb{R}^+$. We now prove that under these conditions travelling wave solutions for the system (2.1)–(2.2) can only exist if the chemotactic sensitivity $\chi(S)$ has a singularity [96].

Theorem 2.1. *If the chemotactic sensitivity $\chi(S) : \mathbb{R}_0^+ \mapsto \mathbb{R}_0^+$ does not have a singularity, the classical Keller-Segel system* (2.1)–(2.2)*, under the assumptions $D_S = 0$ and $k(S) = k_0 \in \mathbb{R}^+$, cannot yield travelling wave solutions in $\mathbb{R}$ that satisfy the boundary conditions* (2.5)*.*

Proof. Let us assume that the chemotactic sensitivity $\chi(S)$ does not have a singularity inside the interval $S \in [0, S_\infty]$. Therefore, χ is bounded and integrable in this interval. Let us moreover assume that a travelling wave solution exists and use the following change of variables to account for this

$$n(t,x) = n_w(x-ct) = n_w(\xi) \qquad \text{and} \qquad S(t,x) = S_w(x-ct) = S_w(\xi)\,. \tag{2.6}$$

Using the simplifying assumptions $D_S = 0$ and $k(S) = k_0$ we can now rewrite the system (2.1)–(2.2) as

$$-cn_w' = D_n n_w'' - \left(n_w S_w' \chi(S_w)\right)', \tag{2.7}$$

$$-cS_w' = -k_0 n_w\,, \tag{2.8}$$

where primes denote derivatives with respect to the wave variable ξ. The boundary conditions for the wave variables are

$$\lim_{\xi\to\pm\infty} n_w(\xi) = 0\,, \qquad \lim_{\xi\to\pm\infty} n_w'(\xi) = 0\,, \qquad \lim_{\xi\to-\infty} S_w(\xi) = 0\,, \qquad \lim_{\xi\to\infty} S_w(\xi) = S_\infty\,. \tag{2.9}$$

Integrating (2.7) with respect to ξ and applying the boundary conditions (2.9) at $\xi \to -\infty$ as well as the boundedness of χ, we obtain

$$n_w(\xi) = Q \exp\left(-\frac{c}{D_n}\xi\right) \exp\left(\frac{g(S_w)}{D_n}\right) ,$$

where $\mathrm{d}g(S)/\mathrm{d}S = \chi(S)$, i.e.

$$g(S) = g(0) + \int_0^S \chi(\bar{S})\,\mathrm{d}\bar{S} ,$$

with $g(0) \in \mathbb{R}$ and $Q \in \mathbb{R}^+$ being arbitrary integration constants. Substituting this result into (2.8) and reordering, we get

$$\frac{S_w'}{\exp\left(\frac{g(S_w)}{D_n}\right)} = \frac{Qk_0}{c} \exp\left(-\frac{c}{D_n}\xi\right) .$$

Integrating this equation over the interval $[\xi, \infty)$ gives

$$\int_\xi^\infty \frac{S_w'(\bar{\xi})}{\exp\left(\frac{g\left(S_w(\bar{\xi})\right)}{D_n}\right)} \mathrm{d}\bar{\xi} = \int_{S(\xi)}^{S_\infty} \exp\left(-\frac{g(\bar{S})}{D_n}\right) \mathrm{d}\bar{S} = \frac{QD_nk_0}{c^2} \exp\left(-\frac{c}{D_n}\xi\right) . \quad (2.10)$$

Considering this equation in the limit $\xi \to -\infty$, we can see that the right-hand side diverges. However, for the left-hand side of the equation we get

$$\begin{aligned} 0 &\leq \int_0^{S_\infty} \exp\left(-\frac{g(S)}{D_n}\right) \mathrm{d}S = \int_0^{S_\infty} \exp\left(-\frac{1}{D_n}\left(g(0) + \int_0^S \chi(\bar{S})\,\mathrm{d}\bar{S}\right)\right) \mathrm{d}S \\ &\leq \int_0^{S_\infty} \exp\left(-\frac{g(0)}{D_n}\right) \mathrm{d}S \; = S_\infty \exp\left(-\frac{g(0)}{D_n}\right) < \infty , \end{aligned}$$

where we used the fact that χ is integrable, bounded and non-negative in the interval $[0, S_\infty]$. This result, however, is a contradiction, as the left-hand side of equation (2.10) is bounded for $\xi \to -\infty$, while the right-hand side diverges. Therefore, a travelling wave solution for the classical Keller-Segel system (2.1)–(2.2) with the simplifying assumptions $D_S = 0$ and $k(S) = k_0$ can only exist if $\chi(S)$ has a singularity. □

Moreover, Keller and Segel showed that the singularity needs to be of order 1 or greater and hence we choose the form $\chi(S) = D_n\kappa/S$ [96], where $\kappa \in \mathbb{R}^+$ is an arbitrary positive constant. Using this form along with the simplifying assumptions made in Theorem 2.1, we can derive the following dimensionless Keller-Segel system that we use throughout the remainder of this section:

$$\frac{\partial n}{\partial t} = \mu \frac{\partial}{\partial x}\left(\frac{\partial n}{\partial x} - \kappa n \frac{1}{S}\frac{\partial S}{\partial x}\right) , \quad (2.11)$$

$$\frac{\partial S}{\partial t} = -n , \quad (2.12)$$

where the dimensionless constant μ is given through

$$\mu = \frac{D_n\, S_\infty}{N\, k_0\, L^2} \in \mathbb{R}^+\,, \quad \text{with} \quad N = \int_\Omega n(0,x)\,\mathrm{d}x\,,$$

and L is a typical domain length. The initial condition (2.3) for S has been simplified to $S_\infty = 1$ and the integral of n over the whole domain was normalised to

$$\int_\Omega n(t,x)\,\mathrm{d}x = 1\,, \qquad t \geq 0\,. \tag{2.13}$$

Theorem 2.2. *If $\kappa > 1$ and $\mu > 0$, the system (2.11)–(2.12) considered in $\Omega = \mathbb{R}$ has a travelling wave solution satisfying the boundary conditions (2.5) with $S_\infty = 1$.*

Proof. Using the wave variables as defined in (2.6), we can rewrite the system (2.11)–(2.12) in the form

$$-cn_w' = \mu n_w'' - \mu\kappa \left(n_w \frac{S_w'}{S_w} \right)' , \tag{2.14}$$

$$-cS_w' = -n_w\,, \tag{2.15}$$

The boundary conditions for this new system are given in (2.9), again using the simplifying assumption $S_\infty = 1$. Integrating (2.14) with respect to ξ and using the boundary conditions (2.9) for $\xi \to \infty$, we obtain

$$cn_w = \mu \left(\kappa n_w \frac{S_w'}{S_w} - n_w' \right) .$$

We can divide this by n_w (> 0) and integrate again with respect to ξ. We get

$$n_w = Q \exp\left(-\frac{c}{\mu}\xi \right) S_w^\kappa\,, \tag{2.16}$$

where Q is a positive constant. We now plug this result back into (2.15) and integrate again in order to get the travelling wave solution $S_w(\xi)$:

$$S_w(\xi) = \left[1 + \frac{\mu}{c^2}(\kappa - 1) Q \exp\left(-\frac{c}{\mu}\xi \right) \right]^{\frac{1}{1-\kappa}} ,$$

where the integration constant was chosen in a way that satisfies the boundary conditions (2.9) for $\xi \to \infty$. The second constant Q can be chosen arbitrarily positive, as it translates the travelling wave solution $S_w(\xi)$ along the ξ-axis. We choose

$$Q = \frac{c^2}{\mu(\kappa - 1)}\,,$$

which results in the following expression for S_w

$$S_w(\xi) = \left[1 + \exp\left(-\frac{c}{\mu}\xi\right)\right]^{\frac{1}{1-\kappa}} . \tag{2.17}$$

Using this result, we obtain the form of the travelling wave solution n_w for the bacteria through (2.16)

$$n_w(\xi) = \frac{c^2}{\mu(\kappa-1)} \exp\left(-\frac{c}{\mu}\xi\right) \left[1 + \exp\left(-\frac{c}{\mu}\xi\right)\right]^{\frac{\kappa}{1-\kappa}} . \tag{2.18}$$

We can see that the travelling wave solution given in (2.17)–(2.18) satisfies the boundary conditions (2.9) for all values of $\mu > 0$ and $\kappa > 1$. □

Corrollary 2.1. *Using the integral constraint* (2.13) *given through the non-dimensionalisation, we can integrate equation* (2.15) *over the whole* $\mathbb{R}$ *in order to obtain the wave speed* c

$$c = \frac{\int_{-\infty}^{\infty} n_w \,\mathrm{d}\xi}{\int_{-\infty}^{\infty} S_w'(\xi)\,\mathrm{d}\xi} = 1 .$$

Hence, the wave speed of the dimensionless Keller-Segel system (2.11)–(2.12) *is equal to* 1.

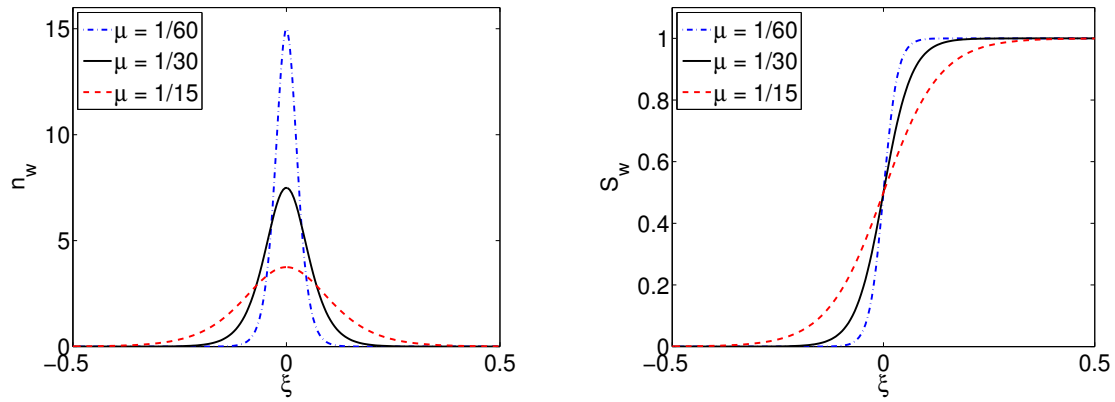
Corrollary 2.2. *For the special case* $\kappa = 2$*, using the result from Corollary 2.1* ($c = 1$)*, the travelling wave solution simplifies to*

$$\begin{aligned} n_w(\xi) &= \mu^{-1} \exp\left(-\frac{\xi}{\mu}\right) \left[1 + \exp\left(-\frac{\xi}{\mu}\right)\right]^{-2} , \\ S_w(\xi) &= \left[1 + \exp\left(-\frac{\xi}{\mu}\right)\right]^{-1} . \end{aligned} \tag{2.19}$$

In particular, we can see that the function $n_w(\xi)$ *is an even function, i.e.*

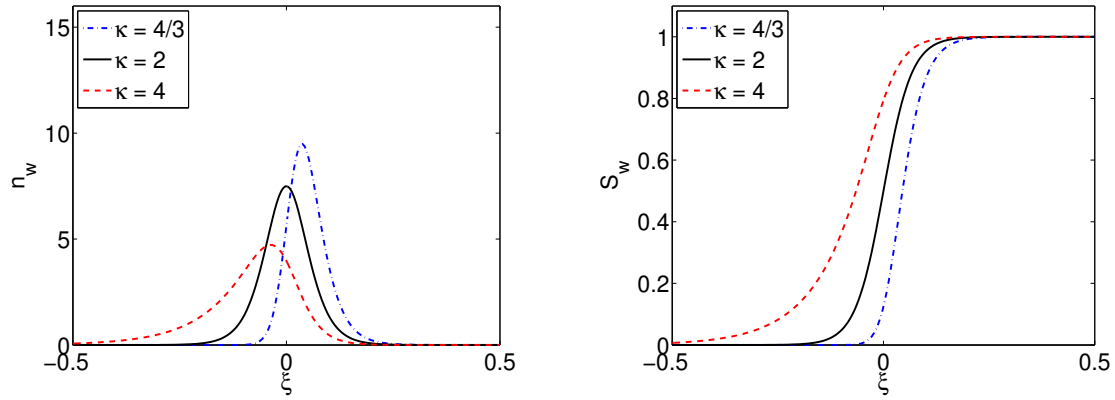
$$\begin{aligned} n_w(-\xi) &= \mu^{-1} \exp\left(\frac{\xi}{\mu}\right) \left[1 + \exp\left(\frac{\xi}{\mu}\right)\right]^{-2} \\ &= \mu^{-1} \exp\left(-\frac{\xi}{\mu}\right) \left[\exp\left(-\frac{\xi}{\mu}\right)\left(1 + \exp\left(\frac{\xi}{\mu}\right)\right)\right]^{-2} \\ &= n_w(\xi) . \end{aligned}$$

In Figures 2.1–2.2 we plot the travelling wave solutions (2.17)–(2.18) of the dimensionless Keller-Segel system (2.11)–(2.12) for varying parameter values. We can see that the value of μ defines the width of the wave, while leaving the shape unchanged. The value of κ, however, changes the shape of the travelling wave solution. In the



(a) $n_w(\xi)$ (b) $S_w(\xi)$

Figure 2.1: *Travelling wave solution of the Keller-Segel model* (2.11)–(2.12) *given in* (2.17)–(2.18) *for different values of the parameter* μ*, where* $\kappa = 2$*.*



(a) $n_w(\xi)$ (b) $S_w(\xi)$

Figure 2.2: *Travelling wave solution of the Keller-Segel model* (2.11)–(2.12) *given in* (2.17)–(2.18) *for different values of the parameter* κ*, where* $\mu = 1/30$*.*

case $\kappa = 2$ the travelling band of bacteria is symmetric, as was shown in Corollary 2.2. For $\kappa > 2$ the wave is steeper in the front (right) and falls slowly in the back (left). Choosing $\kappa < 2$ causes an opposite effect with the wave being bent backwards. We will use the symmetric case $\kappa = 2$ throughout the remainder of this chapter for convenience.

Travelling waves in chemotaxis models have also been recently studied in [106, 107]; we also note the articles [88] and [166] for a review and analysis of travelling waves in PDE-based models. A comparison between mesoscopic (hyperbolic) and macroscopic (parabolic) PDEs has been presented in [112].

2.3 Hybrid models for different species

The bacterial chemotaxis system comprises two species that differ strongly in their nature. On the one hand, extracellular molecules are extremely abundant and follow a simple diffusion process, whilst bacteria on the other hand have a much more complex behaviour, are much bigger and occur in copy numbers of many orders of magnitude less. Because of those inhomogeneities, we are looking to apply the position-based hybrid modelling approach introduced in Section 1.4.3 to this system, with the bacteria being modelled as an agent-based species and the extracellular signal as a continuum.

Additionally, it is reasonable to assume that *E. coli* bacteria only have a very localised sensing [11]. We therefore assume that the agents in this model can only observe concentrations (and derivatives thereof) at their current position $\mathbf{x}^{(i,j)}$, $i = 1, \dots, N_j$, $j = 1, \dots, M_A$, following the definitions in Section 1.4.3. This, along with the step-wise update of agents internal dynamics (and positions) as defined in (1.22), has serious consequences for the nature of the numerical implementation of such hybrid models. As for purely agent-based models the accuracy and efficient implementation of numerical simulations is of great importance, especially because it is only in very rare cases possible to obtain analytic results for such systems.

2.3.1 Numerical implementation

When aiming to implement a hybrid model for the chemotaxis system, one quickly encounters a number of conceptual difficulties that span over all position-based hybrid models. Those problems can be identified when looking at the update performed for each of the species.

For the continuum variables $\mathbf{c}(t, \mathbf{x})$, we assume, similarly to the framework in Section 1.4.2, that the domain $\Omega \in \mathbb{R}^d$, $d = 1, 2, 3$, can be described by a finite number of gridpoints $\mathbf{r}^{(1)}, \dots, \mathbf{r}^{(L)} \in \Omega$. We are again interested in approximating the values of the continuous variables $\mathbf{c}$ at the gridpoints at given times $t_k = k\Delta t$, $k = 1, 2, \dots$. We use the notation $\mathbf{C}_k$ introduced in (1.19).

Additionally, we now need to calculate the values of the internal variables $\mathbf{y}^{(i,j)}$ for each of the agents $i = 1, \dots, N_j$ of each of the agent-based species $j = 1, \dots, M_A$ at the times $t_k = k\Delta t$, $k = 1, 2, \dots$. For each of the two subsystems we now have to answer the question of how to get from t_k to t_{k+1} still guaranteeing an accurate approximation of the system. For the continuous variables this means, we have to

adjust the solver described in (1.20) to contain the agents as follows

$$\{\mathbf{C}_0,\, \ldots,\, \mathbf{C}_k,\, \varrho_\delta(\cdot, t_k)\} \overset{\mathcal{L}_D}{\longmapsto} \mathbf{C}_{k+1}\,,$$

where $\varrho_\delta = \left[\varrho_\delta^{(1)}, \ldots, \varrho_\delta^{(M_A)}\right]^T \in \mathbb{R}^{M_A}$. The operator $\mathcal{L}_D$ now additionally depends on the positions of the agents at time t_k. Note that we have made the implicit assumption that the solver used for (1.21) only takes the positions of the agents at time t_k into account. For the agents, we intend to directly use the update rule given in (1.22) with the restriction that the concentrations $\mathbf{c}$ (and their derivatives) can only be obtained at the position $\mathbf{x}^{(i,j)}$ of the agent considered. This assumption leads to an immediate spatial matching problem, because in the numerical simulations these concentrations only exist on the gridpoints $\mathbf{r}^{(l)}$, $l = 1, \ldots, L$.

Interpolation of c to agent positions

We can reformulate this spatial matching problem into an interpolation problem, because the need to have estimates of $\mathbf{c}$ at all of the N agents positions is equivalent to obtaining an interpolated function $\hat{\mathbf{c}} : \Omega \mapsto \mathbb{R}_0^+$. We can visualise the need for an interpolation operator as follows

$$\left\{ \left(\mathbf{r}^{(1)}, \mathbf{c}\left(\mathbf{r}^{(1)}\right)\right), \ldots, \left(\mathbf{r}^{(L)}, \mathbf{c}\left(\mathbf{r}^{(L)}\right)\right) \right\} \overset{\Theta}{\longmapsto} \hat{\mathbf{c}}(\mathbf{x})\,.$$

One can additionally formulate conditions on the smoothness of the result of Θ, for example, we will always require that $\hat{\mathbf{c}}$ is a continuous function and that it is (at least) once continuously differentiable everywhere except at the gridpoints $\mathbf{r}^{(l)}$, $l = 1, \ldots, L$. Knowing that we have to deal with a straightforward interpolation problem here, we can use results from the well-studied fields of interpolation and approximation theory and one of the many textbooks therein [167]. The choice of interpolation operator depends strongly on the requirements for smoothness of the function $\hat{\mathbf{c}}$, on the mesh used and on the accuracy required. Note that in some cases the choice of numerical solver for the update equation of continuum variables $\mathbf{c}$ inherently solves the interpolation problem, for example a finite element approach inherently calculates the interpolated function $\hat{\mathbf{c}}$ [41].

As the study of bacterial chemotaxis in this thesis is reduced to compact one-dimensional domains $\Omega = [a, b]$, the interpolation problem is further simplified by the assumption that $a = r^{(1)} < r^{(2)} < \cdots < r^{(L)} = b$. As none of the hybrid models presented requires an approximation of the second derivative (or higher) of $\mathbf{c}$, we can use a linear (spline) interpolation operator Θ such that

$$\hat{\mathbf{c}}(x) = \mathbf{c}(r^{(l)}) + \left(\mathbf{c}(r^{(l+1)}) - \mathbf{c}(r^{(l)})\right) \frac{x - r^{(l)}}{r^{(l+1)} - r^{(l)}}\,, \qquad x \in \left[r^{(l)}, r^{(l+1)}\right]\,.$$

Other commonly used interpolation methods are quadratic or cubic spline or polynomial interpolations [167].

Density estimation of agent-based species on gridpoints

The second spatial matching problem that occurs during the numerical update of the concentrations $\mathbf{c}$ on the gridpoints $\mathbf{r}^{(l)}$, $l = 1, \ldots, L$, is due to the fact that ϱ_δ is not a classical function, but an ensemble of Dirac δ functions. Therefore, we need to get an estimate of the agent density for each species at least on the gridpoints, but ideally over the whole domain Ω. The required operator can be written in the form

$$\varrho_\delta^{(j)}(\mathbf{x}) = \sum_{i=1}^{N_j} m^{(i,j)} \delta\left(\mathbf{x} - \mathbf{x}^{(i,j)}\right) \overset{\Psi}{\longmapsto} \varrho^{(j)}(t, \mathbf{x}) : [0, \infty) \times \Omega \mapsto \mathbb{R}_0^+ .$$

Again, the requirements on the smoothness of $\varrho^{(j)}(\mathbf{x})$ can alter depending on the system, but we generally require it to be a continuous function. One way to achieve such a mapping is the so-called *kernel density estimation* [165]. In general this method is used to estimate the probability density function of a random process from a number of realisations of this process. The name stems from the use of a kernel $K(\mathbf{x}) : \mathbb{R}^d \mapsto \mathbb{R}$, which is typically a continuous, symmetric, non-negative and normalised function. For the one-dimensional systems studied in this chapter, these requirements can be written as

$$K \in C^0(\mathbb{R}), \quad K(-x) = K(x), \quad K(x) \geq 0, \quad \forall x \in \mathbb{R}, \quad \int_{-\infty}^{\infty} K(x)\,\mathrm{d}x = 1 .$$

In practice a scaled version of K is used, which leads to the introduction of a bandwidth parameter $\sigma \in \mathbb{R}^+$. We define

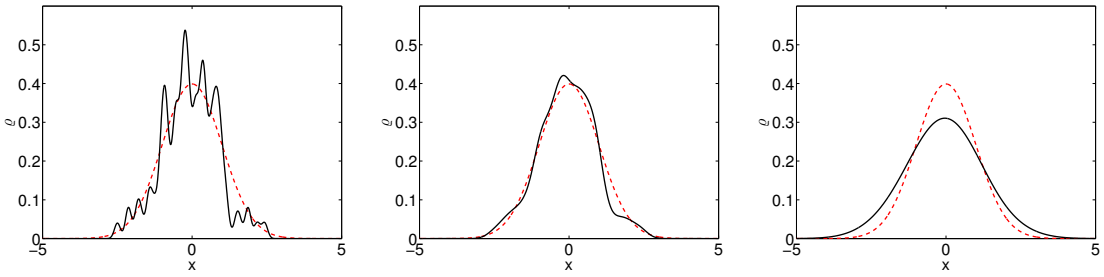
$$K_\sigma(x) = \frac{1}{\sigma} K\left(\frac{x}{\sigma}\right) ,$$

which still satisfies the above conditions. With given positions $\mathbf{x}^{(i,j)}$, an estimate of the probability density function is generally given by

$$\varrho^{(j)}(\mathbf{x}) = \sum_{i=1}^{N_j} m^{(i,j)} K_\sigma\left(\mathbf{x} - \mathbf{x}^{(i,j)}\right) , \qquad j = 1, \ldots, M_A . \tag{2.20}$$

In the one-dimensional case, (2.20) takes the form of a convolution between K_σ and $\varrho_\delta^{(j)}$. The most commonly used kernels are a Gaussian kernel of the form

$$K(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right) , \tag{2.21}$$



(a) $\sigma = 0.1$ (b) $\sigma = 0.25$ (c) $\sigma = 0.9$

Figure 2.3: *Kernel density estimate for $N = 100$ agents, which are placed according to a normal distribution for different bandwidths σ. The dashed (red) line is the underlying Gaussian probability density function and the solid (black) line is the generated estimate by* (2.20).

a piecewise linear kernel with compact support [133]

$$K(x) = \max\{0, 1 - |x|\}\,, \tag{2.22}$$

and a Laplacian kernel [90]

$$K(x) = \frac{1}{2}\exp\left(-|x|\right)\,.$$

Figure 2.3 shows an example of a kernel density estimation for 100 normally distributed random variables using a Gaussian kernel with different bandwidths σ. In Figure 2.3(a) we can see that the choice of a very small σ leads to a highly oscillating estimate, while a very large σ can lead to the estimate being too wide, as shown in Figure 2.3(c). An optimal choice for the parameter σ and the kernel itself always depends on the nature of the problem and the number of samples [91].

In practice, many hybrid models (e.g. Example 2.1) solve the density estimation process using a regular grid and having each agent only contribute to neighbouring gridpoints. This can be represented by the use of the piecewise-linear kernel given in (2.22) with the bandwidth $\sigma = \Delta x$ and can therefore also be seen as a kernel density estimation.

Other aspects of numerical simulations

In addition to the two spatial matching problems discussed thus far, the numerical simulation of hybrid models can present further difficulties. The first problem is the choice of a solver for the continuous variables that is in most cases restricted to explicit methods, because the use of implicit method requires the solution of implicit equations for all agent states and becomes therefore quickly non-feasible. Related

to this is the choice of time stepping. In theory it is possible to use different time steps for each species, whether agent-based or continuous. However, in practice the necessity of accurate estimates of concentrations for the update of all species often makes it easiest to use the same time step for all species.

The discrete nature of the agent-based parts can introduce stochastic effects into the system and we will discuss these effects in Sections 2.4 and 2.5. Because of this, it is important to consider these effects when choosing the time step, spatial resolution, interpolation method and kernel density estimation for the simulation. These choices will depend on the number of agents present in each of the species as well as the complexity of interactions between species.

Example 2.1. ***Numerical realisation of Example 1.2***
In Example 1.2 we presented a hybrid model for chemotaxis of slime mould Dictyostelium discoideum *developed by Dallon and Othmer* [39]. *To generate a discretised operator $\mathcal{L}_D$ they used the particle-in-cell method* [126]. *To generate an estimated particle density on a regular grid, every particle contributes to the four neighbouring gridpoints. If a particle is very close to one of the gridpoints it contributes more to this point than to the three other neighbouring gridpoints. This method can be interpreted as the use of a kernel density estimation with a piecewise linear kernel. Hence*

$$K(\mathbf{x}) = \max\left(1 - \|\mathbf{x}\|_\infty, 0\right) ,$$

where $\|\cdot\|_\infty$ denotes the infinity norm and $\mathbf{x} \in \mathbb{R}^2$. The bandwidth σ for the kernel density estimation is chosen to be equal to the grid size to ensure that every point can only contribute to four neighbouring gridpoints. Dallon and Othmer [39] *employed a fifth order spline interpolation as interpolation operator Θ.*

2.3.2 Initial and boundary conditions

An important aspect of modelling is the incorporation of initial and boundary conditions. Hybrid models necessarily combine the conditions from the two different approaches. For the continuous variables initial data is given similarly to the initial conditions seen in (1.17). For the agents an initial distribution of their position and internal states is given through a probability density. This probability density is then used to generate each agents' internal variables at the beginning of the simulation. These internal variables also define the agents' position inside the domain Ω. In some applications agents can be born during the course of the simulation. In this case, we have to ensure the appropriate initialisation of internal variables.

A similar idea of independent conditions for the continuum and the agent-based parts of the hybrid model can be used for the boundary conditions. For the continuous variables we again use the condition (1.18) defined in Section 1.4.2. For the agents the boundary conditions are often given in a more descriptive manner. For example, agents can leave the domain through one end and automatically reappear on the other end. This *periodic boundary condition* implies that the number of agents in the system is conserved. Periodic boundaries are widely used because of their simplicity and because they exclude all boundary effects. *Reactive boundaries* adsorb agents with a probability p, while reflecting them with probability $1 - p$ [47]. If $p = 0$, one often speaks of a *reflecting boundary*, while for $p = 1$ the condition is called an *adsorbing boundary*. In most application areas it makes sense to use corresponding boundary conditions for all species, e.g. reflective boundaries for agent-based species and no-flow boundaries for continuous species.

2.4 First hybrid models of bacterial chemotaxis

In this section, we put the hybrid framework discussed in Section 2.3 into practice for the example of bacterial chemotaxis that we presented in Section 2.1. This system provides an excellent example system, because the *E. coli* bacteria and its metabolism are extremely well understood and a great number of very detailed models exist [131]. Additionally, we have a system with two species of very different properties: On the one hand the extracellular chemical that acts as a chemoattractant/chemorepellent with more than 10^{10} molecules that follow a simple Brownian motion and on the other hand the bacteria with common copy numbers on the order of $\sim 10^4$ bacteria [55]. Another big advantage of this system is that hybrid and agent-based models of chemotaxis have already been successfully used in the literature [39, 44, 79, 174].

2.4.1 Hybrid Keller-Segel models

The hybrid models we are discussing in this section are derived from the dimensionless Keller-Segel system (2.11)–(2.12). In order to optimally exploit the different properties of the two species involved in this system, the extracellular chemical S is modelled using a continuum approach, whilst the bacterial concentration n gets replaced by an agent-based approach. We are studying three different agent-based descriptions in order to show the flexibility of the hybrid modelling framework introduced in Section 1.4.3. As we are only considering one agent-based species ($M_A = 1$), we can drop the species superscript j in all agent properties, e.g. $x^{(i,j)} \equiv x^{(i)}$ and we

will additionally use $N = N_1$. In all three models we disregard any growing or shrinking effects of individuals and therefore assume that for all agents the mass $m^{(i)} = 1$ stays constant in time. Similarly, we disregard proliferation, i.e. the number of agents N stays constant throughout one simulation. Both proliferation and growth of individuals will be introduced into the hybrid framework in Section 2.5.

According to the general Keller-Segel system (2.1)–(2.2), the update equation for the extracellular signal then takes the form

$$\frac{\partial S}{\partial t} = D_S \frac{\partial^2 S}{\partial x^2} - \frac{k(S)}{N} \sum_{i=1}^{N} K\big(x - x^{(i)}(t)\big)\,, \tag{2.23}$$

where we use the normalisation constant N according to the non-dimensionalisation and K is the kernel used for the density estimation. We are concentrating on the dimensionless system (2.11)–(2.12) and apply the additional assumptions $D_S = 0$ and $k(S) = 1$ made by Keller and Segel [96].

Model I – Brownian motion

The first idea of replacing equation (2.11) with an agent-based system uses a Brownian dynamics formulation and stems from observations made in Section 1.3.1. The idea is to interpret (2.11) as a Fokker-Planck equation and to formulate the corresponding stochastic process as update equation for the position of agent $i = 1, \ldots, N$ as follows

$$\mathrm{d}x^{(i)} = \mu\kappa \frac{1}{S\left(x^{(i)}\right)} \frac{\partial S\left(x^{(i)}\right)}{\partial x}\,\mathrm{d}t + \sqrt{2\mu}\,\mathrm{d}W\,, \tag{2.24}$$

This approach was formulated by Stevens [152] and will be the base for the first version of a hybrid chemotaxis model. The parameters used in (2.24) correspond to the ones in the dimensionless Keller-Segel equation (2.11). This particle-based description of equation (2.11) shows a weaknesses in the original Keller-Segel model. According to (2.24) an agent can obtain very high speeds if the gradient of S is large or the value of S is low at the agent's position. This contradicts some of the basic biological knowledge about the movement of *E. coli* bacteria that we discussed in Section 2.1.

Model II – Velocity jump with gradient sensing

Driven by this weaknesses in Model I and by the biological background described in Section 2.1, we will use the velocity jump process described in Section 1.3.2 in Model II. In particular, we assume that agents move with a constant speed $s \in \mathbb{R}^+$

either to the right or the left of the real line. To incorporate the chemotactic effect, Othmer et al. [128] introduced a biased velocity jump process. In their model the turning frequency λ depends on information gathered at the current position of the individual as follows

$$\lambda^{\pm}(t) = \lambda_0 \mp \frac{\kappa s}{S\left(t, x^{(i)}(t)\right)} \frac{\partial S\left(t, x^{(i)}(t)\right)}{\partial x} = \lambda_0 \mp s\chi(S)\frac{\partial S}{\partial x}\,, \tag{2.25}$$

where the plus stands for a right-moving agent and the minus for a left-moving agent and $\lambda_0 \in \mathbb{R}^+$ is the unbiased turning frequency. Hence, the run-phase is prolonged, if the particle is moving in the direction of increasing gradient and is shortened if the agent is moving against the gradient. The underlying assumption in this model is that cells are able to directly sense the gradient of the extracellular signal S, which is only really true for cells of a reasonable size, such as *Dictyostelium discoideum* as discussed in [39] (see Example 1.2 for more detail). To represent the direction of motion, the velocity $v^{(i)}(t) = \pm s$ is introduced as part of the internal dynamics vector $\mathbf{y}^{(i)} = [x^{(i)}, m^{(i)}, v^{(i)}]$. The update for these internal variables can now be written in the form

$$\begin{aligned} x^{(i)}(t+\Delta t) &= x^{(i)}(t) + v^{(i)}(t)\Delta t\,, \\ v^{(i)}(t+\Delta t) &= \begin{cases} -v^{(i)}(t)\,, & \text{with probability } \dfrac{\lambda^{\pm}(t)}{2}\Delta t\,, \\ v^{(i)}(t)\,, & \text{otherwise}\,, \end{cases} \end{aligned} \tag{2.26}$$

with λ as defined in (2.25). We can write density equations similar to the Telegrapher's equations (1.7) presented in Section 1.3.2 for the biased velocity jump process as follows [54]:

$$\begin{aligned} \frac{\partial p^+}{\partial t} + s\frac{\partial p^+}{\partial x} &= \frac{\lambda^-}{2}p^- - \frac{\lambda^+}{2}p^+\,, \\ \frac{\partial p^-}{\partial t} - s\frac{\partial p^-}{\partial x} &= \frac{\lambda^+}{2}p^+ - \frac{\lambda^-}{2}p^-\,. \end{aligned}$$

Using the definitions (1.8) and the same methods as for the unbiased process, we obtain the equation

$$\begin{aligned} \frac{\partial n}{\partial t} + \quad \frac{\partial j}{\partial x} &= 0\,, \\ \frac{\partial j}{\partial t} + s^2\frac{\partial n}{\partial x} &= s\left(\lambda^- p^- - \lambda^+ p^+\right)\,. \end{aligned}$$

Plugging in the form of $\lambda^{\pm}$ given in (2.25), we can write the second equation as

$$\frac{\partial j}{\partial t} + s^2\frac{\partial n}{\partial x} = -\lambda_0 j + s^2\frac{\kappa}{S}\frac{\partial S}{\partial x}n\,,$$

and we can finally obtain a hyperbolic chemotaxis equation of the form

$$\frac{1}{\lambda_0}\frac{\partial^2 n}{\partial t} + \frac{\partial n}{\partial t} = \frac{s^2}{\lambda_0}\frac{\partial}{\partial x}\left(\frac{\partial n}{\partial x} - n\frac{\kappa}{S}\frac{\partial S}{\partial x}\right). \tag{2.27}$$

This shows that changing the type of random-walk used for the agents can influence the corresponding continuum equation. Nevertheless (2.27) can be used to adjust the parameters of the agent-based model to match the parameters of the Keller-Segel model, as the large time behaviour of (2.27) is given by the classical chemotaxis equation (2.11), where we have $\mu = s^2/\lambda_0$ [93]. Lui and Wang [112] showed that coupling the hyperbolic chemotaxis equation (2.27) with (2.23) for the substrate also yields travelling wave solutions similar to the original Keller-Segel system. An investigation of this case for a more general dependence of the turning frequency is given in [173].

Model III

More accurate descriptions of the individual behaviour of *E. coli* bacteria incorporate the sensing and processing of extracellular signals [10, 131, 151]. Hybrid models with descriptions of these intracellular processes have been used by Dallon and Othmer [39] as well as Xue et al. [173]. Erban and Othmer [55, 56] used an agent with a toy version of the internal dynamics that includes two main features of the sensing process [17]: a fast excitation and a slower adaptation given through the two internal variables z_1 and z_2 as follows

$$\begin{aligned}\frac{\mathrm{d}z_1}{\mathrm{d}t} &= \frac{g\big(S\big(t,x(t)\big)\big) - z_1 - z_2}{t_e}\,,\\ \frac{\mathrm{d}z_2}{\mathrm{d}t} &= \frac{g\big(S\big(t,x(t)\big)\big) - z_2}{t_a}\,,\end{aligned} \tag{2.28}$$

where $g : \mathbb{R} \mapsto \mathbb{R}$ is a sensing function and $t_e, t_a \in \mathbb{R}^+$ are the excitation and adaptation times respectively; we assume $t_e \ll t_a$. In particular the adaptation variable z_2 takes the role of a memory that allows the agent to identify increasing or decreasing signal concentrations when compared to the current concentration $g(S(t, x(t)))$ [55]. Hence, we use again a velocity jump process with constant speed $s \in \mathbb{R}^+$ and the following form for the turning frequency λ

$$\lambda(t) = \lambda_0 + z_1^{(i)}(t)\,. \tag{2.29}$$

Using the assumption $t_e \ll t_a$, we can consider the first equation in (2.28) in its pseudo-steady state

$$z_1^{(i)}(t) = g\big(S\left(t, x^{(i)}(t)\right)\big) - z_2^{(i)}(t)\,.$$

Thus, we can formulate the full hybrid model using the internal dynamics vector $\mathbf{y}^{(i)} = [x^{(i)}, m^{(i)}, v^{(i)}, z_2^{(i)}]$ and the update equations

$$\begin{aligned} x^{(i)}(t+\Delta t) &= x^{(i)}(t) + v^{(i)}(t)\Delta t\,, \\ v^{(i)}(t+\Delta t) &= \begin{cases} -v^{(i)}(t)\,, & \text{with probability } \dfrac{\lambda(t)}{2}\,\Delta t\,, \\ v^{(i)}(t)\,, & \text{otherwise}\,, \end{cases} \\ z_2^{(i)}(t+\Delta t) &= z_2^{(i)}(t) + \frac{g\big(S\big(t, x^{(i)}(t)\big)\big) - z_2^{(i)}(t)}{t_a}\Delta t\,. \end{aligned} \tag{2.30}$$

In the limit $\Delta t \to 0$ and $N \to \infty$ this process can be described by the chemotaxis equation

$$\frac{\partial n}{\partial t} = \frac{s^2}{\lambda_0}\frac{\partial}{\partial x}\left(\frac{\partial n}{\partial x} - \frac{t_a}{1+\lambda_0 t_a}\frac{\mathrm{d}g}{\mathrm{d}S}\frac{\partial S}{\partial x}\right)\,, \tag{2.31}$$

provided that t is large ($t \gg 1/\lambda_0$) and the gradient of S is shallow [55]. Choices for the parameters of this model can be made by matching (2.31) with the classical chemotaxis equation (2.11), which especially indicates that g is given through $\mathrm{d}g/\mathrm{d}S \sim \chi(S)$.

2.4.2 First numerical results - transient travelling waves

Numerical solutions of the three hybrid chemotaxis models presented in Section 2.4.1 are shown in Figure 2.4. For all three models the parameters, given below, were chosen to match the global parameters $\mu = 1/30$ and $\kappa = 2$ in the dimensionless Keller-Segel system (2.11)–(2.12). In all three systems we see the formation of a wave that moves towards the right of the real line with a speed of about 1, which matches with the observation we made in Corollary 2.1. Additionally, comparing the observed wave forms to the travelling wave solution (2.19), as shown in Figure 2.1, we see a good agreement for all three models.

Closer observations of the region behind the travelling waves can be seen in panels (b),(d),(f) of Figure 2.4. These reveal that something interesting is happening here in that the extracellular signal S is completely exploited in that region, while some agents are still present. Due to lack of chemotactic guidance, a number of those agents will never be able to catch up with the wave again and will stay behind indefinitely. Two explanations for this are possible: (1) it could be solely due to agents losing contact when the wave first moves away from the left-hand boundary and some residual signal left in this region that keeps agents there for long enough to miss the wave or (2) it could be an effect inherent in these models. The focus of this section is to understand what causes this effect and how it influences the travelling wave behaviour in the

long-term. Throughout this and the following sections we will repeatedly refer to the loss of agents as *dropout*.

Simulation parameters

In the numerical solutions of the three different models used to obtain Figure 2.4, we used a Gaussian kernel as given in (2.21) with a bandwidth $\sigma = 5 \times 10^{-3}$ and a linear interpolation method as described in Section 2.3.1. We use $N = 10^4$ agents, $L = 401$ regularly dispersed gridpoints to simulate the interval $[0, 1]$, resulting in $r^{(l)} = (l-1)\Delta x$, $l = 1, \dots, L$, with $\Delta x = 2.5 \times 10^{-3}$. As initial conditions, we used $S \equiv 1$ and $x^{(i)}(t) = 10^{-4}|\xi_i|$, $i = 1, \dots, N$, where $\xi_i \sim \mathcal{N}(0; 1)$ are N independent normally distributed random variables. No-flow boundary conditions are applied at both boundaries. The time step was $\Delta t = 10^{-4}$ and the system was simulated until $t = 0.5$, at which point the results are plotted in Figure 2.4. The model specific parameters are:

Model I: We use the Euler-Maruyama method to solve the update equation (2.24) with the parameters $\mu = 1/30$ and $\kappa = 2$, using the linear interpolation to obtain the gradient and the value of S at the agent position $x^{(i)}$, $i = 1, \dots, N$.

Model II: We directly simulate the update equation (2.26) with the turning frequency given in (2.25). We use $\lambda_0 = 1.33 \times 10^2$, $\kappa = 2$ and $s = 6.67$, resulting in $s^2/\lambda_0 = 1/30$, meaning that the hyperbolic chemotaxis equation (2.27) is comparable to the Keller-Segel equation (2.11) with $\mu = 1/30$.

Model III: We directly simulate the update equation (2.30) with the turning frequency given in (2.29). We use $\lambda_0 = 1.33 \times 10^2$, $s = 6.67$, $t_a = 1.5 \times 10^{-3}$ and $g(S) = 1.5\lambda_0 \log(S)$. These parameters were chosen in such a way that the chemotaxis equation (2.31) matches the Keller-Segel equation (2.11) with $\mu = 1/30$ and $\kappa = 2$.

Measuring the dropout

In order to further analyse the dropout of agents from the travelling waves in hybrid Keller-Segel models, we need to find a way of quantifying it. In particular, we are looking for an index set $\Gamma(t)$ that contains the agents that are currently part of the wave.

However, before defining and comparing different conditions for the dropout, we investigate some global statistical values of the agent set. The first measure to indicate

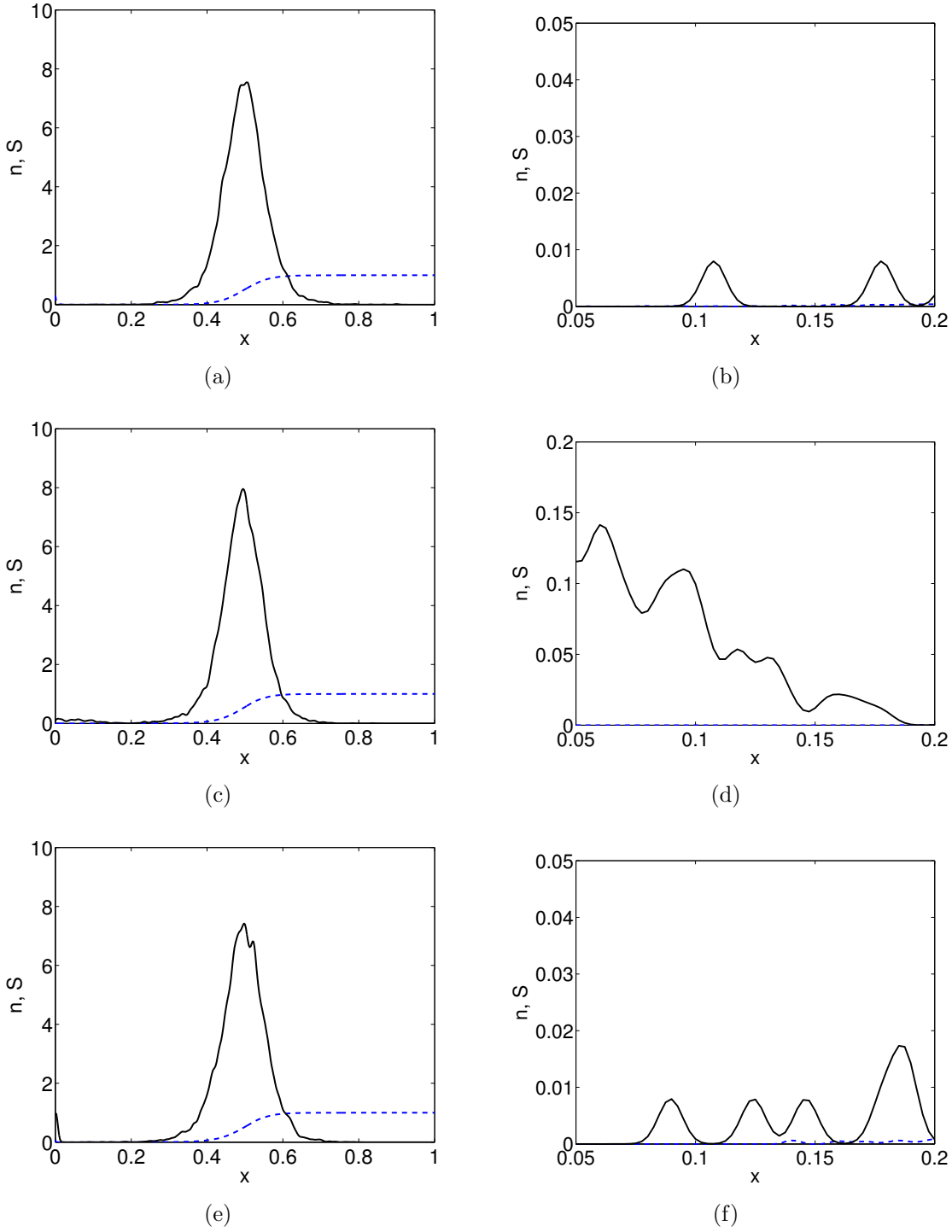


Figure 2.4: *Numerical simulation of the hybrid Keller-Segel models. Panels* (a),(c),(e) *show the full simulation results at* $t = 0.5$ *and panels* (b), (d), (f) *show a zoom into the region [0.05, 0.2]. Solid (black) line:* $n(0.5, x)$*, dashed (blue) line:* $S(0.5, x)$*. Simulation and model parameters as described in the text.*
(a),(b): *Model I.* (c),(d): *Model II.* (e),(f): *Model III.*

the fact that agents have dropped out is the position of the centre of the population $\bar{x}(t)$ defined through

$$\bar{x}(t) = \frac{1}{N} \sum_{i=1}^{N} x^{(i)}(t) \,.$$

From Corollary 2.1 we know that the theoretical wave speed of the dimensionless Keller-Segel system (2.11)–(2.12) is 1. Neglecting the time it takes for the wave to form, the predicted position of the centre of the wave therefore is $\bar{x}_{\text{pred}}(t) = t$. In practice the measure $\bar{x}(t)$ for the centre of the population is biased by those agents that have dropped out, which is why an alternative mean position is given through

$$\bar{x}_\Gamma(t) = \frac{1}{|\Gamma|} \sum_{i \in \Gamma} x^{(i)}(t) \,.$$

The problem with that second measure is that it depends on the index set $\Gamma(t)$ that we have not defined yet and that might in turn depend on the position of the centre of the wave. An alternative way of measuring the position of the wave is to use the wave front $\hat{x}(t)$ instead of the centre of the population $\bar{x}(t)$, as this is easier to measure in practice through

$$\hat{x}(t) = \max \left\{ x^{(i)}(t) \;:\; i = 1, \ldots, N \right\} \,.$$

Assuming that the wave front $\hat{x}(t)$ stays close to the centre of the actual wave, i.e. that the front-most agent stays in the vicinity of the wave[1], we have a means to define Γ without using the (unknown) centre of the wave

$$\Gamma_\delta(t) = \left\{ i = 1, \ldots, N \;:\; x^{(i)} \geq \hat{x}(t) - \delta \right\} \,,$$

where $\delta \in \mathbb{R}^+$ is a positive constant that should be chosen in accordance to the width of the wave. We can now define the dropout $o_\delta(t)$ as the fraction of agents that are not in Γ through

$$o_\delta(t) = 1 - \frac{|\Gamma_\delta(t)|}{N(t)} \,.$$

Large time behaviour

We use these measures to further investigate the dropout effect by considering long-time simulations. In particular we simulate Model III on the interval $\Omega = [0, 51]$ with $L = 20401$ gridpoints resulting in $\Delta x = 2.5 \times 10^{-3}$. In order to minimise the effect of the initial condition on the results, we initialise the system close to the travelling

[1] In practice, this assumption is not always applicable. A more robust definition of $\hat{x}(t)$ can be achieved by considering for example the frontmost 5% of agents.

wave solution (2.19) of the Keller-Segel system centred around $x = 0.5$. In particular this means that $S_0(r^{(l)})$ is given through

$$S_0\left(r^{(l)}\right) = \left(1 + \exp\left(-\frac{(l-1)\Delta x - 0.5}{\mu}\right)\right)^{-1}.$$

Similarly, the agents are initially distributed according to the density function (2.19) with random velocities and in the state of perfect adaptation $z_2^{(i)}(0) = S\left(x^{(i)}(0)\right)$. We plot the results in form of the various measures for the wave position in Figure 2.5(a) and it is clearly visible that the mean position of all agents $\bar{x}$ starts to strongly differ from the position of the front $\hat{x}$ and the related $\bar{x}_\Gamma$ indicating a significant dropout. We can also see that the front of the wave $\hat{x}$ does not move at constant speed as predicted for the continuum system. Further details of this are shown in Figure 2.5(b), where we plot the speed of the wave-front $\hat{v}(t)$ estimated through

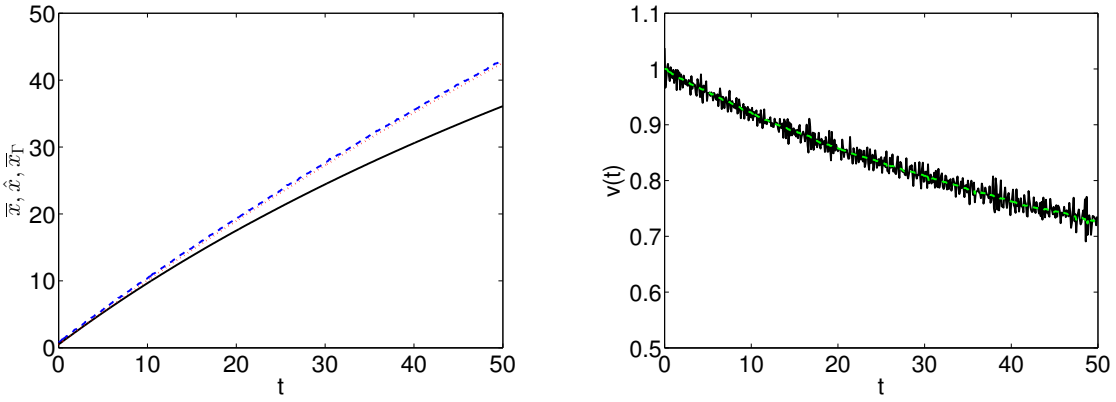
$$\hat{v}(t) = \frac{\hat{x}(t + \Delta T) - \hat{x}(t)}{\Delta T}, \tag{2.32}$$

where we use $\Delta T = 0.1$ to get reasonably smooth results. We compare $\hat{v}(t)$ to the number of agents still in the wave given through $1 - o_\delta(t)$ with $\delta = 0.8$ and see that they closely align indicating that the slowing of the wave is due to the loss of agents in the wave, as fewer agents take longer to consume the extracellular signal present and therefore advance slower. Figure 2.5 also shows that the dropout effect cannot solely be attributed to initial conditions, as the loss persists over long times.

The dropout is an interesting effect, because it shows a qualitative difference between the hybrid model and the original Keller-Segel model, as the hybrid model only yields transient travelling wave solutions. A possible way of explaining this discrepancy is that the copy numbers of bacteria in the region behind the wave are very low and that in this region stochastic effects play an enhanced role, thereby rendering the continuum assumptions of the Keller-Segel system invalid. The stochastic effects can lead to a distorted or completely exploited extracellular signal and therefore to the loss of information for bacteria to be able to catch up with the wave again.

2.4.3 Dropout under changing system parameters

In this section, we investigate the influence of the kernel density estimation as discussed in Section 2.3.1 on the dropout effect. To that aim we run simulations of Model III on the interval $\Omega = [0, 2]$ starting the system close to the travelling wave solution at $x = 0.5$, as explained above and letting it run until $t = 1$, at which point the wave is expected to be close to $x = 1.5$. We use $o_\delta(1)$, $\delta = 0.8$, to measure the



(a) (b)

Figure 2.5: *Numerical solution of hybrid chemotaxis Model III for large time ($t = 50$). Parameters as given in the text.*
(a) *Wave measures $\bar{x}$ (solid black line), $\hat{x}$ (dashed blue line), $\bar{x}_\Gamma$ (dotted red line) with $\Gamma = \Gamma_{0.8}(t)$.*
(b) *Velocity of the front of the wave $\hat{v}(t)$ (solid black line) compared to $1 - o_{0.8}(t)$ (dashed green line).*

dropout at this point and plot its dependence on the number of agents N ($L = 801$, $\Delta x = 2.5 \times 10^{-3}$) and the number of gridpoints L ($N = 10^4$) in Figure 2.6. For each value we run 100 simulations (crosses) and plot the ensemble average (circles). The first row shows the results using the nearest-neighbour kernel that was discussed in Example 2.1 and that can be found in various models in the literature. The remaining two rows show the results using a Gaussian kernel as defined in (2.21) with $\sigma = 2.5 \times 10^{-3}$ (second row) and $\sigma = 5 \times 10^{-3}$ (third row).

In panels (a),(c) and (e) we can see that for each of the methods, the relative dropout decreases for increasing numbers of agents N and closer investigation shows that we have $o_\delta(1) \sim N^{-1/2}$. This reinforces the idea of stochasticity causing the dropout, as the central limit theorem predicts noise in the number of particles per gridpoint to roughly behave like $N^{1/2}$ and dividing by N in (2.23) would predict the factor of $N^{-1/2}$. This also indicates that one recovers a non-transient travelling wave in the limit $N \to \infty$ as predicted by Keller and Segel [96].

On the other hand, the choice of a different kernel, can have significant effect on the dropout. While the dropout seems to level off for increasing L when using the Gaussian kernel (panels (d), (f)), it continues to increase for the nearest-neighbour kernel (panel (b)). This result justifies the use of the kernel density estimation with a bandwidth σ that is independent of Δx, as seemingly only in this case the system gives a meaningful result in the limit $\Delta x \to 0$. The use of the nearest-neighbour kernel

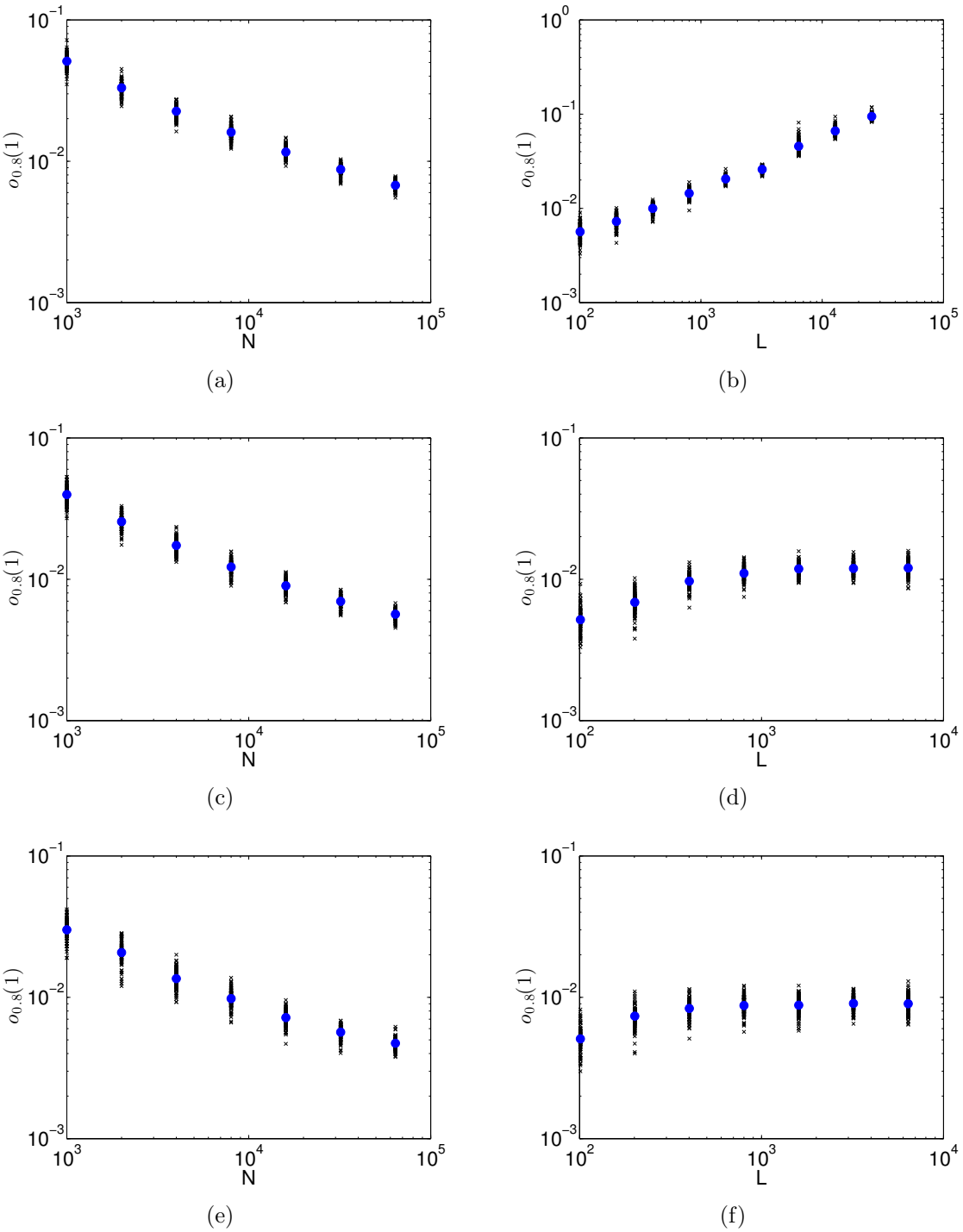


Figure 2.6: *Dropout* $o_{0.8}(1)$ *as a function of* N *(panels (a), (c), (e)) and* L *(panels (b), (d), (f)). Crosses: individual simulation results, dots: average over 100 realisations. All parameters as explained in the text.*
(a),(b) *Nearest neighbour kernel as defined in* (2.22) *with* $\sigma = \Delta x$.
(c),(d) *Gaussian kernel as defined in* (2.21) *with* $\sigma = 2.5 \times 10^{-3}$.
(e),(f) *Gaussian kernel as defined in* (2.21) *with* $\sigma = 5 \times 10^{-3}$.

on the other hand does not seem to stabilise under increasing L, thereby indicating that the system breaks down as $\Delta x \to 0$. The increase of dropout is proportional to $\sqrt{L}$, which one can again explain using the central limit theorem when considering that the number of agents per gridpoint is proportional to N/L and therefore that the noise in the estimated agent density is proportional to $\sqrt{L/N}$.

2.4.4 First steps into an analytic understanding of dropout

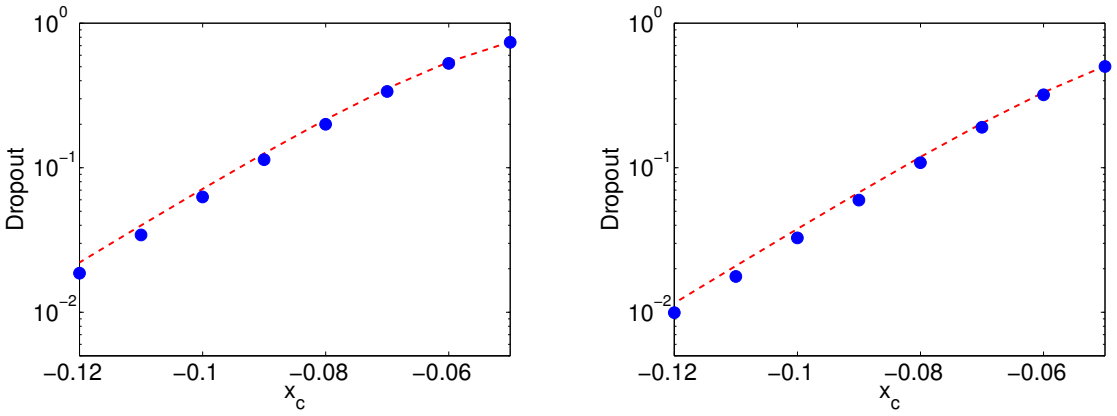
More theoretical insights into the dropout effect can be obtained by considering a simplified system, where the concentration of extracellular signal S is a given function that does not change over time. A natural choice for the function $S(x)$ is the travelling wave solution found by Keller and Segel [96]. Using the knowledge of the exploited region behind the wave, we can adjust this function slightly to allow for the analysis of the dropout effect. We therefore define S to be equal to 0 for x smaller than some critical position x_c and to take the form of the travelling wave solution (2.18) everywhere else. In this section we will use $\kappa = 2$, in which case the travelling wave solution takes the form shown in (2.19). Hence, we have

$$S(x) = \begin{cases} \left(1 + \exp(-x/\mu)\right)^{-1}, & x \geq x_c\,, \\ 0\,, & x < x_c\,. \end{cases}$$

To be able to use a time-independent function for S we need to offset the chemotactic drive towards the right of the real axis by adjusting the hybrid model. Therefore, we subtract the expected wave speed of 1 from the movement velocity of the agents in order to keep them in a position that is realistic for the travelling wave. In other words, we use a coordinate system that moves with the travelling wave solution. For an agent of Model I the evolution equation (2.24) becomes

$$\mathrm{d}x^{(i)} = \left(\mu \frac{\kappa}{S\left(x^{(i)}\right)} \frac{\mathrm{d}S\left(x^{(i)}\right)}{\mathrm{d}x} - 1 \right) \mathrm{d}t + \sqrt{2\mu}\,\mathrm{d}W\,,$$

whilst for agents of Model II and III we use $v = \pm s - 1$. With the help of this simplified system we can now make further analytic and simulative investigations into the effect of different x_c on the quantity of the dropout. If an agent enters the exploited region $x < x_c$, two behaviours are considered. In the first case, the agent would be considered dropped out and is adsorbed by the boundary, so that it has no chance of becoming part of the wave again. The second case allows the agent to randomly diffuse inside the exploited area and therefore allows the agent to enter the non-exploited region again. For both cases we perform 100 simulations of



(a) (b)

Figure 2.7: *Fraction of agents with* $x^{(i)} < x_c$ *at* $t = 0.5$. *In each figure we show an ensemble average (blue dots) over* 100 *simulations. All parameters as described in the text.*
(a) *Simulations where no comeback from* $x = x_c$ *is allowed. The dashed (red) line is a result of the theoretical analysis given by* (2.35).
(b) *Simulations where dropout agents can return. The dashed (red) line is 50% of the dropout predicted by* (2.35).

Model III for each of the considered values of x_c and measured the fraction of agents with $x^{(i)} < x_c$ at time $t = 0.05$. We use $N = 10^4$ and start all agents at $x = 0$. The ensemble averages of these results are shown in Figure 2.7 as circles. To analyse the case of an adsorbing boundary at $x = x_c$ we consider the system in the limit $N \to \infty$, which is described by the following equation (cf. (2.11))

$$\frac{\partial n}{\partial t} - \frac{\partial n}{\partial x} = \mu \frac{\partial}{\partial x}\left(\frac{\partial n}{\partial x} - n\frac{\kappa}{S}\frac{\mathrm{d}S}{\mathrm{d}x}\right) . \tag{2.33}$$

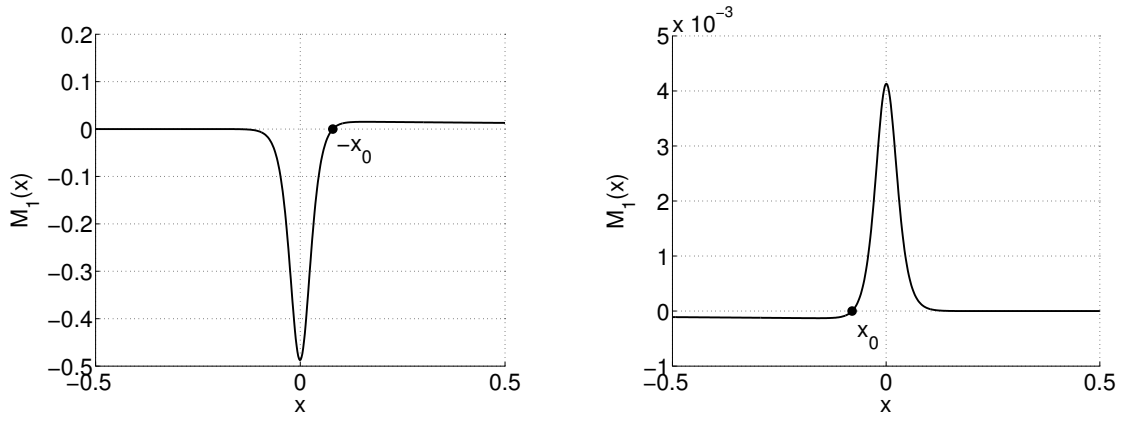
The boundary condition on the left-hand boundary can be written in the form $n(x_c) = 0$. Further conditions for $x \to \infty$ can be introduced. We look for a separable solution of the form

$$n(t, x) = \exp(-\zeta t) M(x) ,$$

where ζ is a positive constant. Plugging this ansatz into (2.33) leads to

$$\mu M'' + M' - \kappa\mu \left(M \frac{S'}{S}\right)' + \zeta M = 0 , \tag{2.34}$$

where primes denote derivatives with respect to x. For the ODE (2.34) a non-negative solution is sought that satisfies $M(x_c) = 0$ and $M(x) \to 0$ as $x \to \infty$. The general



(a) $M_1(x)$ (b) $M_2(x)$

Figure 2.8: *Solutions $M_1(x)$ and $M_2(x)$ of standing wave experiments. Parameters are $\mu = 1/60$, $\zeta = 0.5$. x_0 indicates the root the functions.*

solution for (2.34) with $\kappa = 2$ is

$$\begin{aligned} M(x) &= C_1 M_1(x) + C_2 M_2(x)\,, \qquad \text{with} \\ M_1(x) &= \frac{2\zeta\mu \exp\left(x\frac{3+\gamma}{2\mu}\right) + (2\zeta\mu - 1 - \gamma)\exp\left(x\frac{1+\gamma}{2\mu}\right)}{\left(\exp\left(\frac{x}{\mu}\right)+1\right)^2}\,, \\ M_2(x) &= \frac{2\zeta\mu \exp\left(x\frac{3-\gamma}{2\mu}\right) + (2\zeta\mu - 1 + \gamma)\exp\left(x\frac{1-\gamma}{2\mu}\right)}{\left(\exp\left(\frac{x}{\mu}\right)+1\right)^2}\,. \end{aligned}$$

In these equations we used $\gamma = \sqrt{1-4\zeta\mu}$ to simplify the notation. We assume values for $\zeta\mu$ such that γ is real and hence $0 \le \gamma \le 1$. The two solutions M_1 and M_2 are plotted in Figure 2.8 and we can see that except for a different scaling they are mirrored along the y-axis and x-axis, i.e. $-M_1(-x) \sim M_2(x)$ for all $x \in \mathbb{R}$.

We seek a solution that satisfies both boundary condition $M(x_c) = 0$ and $M(x) \to 0$ as $x \to \infty$ as well as a non-negativity condition $M(x) \ge 0$ for $x \in [x_c, \infty)$. Using these conditions we are hoping to get information about the three unknowns in the system, namely C_1, C_2 and ζ. Let us first note that both $M_1(x)$ and $M_2(x)$ decay to 0 for $x \to \infty$. Therefore the boundary condition for $x \to \infty$ is satisfied for all values of C_1 and C_2. We can, however, use the boundary condition at x_c to derive the value of C_1/C_2 as a function of ζ.

Let us now take a closer look at the limit $x \to \infty$. For large x we have

$$M_1(x) \sim \exp\left(-x\frac{1-\gamma}{2\mu}\right) \gg \exp\left(-x\frac{1+\gamma}{2\mu}\right) \sim M_2(x)\,,$$

which means that M_1 decays slower than M_2 and therefore dominates the limiting process. In order to ensure a positive solution for $M(x)$, we therefore require the integration constant C_1 to be non-negative. Hence, we also need $C_2 \geq 0$ in order to ensure a non-negative value of $M(0)$. With this argument we can see that the root of $M_2(x)$, which we will call $x_0(\zeta)$, cannot be larger than $x_c < 0$ and hence we get a restriction on ζ through

$$x_0(\zeta) = \mu \ln\left(\frac{1-\gamma-2\zeta\mu}{2\zeta\mu}\right) \leq x_c \quad \Rightarrow \quad 0 \leq \zeta \leq \zeta_c(x_c)\,, \quad \text{with}$$
$$\zeta_c(x_c) = -\frac{1}{\mu}\exp\left(-\frac{x_c}{\mu}\right)\left(1+\exp\left(-\frac{x_c}{\mu}\right)\right)^{-2} = -S'(x_c)$$

We now use this value $\zeta_c(x_c)$ to get a predicted value of the dropout $o_{\text{pred}}(x_c, t)$ via

$$o_{\text{pred}}(t, x_c) = 1 - \exp(\zeta_c(x_c)t)\,. \tag{2.35}$$

The function $o_{pred}(0.5, x_c)$ is plotted as the dashed (red) line in Figure 2.7(a). We can see that it matches well with the simulation results. The slight overestimation given by $o_{pred}(0.5, x_c)$ can be explained through the time it takes before the first agents reach the critical position x_c from the starting position at $x = 0$.

For the situation with comeback, we choose a value $\zeta = \alpha\zeta_c(x_c)$ to predict the dropout, where α is a constant. Matching this with the simulation results as shown in Figure 2.7(b) we found that $\alpha \approx 0.5$, which indicates that about 50% of agents come back into the wave after they have dropped out. This effect could be modelled by using a semi-reflective boundary [47] instead of the free diffusion zone behind the wave. Half the agents that cross the boundary would be reflected, while the other half is considered as dropped out.

2.5 Introduction of growth processes into hybrid models

Thus far we have studied hybrid versions of the classical Keller-Segel model (2.1)–(2.2). We have seen that this model necessitates a singularity in the chemotactic sensitivity to generate stationary travelling waves: a requirement that allows bacteria behind the wave to acquire biologically unrealistic speed. This singularity also leads to a dropout effect when considering hybrid versions of the classical Keller-Segel model.

The singularity requirement can be circumvented by incorporating other processes. Studies have shown that introduction of more complex nutrient terms can give rise to

travelling waves [145, 146]. An experimental system which also included two chemicals – a chemoattractant and a nutrient source – was presented in [24, 25], with stationary or transient travelling waves obtained according to the formulation of the model [19, 173]. On the other hand, the well known Fisher's equation [60] demonstrates travelling waves in systems coupling diffusion with logistic growth terms. Parabolic chemotaxis models with non-singular sensitivities but incorporating either logistic [101, 103, 121] or non-logistic [97, 147] growth terms also admit travelling wave solutions.

Because of the importance of growth terms in explaining stable travelling waves, we extend the concept of hybrid models to incorporate growth processes in this section. We start the section by presenting different possibilities for incorporating growth into the the hybrid concept presented in Section 2.3 and then present the hybrid model we want to study in more detail. We formulate corresponding continuum models in Section 2.5.3 and analyse their travelling wave properties in Section 2.5.4. In Section 2.5.5 we perform a numerical comparison of wave speeds between hybrid and continuum models.

2.5.1 Extension of the hybrid concept

Just like in Sections 2.3 and 1.4.3, we assume that the hybrid system consists of M_C continuum and M_A agent-based species. Let us additionally assume that through a density estimation the concentrations $\varrho^{(j)}$, $j = 1, \dots, M_A$, are given in addition to the concentrations c_j, $j = 1, \dots, M_C$, of the continuum species. We then assume that for each of the agent-based species a growth function $h_j(t, \mathbf{x}, \mathbf{c}, \varrho)$ is given. These functions h_j act as growth factors such that in absence of all other influences an exponential growth proportional to $\exp(h_j t)$ is achieved. We denote h_j as growth functions despite the fact that they are not restricted to positive values and can therefore shrink the mass of a species at a given point. In this section we present two different ways of incorporating these growth functions into the hybrid modelling concept.

Growth Concept I: Direct proliferation

The first growth concept is motivated by mass action kinetics and acts only on the number of agents, but not on the mass of each agent. The main idea is that each agent can produce a copy of itself in each time step. In order to obtain the exponential growth desired, the probability of agent (i, j), $i = 1, \dots, N_j$, $j = 1, \dots, M_A$,

proliferating in $[t, t + \Delta t)$ is given through

$$0 \leq \Delta t \, h_j \left(t, \mathbf{x}^{(i,j)}, \mathbf{c}(t, \mathbf{x}^{(i,j)}), \varrho(t, \mathbf{x}^{(i,j)})\right) \ll 1 \,,$$

in the case of $h_j \geq 0$. In order to allow for negative values of h_j, we assume that an agent can die with the probability

$$0 \leq -\Delta t \, h_j \left(t, \mathbf{x}^{(i,j)}, \mathbf{c}(t, \mathbf{x}^{(i,j)}), \varrho(t, \mathbf{x}^{(i,j)})\right) \ll 1 \,,$$

in the case of $h_j < 0$. In the case of proliferation, we assume that an exact copy of the proliferating agent is produced and that the total number of agents is increased by one. Death of an agent simply means that it ceases to exist and that $N(t)$ is reduced by one.

Growth Concept II: Proliferation through mass growth

In the case of biological applications, Growth Concept I is not very realistic, because newly born agents are able to reproduce again straight away. However, in biological systems, one observes a maturing phase where the individual undergoes growth before it can proliferate again. The easiest way of implementing this within the scope of the existing hybrid concept is to use the the mass $m^{(i,j)}(t)$ introduced in Section 1.4.3 to account for the growth through

$$\frac{\mathrm{d}m^{(i,j)}}{\mathrm{d}t} = h_j \left(t, \mathbf{x}^{(i,j)}, \mathbf{c}(t, \mathbf{x}^{(i,j)}), \varrho(t, \mathbf{x}^{(i,j)})\right) \,.$$

Once an agent has reached a critical mass m_p, we assume that it automatically splits into two identical agents, with the mass $m^{(i,j)}$ being equally split between the existing and the newly created agent. This process is similar to cell division [143]. In the same scope, we introduce the concept of starving agents: if the mass $m^{(i,j)}$ drops below the starvation mass m_s, the agent dies automatically. This extension has the computational advantage that the number of agents does not increase infinitely.

2.5.2 An adjusted hybrid model for bacterial chemotaxis

In this section we present the hybrid chemotaxis model that we study throughout the remainder of this chapter. As in all hybrid chemotaxis models studied thus far, we consider two species: an agent-based bacteria population and a continuous extracellular signal concentration. The model is based on Model III presented in Section 2.4.1 and in particular we conserve the update equations (2.23) for the extracellular signal (with the exception that we use $N(0)$ instead of N as normalisation variable) $S(t, x)$

as well as the update equations for the agents given in (2.30). We do, however, use a different form for the turning frequency λ that was used in [69, 173] and results in a non-singular and therefore more biologically realistic behaviour:

$$\lambda = \lambda_0 \left(1 - \frac{S\left(t, x^{(i)}(t)\right) - z_2^{(i)}(t)}{\nu + \left| S\left(t, x^{(i)}(t)\right) - z_2^{(i)}(t) \right|} \right) , \tag{2.36}$$

where $\nu \in \mathbb{R}^+$ is a positive constant that controls the strength of the chemotactic reaction. For the consumption function $k(S)$ and the growth function $h(S)$, respectively we use

$$k(S) = \beta S \, , \qquad h(S) = \alpha(S - S_c) \, , \tag{2.37}$$

where $\alpha, \beta \in \mathbb{R}^+$ are positive constants.

Numerical methods

We perform a first set of numerical simulations in order to get an idea of the behaviour of the adjusted system presented above. We simulate the system on the domain $[0, 100]$ with $L = 401$ regularly spaced gridpoints, resulting in $\Delta x = 0.25$ and $r^{(l)} = (l-1)\Delta x$, $l = 1, \ldots, L$. We initialise the extracellular signal using $S \equiv S_\infty > 0$. We start the simulation with $N(0) = N_0$ agents that are initialised at $x^{(i)}(0) = 0.5|\xi_i|$, $i = 1, \ldots, N_0$, where $\xi_i \sim \mathcal{N}(0; 1)$ are N independent normally distributed random variables. The agents are further initialised with a random velocity direction, in the state of total adaptation, i.e. $z_2^{(i)}(0) = S_\infty$ and with $m^{(i)}(0) = 1$. Reflective (no-flow) boundary conditions are employed on both ends of the domain Ω. We use $\Delta t = 10^{-3}$ and simulate the system until $t = 100$. For the coupling between continuum and agent-based components we use again a Gaussian kernel as defined in (2.21) with bandwidth $\sigma = 0.5$ and a linear interpolation method.

For all simulations, if not explicitly stated otherwise, we used the following set of parameters: $\alpha = \beta = s = 1$, $S_c = 0.5$, $S_\infty = 1$, $\lambda_0 = 10$, $\nu = 0.01$, $D_S = 0$, $t_a = 0.1$, $N_0 = 10^4$. For Growth Concept II we use $m_\mathrm{p} = 2$ and $m_\mathrm{s} = 10^{-3}$.

First simulation results

We use the methods and parameters given above to perform a first set of simulations and plot the results in Figure 2.9. Panels (a) and (c) show the shape of S and the density estimate of agents at $t = 100$, whilst panels (b) and (d) show the speed of the wave front as defined in (2.32) using $\Delta T = 0.2$. Note that the high noise seen in the plots is due to high fluctuations in the position of the wave front $\hat{x}(t)$.

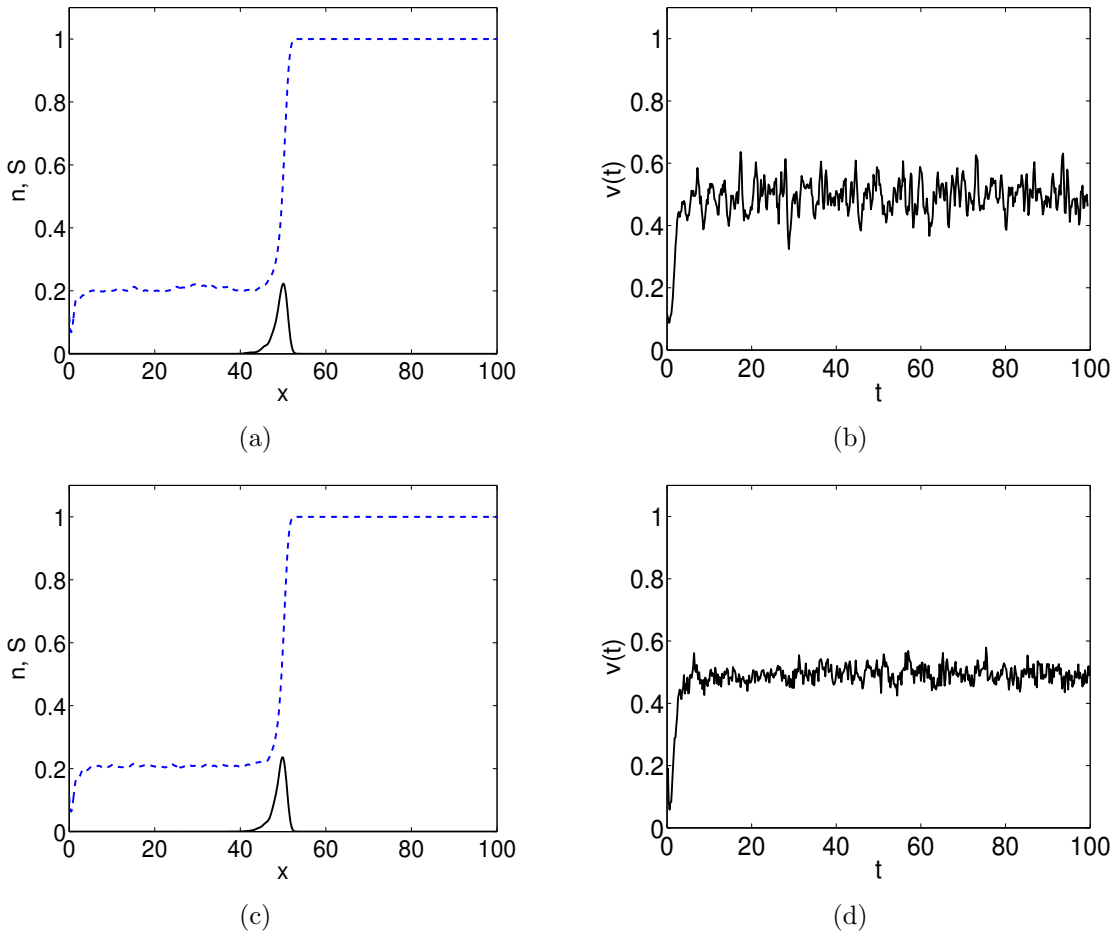


Figure 2.9: *Numerical solutions of the hybrid chemotaxis model presented in Section* 2.5.2 *for different growth concepts. Panels* (a), (c) *show the wave form at* $t = 100$. *Solid (black) line:* $n(100, x)$*; dashed (blue) line:* $S(100, x)$. *Panels* (b), (d) *show the measured speed of the wave front. All parameters as explained in the text.*
(a), (b) *Growth Concept I.* (c), (d) *Growth Concept II.*

Panels (a) and (b) in Figure 2.9 show the simulation results obtained using Growth Concept I. We can clearly see the formation of a stable travelling wave that moves with a constant speed close to $\hat{v} \approx 0.5$.

These results match closely with those for Concept II seen in panels (c) and (d) of Figure 2.9. Again, a travelling band forms and moves with a constant speed to the right of the real axis. Both, wave forms and speeds, match very well. We conclude from this that Concepts I and II present reliable methods to implement growth in hybrid models. In all simulation in the remainder of this Section we will use Concept I due to the equivalence between Concepts I and II and its easier implementation.

Influence of the growth term

To investigate the influence of the growth term on the existence of travelling waves in this system, we simulate the hybrid model presented above including ($\alpha = 1$) and excluding ($\alpha = 0$) the growth process and compare the results in Figure 2.10. In Figure 2.10(a) the position of the wave front $\hat{x}(t)$ is plotted. The dashed (red) line shows results for $\alpha = 1$ and represents a straight line, indicating a wave moving with constant speed. On the other hand, using $\alpha = 0$ (solid black line), the wave front initially moves with a similar speed but then slows down gradually.

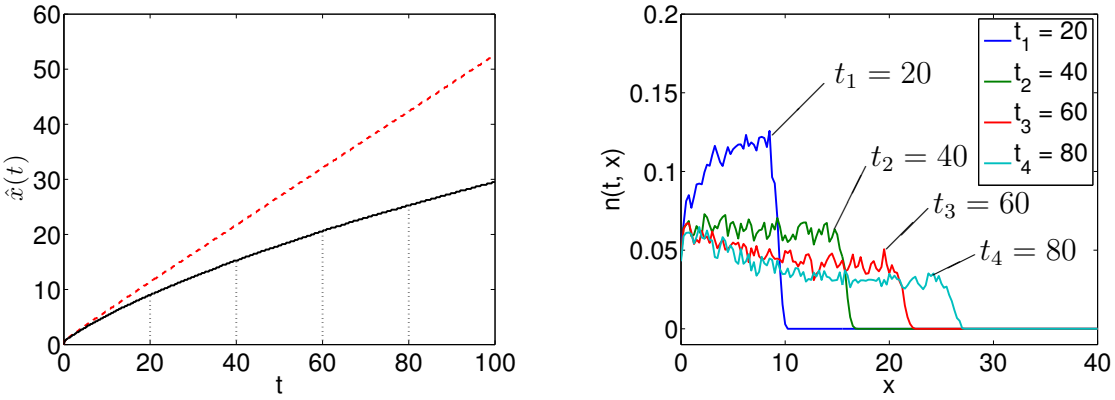
To further illustrate this slowing down effect, we plot the estimated density of bacteria at different times for the case $\alpha = 0$ in Figure 2.10(b). We clearly see that no true travelling wave forms, with many agents being left behind the wave front, leading to its slowing down. Thus, we can interpret the growth term as a stabilising factor on the wave profile: although not all agents can keep up with the wave, new agents are constantly created at the front and the agents that drop out eventually die due to starvation, resulting in a travelling band of agents.

Note, that the results for the case $\alpha = 0$ indirectly correspond to the results shown in Theorem 2.1 as the chosen form of the turning frequency λ given in (2.36) does not have a singularity and therefore should not lead to travelling waves.

Influence of signal diffusion

Thus far in this chapter all simulations excluded the possibility of signal diffusion using $D_S = 0$. Let us therefore investigate whether this is a reasonable assumption for the current model, or whether including diffusion completely changes the outcome of the model presented above.

The first step to introducing diffusion in the hybrid system is to change the update equation of extracellular signal S given in (2.23). We use the explicit Euler method



(a) (b)

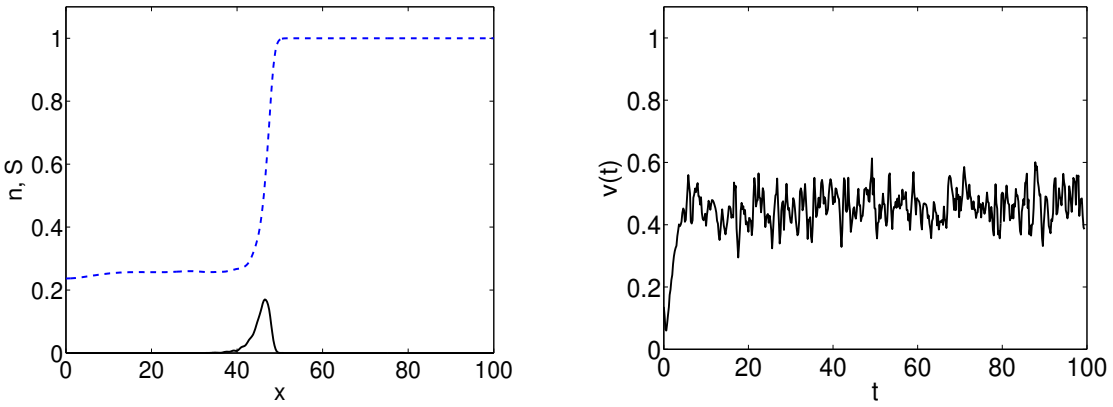
Figure 2.10: *Numerical solutions of the hybrid chemotaxis model presented in Section* 2.5.2 *including* ($\alpha = 1$) *and excluding* ($\alpha = 0$) *the growth process. All other parameters as given in the text.*
(a) *Position of wave front* $\hat{x}(t)$*. Solid (black) line:* $\alpha = 0$*; dashed (red) line:* $\alpha = 1$.
(b) *Wave form at different times for* $\alpha = 0$*. From left to right:* $t = 20, 40, 60, 80$.

to solve the reaction-diffusion equation as follows

$$
\begin{aligned}
S\left(r^{(l)}, t+\Delta t\right) =& S\left(r^{(l)}, t\right) + D_S\,\Delta t \frac{S\left(r^{(l-1)}, t\right) + S\left(r^{(l+1)}, t\right) - 2S\left(r^{(l)}, t\right)}{(\Delta x)^2} \\
& - k\left(S\left(r^{(l)}, t\right)\right)\,\Delta t \sum_{i=1}^{N(t)} K\left(r^{(l)} - x^{(i)}(t)\right)\,, \qquad l = 2\,\ldots, L-1\,, \\
S\left(r^{(1)}, t+\Delta t\right) =& S\left(r^{(1)}, t\right) + D_S\,\Delta t \frac{S\left(r^{(2)}, t\right) - S\left(r^{(1)}, t\right)}{(\Delta x)^2} \\
& - k\left(S\left(r^{(1)}, t\right)\right)\,\Delta t \sum_{i=1}^{N(t)} K\left(r^{(1)} - x^{(i)}(t)\right)\,, \\
S\left(r^{(L)}, t+\Delta t\right) =& S\left(r^{(L)}, t\right) + D_S\,\Delta t \frac{S\left(r^{(L-1)}, t\right) - S\left(r^{(L)}, t\right)}{(\Delta x)^2} \\
& - k\left(S\left(r^{(L)}, t\right)\right)\,\Delta t \sum_{i=1}^{N(t)} K\left(r^{(L)} - x^{(i)}(t)\right)\,,
\end{aligned}
$$

where the second and third equations implement the no-flow boundary conditions.

In [13] the average turning frequency of an *E. coli* bacterium is given as $\lambda \approx 1\,\mathrm{sec}^{-1}$ and the speed of movement as $s \approx 21\,\mu\mathrm{m}\,\mathrm{sec}^{-1}$, resulting in an effective diffusion of $D_n \approx 4.41\,\mathrm{cm}^2\,\mathrm{sec}^{-1}$, which is about half as fast as the diffusion of extracellular signal in water given through $D_S \approx 8.9\,\mathrm{cm}^2\,\mathrm{sec}^{-1}$ [12]. In the simulations, we have been using dimensionless values of $\lambda_0^* = 10$ and $s^* = 1$, resulting in an effective



(a) (b)

Figure 2.11: *Numerical solutions of the hybrid chemotaxis model presented in Section* 2.5.2 *including diffusion of extracellular signal ($D_S = 0.2$). Panel* (a) *shows the wave form at $t = 100$. Solid (black) line: $n(100, x)$; dashed (blue) line: $S(100, x)$. Panel* (b) *shows the measured speed of the wave front. All parameters (other than D_S) as explained in the text.*

diffusion of $D_n^* = 0.1$. Hence, the dimensionless value of diffusion for extracellular signal according to the data above should be $D_S^* \approx 0.2$.

Simulation results for the hybrid system including chemotaxis with $D_S = 0.2$ are shown in Figure 2.11. We can see that the inclusion of diffusion does not change the qualitative behaviour of the system, but does slightly change the wave form and the speed of movement. The average wave speed can be measured as $\hat{v}(t) \approx 0.45$ and is therefore about 10% slower than for $D_S = 0$. As the change is reasonably small and the qualitative behaviour is conserved, we will neglect diffusion for the remainder of this section assuming $D_S = 0$.

2.5.3 From hybrid models to macroscopic PDEs

In this section we derive macroscopic PDEs for the spatio-temporal density of bacteria $n(t, x)$ at given position $x \in \mathbb{R}$ and time $t \geq 0$. An implicit assumption of the derivation is spatial independence of bacteria, which allows formulation of a continuous mesoscopic system. We then use results from [55] to obtain the macroscopic equations. To illustrate the successive formulation of models we construct two systems of PDEs – denoted System **(A)** and System **(B)** – to be referred to in the remainder of this section.

System (A)

We define the mesoscopic densities $p^{\pm}(t, x, z_2)$ for left and right-moving bacteria, depending on their position $x \in \mathbb{R}$, their internal variable $z_2 \in \mathbb{R}$ and $t \geq 0$. If the signal profile $S \equiv S(t, x)$ was uninfluenced by bacteria, the densities $p^{\pm}$ would satisfy the following system of hyperbolic PDEs:

$$\begin{aligned} \frac{\partial p^+}{\partial t} + s\frac{\partial p^+}{\partial x} + \frac{\partial}{\partial z_2}\left(\frac{S(t,x) - z_2}{t_a} p^+\right) &= -\lambda p^+ + \lambda p^- + h(S(t,x))p^+ \,, \\ \frac{\partial p^-}{\partial t} - s\frac{\partial p^-}{\partial x} + \frac{\partial}{\partial z_2}\left(\frac{S(t,x) - z_2}{t_a} p^-\right) &= \quad \lambda p^+ - \lambda p^- + h(S(t,x))p^- \,, \end{aligned} \tag{2.38}$$

where λ is defined in (2.36). The corresponding signal update takes the form $p^{\pm}$ as

$$\frac{\partial S}{\partial t} = D_S \frac{\partial^2 S}{\partial x^2} - k(S) \int_{\mathbb{R}} (p^+(t, x, z_2) + p^-(t, x, z_2)) \, \mathrm{d}z_2 \,. \tag{2.39}$$

We denote the system of equations (2.38)–(2.39) as System **(A)**.

The system (2.38) (for the one-particle distribution) was derived by integrating the probability distribution function $p(x^{(1)}, v^{(1)}, z_2^{(1)}; x^{(2)}, v^{(2)}, z_2^{(2)}; \ldots \mid S(t, x))$ for the many-particle system, assuming that movement of individuals is biased by the signal function $S(t, x)$, but independent from each other. However, for the hybrid chemotaxis model described in Section 2.5.2, individual bacteria interact via the extracellular signal S which implies that (2.38) can only be considered as an approximation to the one particle distributions of the interacting system.

In [53], a kinetic description has been derived for a model of interacting locusts, using a modified version of the BBGKY hierarchy from the classical kinetic theory of gases [27]. The system we consider here, however, is much more complicated to analyse than the locust model studied in [53], due to the variable number of bacteria and internal variables, which is why the approximated dynamics in (2.38) are used and later verified using numerical results.

The capacity of the above mesoscopic system to generate travelling bands analogous to those observed in the hybrid model is illustrated in Figure 2.12(a)-(b). For details of the numerical method employed for this and other simulations of the continuous model, we refer to [173]. The qualitatively and quantitatively close correspondence in solutions under equivalent parameters and initial conditions corroborates the use of the above approximation.

System (B)

We consider a macroscopic model in this section. Define the macroscopic densities

$$p^{\pm}(t,x) = \int_{\mathbb{R}} p^{\pm}(t,x,z_2)\,\mathrm{d}z_2\,, \tag{2.40}$$

and let them satisfy the following system

$$\begin{aligned} \frac{\partial p^+}{\partial t} + s\frac{\partial p^+}{\partial x} &= -\lambda^+\left(\frac{\partial S}{\partial x}\right)p^+ + \lambda^-\left(\frac{\partial S}{\partial x}\right)p^- + h(S)p^+\,, \\ \frac{\partial p^-}{\partial t} - s\frac{\partial p^-}{\partial x} &= \quad \lambda^+\left(\frac{\partial S}{\partial x}\right)p^+ - \lambda^-\left(\frac{\partial S}{\partial x}\right)p^- + h(S)p^-\,, \end{aligned} \tag{2.41}$$

where the turning rates $\lambda^{\pm}$ are given by

$$\lambda^{\pm} = \lambda_0\left(1 \mp \chi\frac{\partial S}{\partial x}\right) \quad \text{with} \quad \chi = \frac{st_a}{\nu\lambda_0(1+2\lambda_0 t_a)}\,. \tag{2.42}$$

Using (2.40), equation (2.39) can be written as

$$\frac{\partial S}{\partial t} = D_S\frac{\partial^2 S}{\partial x^2} - k(S)(p^+(t,x) + p^-(t,x))\,. \tag{2.43}$$

We will denote (2.41) and (2.43) along with the definition of $\lambda^{\pm}$ in (2.42) as System **(B)**. The form (2.42) for $\lambda^{\pm}$ is derived by first linearising the turning frequency (2.36) under the assumption $|S(x) - z_2| \ll \nu$, i.e. we assume that the external signal (S) changes slowly enough such that cells are close to their fully adapted state. Results in [55, 174] show that System **(B)** is quantitatively consistent with this linearised form of System **(A)** when the gradient of the extracellular signal is sufficiently shallow such that (2.42) does not become negative.

The number of parameters of the above models can be reduced by setting s, S_∞, α, β to one through rescaling. We show this in detail for System **(B)** as follows. Rescaling the variables $S = \hat{S}S_\infty$, $p^{\pm} = \hat{p}^{\pm}\alpha S_\infty/\beta$, $t = \hat{t}/(\alpha S_\infty)$, $x = \hat{x}s/(\alpha S_\infty)$ and the parameters $S_c = \hat{S}_c S_\infty$, $\lambda_0 = \hat{\lambda}_0\alpha S_\infty$, taking (2.37) and substituting into System **(B)** we obtain, after dropping hats for notational simplicity and applying $D_S = 0$ as discussed in Section 2.5.2,

$$\begin{aligned} \frac{\partial p^+}{\partial t} + \frac{\partial p^+}{\partial x} &= -\lambda^+\left(\frac{\partial S}{\partial x}\right)p^+ + \lambda^-\left(\frac{\partial S}{\partial x}\right)p^- + (S - S_c)p^+\,, \\ \frac{\partial p^-}{\partial t} - \frac{\partial p^-}{\partial x} &= \quad \lambda^+\left(\frac{\partial S}{\partial x}\right)p^+ - \lambda^-\left(\frac{\partial S}{\partial x}\right)p^- + (S - S_c)p^-\,, \\ \frac{\partial S}{\partial t} &= -S(p^+ + p^-)\,. \end{aligned} \tag{2.44}$$

We are interested in travelling wave solutions that develop from a pointwise inoculation of cells into a domain containing uniformly distributed nutrient S. In this scenario, $p^{\pm}$ (defined as in each system) should form travelling pulses while S forms a travelling front and relevant boundary conditions will be

$$\begin{aligned} p^{\pm}, \frac{\partial p^{\pm}}{\partial x}, \frac{\partial S}{\partial x} &\to 0 \quad \text{as} \quad x \to \pm\infty\,, \\ S &\to 1 \quad\;\; \text{as} \quad x \to +\infty\,, \\ S &\to S_- \quad\;\; \text{as} \quad x \to -\infty\,. \end{aligned} \tag{2.45}$$

Note that S_- is currently unknown; we determine its value in the travelling wave analysis of Section 2.5.4. Since $p^{\pm}$ and S are physical quantities, we search for non-negative travelling wave solutions, i.e.

$$p^{\pm} \geq 0, \quad S \geq 0.$$

In Figures 2.12(c), (d) we show that System **(B)** yields travelling waves with constant speed. We, however, already see a quantitative difference between the wave speeds of Systems **(A)** and **(B)** that we investigate further in Section 2.5.5.

It is clear that a travelling wave of this form cannot exist for $S_c \geq 1$ (extinction of bacteria) or for $S_c \leq 0$ (infinite growth) and we therefore only consider systems that satisfy $S_c \in (0, 1)$. In the next section we analyse System **(B)** with respect to travelling wave solutions in order to obtain further insight. To do that, we use the rescaled system (2.44).

2.5.4 Travelling wave analysis

In this section we first apply the standard travelling wave ansatz to System (2.44) and derive a necessary condition for the existence of non-negative travelling wave solutions. We then identify an invariant manifold and use this to reduce the resulting ODE system to two components. Finally we use phase plane methods to analyse the existence and properties of travelling wave solutions.

A necessary condition for the existence of travelling wave solutions

Let us apply the travelling wave ansatz $p^{\pm}(t, x) = p^{\pm}(\xi) = p^{\pm}(x - ct)$ and $S(t, x) = S(\xi) = S(x - ct)$, where c is the unknown wave speed [119, 120]. System (2.44) becomes

$$\begin{aligned} (1-c)(p^+)' &= -\lambda_0\,(1 - \chi\,S')\,p^+ + \lambda_0\,(1 + \chi\,S')\,p^- + (S - S_c)p^+\,, \\ -(1+c)(p^-)' &= \;\;\lambda_0\,(1 - \chi\,S')\,p^+ - \lambda_0\,(1 + \chi\,S')\,p^- + (S - S_c)p^-\,, \\ -cS' &= -S(p^+ + p^-)\,, \end{aligned} \tag{2.46}$$

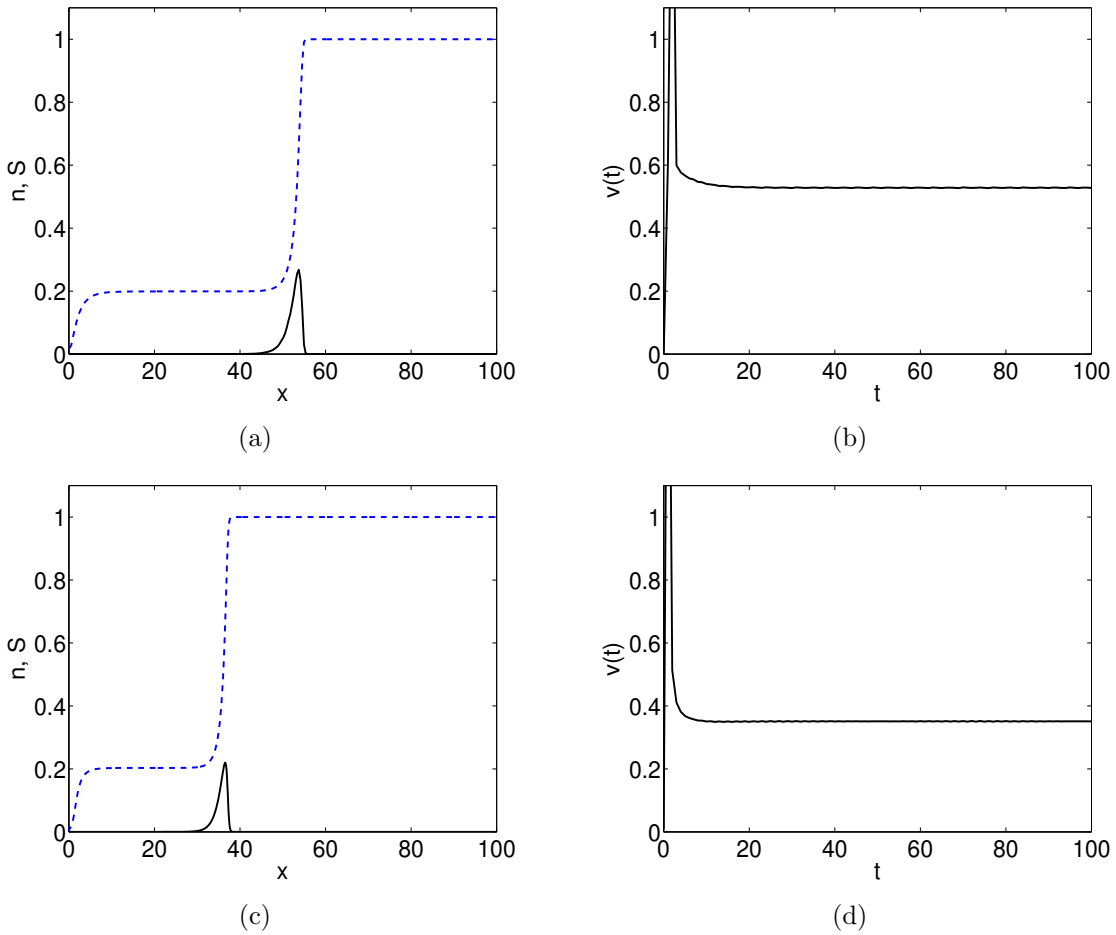


Figure 2.12: *Numerical solutions of PDE systems* ***(A)*** *and* ***(B)****. Panels* (a), (c) *show the wave form at* $t = 100$*. Solid (black) line:* $n(100, x)$*; dashed (blue) line:* $S(100, x)$*. Panels* (b), (d) *show the measured speed of the wave front. All parameters as explained in the text.*
(a), (b) *PDE system* ***(A)*** *given through* (2.38)–(2.39).
(c), (d) *PDE system* ***(B)*** *given through* (2.44) *and* (2.42).
Note that the high value of the velocities at small t *in panels (b) and (d) is an artefact of the way velocity is measured in these PDE models.*

where the primes denote derivatives with respect to the travelling wave variable ξ. Note that any point on the S-axis is a steady state of the system (2.46) and that linear stability of such a steady state, $(p^+, p^-, S) = (0, 0, S_*)$, is governed by the eigenvalues of the matrix $A^{-1}B$, where

$$A = \begin{pmatrix} 1-c & 0 & 0 \\ 0 & -1-c & 0 \\ 0 & 0 & -c \end{pmatrix}, \qquad B = \begin{pmatrix} -\lambda_0 + S_* - S_c & \lambda_0 & 0 \\ \lambda_0 & -\lambda_0 + S_* - S_c & 0 \\ -S_* & -S_* & 0 \end{pmatrix}.$$

The eigenvalues of $A^{-1}B$ are

$$\zeta_1 = 0, \qquad \zeta_{2,3} = \frac{c(-\lambda_0 + S_* - S_c) \pm \sqrt{\Delta_1(S_*)}}{1-c^2}, \tag{2.47}$$

where

$$\Delta_1(S_*) = c^2\lambda_0^2 + (S_* - S_c - 2\lambda_0)(S_* - S_c). \tag{2.48}$$

Under the boundary conditions (2.45) we look for non-negative solutions to (2.46) connecting steady states $(p^+, p^-, S) = (0, 0, S_-)$ and $(p^+, p^-, S) = (0, 0, 1)$. To admit such a solution the latter must be a stable node, since a stable spiral would imply negative values for $p^\pm$. Hence, a necessary condition is $\Delta_1(1) \geq 0$, which is equivalent to

$$c \geq c^* = \frac{1}{\lambda_0}\sqrt{(2\lambda_0 - 1 + S_c)(1 - S_c)}. \tag{2.49}$$

Given $2\lambda_0 > (1 - S_c)$ it is easy to show that $c^* \in [0, 1]$.

Theorem 2.3. *A necessary condition for the existence of non-negative travelling wave solutions of the system* (2.44) *is*

$$2\lambda_0 > (1 - S_c). \tag{2.50}$$

The above condition is reasonable, as we expect the run duration to occur on a much faster time scale than proliferation processes.

Dimension reduction

Let us now perform a change of variables by introducing the cell density $n = p^+ + p^-$ and the cell flux[2] $j = p^+ - p^-$. The travelling wave system (2.46) can then be written as

$$-cn' + j' = (S - S_c)n, \tag{2.51}$$

$$-cj' + n' = 2\lambda_0\chi\, S'n + (S - S_c - 2\lambda_0)j, \tag{2.52}$$

$$-cS' = -Sn, \tag{2.53}$$

[2]This definition is consistent with $j = s(p^+ - p^-)$ used in Section 1.3.2, because we have $s = 1$.

where the boundary conditions for this system are

$$\begin{aligned} n, j, \frac{\partial n}{\partial x}, \frac{\partial j}{\partial x}, \frac{\partial S}{\partial x} \to 0 \quad & \text{as} \quad \xi \to \pm\infty\,, \\ S \to 1 \quad & \text{as} \quad \xi \to +\infty\,, \\ S \to S_- \quad & \text{as} \quad \xi \to -\infty\,. \end{aligned}$$

From (2.53), we have $Sn = cS'$ and, hence, $n = c(\ln S)'$. Substituting into (2.51) we obtain

$$-cn' + j' = cS' - cS_c(\ln S)'\,.$$

Integrating and applying the boundary conditions at $\xi \to +\infty$, an invariant manifold of the problem is given by

$$-cn + j = c(S-1) - cS_c \ln S\,.$$

With the definition $f(S) \equiv S - 1 - S_c \ln S$, we obtain $j = cn + cf(S)$, which can be used to eliminate j from system (2.51)–(2.53). For $c \neq 1$ we can solve for n' and obtain the reduced system

$$n' = \frac{c}{1-c^2}\left[\frac{2\lambda_0\chi\, Sn^2}{c^2} + 2\,n(S - S_c - \lambda_0) + (S - S_c - 2\lambda_0)f(S)\right], \tag{2.54}$$

$$S' = \frac{1}{c}Sn\,. \tag{2.55}$$

For $c = 1$, we obtain

$$\begin{aligned} n &= \frac{\lambda_0 - S + S_c - \sqrt{(\lambda_0 - S + S_c)^2 - 2\lambda_0\chi\, S(S - S_c - 2\lambda_0)f(S)}}{2\lambda_0\chi\, S}\,, \\ S' &= Sn\,, \end{aligned} \tag{2.56}$$

where we chose the solution to the quadratic equation for n that satisfies the boundary conditions $n \to 0$ as $\xi \to \pm\infty$.

It can be easily shown that $f(S) = 0$ has two solutions in the region $(0, 1]$ for all $S_c \in (0, 1)$ as follows. Since $f'(S) = 1 - S_c/S$, $f(S)$ is monotonically decreasing for $S \in (0, S_c)$ and monotonically increasing for $S \in (S_c, 1]$. With $f(1) = 0$, this implies $f(S_c) < 0$ and, using $f(S) \to \infty$ for $S \to 0$, we obtain the existence and uniqueness of the second root of $f(S) = 0$: we call it $S_1 \in (0, S_c)$. The existence of S_1 and the negativity of $f(S)$ for $S \in (S_1, 1)$, together with the condition $2\lambda_0 > 1 - S_c$, implies that n as given in (2.56) is positive everywhere, and that the given solution therefore satisfies the non-negativity condition.

Steady states and their linear stability

Using the two roots of $f(S) = 0$ and under the condition (2.50), it is clear that there are two steady states of the system (2.54)-(2.55): $(n, S) = (0, 1)$ and $(n, S) = (0, S_1)$. Linearising the system (2.54)-(2.55) about its steady states generates a system of the form

$$\begin{pmatrix} n \\ S \end{pmatrix}' = A \begin{pmatrix} n \\ S \end{pmatrix},$$

where, for the general steady state $S_* \in \{S_1, 1\}$, we have

$$A = \begin{pmatrix} \dfrac{2c}{1-c^2}\,(S_* - S_c - \lambda_0) & \dfrac{c}{1-c^2}(S_* - S_c - 2\lambda_0)\dfrac{S_* - S_c}{S_*} \\ \dfrac{S_*}{c} & 0 \end{pmatrix},$$

with

$$\operatorname{trace} A = \frac{2c}{1-c^2}\,(S_* - S_c - \lambda_0)\,, \quad \det A = -\frac{1}{1-c^2}(S_* - S_c - 2\lambda_0)(S_* - S_c)\,.$$

The eigenvalues of A are identical to $\zeta_{2,3}$ as given in (2.47). The steady state $(0, 1)$ is therefore a stable node for all $c \in (c^*, 1)$ with c^* as defined in (2.49). Similarly, it can be seen that the steady steady $(0, S_1)$ is a saddle point. The eigenvectors corresponding to the eigenvalues $\zeta_{2,3}$ take the form

$$\mathbf{v}_{1,2} = \left(\zeta_{2,3}\,,\quad \frac{S_*}{c}\right)^T.$$

In the $n - S$ plane, the slopes of the eigenvectors are given by

$$k_{1,2}(c) = \frac{\zeta_{2,3} c}{S_*}\,.$$

For the steady state $(n, S) = (0, 1)$ this slope can be written in the form

$$k_{1,2}(c) = \frac{c^2\lambda_0^2}{1 - S_c - \lambda_0 \mp \sqrt{\Delta}}\,, \tag{2.57}$$

where we define $\Delta = c^2\lambda_0^2 + (1 - S_c - 2\lambda_0)(1 - S_c)$ similarly to (2.48).

Case I: No chemotaxis ($\nu = \infty$)

We first consider the case where the chemotactic sensitivity χ (given by (2.42)) vanishes, i.e cells do not respond chemotactically to changes in S. Here, travelling waves

are generated solely through proliferation of bacteria at the wave front. To understand the wave behaviour we perform a phase plane analysis for the ODE system (2.54)–(2.55). Using $\nu = \infty$ (i.e. $\chi = 0$), it reduces to

$$\begin{aligned} n' &= \frac{c}{1-c^2}\Big[2n(S - S_c - \lambda_0) + (S - S_c - 2\lambda_0)f(S)\Big]\,, \\ S' &= \frac{1}{c}Sn\,. \end{aligned} \tag{2.58}$$

Thus, the slope of a trajectory in the $n - S$ plane can be written as

$$\frac{\mathrm{d}n}{\mathrm{d}S} = \frac{c^2}{1-c^2}\frac{2n(S - S_c - \lambda_0) + (S - S_c - 2\lambda_0)f(S)}{Sn}\,.$$

Additionally, an expression for the $n-$nullcline Λ_n is given by

$$n = -\frac{S - S_c - 2\lambda_0}{2(S - S_c - \lambda_0)}f(S)\,,$$

and the S-nullcline is simply

$$n = 0, \quad \text{or} \quad S = 0\,.$$

Let us now show that travelling waves exist for the reduced system (2.58).

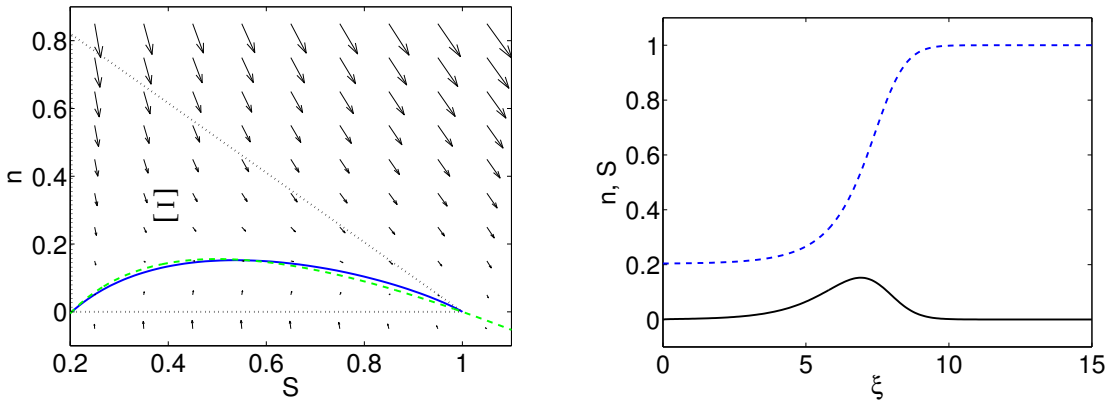
Theorem 2.4. *For the case $\chi = 0$ (which is equivalent to $\nu = \infty$), a unique travelling wave solution for system* (2.44) *exists for all $c \in (c^*, 1)$.*

Proof. For any $c \in (c^*, 1)$ we can define a region Ξ (see Figure 2.13(a)), enclosed by the line $n = k_2(S - 1)$ (with k_2 defined in (2.57)), the S-nullcline $n = 0$ and the line $S = S_1$. We will first show that Ξ is an invariant region of the system (2.44). Since S is non-decreasing everywhere in Ξ and n' is non-negative for $n = 0$ and $S \in [S_1, 1]$, we need only to show that the direction field on the segment $\Lambda_1 = \{(n, S) : n = k_2(S - 1), S \in [S_1, 1)\}$ points from the top half of the plane above this segment towards the bottom. Since S is strictly increasing we require

$$\left.\frac{\mathrm{d}n}{\mathrm{d}S}\right|_{\Lambda_1} \;\le\; k_2 \;\le\; 0\,.$$

Indeed,

$$\begin{aligned} \frac{1-c^2}{c^2}\left.\frac{\mathrm{d}n}{\mathrm{d}S}\right|_{\Lambda_1} &= 2\frac{S - S_c - \lambda_0}{S} + \frac{(S - S_c - 2\lambda_0)f(S)}{S(S-1)k_2} \\ &= 2\frac{S - S_c - \lambda_0}{S} + \frac{(S - S_c - 2\lambda_0)f(S)}{S(S-1)c^2\lambda_0^2}(1 - S_c - \lambda_0 + \sqrt{\Delta}) \\ &\le 2\frac{S - S_c - \lambda_0}{S} + \frac{(S - S_c - 2\lambda_0)(1 - S_c)}{Sc^2\lambda_0^2}(1 - S_c - \lambda_0 + \sqrt{\Delta})\,, \end{aligned}$$



(a) (b)

Figure 2.13: *Illustration of the travelling wave solution calculated using the ODE system* (2.58) *for* $\lambda_0 = 10$, $c = c^* = 0.3122$ *and* $S_c = 0.5$.
(a) *Trajectory of travelling wave solution. Solid (blue) line: trajectory; dashed (green) line: n-nullcline; dotted (black) line: circumference of invariant region* Ξ *introduced in the proof of Theorem* 2.4.
(b) *Travelling wave solution in* ξ. *Solid (black) line:* n, *dashed (blue) line:* S.

where we used (2.57) in the first step and the relation $f(S)/(S-1) \leq 1 - S_c$ for all $S \in [S_1, 1]$. Using the fact that k_2 and $(S - S_c - 2\lambda_0)$ are negative, we can use the definition of c^* and the fact that $S \leq 1$ to obtain

$$\begin{aligned}
\frac{1-c^2}{c^2}\left.\frac{\mathrm{d}n}{\mathrm{d}S}\right|_{\Lambda_1} &\leq 2\frac{S - S_c - \lambda_0}{S} - \frac{2\lambda_0 + S_c - S}{S(2\lambda_0 + S_c - 1)}(1 - S_c - \lambda_0 + \sqrt{\Delta}) \\
&= \frac{-2\lambda_0^2 + 3\lambda_0(S - S_c) - (S - S_c)(1 - S_c)}{S(2\lambda_0 + S_c - 1)} \\
&\quad - \frac{2\lambda_0 + S_c - S}{S(2\lambda_0 + S_c - 1)}\sqrt{\Delta} \\
&\leq \frac{-2\lambda_0^2 + 2\lambda_0(S - S_c) + \lambda_0(1 - S_c) - (S - S_c)(1 - S_c)}{S(2\lambda_0 + S_c - 1)} \\
&\quad - \frac{2\lambda_0 + S_c - S}{2\lambda_0 + S_c - 1}\sqrt{\Delta} \\
&= -\frac{(2\lambda_0 + S_c - 1)(\lambda_0 + S_c - S)}{S(2\lambda_0 + S_c - 1)} - \frac{2\lambda_0 + S_c - S}{2\lambda_0 + S_c - 1}\sqrt{\Delta} \\
&\leq -\lambda_0 + 1 - S_c - \sqrt{\Delta} = \frac{1 - c^2}{c^2}k_2\,,
\end{aligned}$$

where we used $S \leq 1$ throughout the derivation. We can therefore conclude that Ξ is an invariant region of the system (2.44). Noting that at the steady state $(n, S) = (0, S_1)$ the unstable manifold has a positive slope ($k_{1,2} = \zeta_{2,3}c/S_*$), i.e. it points into the region Ξ, and using the fact that S is strictly increasing inside Ξ for $n > 0$ we can

conclude that, for each $c \in (c^*, 1)$, there is a heteroclinic orbit starting from $(0, S_1)$ and finishing at $(0, 1)$, corresponding to a travelling wave solution of the PDE system (2.44). □

Case II: Increasing chemotaxis ($0 < \nu < \infty$)

Decreasing ν corresponds to an increase in the chemotactic sensitivity χ in the ODE system (2.54)–(2.55) and the slope of trajectories in the $n - S$ plane is determined by

$$\frac{\mathrm{d}n}{\mathrm{d}S} = \frac{c^2}{1-c^2} \frac{2n(S - S_c - \lambda_0) + (S - S_c - 2\lambda_0) f(S)}{Sn} + \frac{2\lambda_0 \chi}{1-c^2} n \,.$$

It is noted that the above slope is larger than that for the non-chemotaxis case within the region of interest $n > 0$. Due to this increase the region Ξ for the proof of Theorem 2.4 is no longer invariant for this system and a travelling wave solution to (2.44) does not necessarily exist for all $c \in (c^*, 1)$. The n-nullcline for the full ODE system (2.54)–(2.55) is given as the solution of the quadratic equation

$$\frac{2\lambda_0 \chi S}{c} n^2 + 2c(S - S_c - \lambda_0) n + c(S - S_c - 2\lambda_0) f(S) = 0 \,.$$

For a given wave speed c, the n-nullcline can therefore be calculated as

$$n = \frac{c}{2\lambda_0 \chi S} \left[c(\lambda_0 + S_c - S) \pm \sqrt{\Delta_2(S)} \right] ,$$

with

$$\Delta_2(S) = c^2(\lambda_0 + S_c - S)^2 - 2\lambda_0 \chi \, S(S - S_c - 2\lambda_0) f(S) \,.$$

We can see that $\Delta_2(S) \to -\infty$ as $S \to \infty$ due to its leading order term $-2\lambda_0 \chi \, S^3$. Therefore, as S becomes large, no n-nullcline exists and n' is positive everywhere. Additionally, $\Delta_2(S)$ might have further roots and, in particular, $\Delta_2(S)$ might be negative in parts (or the whole) of region $S \in [S_1, 1]$. This again means that n is strictly growing in these parts of the domain.

We detect three different types of behaviours of trajectories starting close to $(n, S) = (0, S_1)$, plotted in Figure 2.14. In particular, we can see each of these behavioural types for different values of χ and despite different configurations of the nullclines. In panels (a) and (b) of Figure 2.14 we present the case of a diverging solution. Examining ODE (2.54), we observe that for large n, n grows quicker than $\mathcal{O}(n^2)$ and the divergence can be identified as a finite-time blow-up. In the second case, depicted in panels (c) and (d) of Figure 2.14, the trajectory converges to the steady state $(0, 1)$, but does so after entering the region $S > 1$ and thereafter the

region $n < 0$. Note that the steady state $(0, 1)$ is still a stable node in this case and that this overshoot is therefore not a spiralling effect. Since these trajectories do not correspond to a non-negative solution of the ODE system (2.54)–(2.55), they do not represent biologically relevant travelling wave solutions to the original problem. The last case, presented in panels (e) and (f) of Figure 2.14, corresponds to an acceptable solution and is characterised by the convergence to $(0, 1)$ without crossing the line $S = 1$.

Case III: Infinite chemotactic sensitivity ($\nu = 0$)

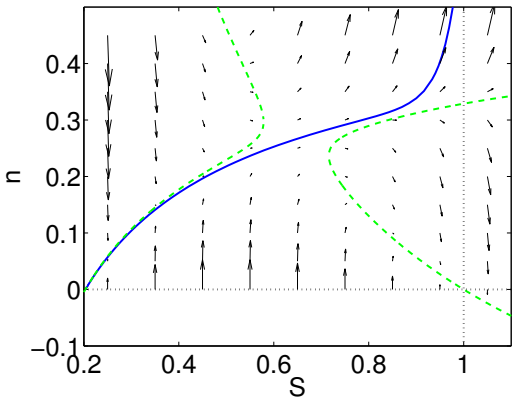
As ν decreases further, we observe that the minimal wave speed necessary to allow a non-negative travelling wave solution of (2.44) increases. In the limit $\nu \to 0$, the ODE system (2.54)–(2.55) undergoes blow-up in finite time for all values of $c \in (0, 1)$. This is because when ν goes to 0, χ goes to infinity, and thus the first term on the right-hand side of (2.54), which is quadratic in n, becomes the leading order term of (2.54). However, in this limit the linearisation assumption leading to these ODEs and the system (2.44) is no longer valid and we must consider the original turning frequency as defined in (2.36). In the limit $\nu \to 0$ the turning rate in the hybrid model becomes

$$\lambda = \begin{cases} 0\,, & S - z_2 > 0\,, \\ 2\lambda_0\,, & S - z_2 < 0\,. \end{cases}$$

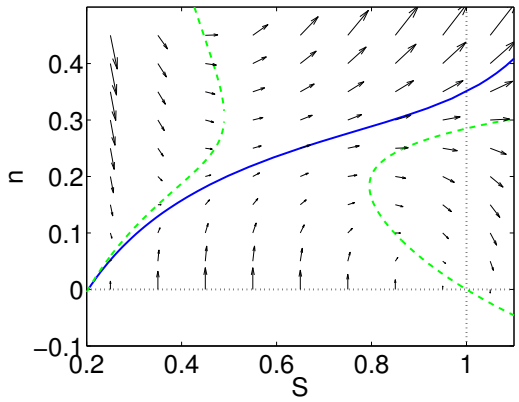
Hence, bacteria moving in a favourable direction do not turn, indicating that the wave speed achieved in this limit should evolve to $c = s = 1$. In [173] it was shown, for a slightly different turning kernel, that travelling waves can exist even without growth terms and that their wave speed satisfies $c = s$.

2.5.5 Computational analysis of the wave speed

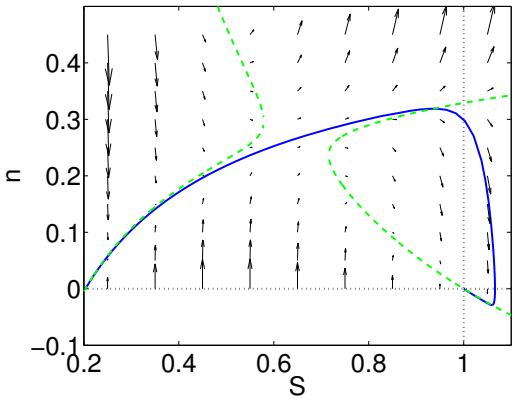
In this section we computationally compare wave speeds from the hybrid model with those of the fully continuous models. Specifically, we investigate the regimes in which the latter provide an acceptable insight into the travelling wave behaviour of the hybrid model, and where they differ. We begin by investigating the non-chemotaxis case, where the minimum wave speed c^* for the continuum systems was determined in (2.49). We show how the wave speed depends on the value of ν, and correspondingly the chemotactic sensitivity χ in the macroscopic model.



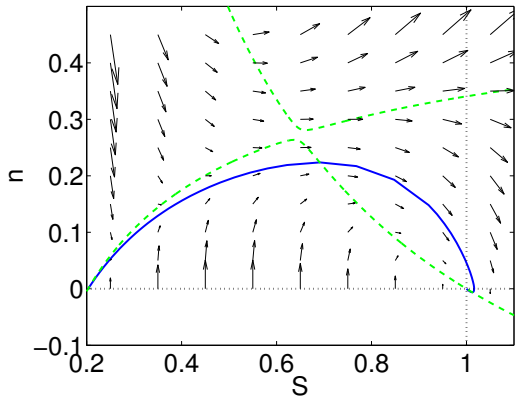
(a) $\chi = 1, c = 0.5884$



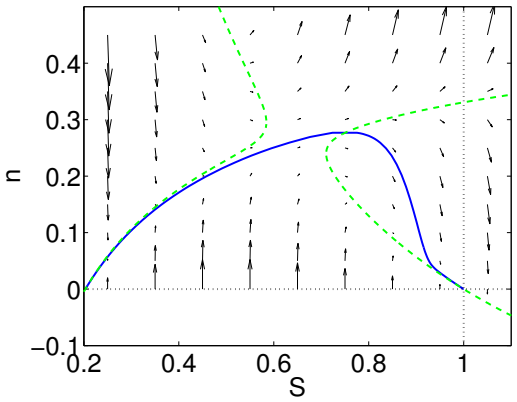
(b) $\chi = 0.3, c = 0.3$



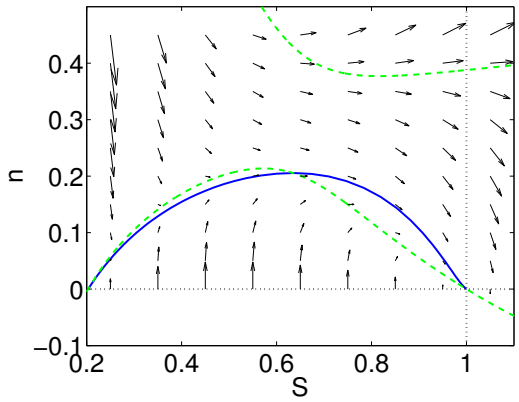
(c) $\chi = 1$, $c = 0.5885$



(d) $\chi = 0.3$, $c = 0.328$



(e) $\chi = 1$, $c = 0.59$



(f) $\chi = 0.3$, $c = 0.35$

Figure 2.14: *Trajectories of the ODE system* (2.54)–(2.55) *that highlight the three different cases discussed in the text. Parameters in all plots are* $\lambda_0 = 10$, $S_c = 0.5$. *Solid (blue) lines: trajectories; dashed (green) lines: n-nullclines; dotted (black) lines:* $n = 0$ *and* $S = 1$.

Case I: No chemotaxis ($\nu = \infty$)

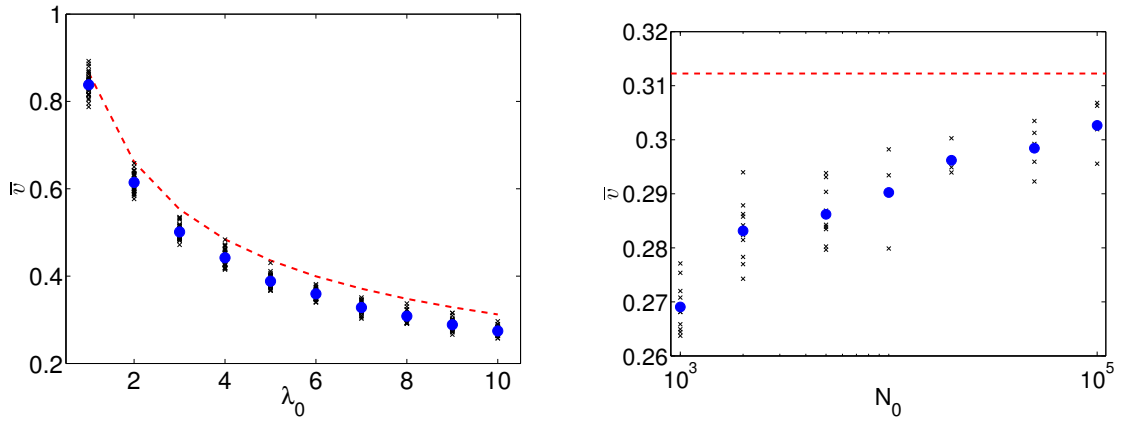
In Section 2.5.4 we analysed the macroscopic PDEs in the absence of chemotaxis. Travelling wave solutions were shown to exist for all wave speeds $c \in (c^*, 1)$, with c^* determined by (2.49). In Figure 2.15(a), (2.49) is illustrated as a function of λ_0; we note that wave speeds determined through simulation of the PDE systems correspond exactly (to accuracy of the numerical approximation) with the analytical wave speeds. We now numerically investigate the wave speed for the case $\chi = 0$ in the hybrid model.

We use the same parameters as for the simulations in Section 2.5.2. We simulate the system until the time t_{end} when the value of S at $x = 60$ falls below 0.5. We estimate the wave speed using the time t_{st} when S at $x = 20$ falls below 0.5 through $\bar{v} = 40/(t_{\text{end}} - t_{\text{st}})$.

The measured wave speed for varying λ_0 is illustrated in Figure 2.15(a), along with c^* as predicted from the travelling wave analysis. While the relationship is similar in shape, we note that at all values of λ_0 tested the measured wave speed lies below the analytical value c^*. In the literature it has been observed that inaccuracies in numerical schemes can lead to an increase in wave speeds [139], therefore rendering the lower wave speed seen in Figure 2.15(a) as counterintuitive.

Nevertheless, we can provide the following explanation for the differing values in the continuum and hybrid models. For the zero-chemotaxis case, wave generation and movement is solely determined by growth ahead and death behind the wave. In the continuum model an outermost 'fractional bacteria population' can extend significantly beyond the wave front, since some proportion of the initial population never turns left, and hence far into the region where S is very close to its initial value of 1. Yet this fractional population undergoes exponential growth ($\partial p^{\pm}/\partial t \approx (1-S_c)p^{\pm}$), thereby seeding expansion of the population towards the right of the real axis. The finite/discrete nature of the hybrid model precludes any fractional bacterium: the forward 'tail' is necessarily finite and growth will not occur beyond the outermost individual.

For the above explanation to hold we would expect a dependence of the measured wave speed on the initial number of bacteria N_0: continuous densities provide a closer approximation under larger numbers of bacteria and we would expect convergence in the wave speed to c^*. Simulations in Figure 2.15(b) demonstrate this property, corroborating the above interpretation.



(a) (b)

Figure 2.15: *Measured wave speed in the hybrid model. (Black) crosses: individual simulations; (blue) dots: ensemble averages; dashed (red) line: c^* given by* (2.49)*. Parameters are as described in the text.*
(a) *Wave speed as a function of* λ_0 *for* $N_0 = 10^3$*.*
(b) *Wave speed as a function of* N_0 *for* $\lambda_0 = 10$*.*

Case II: Increasing chemotaxis ($0 < \nu < \infty$)

In the second set of numerical experiments we measure the dependency of the wave speed on the critical parameter ν, i.e. we determine the effect of increasing chemotaxis as ν decreases. We compare the results measured for the hybrid system with the continuous Systems **(A)** and **(B)**.

We use the same parameters as before and results are shown in Figure 2.16. The results demonstrate the regimes where correspondence across the varying modelling levels occurs: while the hybrid model (blue dotted line) corresponds well with its closest continuous version (mesoscopic System **(A)**, red solid line) over a wide range of ν, it only corresponds with System **(B)** (black dashed line) for larger ν, diverging as ν decreases. Note that the turning rate (2.42) used for System **(B)** becomes negative at small values of ν and we limit the range of ν studied accordingly.

At larger ν all three models converge to a value close to c^* as ν grows: in this regime the main assumption proposed for the linearisation ($|S(x) - z_2| \ll \nu$) holds and we obtain good quantitative agreement. While this assumption becomes less acceptable as we decrease ν, leading to the divergent behaviour described above, we note that all models show the same qualitative agreement: increasing chemotactic responses leads to an increase in the wave speed. Note that the results for System **(B)** can be identically replicated using the ODE system (2.54)–(2.55) and a search algorithm for the smallest value of c that admits a non-negative solution to the system.

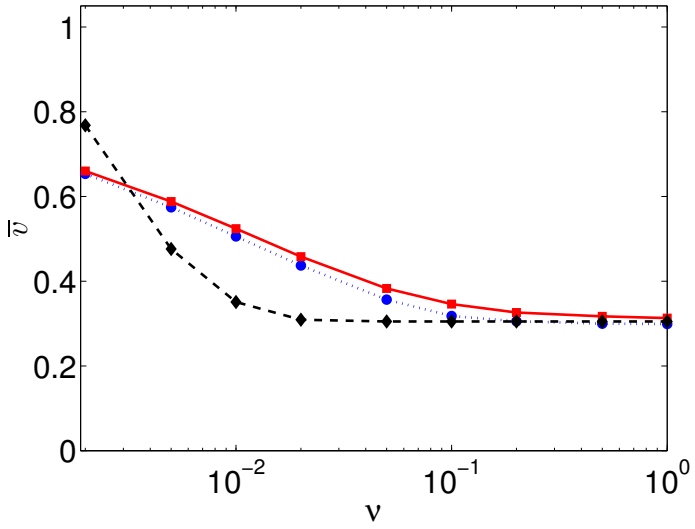


Figure 2.16: *Comparison between wave speeds of the various models in dependence of ν. Dotted (blue) line: hybrid model; solid (red) line: mesoscopic System* ***(A)****; dashed (black) line: linearised System* ***(B)****. Parameters are as described in the text.*

These numerical experiments demonstrate that chemotaxis has a significant effect on the speed of movement and that the waves cannot solely be explained by growth terms. Rather, we interpret birth and death processes as stabilisers to what would otherwise be transient waves [66, 173]. This interpretation is in agreement with the results presented in Figure 2.10, as the initial wave speed for the system without growth seems to be similar to the wave speed of the system including growth terms.

2.5.6 Oscillations in the wave speed

An additional observation we made during the numerical experiments of the hybrid model is that for increasing values of the adaptation time t_a, the wave speed starts to differ strongly from the mesoscopic System **(A)**, an effect that we identified to be due to oscillations in the behaviour of the wave. In Figure 2.17(a) we present an example of strongly oscillating wave speeds (where the wave speed is measured as rate of change of the average position of bacteria). This example occurred for the parameters $S_c = 0.5$, $\lambda_0 = 10$, $\nu = 0.001$ and $t_a = 4$. We can also clearly see that the wave speed is correlated to the current number of agents in the system. In the literature similar effects of oscillating waves in stochastic models have been observed [114, 130].

In Figure 2.17(b), we present the form of the wave at different times throughout the simulation. It is clearly visible that the shape differs significantly at different

times. One reason these oscillations occur when t_a is very high is that a bacterium that happens to be in front of the wave experiences a very high value of S, whilst its internal dynamics only adapt very slowly. This, in combination with the low value of ν, leads to a bacterium that does not switch direction for a long time and will proliferate at a high rate. This implies that a spike of bacteria forms in front of the wave that moves faster than the rest of the wave. We can clearly see such a spike in the left-most waveform in Figure 2.17(b). Once the frontrunning bacterium and its copies have turned, the wave goes into a reordering phase (second and third waveform), until, eventually, a new spike emerges (4th waveform).

In Figure 2.17(c) we plot the wave speed over time for a smaller value of t_a. We can see that the oscillations are less severe and more frequent than in Figure 2.17(a), which is in agreement with the explanation above. As we decrease t_a the frontrunning bacteria will adapt quicker to their surroundings and are thereby more likely to turn. We show the influence of changing N_0 on the oscillating behaviour in Figure 2.17(d). The oscillations seem to occur with a similar frequency but more regular to those before, which can be explained by the increased likelihood of frontrunning bacteria with a higher number of agents and reduced noise in the system.

2.6 Discussion

In this chapter we have extended the hybrid modelling framework formulated in Section 1.4.3 for the special use on a number of species with different modelling frameworks sharing the same domain. We present different ways of coupling the PDE and the agent-based species as well as how to incorporate growth processes into the agent-based system. All concepts were illustrated using the example of bacterial chemotaxis and in particular the movement of *E. coli* bacteria in reaction to an extracellular chemoattractant.

The main result of the study of hybrid Keller-Segel models is that travelling waves still form, but that in contrast to the classical model these waves are not constant in speed and agents are lost in the back of the wave. A more detailed study of this dropout effect revealed that stochastic effects in the region of low copy numbers behind the wave render continuum assumptions as inaccurate. Comparing different kernel density estimations for the coupling, we can see that using a different space scale for the kernel than for the gridpoints can result in a method that is stable in the limit $\Delta x \to 0$, whilst shrinking the kernel proportionally to Δx results in a system break down. This additional space scale could be interpreted as the size of an

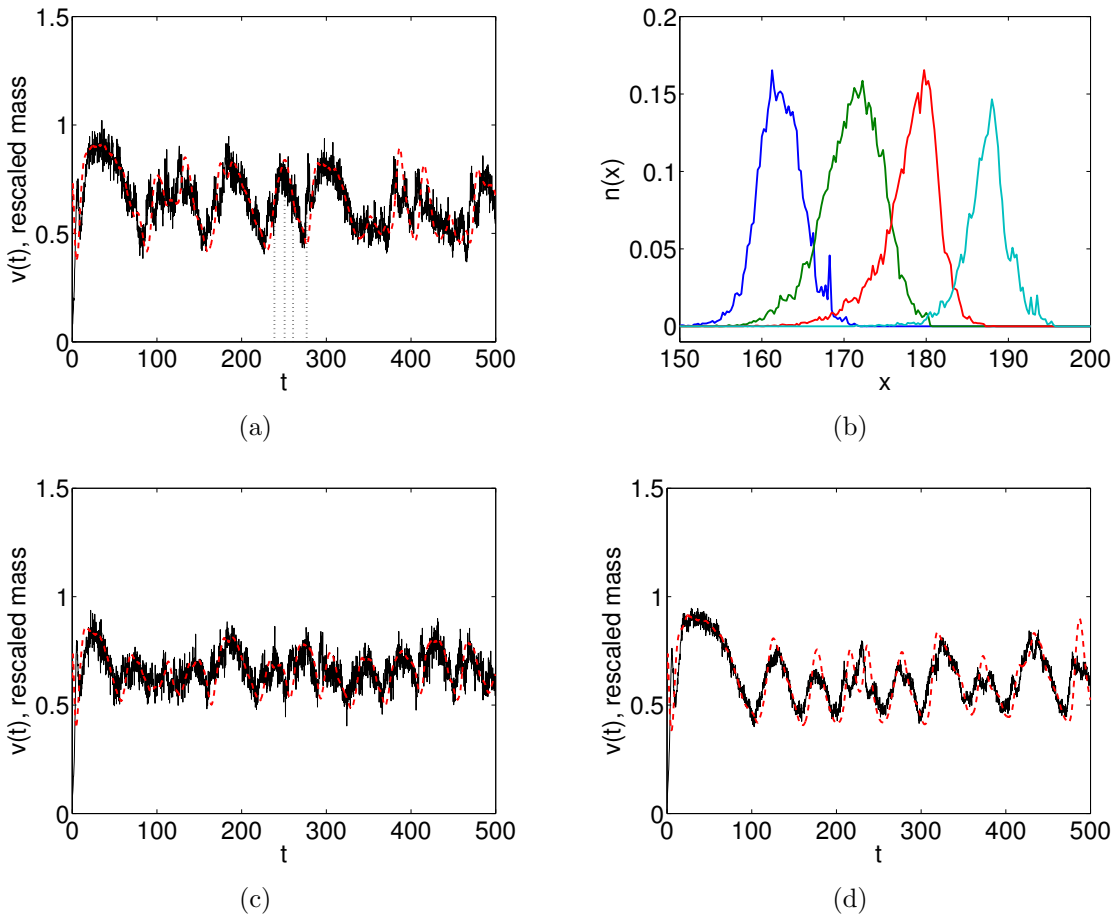


Figure 2.17: *Oscillations in the wave speed of the hybrid model presented in Section 2.5.2. Parameters as shown in the text.*
(a) *Wave speed in comparison to current number of particles for* $N_0 = 10^4$, $t_a = 4$. *Solid (black) line: wave speed; dashed (red) line: number of particles; dotted (black) lines: times of wave forms shown in panel (b).*
(b) *Waveform at 4 distinct times marked in panel (a) from left to right.*
(c) *As in (a) with* $N_0 = 10^4$, $t_a = 2$.
(d) *As in (a) with* $N_0 = 5 \times 10^4$, $t_a = 4$.
Other parameters are given in Section 2.5.6.

individual bacterium as it defines the region where consumption (and deposition) of signal takes place.

In the second part, we study hybrid systems of bacterial chemotaxis under the inclusion of growth terms. We show how these growth terms can stabilise chemotaxis systems with a biologically realistic turning kernel introduced in [173] to obtain travelling wave solutions. We compare two different methods of incorporating the growth process into the hybrid system and conclude that both give consistent results. These results are compared to mesoscopic and macroscopic PDE systems and we obtain a qualitative correspondence. A travelling wave analysis for these PDE systems is performed, revealing a quantitative agreement for the case of no chemotaxis and an increasing wave speed as chemotaxis is increased. We conclude that the growth process has a stabilising effect on the otherwise transient travelling wave system. Additional simulations showed the existence of oscillations in the speed and mass of the travelling waves for certain parameter values, an effect that has been seen in similar systems in the literature [130] and that cannot be explained using mean-field approximations.

To date, travelling waves in chemotaxis models have mainly been analysed from the perspective of macroscopic PDE models of chemotaxis [87, 88]. The existence of travelling waves for continuum models with growth terms is well established [101, 121, 147]. While hybrid models have been used to study pattern formation in bacterial chemotaxis [79, 172], these studies have not analysed the travelling wave patterns observed in bacterial cell populations. Recently, experimental studies using microfluidic techniques tracked cell trajectories within a travelling pulse, and revealed that persistence of direction in cell movement accounts for 30% of the macroscopic speed of the travelling pulse [146]. The hybrid model framework studied here provides a natural method for direct comparison of model predictions with experimental measurements of cell trajectories.

Looking at the bigger picture of obtaining a unified hybrid modelling scheme for a large number of systems, the methods presented can generally be applied for all reaction-diffusion systems with reactions that are first order in the density of agent species. In Chapter 3, we extend this to incorporate additional types of reactions and an additional possibility of how to split the system into continuum and agent-based species. In Chapter 4 we apply bacterial chemotaxis models to the target finding problem in robot swarms. Whilst the models in this chapter do not consider direct interactions between individuals, we introduce volume exclusion into the velocity jump framework in Chapter 5.

Chapter 3

PDE-assisted Brownian dynamics

In Chapter 2 we extended the hybrid framework presented in Section 1.4.3 to use different modelling approaches for different species in a common domain Ω. In this chapter we want to extend hybrid models into a different direction, namely we derive an algorithm that uses different levels of detail in different parts of the domain. We start by discussing possible application areas of this hybrid approach and then introduce the extension of the framework presented in Section 1.4.3 along with a mathematical formulation of the problems involved. Section 3.3 introduces a first model for purely diffusive systems. We extend this initial model by introducing simple reactions and show a number of numerical results to demonstrate the correct working of the presented algorithm. First steps for an extension into higher dimensions are shown in Section 3.6 and a discussion of the results of this chapter follows.

Most of the results presented in this chapter were published in the *SIAM Journal of Applied Mathematics* paper [67].

3.1 Motivation and examples

Whilst Chapter 2 concentrated on the active motion of bacteria in response to extracellular signal, this chapter initially studies simpler types of motion in the form of reaction-diffusion models. Those models have been widely used for the description of biological systems [120] and allow each species to diffuse and to react by itself or in combination with other species. Often continuum approaches, written in the form of reaction-diffusion PDEs, are used due to their simplicity and the vast number of ready-to-use numerical solvers. A general reaction-diffusion model takes the PDE form

$$\frac{\partial c_j}{\partial t} = \nabla_{\mathbf{x}} \cdot (D_j(x) \nabla_{\mathbf{x}} c_j) + R_j(c_1, \ldots, c_M)\,, \tag{3.1}$$

where $c_j(t, \mathbf{x})$ is the concentration of species $j = 1, \ldots, M$, D_j is its diffusion constant and R_j represents reaction terms. As in Chapter 2, not all biological effects can be described by deterministic PDE-based models, especially in the case of low copy numbers and the inherent fluctuations in spatial concentrations in that case [110, 155]. If low copy numbers are present in a reaction-diffusion network, stochastic models such as mesoscopic compartment-based algorithms [42, 81] or Brownian dynamics (BD) simulations as introduced in Section 1.3.1 may be deployed [8, 162].

In many situations individual trajectories are important only in certain parts of the domain, whilst in the remainder of the domain a coarser, less detailed, description can be used [62]. This is the case, for example, in the modelling of ion-channels [118]. Ions pass through a channel in single file and an individual-based model has to be used to accurately compute the discrete, stochastic, current in the channel [31]. The positions of individual ions are less important away from the channel where copy numbers may be very large (rendering a detailed Brownian dynamics description infeasible) [36]. Another example is the stochastic reaction-diffusion modelling of filopodia which are dynamic finger-like protrusions used by eukaryotic motile cells to probe their environment and guide cell motility [175]. These relatively small protrusions are connected to a larger cytosol compartment. If a modeller is interested in understanding the dynamics of filopodia, then there is a potential to decrease the computational cost of simulations by using a coarser model in the cytosol [52]. In both examples, it is important to understand how models with a different level of detail can be used in different parts of the computational domain [62].

In this chapter, we develop two algorithms that calculate BD paths in a desired part of the domain, whilst using a continuum PDE-based model in the remainder. Similarly to the hybrid models studied in Chapter 2 this *PDE-assisted Brownian dynamics (PBD)* methodology has the advantage that efficient methods for solving PDEs can be used for large parts of the system, whilst BD data is available where required. The main goal of the PBD methodology is to obtain the same statistics (means and variances) in the BD subdomain as one would get using BD simulations in the whole domain. In particular, the correct coupling between the two parts of the domain is of vital importance for the accuracy of a PBD algorithm.

3.2 Adaptation of the hybrid modelling framework

The development of hybrid methods of the desired form is based on the general framework for spatial hybrid models described in Section 1.4.3. Therefore, we again have

M_C continuum species described through the concentrations $c_j(t, \mathbf{x})$, $j = 1, \ldots, M_C$, defined for $\mathbf{x} \in \Omega \subset \mathbb{R}^d$, $d = 1, 2, 3$, as well as M_A agent-based species defined by the position $\mathbf{x}^{(i,j)} \in \Omega$ of particle $i = 1, \ldots, N_j$ of species $j = 1, \ldots, M_A$.

However, in contrast to the hybrid models we have seen in Chapter 2, agents and continuum descriptions no longer share the whole domain Ω, but each have a dedicated area. We therefore define the subdomains $\Omega_P, \Omega_B \subset \Omega$ and the interface between the two domains $I \subset \Omega$ so that the three sets satisfy the conditions

$$\Omega = \overline{\Omega_B} \cup \overline{\Omega_P}\,, \qquad I = \overline{\Omega_B} \cap \overline{\Omega_P}\,. \tag{3.2}$$

We denote Ω_P as the *PDE* or *continuum subdomain* and Ω_B as the *BD* or *particle-based subdomain* of Ω. For the defining quantities of the different species this means in particular that the concentrations $c_j(t, \mathbf{x})$, $j = 1, \ldots, M_C$, are only defined for $\mathbf{x} \in \Omega_P$ whilst particle positions need to satisfy $\mathbf{x}^{(i,j)} \in \Omega_B$, $i = 1, \ldots, N_j$, $j = 1, \ldots, M_A$.

Another difference compared to the hybrid models seen before is that we allow for one species to be modelled by both modelling approaches and in particular we require that $M_C = M_A = M$ and that the concentration $c_j(t, \mathbf{x})$ represents the same species as particles (i, j), $i = 1, \ldots, N_j$, for $j = 1, \ldots, M$. The main focus of this chapter is the question of how to correctly interchange mass between the two subdomains whilst maintaining a consistent concentration profile. Because we are dealing with reaction-diffusion systems, we can make further simplifying assumptions on the behaviour of the different species. In particular, we assume that the update of the concentrations $c_j(t, \mathbf{x})$, $j = 1, \ldots, M$, is given through the reaction-diffusion PDE (3.1) and that the particles follow the Euler-Maruyama discretisation of the classical Brownian motion given through

$$\mathbf{x}^{(i,j)}(t + \Delta t) = \mathbf{x}^{(i,j)}(t) + \sqrt{2D_j \Delta t}\, \xi\,, \tag{3.3}$$

where $\xi \in \mathbb{R}^d$ is a vector of d independent normally distributed random variables. Additionally, the reaction-diffusion framework dictates that particles do not carry further information than their current positions $\mathbf{x}^{(i,j)}$, i.e. no further internal dynamics are possible.

In this chapter, we present PBD algorithms for the following two situations:

[A] Ω_B and Ω_P do not overlap, i.e. $\Omega_B \cap \Omega_P = \emptyset$;

[B] there exists an overlap region where the PDE description and BD simulations exist in parallel, i.e. $O = \Omega_B \cap \Omega_P \neq \emptyset$.

The case [A] will lead to the PBD algorithm (A1)–(A5) presented in Table 3.1. The case [B] is implemented in the second PBD algorithm (B1)–(B5) which is presented

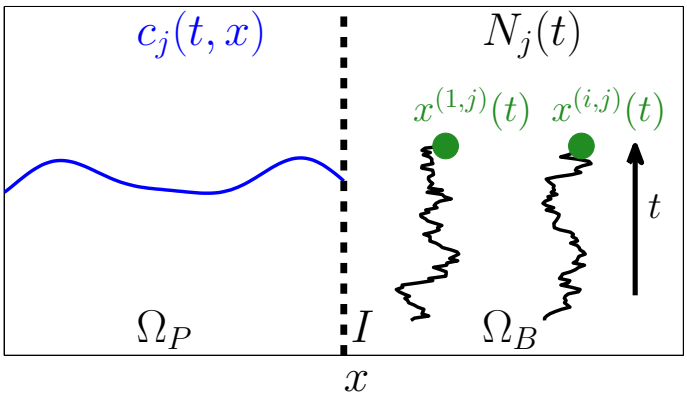


Figure 3.1: *Sketch of the first PBD algorithm and the notation related to it. In Ω_P, molecules are described by their concentration $c_j(t,x)$ and in the particle-based domain Ω_B are described by the number $N_j(t)$ of molecules and their positions $x^{(i,j)}(t)$, $i = 1, 2, \ldots, N_j(t)$, $j = 1, \ldots, M$. The interface between these domains is called I and is defined in* (3.2).

in Table 3.2. We will start the discussion with case [A] because it is less technical to implement than the case [B].

To simplify presentation, we will consider that Ω is a 'narrow' three-dimensional domain, and hence only consider the process mapped onto an effective one-dimensional domain $\Omega \subset \mathbb{R}$ by assuming that the system is well mixed in the other two dimensions. A sketch of the mixed PDE, particle-based system along with the notation for case [A] is shown in Figure 3.1 for one chemical species.

Mathematical problem formulation

As we mentioned in Section 3.1, the main aim of the PBD method is to maintain the same statistics in the particle-based regime Ω_B as a purely agent-based system would achieve. Let us therefore consider a general BD reaction-diffusion simulation with M chemical species in the full domain $\Omega \subset \mathbb{R}^d$, $d = 1, 2, 3$. We maintain the main notation from the hybrid system, such that species $j = 1, \ldots, M$ is represented by N_j particles located at $\mathbf{x}^{(i,j)}(t) \in \Omega$. After setting off the particles according to an arbitrary distribution $n_0(\mathbf{x})$, BD simulations can be used to evaluate the following random variable for every set $A \subset \Omega$

$$\varphi_j(t, A) = \left|\left\{\mathbf{x}^{(i,j)}(t) \in A \,, i = 1, 2, \ldots, N_j\right\}\right| \,, \quad j = 1, \ldots, M \,,$$

representing the number of molecules of the j-th chemical species inside the subset A at time t. We denote by $n_j(t, \mathbf{x}) : [0, \infty) \times \Omega \mapsto \mathbb{R}_0^+$, $j = 1, 2, \ldots, M$, the expected spatio-temporal concentration of the j-th chemical species at the position $\mathbf{x}$ and time

t over the domain Ω. Hence, we have

$$\mathbb{E}\left[\varphi_j(t, A)\right] = \int_A n_j(t, \mathbf{x})\,\mathrm{d}x\,.$$

We also define the expected spatio-temporal variance of the number of molecules of the j-th chemical species inside $A \subset \Omega$ by

$$\mathrm{var}\left[\varphi_j(t, A)\right] = \mathbb{E}\left[\left(\varphi_j(t, A)\right)^2\right] - \left(\int_A n_j(t, \mathbf{x})\,\mathrm{d}x\right)^2 .$$

Using these two quantities, we can now formulate three conditions that we require PBD methods to satisfy in order to be considered accurate.

Condition (C.1): We require that the expected distribution of molecules in $\Omega_B \setminus \Omega_P$, i.e. the part of the domain where only a particle-based description is used, of the PBD method matches that of the expected distribution $n_j(t, \mathbf{x})$, i.e. the distribution which we would obtain if we used detailed BD simulations in the whole domain Ω. In particular this means that for every set $A \subset \Omega_B \setminus \Omega_P$, the expected number of particles in A at time $t > 0$ has to satisfy

$$\mathbb{E}\left[\left|\left\{\mathbf{x}^{(i,j)}(t) \in A\,, i = 1, \ldots, N_j\right\}\right|\right] = \int_A n_j(t, \mathbf{x})\,\mathrm{d}\mathbf{x}\,, \qquad \forall A \subset \Omega_B \setminus \Omega_P\,. \tag{3.4}$$

In the case $d = 1$ mainly considered in this chapter, the choice of an arbitrarily small but finite interval $A = [x, x + \mathrm{d}x)$ for $x \in \Omega_B \setminus \Omega_P$ leads to an alternative formulation of this condition

$$\mathbb{E}\left[\left|\left\{x^{(i,j)}(t) \in [x, x + \mathrm{d}x)\,, i = 1, \ldots, N_j\right\}\right|\right] = n_j(t, x)\,\mathrm{d}x\,, \qquad \forall x \in \Omega_B \setminus \Omega_P\,.$$

Condition (C.2): In addition to the expected outcomes, we also want the variances of the molecule distribution in $\Omega_B \setminus \Omega_P$ to match that which would be expected if a BD simulation were to be performed over the entire domain Ω. Hence, we require

$$\mathrm{var}\left[\left|\left\{\mathbf{x}^{(i,j)}(t) \in A\,, i = 1, \ldots, N_j\right\}\right|\right] = \mathrm{var}\left[\varphi_j(t, A)\right]\,, \tag{3.5}$$

for all $A \subset \Omega_B \setminus \Omega_P$ and $t > 0$.

In $\Omega_P \setminus \Omega_B$, the system is described by the concentration vector c_j, $j = 1, 2, \ldots, M$, which evolves according to the reaction-diffusion PDE (3.1). This distribution, whilst continuous, is not strictly deterministic since it is coupled with the stochastic outcomes of the BD subdomain Ω_B. If the stochastic reaction-diffusion model only includes zero-order or first-order reactions, then the mean-field PDE (3.1) describes

the expected behaviour of stochastic models [51]. In this case it is reasonable to require the following additional condition.

Condition (C.3): We require

$$\mathbb{E}\left[c_j(t,\mathbf{x})\right] = n_j(t,\mathbf{x})\,, \qquad \forall x \in \Omega_P \setminus \Omega_B\,,\ t>0\,. \tag{3.6}$$

In the case [A], the conditions (C.1)–(C.3) can be simplified by observing that $\Omega_B = \Omega_B \backslash \Omega_P$ and $\Omega_P = \Omega_P \backslash \Omega_B$. In the case [B], we will also require that the PBD algorithm gives the correct mean distribution of molecules in the overlap region $O = \Omega_B \cap \Omega_P$, i.e.

$$\mathbb{E}\left[\int_A c_j(t,\mathbf{x})\,\mathrm{d}\mathbf{x}\right] + \mathbb{E}\left[\left|\left\{\mathbf{x}^{(i,j)}(t) \in A\,, i=1,\ldots,N_j\right\}\right|\right] = \int_A n_j(t,\mathbf{x})\,\mathrm{d}\mathbf{x}\,,$$

for all sets $A \subset O = \Omega_B \cap \Omega_P$ and $t > 0$.

Note that we did not make any explicit assumptions for the solutions at the interface I. Instead, we formulate these assumptions implicitly using Conditions (C.1)–(C.3). Conditions (C.1) and (C.3) imply that PBD algorithms guarantee the continuity of the expected spatio-temporal concentration across the interface I.

3.3 PBD simulation of diffusion

In this section, we concentrate on a case [A] PBD algorithm (i.e. $\Omega_B \cap \Omega_P = \emptyset$) using a system of diffusing non-interacting molecules of a single chemical species in a quasi one-dimensional domain $\Omega \subset \mathbb{R}$. Therefore, for all further discussions in this and the following section we will drop the index j representing the species. Then the macroscopic PDE (3.1) for this system becomes the diffusion equation

$$\frac{\partial c}{\partial t} = D\frac{\partial^2 c}{\partial x^2}\,. \tag{3.7}$$

We will consider the infinite domain $\Omega = \mathbb{R}$ for simplicity. Without loss of generality, we assume that $\Omega_P = (-\infty, 0)$ and $\Omega_B = (0, \infty)$, i.e. the internal boundary $I = \{0\}$ is situated at the origin $x = 0$. In the case of diffusion only, the total mass N_{tot} in the system is conserved, i.e.

$$N_{\mathrm{tot}} = \int_\Omega n(t,x)\,\mathrm{d}x\,, \qquad \text{for all } t \geq 0\,. \tag{3.8}$$

Hence, for the PBD algorithm the conservation of mass condition takes the form

$$N_{\mathrm{tot}} = \int_{\Omega_P} c(t,x)\,\mathrm{d}x + N(t)\,, \qquad \text{for all } t \geq 0\,, \tag{3.9}$$

where $N(t)$ represents the mass contribution of the particle-based component and the integral represents that of the continuum part. Since the diffusing molecules are non-interacting, we can evaluate the variance of φ by assuming that every particle has the (identical) probability $P_A = \int_A n(t,x)\,\mathrm{d}x/N_{\mathrm{tot}}$ of being in the set A at time t. Consequently,

$$\mathrm{var}\left[\varphi_j(t,A)\right] = N_{\mathrm{tot}} P_A(1-P_A)\,.$$

Substituting for P_A and using (3.8), equation (3.5) takes the form

$$\mathrm{var}\left[\left|\left\{x^{(i)}(t)\in A\,, i=1,\ldots,N\right\}\right|\right] = \int_A n(t,x)\,\mathrm{d}x\left(1-\frac{\int_A n(t,x)\,\mathrm{d}x}{\int_\Omega n(t,x)\,\mathrm{d}x}\right)\,,$$

for all $A\subset\Omega_B$ and $t>0$. Using again $A=[x,x+\mathrm{d}x)$, we obtain the alternative formulation up to order $(\mathrm{d}x)^2$

$$\mathrm{var}\left[\left|\left\{x^{(i)}(t)\in [x,x+\mathrm{d}x)\,, i=1,\ldots,N\right\}\right|\right] = n(t,x)\,\mathrm{d}x\,.$$

In Sections 3.3.1 and 3.3.2, we present one update step of the continuum and the particle-based simulations, respectively, before the full PBD algorithm (A1)–(A5) is formulated in Section 3.3.3. In Section 3.3.4, we will discuss the accuracy of the algorithm with respect to the Conditions (C.1)–(C.3).

3.3.1 Updating the PDE regime in Ω_P

At time t, we have the concentration $c(t,x)$ for $x\in\Omega_P$ and are aiming to calculate the concentration $c(t+\Delta t,x)$ that corresponds to a realisation of one time step Δt of the diffusion process (3.7). Let us therefore define in an intermediate step the exact outcome of the diffusion process in the full domain Ω given initial data $c(t,x)$:

$$\widetilde{c}(t+\Delta t,x) = \int_{\Omega_P} K(\Delta t, x-x')\,c(t,x')\,\mathrm{d}x'\,, \tag{3.10}$$

where $K(\Delta t, x-x')$ is the diffusion kernel

$$K(\Delta t,\xi) = \frac{1}{\sqrt{4\pi D\Delta t}}\exp\left(-\frac{\xi^2}{4D\Delta t}\right)\,. \tag{3.11}$$

The intermediate function $\widetilde{c}(t+\Delta t,x)$ represents the exact outcome of diffusion of the PDE component $c(t,x)$ on the time interval $[t,t+\Delta t)$ and therefore has support Ω. We are, however, aiming to obtain a distribution $c(t+\Delta t,x)$ that has support only in Ω_P and therefore need to consider the mass that has flowed over the interface I during the diffusion step. Let us therefore define the *overflow* $\alpha(t+\Delta t)$ through

$$\alpha(t+\Delta t) = \int_{\Omega_B}\widetilde{c}(t+\Delta t,x)\,\mathrm{d}x\,. \tag{3.12}$$

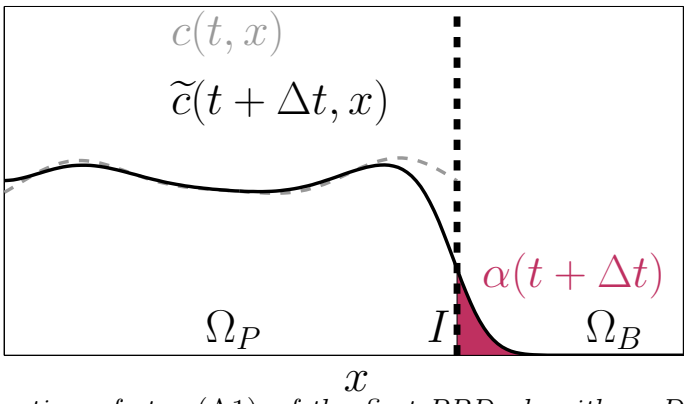


Figure 3.2: *Illustration of step* (A1) *of the first PBD algorithm. Dashed (grey) line:* $c(t,x)$*; solid (black) line:* $\widetilde{c}(t+\Delta t,x)$ *as defined in* (3.10)*; shaded (red) area:* $\alpha(t+\Delta t)$ *as defined in* (3.12)*.*

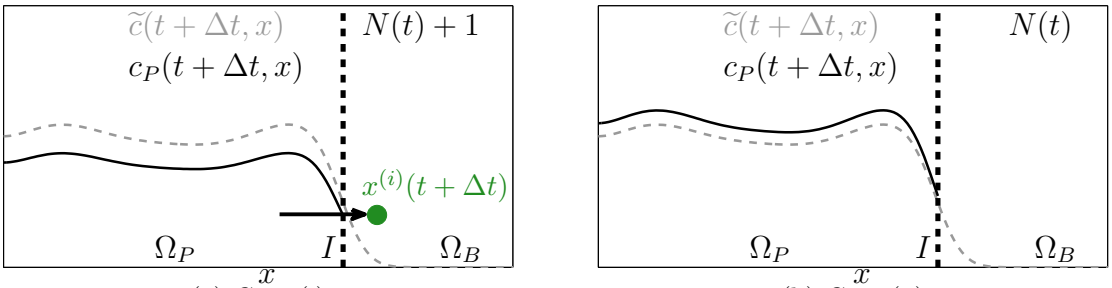
A graphical explanation of this overflow can be seen in Figure 3.2. The only way to account for mass that has crossed the interface is by creating new particles inside Ω_B, and in order to get the correct expected mass flow, we need to create an expected number of $\alpha(t+\Delta t)$ new particles. Since all molecules in Ω_P are identical (of the same chemical species and have the same probability distribution proportional to $\widetilde{c}(t+\Delta t,x)$), the number of molecules that cross the interface from Ω_P to Ω_B in the time interval $[t,t+\Delta t)$ is Poisson distributed with mean $\alpha(t+\Delta t)$. Let us assume that Δt has been chosen small enough to ensure that $\alpha(t+\Delta t) \ll 1$. In this case the probability of more than one particle crossing the interface is negligible and we need to consider two cases:

(i) one particle gets created in Ω_B with the probability $\alpha(t+\Delta t)$;
(ii) no particle is created with the probability $1-\alpha(t+\Delta t)$.

For both cases we need to calculate the updated concentration $c(t+\Delta t,x)$ with support in Ω_P. The main idea to calculate this updated concentration is to consider $c(t,x)$ as the distribution of $N_{\text{tot}}-N$ identically distributed particles at time t. Therefore each of these particles has at time t the probability distribution $c(t,x)/(N_{\text{tot}}-N)$. After one time step each of these particles has the probability distribution $\widetilde{c}(t+\Delta t,x)/(N_{\text{tot}}-N)$. Hence, for each particle, its probability distribution given that it did not leave Ω_P can be calculated as

$$p_1(t+\Delta t,x) = \frac{\widetilde{c}(t+\Delta t,x)}{N_{\text{tot}}-N(t)-\alpha(t+\Delta t)}\,, \qquad x\in\Omega_P\,.$$

On the other hand, if a particle does leave the domain Ω_P, then its distribution function becomes zero for $x\in\Omega_P$. Instead the particle is introduced into Ω_B at time



(a) Case (i) (b) Case (ii)

Figure 3.3: *Illustration of steps* (A2) *and* (A3) *of the first PBD algorithm. Dashed (grey) line:* $\widetilde{c}(t+\Delta t, x)$*; solid (black) line:* $c_P(t+\Delta t, x)$*; (green) dot: created particle.*

$t + \Delta t$ at a location x with the probability distribution given by

$$p_2(t+\Delta t, x) = \frac{\widetilde{c}(t+\Delta t, x)}{\alpha(t+\Delta t)}, \qquad x \in \Omega_B\,. \tag{3.13}$$

This updating process is similar to collapsing a wave function in quantum mechanics [163]. We sample from its probability distribution to know if a particle has crossed the boundary into Ω_B: if it has crossed the interface, we sample its exact position and its distribution function collapses to a δ function at its new position, while if it is not, then its distribution function becomes zero in Ω_B and is rescaled in Ω_P.

Combining these arguments for all the particles, we see that the last update step is a simple rescaling of the probability distribution so that the updated distribution satisfies conservation of mass according to (3.8). We therefore have

$$c_P(t+\Delta t, x) = \begin{cases} \beta_{(\mathrm{i})}\,\widetilde{c}(t+\Delta t, x), & \text{in the case (i)}, \\ \beta_{(\mathrm{ii})}\,\widetilde{c}(t+\Delta t, x), & \text{in the case (ii)}, \end{cases} \qquad x \in \Omega_P\,, \tag{3.14}$$

with the rescaling constants $\beta_{(\mathrm{i})}$, $\beta_{(\mathrm{ii})}$ given respectively by

$$\beta_{(\mathrm{i})} = \frac{N_{\mathrm{tot}} - N - 1}{N_{\mathrm{tot}} - N - \alpha(t+\Delta t)}, \qquad \beta_{(\mathrm{ii})} = \frac{N_{\mathrm{tot}} - N}{N_{\mathrm{tot}} - N - \alpha(t+\Delta t)}\,. \tag{3.15}$$

Note that the new concentration $c_P(t+\Delta t, x)$ is only an intermediate concentration and not the new concentration $c(t+\Delta t, x)$ as we still need to consider possible contributions from the particle based subdomain Ω_B. The update for the continuum regime satisfies conservation of mass (3.8) in both cases (i) and (ii). An illustration of these two cases can be seen in Figure 3.3.

3.3.2 Updating the BD regime in Ω_B

We use the discretised version of the overdamped Langevin equation introduced in (3.3) to update the positions of particles in Ω_B. In the one-dimensional case considered

here, it takes the form

$$x^{(i)}(t+\Delta t) = x^{(i)}(t) + \sqrt{2D\,\Delta t}\xi\,, \quad \xi \sim \mathcal{N}(0;1)\,. \tag{3.16}$$

If the position $x^{(i)}(t+\Delta t)$ of particle $i = 1, \ldots, N$, computed through (3.16), is in Ω_B, then we will continue representing it as a particle. Note that a particle that crossed the interface I and came back into Ω_B during the time step $[t, t+\Delta t)$ is also captured by this case. We have to be more careful whenever the position $x^{(i)}(t+\Delta t)$, computed by (3.16), is inside the PDE subdomain Ω_P at time $t+\Delta t$, i.e. $x^{(i)}(t+\Delta t) \in \Omega_P$. Because, this particle can no longer be represented through BD simulations, it needs to be taken into account for the calculation of the final concentration $c(t+\Delta t, x)$ in Ω_P. As we know the exact position of this particle i at time $t+\Delta t$ we transform it into a Dirac δ function at the position $x^{(i)}(t+\Delta t) \in \Omega_P$. Therefore we compute the final concentration $c(t+\Delta t, x)$ in Ω_P by

$$c(t+\Delta t, x) = c_P(t+\Delta t, x) + \sum_{x^{(i)}(t+\Delta t)\in\Omega_P} \delta\left(x - x^{(i)}(t+\Delta t)\right)\,,$$

with $c_P(t+\Delta t, x)$ given in (3.14).

3.3.3 The first PBD algorithm

Let us now put the individual updates presented in Sections 3.3.1 and 3.3.2 into the first complete PBD algorithm. This algorithm tracks the concentration $c(t,x)$ for $x \in \Omega_P$, and the number $N(t)$ and positions of BD particles $x^{(i)}(t) \in \Omega_B$, $i = 1, 2, \ldots, N$. One time step of the first PBD algorithm is presented in Table 3.1 as algorithm (A1)–(A5). In order to simplify the presentation of this algorithm, we assume that the time step Δt is chosen so small that $\alpha(t+\Delta t) \ll 1$. In particular, we only need to implement cases (i)–(ii) presented in Section 3.3.1, because the probability that two or more molecules are initiated in Ω_B during one time step is negligible.

The auxiliary distribution $\widetilde{c}(t+\Delta t, x)$ in step (A1) is in practice calculated using a numerical approximation algorithm. To calculate $\alpha(t+\Delta t)$ we can either use this numerical approximation of $\widetilde{c}(t+\Delta t, x)$, or (more efficiently) we can approximate $\alpha(t+\Delta t)$ analytically using a boundary layer expansion in the vicinity of the interface I.

By construction, the algorithm (A1)–(A5) satisfies the conservation of mass condition (3.9) and the concentration $c(t,x)$ satisfies non-negativity. We will now show that this algorithm also guarantees the correct expected outcome and therefore satisfies Conditions (C.1) and (C.3).

(A1) Calculate $\widetilde{c}(t+\Delta t, x)$ using (3.10) and $\alpha(t+\Delta t)$ using (3.12).

(A2) Generate a uniformly distributed random number $r \in (0,1)$.

(i) If $r < \alpha(t+\Delta t)$, then create a new particle in Ω_B according to the probability density $p_2(t+\Delta t, x)$ defined in (3.13). Set $\beta = \beta_{(\mathrm{i})}$, where $\beta_{(\mathrm{i})}$ is given by (3.15). Set $N_{\mathrm{new}} = 1$.

(ii) If $r \geq \alpha(t+\Delta t)$, then set $\beta = \beta_{(\mathrm{ii})}$, where $\beta_{(\mathrm{ii})}$ is given by (3.15). Set $N_{\mathrm{new}} = 0$.

(A3) Compute the positions $x^{(i)}(t+\Delta t)$, $i = 1, 2, \ldots, N(t)$, of the BD particles according to (3.16).

(A4) Compute the new concentration in Ω_P by

$$c(t+\Delta t, x) = \beta\, \widetilde{c}(t+\Delta t, x) + \sum_{x^{(i)}(t+\Delta t) \in \Omega_P} \delta\left(x - x^{(i)}(t+\Delta t)\right), \quad x \in \Omega_P\,.$$

(A5) Update the number of BD particles by

$$N(t+\Delta t) = N(t) + N_{\mathrm{new}} - \left|\left\{x^{(i)}(t+\Delta t) \in \Omega_P\,, i = 1, \ldots, N\right\}\right|\,.$$

Terminate computation of the trajectories of BD molecules which landed in Ω_P (i.e. the BD particles which satisfy $x^{(i)}(t+\Delta t) \in \Omega_P$.)
Then continue with step (A1) for time $t + \Delta t$.

Table 3.1: *One time step of the first PBD algorithm for a system of diffusing molecules.*

Theorem 3.1. *Consider the BD simulation of N_{tot} diffusing molecules in the computational domain Ω which is divided into subdomains $\Omega_B \subset \Omega$ and $\Omega_P \subset \Omega$ satisfying* (3.2) *and the case* [A]. *Suppose that the $N(0)$ particles are initially distributed in Ω_B at positions $x^{(i)}(0)$, $i = 1, 2, \ldots, N(0)$, according to $n(0,x)$, $x \in \Omega_B$ and that the remaining $N_{\mathrm{tot}} - N(0)$ particles have concentration $c(0,x)$ in Ω_P, such that $c(0,x) = n(0,x)$ for $x \in \Omega_P$. Then the expected outcome of the PBD algorithm* (A1)–(A5) *(presented in Table* 3.1*) satisfies Conditions* (C.1) *and* (C.3) *for arbitrary $\Delta t > 0$.*

Proof. We will show that the identities (3.4) and (3.6) hold during one iteration (A1)–(A5) presented in Table 3.1. It will then follow by induction that they hold for all times $k\Delta t$, $k = 0, 1, 2, \ldots$.

Let us assume that the expected probability distribution of particles in Ω_B at time

t is equal to $n(t,x)$, $x \in \Omega_B$ and that the expected concentration $c(t,x)$, $x \in \Omega_P$, at time t is equal to $n(t,x)$. The conditional expected value of $c(t+\Delta t, x)$ for $x \in \Omega_P$ is given by

$$\begin{aligned}\mathbb{E}\left[\, c(t+\Delta t,x) \,|\, c(t,x) \,\right] &= \alpha(t+\Delta t)\,\beta_{(\mathrm{i})}\,\widetilde{c}(t+\Delta t,x) \\ &\quad + (1-\alpha(t+\Delta t))\,\beta_{(\mathrm{ii})}\,\widetilde{c}(t+\Delta t,x) \\ &\quad + \int_{\Omega_B} K(\Delta t, x-x')\,n(t,x')\,\mathrm{d}x'\,,\end{aligned}$$

where $\beta_{(\mathrm{i})}$ and $\beta_{(\mathrm{ii})}$ are given by (3.15), $K(\Delta t, x-x')$ is the diffusion kernel given in (3.11) and the last term represents particles that crossed the interface I, during the time step $[t, t+\Delta t)$. Using (3.15), we obtain

$$\mathbb{E}\left[\, c(t+\Delta t,x) \,|\, c(t,x) \,\right] = \widetilde{c}(t+\Delta t,x) + \int_{\Omega_B} K(\Delta t, x-x')\,n(t,x')\,\mathrm{d}x'\,.$$

Using (3.10) and the law of total expectation (law of iterated expectations), we get

$$\mathbb{E}\left[c(t+\Delta t,x)\,\right] = \int_{\Omega_P} K(\Delta t, x-x')\,\mathbb{E}[c(t,x')]\,\mathrm{d}x' + \int_{\Omega_B} K(\Delta t, x-x')\,n(t,x')\;\mathrm{d}x'\,.$$

Using the induction assumption that $\mathbb{E}[c(t,x)] = n(t,x)$, we obtain

$$\mathbb{E}\left[c(t+\Delta t,x)\,\right] = \int_{\Omega} n(t,x')K(\Delta t, x-x')\,\mathrm{d}x' = n(t+\Delta t,x)\,, \qquad x \in \Omega_P\,,$$

i.e. we have shown that (A1)–(A5) satisfies (3.6) and thereby (C.3).

Let us for the second part consider a set $A \subset \Omega_B$. The conditional expected number of particles in A at time $t+\Delta t$ is

$$\begin{aligned}\mathbb{E}\left[\,\left|\left\{x^{(i)}(t) \in A\,, i=1,\ldots,N\right\}\right| \,\middle|\, c(t,x)\,\right] &= \int_A \alpha(t+\Delta t)\,p_2(t+\Delta t,x)\,\mathrm{d}x \\ &\quad + \int_A \int_{\Omega_B} K(\Delta t, x-x')\,n(t,x')\,\mathrm{d}x'\,\mathrm{d}x\,,\end{aligned}$$

where the first term represents newly created particles from the PDE regime Ω_P and the second term represents the movement of particles inside Ω_B. Using (3.10), (3.13) and the law of total expectation, we obtain

$$\begin{aligned}\mathbb{E}\left[\,\middle|\, \left\{x^{(i)}(t) \in A\,, i=1,\ldots,N\right\} \,\middle|\,\right] &= \int_A \int_{\Omega} K(\Delta t, x-x')\,n(t,x')\,\mathrm{d}x'\,\mathrm{d}x \\ &= \int_A n(t+\Delta t,x)\,\mathrm{d}x\,,\end{aligned}$$

which is equivalent to (3.4). Thus we have showed that both Conditions (C.1) and (C.3) are satisfied. □

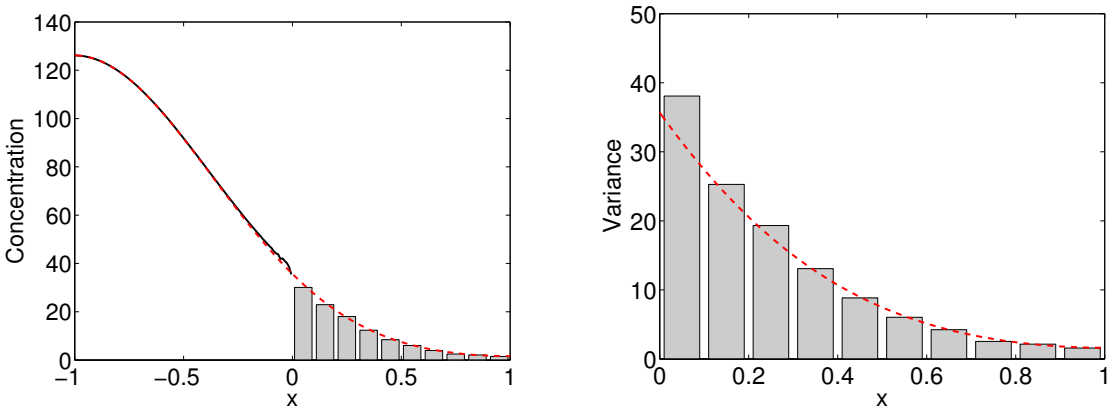
Theorem 3.1 also holds if the algorithm (A1)–(A5) is extended to the creation of more than one new particle per time step as long as the expected value of the number of created particles is $\alpha(t + \Delta t)$ and the rescaling is done accordingly. This has the advantage that one is not forced to choose Δt small enough to ensure that $\alpha(t + \Delta t)$ is small. In fact, this extension will be applied in all presented simulation results throughout this chapter. The algorithm (A1)–(A5) and the proof can also be easily extended for a finite domain $\Omega = [0, L]$ with no flux boundary conditions by redefining the kernel $K(\Delta t, \xi)$ in (3.11) accordingly.

3.3.4 Discussion of the PBD algorithm (A1)–(A5)

In Theorem 3.1 we showed that the PBD algorithm (A1)–(A5) satisfies Conditions (C.1) and (C.3). However, we still need to check whether Condition (C.2) on the variances is also satisfied.

To investigate the variances created by this algorithm, we show the outcome of an illustrative numerical example in Figure 3.4. We simulate the diffusion of $N_{\text{tot}} = 100$ molecules in the domain $\Omega = [-1, 1]$. All molecules are initialised at $x = -0.95$, i.e. the initial condition takes the form $n(0, x) = 100\,\delta(x+0.95)$. We use no flux boundary conditions. We test the algorithm (A1)–(A5) with $\Omega_P = [-1, 0)$, $\Omega_B = (0, 1]$ and $I = \{0\}$. For the calculation of $\widetilde{c}(\Delta t, x)$ in step (A1) we use an implicit Euler-scheme with $\Delta x = 0.01$ and a numerical time step of $\widetilde{\Delta t} = 10^{-6}$. The time step Δt used by the algorithm (A1)–(A5) is $\Delta t = 10^{-3}$ and we simulate the system until $t = 0.2$. The time step $\widetilde{\Delta t} \ll \Delta t$ for the approximation of $\widetilde{c}(t + \Delta t, x)$ was chosen in order to minimise numerical errors in the calculation of $\alpha(t + \Delta t)$. We run 10^3 realisations of this process and measure the number of particles in 10 intervals ('bins') of the size 0.1 in Ω_B. We calculate the mean value and the variance of the particle number for each of the bins at time $t = 0.2$. The results are presented in Figure 3.4 as grey histograms. In this example, it is easy to calculate the correct distribution $n(t, x)$ which the algorithm (A1)–(A5) tries to approximate. It can be obtained by solving the diffusion equation (3.7) in the domain $\Omega = [-1, 1]$. The expected values for both means and variances are plotted as dashed (red) lines in Figure 3.4.

In Figure 3.4(a), we see that the mean value of the simulation results matches well with the solution of the diffusion process, with only small fluctuations close to the internal boundary I at $x = 0$ due to stochastic effects and inaccuracies caused by the numerical approximation of $\widetilde{c}(t + \Delta t, x)$. However, in Figure 3.4(b) it becomes clear that the variance between different realisations is higher than the desired value, in particular close to the internal interface. These kind of numerical artefacts were

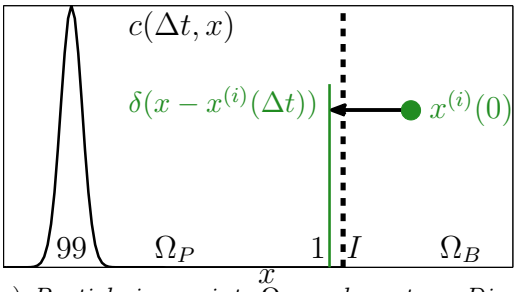


(a) Mean (b) Variance

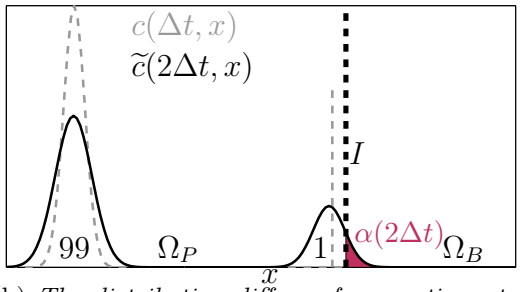
Figure 3.4: *Simulation results of diffusion of* 100 *particles in* $(-1, 1)$ *with no flux boundary conditions initialized at* $x = -0.95$*, i.e.* $n(0, x) = 100\,\delta(x + 0.95)$*. Results averaged over* 10^3 *realisations. Dashed (red) line: expected outcome.* (a) *Solid (black) line: mean value in* Ω_P*; (grey) histograms: particle concentrations in* Ω_B*.* (b) *(Grey) histograms: measured variances in particle concentrations. Parameter values are described in the text.*

always to be expected when coupling an inherently stochastic BD simulation (with a single particle resolution) with a deterministic PDE system (that only considers concentrations). In particular, we identified one such effect and will explain it using a thought experiment.

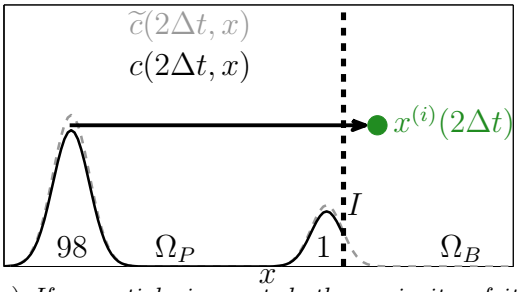
Let us consider a situation where the concentration $c(0, x)$ is 0 close to the internal boundary I and has a peak of mass 99 arbitrarily far away from I. Additionally, we assume that one particle is situated in Ω_B close to the interface I. Assuming the particle crosses the interface in the first simulation step, a Dirac δ function is created in Ω_P close to the interface, as illustrated in Figure 3.5(a). This δ function has a large impact on the region in Ω_P that is close to the interface, as the concentration $c(\Delta t, x)$ is negligible in this region. Immediately after incorporating the particle into $c(\Delta t, x)$, all its information is lost and we are forced to assume 100 independent particles, each following the probability distribution $c(\Delta t, x)/N$ in Ω_P. In particular this implies that every particle has a 1% chance of being at the position of the δ and 99% chance of being in the bulk distribution far away from the boundary. This is indeed not the case, since we know that there is exactly one molecule near the interface and 99 molecules away from the interface, but the nature of the continuum distribution means that this information must be lost as one would otherwise have to keep track of a separate distribution for all molecules in Ω_P, which would be equivalent to a BD simulation.



(a) *Particle jumps into Ω_P and creates a Dirac δ function at the landing position.*



(b) *The distribution diffuses for one time step and an amount of the δ flows back across the boundary.*



(c) *If a particle is created, the majority of its mass is virtually taken from the bulk distribution and the peak near the boundary remains.*

Figure 3.5: *Thought experiment that leads to errors in the variance.*

In the second diffusion step, we calculate $\widetilde{c}(2\Delta t, x)$ according to (3.10) and some mass $\alpha(2\Delta t)$ may have drifted across the interface I (see Figure 3.5(b)). Because the bulk distribution is far away from the boundary, almost all of this mass $\alpha(2\Delta t)$ comes from the δ function close to I. Let us imagine that a new particle is now created in Ω_B (according to the probability $\alpha(2\Delta t)$), in which case the whole distribution needs to be rescaled, as shown in Figure 3.5(c). Because 99% of the mass in $c(\Delta t, x)$ is situated in the bulk far away from the boundary, the majority of the rescaling happens in this region, such that effectively the mass needed to create the particle is almost entirely taken from the bulk, rather than from the region close to the interface. This also implies that most of the mass close to the boundary will stay and it is therefore possible to create another particle from this mass in further time steps. This is in contradiction to the result that would be expected if information was not lost in the first time step. That is, the distribution close to the interface should disappear and the bulk far from the interface is left alone. This effect is the main reason a higher than expected variance can be measured in Ω_B. Note, however, that this does not effect the expected values, as shown in Theorem 3.1.

Of course, this thought example is an extreme case: we would expect that in prac-

tical cases there would be a significant density of particles throughout the continuum region (otherwise we could be tracking them individually). Nevertheless the fact that all information about an individual particle is lost as soon as it crosses the interface does generate an error in the variance of particle numbers near the interface as shown in Figure 3.4(b), and the effect becomes more pronounced when the concentrations in Ω_P close to I are low.

Dependence of the variance on the system parameters

We want to quantify the error in the variance as a function of the system parameters, which are the size of the domain $[-L, L]$, the diffusion constant D, the simulated time t and the total mass N_{tot}. After a non-dimensionalisation the macroscopic PDE (3.7) can be written in the form

$$\frac{\partial n}{\partial t} = \frac{\partial^2 n}{\partial x^2}\,, \qquad x \in [-1, 1]\,,$$

where the simulation is run until

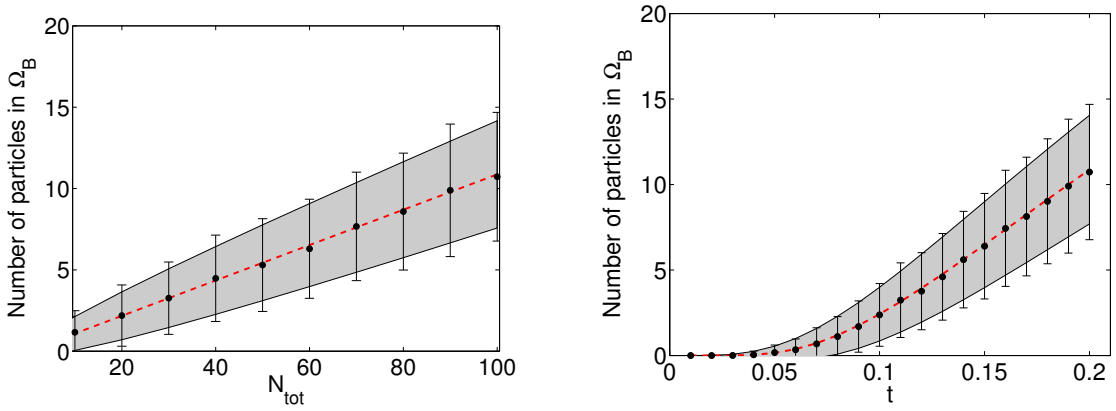
$$t^* = t\frac{L^2}{D}\,.$$

Hence, the system only has two parameters that need to be investigated: the simulation time t and the total number of molecules N_{tot}.

As in Figure 3.4, we use PBD algorithm (A1)–(A5) with $\Omega_P = (-1, 0)$, $\Omega_B = (0, 1)$, $I = \{0\}$ and initial condition $n(0, x) = 100\,\delta(x + 0.95)$. For each value of N_{tot} we simulate the system 10^3 times until $t = 0.2$ and measure in each case the number of particles in Ω_B at the end of the simulation, i.e. the value $N(t)$. For $N_{\text{tot}} = 100$ we also collect the intermediate values of $N(t)$. In Figure 3.6, we see that the measured mean values match well with the expected outcomes. For the standard deviations, however, we see that for all values of t and N_{tot} the measured outcomes are higher than expected. This is an undesired effect and the next section will discuss a way to reduce this artefact.

3.4 A PBD algorithm with an overlap region

In Section 3.3.4 we saw that the immediate return of particles from Ω_P into Ω_B in combination with relatively low concentrations close to the interface can lead to errors in the variance of particle concentrations in Ω_B. One way to overcome this problem is the introduction of an overlap region where BD simulation and a continuum description exist in parallel, i.e. we will consider the case [B] defined in Section 3.2 by



(a) $N_{\text{tot}} = 10, 20, \ldots, 100, \quad t = 0.2$ (b) $t = 0.01, 0.02, \ldots, 0.2, \quad N_{\text{tot}} = 100$

Figure 3.6: *Mean values and standard deviations of the number of particles $N(t)$ depending on t and N_{tot}. Dashed (red) line: expected number of particles; (grey) shaded area: expected standard deviation; (black) dots: measured mean values; (black) error bars: measured standard deviations. Other parameters as described in the text.*

$\Omega_B \cap \Omega_P \neq \emptyset$. A sketch of this new set up can be seen in Figure 3.7 with the overlap region denoted as $O = \Omega_B \cap \Omega_P$. We also denote the interfaces I_1 and I_2 by

$$I_1 = \partial\Omega_B \cap \Omega_P, \qquad I_2 = \Omega_B \cap \partial\Omega_P. \tag{3.17}$$

For the case of pure diffusion, this new set up requires only subtle changes in the algorithm. The Brownian particles move throughout the whole of Ω_B and the boundary I_2 has no effect on their behaviour. They only get incorporated into the continuum part when they cross the far boundary I_1. In this case, we again place a δ at their exact position in $\Omega_P \setminus \Omega_B$.

For the continuum concentration $c(t, x)$ in Ω_P, the boundary I_1 takes the role of I in the algorithm (A1)–(A5), whilst I_2 does not effect the update behaviour. This implies that the update step is identical to the previous one described in (3.3.1) and illustrated in Figures 3.2 and 3.3 when I is replaced by I_1. In terms of update equations the overlap region has minor effects. The definition of $\widetilde{c}$ is still equal to (3.10), but we have to redefine $\alpha(t + \Delta t)$ and $p_2(t + \Delta t, x)$ as follows

$$\alpha(t + \Delta t) = \int_{\Omega_B \setminus \Omega_P} \widetilde{c}(t + \Delta t, x)\, \mathrm{d}x\,, \tag{3.18}$$

$$p_2(t + \Delta t, x) = \frac{\widetilde{c}(t + \Delta t, x)}{\alpha(t + \Delta t)}\,, \qquad \text{for } x \in \Omega_B \setminus \Omega_P\,. \tag{3.19}$$

The introduction of the overlap region prevents undesired effects generated by molecules crossing over and coming back straight away, as discussed in the thought experiment in Section 3.3.4. In particular, a molecule initialized as a Dirac δ function in

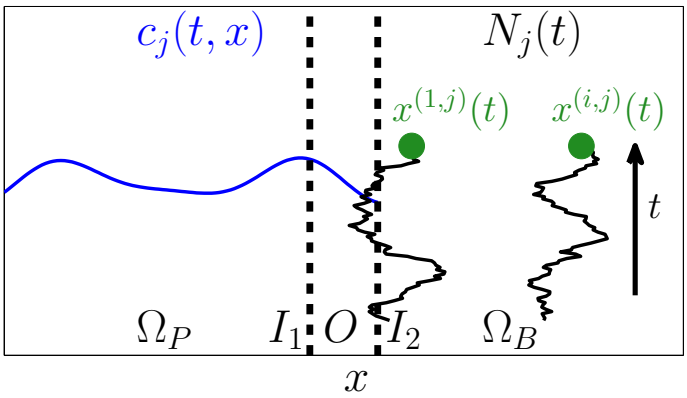


Figure 3.7: *Sketch of the PBD algorithm with overlap region O and the notation related to it. In Ω_P, molecules are described by their concentration $c_j(t,x)$, in the microscopic domain Ω_B described by the number $N_j(t)$ of molecules and their positions $x^{(i,j)}(t)$, $i = 1,\dots,N_j(t)$, $j = 1,\dots,M$. In the overlap region O, both descriptions exist in parallel. The interfaces between the subdomains are denoted I_1 and I_2 as defined in* (3.17).

$\Omega_P \setminus \Omega_B$ initially contributes very little to the overall probability density near the interface I_2; by the time it has a significant probability of crossing I_2 its distribution has become sufficiently spread that it is 'lost' in the subdomain Ω_P as is required for the continuous distribution. One time step of the second PBD algorithm is presented in Table 3.2. As before, (B1)–(B5) is written for the case that Δt is chosen so small that $\alpha(t+\Delta t) \ll 1$, but is easily extendible for larger values of Δt and $\alpha(t+\Delta t)$.

In order to highlight the advantages of the overlap region, we simulate the same diffusion process as in Figure 3.4 with $\Omega_P = [-1,0)$ and $\Omega_B = (-0.1,1]$. Then the overlap region is $O = \Omega_P \cap \Omega_B = (-0.1,0)$. The results are shown in Figure 3.8. As before, the mean outcome matches well with the exact solution, with stochastic fluctuations inside the overlap region O due to the mixed description. In Figure 3.8(b) we see that the introduction of the overlap region indeed solved the problem of high variances inside $\Omega_B \setminus \Omega_P$. A proof similar to Theorem 3.1 can be used to show that the PBD algorithm (B1)–(B5) satisfies Conditions (C.1) and (C.3) for all $\Delta t > 0$. We will now further show that the algorithm (B1)–(B5) describes a diffusion process exactly in the limit $\Delta t \to 0$.

Theorem 3.2. *Suppose that a PDE-description of the system is used in $\Omega_P = (-\infty,0)$, and a BD simulation in $\Omega_B = (-l,\infty)$ with the overlap region $O = (-l,0)$ where $l > 0$. In the limit that $\Delta t \to 0$ the expected concentration distribution $P_P(t,x) = \mathbb{E}[c(t,x)]$ in the continuum regime and the expected concentration dis-*

(B1) Calculate $\widetilde{c}(t+\Delta t, x)$ using (3.10) and $\alpha(t+\Delta t)$ using (3.18).

(B2) Generate a uniformly distributed random number $r \in (0,1)$.

(i) If $r < \alpha(t+\Delta t)$, then create a new particle in $\Omega_B \setminus \Omega_P$ according to the probability density $p_2(t+\Delta t, x)$ defined in (3.19). Set $\beta = \beta_{(\mathrm{i})}$, where $\beta_{(\mathrm{i})}$ is given by (3.15). Set $N_{\mathrm{new}} = 1$.

(ii) If $r \geq \alpha(t+\Delta t)$, then set $\beta = \beta_{(\mathrm{ii})}$, where $\beta_{(\mathrm{ii})}$ is given by (3.15). Set $N_{\mathrm{new}} = 0$.

(B3) Compute the positions $x^{(i)}(t+\Delta t)$, $i = 1, 2, \ldots, N(t)$, of the BD particles according to (3.16).

(B4) Compute the new concentration in Ω_P by

$$c(t+\Delta t, x) = \beta\, \widetilde{c}(t+\Delta t, x) + \sum_{x^{(i)}(t+\Delta t) \in \Omega_P \setminus \Omega_B} \delta\left(x - x^{(i)}(t+\Delta t)\right), \quad x \in \Omega_P\,.$$

(B5) Update the number of BD particles by

$$N(t+\Delta t) = N(t) + N_{\mathrm{new}} - \left|\left\{x^{(i)}(t+\Delta t) \in \Omega_P \setminus \Omega_B\,, i = 1, \ldots, N\right\}\right|\,.$$

Terminate computation of the trajectories of BD molecules which landed in $\Omega_P \setminus \Omega_B$ (i.e. the BD particles which satisfy $x^{(i)}(t+\Delta t) \in \Omega_P \setminus \Omega_B$.) Then continue with step (B1) for time $t + \Delta t$.

Table 3.2: *One time step of the PBD algorithm with overlap region for a system of diffusing molecules.*

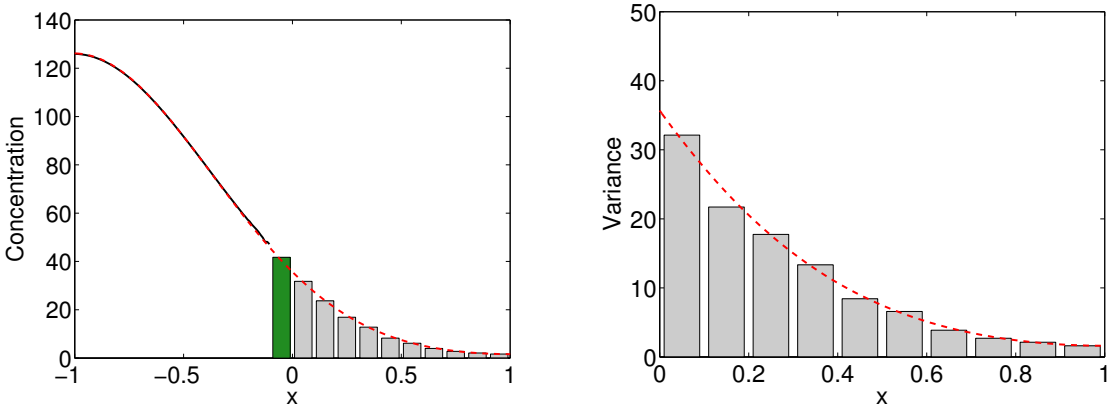
tribution $P_B(t,x)$ in the BD regime obey the equations

$$\frac{\partial P_P}{\partial t} = D\frac{\partial^2 P_P}{\partial x^2} + D\frac{\partial P_B}{\partial x}\bigg|_{x\to -l_+} \delta\,(x+l)\,, \quad x \in \Omega_P\,,$$
$$\frac{\partial P_B}{\partial t} = D\frac{\partial^2 P_B}{\partial x^2} - D\frac{\partial P_P}{\partial x}\bigg|_{x\to 0_-} \delta\,(x)\,, \qquad x \in \Omega_B\,,$$

where

$$P_P(t,0) = 0\,, \qquad P_B(t,-l) = 0\,, \qquad \textit{for} \quad t > 0\,.$$

Extend each distribution to the whole line by defining $P_P(t,x) = 0$ for $x \in (0,\infty)$ and $P_B(t,x) = 0$ for $x \in (-\infty, -l)$. Then the sum of these two distributions $n(t,x) = P_P(t,x) + P_B(t,x)$, $x \in \Omega$, $t > 0$ satisfies the diffusion equation (3.7).



(a) Mean (b) Variance

Figure 3.8: *Simulation results of a diffusion process in* $\Omega = (-1, 1)$ *obtained by* (B1)–(B5) *with no flux boundary conditions and initial conditions* $n(0, x) = 100\,\delta(x + 0.95)$ *with* $\Omega_P = (-1, 0)$, $\Omega_B = (-0.1, 1)$ *and* $O = (-0.1, 0)$ *averaged over* 10^3 *realisations. Dashed (red) line: expected outcome.* (a) *Solid (black) line: mean value in* Ω_P*; histograms: particle concentrations in* $\Omega_B \setminus O$ *(grey) and* O *(green).* (b) *(Grey) histograms: measured variances in particle concentrations. Parameter values as for Figure* 3.4.

Proof. Consider the function

$$n(t, x) = P_P(t, x) + P_B(t, x)\,.$$

Clearly

$$\frac{\partial n}{\partial t} = D\frac{\partial^2 n}{\partial x^2} \qquad \text{for } x \in (-\infty, -l) \cup (-l, 0) \cup (0, \infty)\,.$$

Since each of P_P and P_B is continuous at $x = -l$ and $x = 0$, function n will be continuous there. Moreover, since

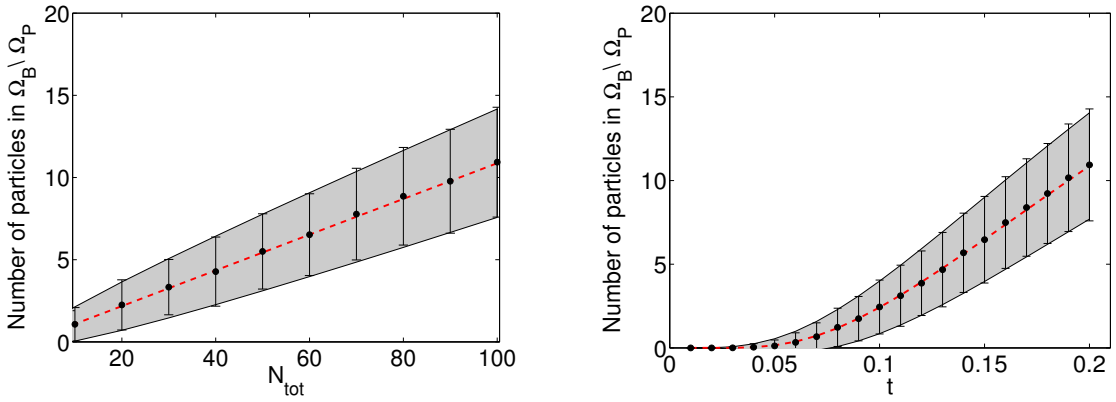
$$\left[\frac{\partial P_P}{\partial x}\right]_{x=0_-}^{x=0_+} = -\frac{\partial P_P}{\partial x}\,(t, 0_-)\,, \qquad \left[\frac{\partial P_B}{\partial x}\right]_{x=0_-}^{x=0_+} = \frac{\partial P_P}{\partial x}\,(t, 0_-)\,,$$

$\partial n/\partial x$ is continuous at $x = 0$. Similarly, since

$$\left[\frac{\partial P_P}{\partial x}\right]_{x=-l_-}^{x=-l_+} = -\frac{\partial P_B}{\partial x}\,(t, -l_+)\,, \qquad \left[\frac{\partial P_B}{\partial x}\right]_{x=-l_-}^{x=-l_+} = \frac{\partial P_B}{\partial x}\,(t, -l_+)\,,$$

$\partial n/\partial x$ is also continuous at $x = -l$. By standard regularity results this is enough to guarantee that n satisfies the diffusion equation on the whole real line. □

Remark. For the overlap region algorithm to give a different result from the simple PBD algorithm (A1)–(A5), we should choose l much larger than the mean displacement $\sqrt{2D\Delta t}$ of a particle given the time step Δt as defined in (3.16).



(a) $N_{\text{tot}} = 10, 20, \ldots, 100, \quad t = 0.2$ (b) $t = 0.01, 0.02, \ldots, 0.2, \quad N_{\text{tot}} = 100$

Figure 3.9: *Mean values and standard deviations of the number of particles in* $\Omega_B \setminus \Omega_P$ *depending on* t *and* N_{tot}*. Dashed (red) line: expected number of particles; (grey) shaded area: expected standard deviation; (black) dots: measured mean values; (black) error bars: measured standard deviations. Parameters as for Figure* 3.8*.*

Note that the result in Theorem 3.2 represents another advantage of algorithm (B1)–(B5) over (A1)–(A5), as the algorithm without overlap region does not work in the limit $\Delta t \to 0$, because particles would have to be created on the interface and would then constantly switch between descriptions. In practice this means that decreasing Δt in (A1)–(A5) increases the inaccuracies in variances.

Dependence of the variance on the system parameters

In order to show that the variances are indeed accurately reproduced by the PBD algorithm (B1)–(B5), we repeat the numerical experiment conducted in Figure 3.6. We choose $\Omega_P = [-1, 0)$, $O = (-0.1, 0)$ and $\Omega_B = (-0.1, 1]$ and measure the number of molecules situated in $\Omega_B \setminus \Omega_P = [0, 1]$ at the end of the simulation. We again calculate the mean values and standard deviation and present the results in Figure 3.9. We can clearly see that the adjusted algorithm produces more accurate standard deviations than the original algorithm, whilst keeping the mean values correct. We conclude that the algorithm (B1)–(B5) produces an accurate BD simulation inside $\Omega_B \setminus \Omega_P$ and therefore satisfies the Conditions (C.1)–(C.3). We will use the PBD algorithm (B1)–(B5) throughout the remainder of this chapter.

3.5 Reaction-diffusion systems

In the next step we introduce chemical reactions into the system presented in Section 3.4. We will concentrate on zero-order and first-order reactions [51]. First-order

reactions are reactions which only have one reactant, for example,

$$X_1 \xrightarrow{k_1} X_2 \qquad \text{or} \qquad X_1 \xrightarrow{k_2} X_4 + X_5\,, \tag{3.20}$$

where X_i denote chemical species and k_1 (resp. k_2) the corresponding rate constant (which has physical units $[\text{sec}^{-1}]$). In what follows, we will denote by $\emptyset$ chemical species which are of no interest to the modeller. Then, considering that X_1 is the only chemical species of interest, we can rewrite reactions (3.20) as

$$X_1 \xrightarrow{k_d} \emptyset\,, \tag{3.21}$$

where $k_d = k_1 + k_2$. We will also consider zero-order reactions. An example is

$$\emptyset \xrightarrow{k_p} X_1\,, \tag{3.22}$$

where the rate constant k_p has physical units $[\,\text{m}^{-3}\,\text{sec}^{-1}]$, i.e. it is the production rate per unit of volume and unit of time. It is relatively straightforward to implement zero-order and first-order chemical reactions in the PBD algorithms (A1)–(A5) and (B1)–(B5), because these reactions can be treated in the individual parts of the system (continuum and BD simulation) independently. Note that for higher-order reactions this is not necessarily the case, as particles could react with the continuum inside the overlap region O. We will discuss this situation in more depth in Section 3.7.

In the continuum regime reactions are represented by the term $R_j(p_1, p_2, \ldots, p_M)$ on the right-hand side of the reaction-diffusion PDE (3.1). For example, if the chemical species X_1 is subject to chemical reactions (3.21)–(3.22), then the reaction-diffusion PDE (3.1) takes the form

$$\frac{\partial c_1}{\partial t} = D_1 \frac{\partial^2 c_1}{\partial x^2} - k_d\, c_1 + k_p\,.$$

In the BD simulations, the molecules act independently and the reactions can therefore be treated individually. A summary of how to implement various reactions in BD simulations can be found in [51].

Although an implementation of the zero-order and first-order reactions is relatively straightforward, one has to still consider some special effects that are related to the coupling of the two parts of the domain. First, what happens when a particle that is supposed to react at time t_1 crosses the interface I_1 at an earlier time $t_2 < t_1$? Since we assumed that all information about particles is lost as soon as they cross the interface I_1, we incorporate it into the continuum and the reaction at time t_1 does not happen. Second, when a creation process is present inside the overlap region O, this

cannot simply be implemented in both parts, because this would double the creation. A number of solutions to this problem are possible: one could split the creation, or declare creation to only contribute to either the continuum or the molecular-based description. We will here assume that all creation inside the overlap region occurs in the form of molecules with exact positions.

Finally, let us note that (reactive) boundary conditions on the external boundary $\partial\Omega$ can be treated according to the corresponding modelling regime, i.e. whether the corresponding segment of $\partial\Omega$ is part of $\partial\Omega_P$ or $\partial\Omega_B$. Derivation of reactive boundary conditions of BD simulations which are consistent with the PDE description can be found in [47]. External boundaries slightly modify the computation of $\widetilde{c}(t+\Delta t, x)$ in Ω_P. It is still given by (3.10) but the kernel (3.11) has to be updated to take into account the boundary condition imposed on the external boundary $\partial\Omega$. We have already done this when we showed simulations of the diffusion process in Figures 3.4 and 3.8 in the finite interval $[-1,1]$ with no flux boundary conditions. We conclude this section with four simulation examples which illustrate the behaviour of the PBD algorithm (B1)–(B5) for realistic reaction-diffusion systems. They include the modelling of morphogen gradients, chemisorption and a revisit of bacterial chemotaxis introduced in Chapter 2.

3.5.1 Example 1: Morphogen gradient

In the first example we compute a steady state for a morphogen gradient model [89, 159, 169]. We consider one chemical species (morphogen) inside the domain $\Omega = [-1,1]$. All parameters are dimensionless for simplicity. The only reaction inside $\Omega = [-1,1]$ is the degradation (3.21). Additionally to this reaction, we assume a constant influx J/D through the left-hand boundary $x = -1$ to the continuum subdomain $\Omega_P = [-1,0)$. We use $O = (-0.1, 0)$ and $\Omega_B = (-0.1, 1]$ with a no flux boundary at $x = 1$. Since we only have first-order reaction (3.21), the exact solution $n(t,x)$ satisfies the system

$$\frac{\partial n}{\partial t} = D\frac{\partial^2 n}{\partial x^2} - k_d n\,, \qquad \frac{\partial n}{\partial x}(t,-1) = -J\,, \qquad \frac{\partial n}{\partial x}(t,1) = 0\,. \tag{3.23}$$

This system is initialised with $n(0,x) = 0$, $x \in \Omega$ and we are interested in the steady state solution. We use $D = 1$, $J = 10^3$ and $k_d = 1$. The second PBD algorithm (B1)–(B5) is run with the parameters presented in Section 3.4 until $t = 20$, after which the system can be considered to have (approximately) settled into its steady state. The reaction (3.21) is simulated in a time-driven manner in Ω_B, which means

that for each morphogen molecule it is decided randomly at the end of each time step whether it was degraded or not (the probability of degradation of each molecule is equal to $k_d\,\Delta t$ provided that $k_d\,\Delta t \ll 1$).

The result of a single simulation of the PBD algorithm (B1)–(B5) is plotted in Figure 3.10(a). We plot the PDE solution in $\Omega_P \setminus O$ as a solid (black) line and the particle concentration in $\Omega_B \setminus O$ as the (grey) histograms. In the overlap region O, we compute the total mass by

$$N_O(t) \equiv \int_O c(t,x)\,\mathrm{d}x + \left|\left\{x^{(i)}(t) \in O\,, i = 1,\ldots,N\right\}\right|. \tag{3.24}$$

The value of $N_O(t)$ is plotted as the green bar in Figure 3.10(a). The results of a single simulation of the PBD algorithm (B1)–(B5) are compared with the steady state solution $n(\infty,x)$ of (3.23) given through

$$n(\infty,x) = -\frac{J}{\sqrt{k_d/D}}\frac{\exp\left(\sqrt{k_d/D}\,(x-2)\right) + \exp\left(-\sqrt{k_d/D}\,x\right)}{\exp\left(-3\sqrt{k_d/D}\right) - \exp\left(\sqrt{k_d/D}\right)}\,.$$

This steady state solution is presented as dashed (red) line in Figure 3.10.

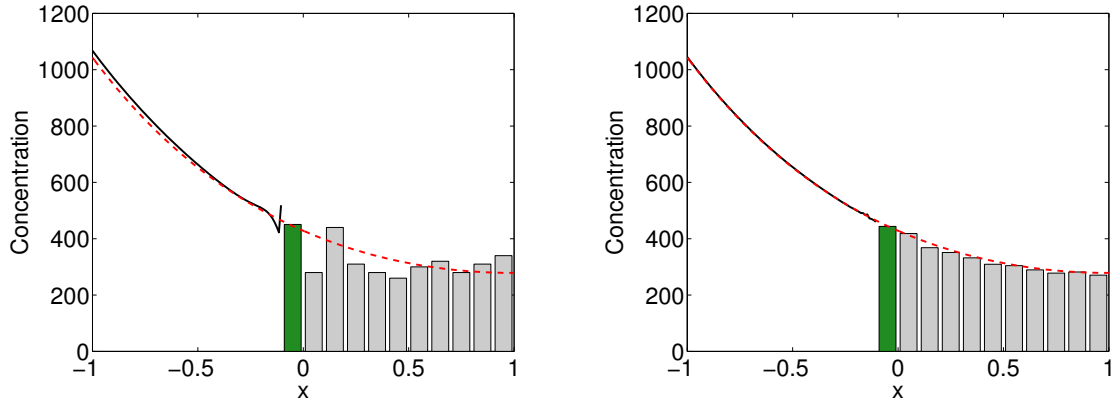
Note that the jagged appearance of the continuum solution close to the interface is not numerical error, but represents the fact that as molecules cross from the discrete to the continuum side information about their exact location is lost gradually over time (remember that Figure 3.10(a) shows just one realisation of the stochastic process). The corresponding distribution on the discrete side $\Omega_B \setminus \Omega_P$ would be δ function spikes at the location of the particles, which we have in effect locally averaged by the binning process. Thus the jaggedness can be seen as a gradual transition in the solution from isolated discrete particles to a continuum density distribution. An ensemble average over 100 simulations of the PBD algorithm (B1)–(B5) is shown in Figure 3.10(b). The stochastic fluctuations are reduced compared to the single simulation and the results compare well with the exact solution for the expected probability density (3.23).

3.5.2 Example 2: Reversed morphogen gradient

In this example we introduce a second reaction in addition to (3.21) – a local production of molecules:

$$\emptyset \xrightarrow{k_p} A \qquad \text{in } [x_s, 1]\,, \tag{3.25}$$

where x_s defines the size of the creation zone. As before we consider all parameters to be dimensionless: k_p is defined as the rate of production per unit length. For



(a) Single simulation

(b) Average over 100 realisations

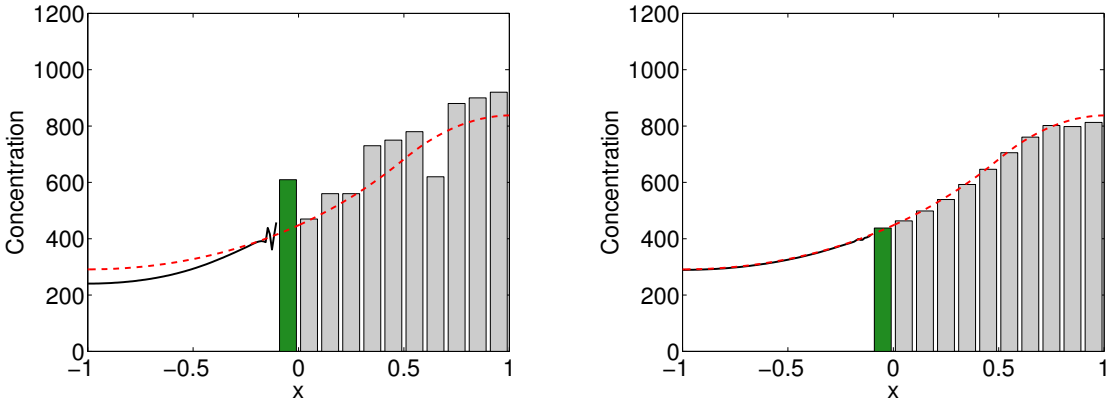
Figure 3.10: *Example* 1. *Morphogen gradient: Concentration. Dashed (red) line: exact solution given by* (3.23)*; solid (black) line:* $c(t,x)$*; (grey) histograms: spatial concentration of particles at* $t = 20$ *in* $\Omega_B \setminus O$*; (green) bar:* $N_O(t)$ *given by* (3.24)*. Parameters as described in the text.*

this system we use no flux boundary conditions on both ends. The combination of localized production (3.25) and degradation (3.21) ensures that the system settles into a non-trivial steady-state which we will compute with the PBD algorithm (B1)–(B5). The exact solution (which the PBD algorithm (B1)–(B5) approximates) satisfies the system

$$\frac{\partial n}{\partial t} = D\frac{\partial^2 n}{\partial x^2} - k_d n + k_p \chi_{[x_s,1]}\,, \qquad \frac{\partial n}{\partial x}(t,-1) = 0\,, \qquad \frac{\partial n}{\partial x}(t,1) = 0\,, \tag{3.26}$$

where $\chi_{[x_s,1]}$ is the characteristic function for the interval $[x_s,1]$ that takes the value 1 inside and 0 outside of the interval. The production reaction (3.25) was implemented in the BD simulations in an event-driven way, such that particles can be created at any time in-between time steps and the number of particles created in one time step is not limited. We used $D = 1$, $k_p = 1$, $k_d = 2 \times 10^3$ and $x_s = 0.5$. For the PBD simulations we use the same parameters as in Figure 3.8. In particular, we have $\Omega_P = [-1,0)$, $O = (-0.1,0)$ and $\Omega_B = (-0.1,1]$.

A single realisation of this process is plotted in Figure 3.11(a). We plot the PDE solution in $\Omega_P \setminus O$ as a black line. The concentration of molecules in $\Omega_B \setminus O$ is visualized as (grey) histograms. In the overlap region O, we plot $N_O(t)$ given by (3.24). The concentration gradient is now reversed, as the creation of particles happens near the right-hand boundary. Again, one can clearly see the stochastic fluctuations in this plot which also have effects on the value of $c(t,x)$ far from the overlap region O. However, as we draw an ensemble average over 100 simulations in Figure 3.11(b),



(a) *Single simulation* (b) *Average over 100 simulations*

Figure 3.11: *Example* 2. *Reversed morphogen gradient: Concentration. Dashed (red) line: exact solution given by* (3.26)*; solid (black) line:* $c(t, x)$ *in* $\Omega_P \setminus O$*; (grey) histograms: spatial concentration of particles at* $t = 20$ *in* $\Omega_B \setminus O$*; (green) bar:* $N_O(t)$ *given by* (3.24)*. Parameters as described in the text.*

the results converge towards the steady state solution of (3.26), which is plotted as a dashed (red) line in Figure 3.11.

3.5.3 Example 3: Chemisorption

In this example we study polymer coating of a virus surface [50, 59]. We will describe it as irreversible adsorption (chemisorption) of polymers to a two-dimensional surface as was introduced in [48]. This example presents a typical application area of PBD algorithms. A detailed model is used close to the reactive boundary where positions of individual molecules influence the dynamics of diffusion-driven adsorption. On the other hand, a less detailed model can be used far away from the adsorbing surface. In the bulk the behaviour of reactive polymers can be described by the macroscopic reaction-diffusion PDE (3.1) in the form

$$\frac{\partial c}{\partial t} = D\frac{\partial^2 c}{\partial z^2} - k_d\, c\,. \tag{3.27}$$

in the semi-infinite domain $\Omega = (0, \infty)$ (here, z is the distance from the reactive surface which is at $z = 0$). This equation takes into account two processes which mainly influence chemisorption dynamics [48]: diffusion of polymer molecules and the hydrolysis of reactive groups in the solution. Both processes can be implemented in the BD context as we saw in the previous examples. However, this level of detail is only needed close to the virus surface.

Whenever a polymer molecule interacts with the surface, it is either reflected or (irreversibly) adsorbed. The chemisorption is modelled by a random sequential adsorption (RSA) algorithm [46]: we check whether the corresponding binding site on the surface is free and then the reaction occurs with a certain probability. This probability is related to the reaction rate constant of the binding reaction as given in [47]. The reader can find more details about the model in [48]. In this example, we show that the PBD algorithm (B1)–(B5) can be used to reproduce the results from [48]. We use the same parameters as in [48], namely $D = 5 \times 10^{-5}\,\mathrm{mm}^2\,\mathrm{s}^{-1}$, $k_d = 1.3 \times 10^{-4}\,\mathrm{s}^{-1}$ and $\Delta t = 0.01\,\mathrm{s}$. Then the mean displacement per time step according to (3.16) is $\sqrt{2D\Delta t} = 10^{-3}\,\mathrm{mm}$ and we therefore choose the size of the overlap region of the PBD algorithm (B1)–(B5) as $|O| = 10^{-2}\,\mathrm{mm}$. From the results in [48], we estimate that a maximum length of $L = 2\,\mathrm{mm}$ is enough to simulate the binding process and use the Dirichlet boundary condition

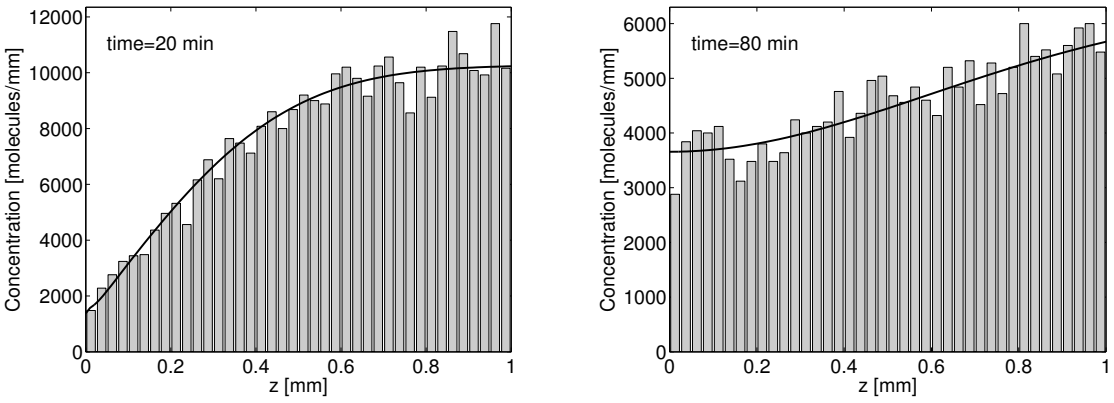
$$n(t, L) = c_0 \exp(-k_d t)\,,$$

where $c_0 = 1.2 \times 10^4\,\mathrm{molecules/mm}$ is the initial concentration of molecules (i.e. $n(0, z) \equiv c_0$ for $z \in \Omega$). To apply the PBD algorithm (B1)–(B5), we choose $\Omega_B = [0, 1.01)\,\mathrm{mm}$, $O = (1, 1.01)\,\mathrm{mm}$ and $\Omega_P = (1, 2)\,\mathrm{mm}$. The RSA algorithm was performed using a nearest neighbour exclusion on a 100×100 grid of receptor binding positions on the surface [46].

In Figure 3.12 we plot the concentration profile inside Ω_B of a single simulation at two different times (grey histograms). We compare the results of the PBD algorithm (B1)–(B5) with the results of the RSA-PDE model presented in [48] (black lines). As shown in [48], the RSA-PDE model also compares well with the full BD simulation. The number of molecules which are attached to the surface as a function of time is plotted in Figure 3.13 (six realisations of the PBD algorithm (B1)–(B5) are plotted as green solid lines). Again, we see an excellent agreement with the result from [48] which is plotted as the dashed (black) line.

3.5.4 Example 4: Bacterial chemotaxis

We use this last example to show the capabilities of the PBD method by applying it to a more complex situation than purely diffusing particles. At the same time this example connects the results presented in this chapter with the system studied in Chapter 2, as we implement a combination of hybrid models of the type presented here and of hybrid models of the type seen before.



(a) $t = 20\,\mathrm{min}$ (b) $t = 80\,\mathrm{min}$

Figure 3.12: *Example* 3. *Chemisorption to virus surface: Concentration. (Grey) histograms: concentration profile in molecules/mm at two given times computed by the PBD algorithm* (B1)–(B5)*; solid (black) line: results of the RSA-PDE model presented in* [48]*. Parameters as shown in the text.*

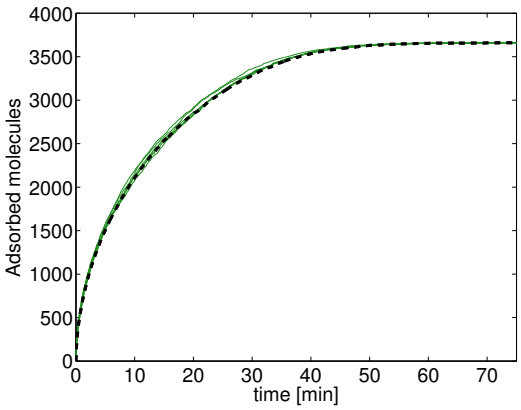


Figure 3.13: *Example* 3. *Chemisorption to virus surface: Number of polymer molecules which are bound to the virus surface as a function of time. Solid (green) lines: six realisations computed by the PBD algorithm* (B1)–(B5)*; dashed (black) line: results of the RSA-PDE model presented in* [48]*. Parameters as shown in the text.*

We start the example with the dimensionless version of the Keller-Segel model (2.11)–(2.12) in standing-wave form

$$\begin{aligned}\frac{\partial n}{\partial t} - \frac{\partial n}{\partial x} &= \mu \frac{\partial}{\partial x}\left(\frac{\partial n}{\partial x} - \kappa n \frac{1}{S}\frac{\partial S}{\partial x}\right), \\ \frac{\partial S}{\partial t} - \frac{\partial S}{\partial x} &= -n\,,\end{aligned}$$

where $n(t,x)$ denotes the concentration of bacteria and $S(t,x)$ denotes the concentration of extracellular signal. As was shown in Theorem 2.1, the wave speed of the travelling wave is 1 for the dimensionless version. The advantage of considering this standing-wave form is that the domain containing the bacteria stays relatively constant, which in turn allows for the use of a PBD method with an individual-based model for the particles behind the wave. One can also interpret this standing-wave form as a simulation in a co-travelling frame [94]. In contrast to all other systems in this chapter, the particles in this example do not undergo a simple diffusion process, but follow the biased Brownian motion

$$\mathrm{d}x^{(i)} = \left(\mu\kappa \frac{1}{S(x^{(i)})}\frac{\partial S(x^{(i)})}{\partial x} - 1\right)\mathrm{d}t + \sqrt{2\mu}\,\mathrm{d}W\,. \tag{3.28}$$

The system can be fully described through the concentration of extracellular signal $S(t,x) : [0,\infty) \times \Omega \mapsto \mathbb{R}_0^+$, the concentration of bacteria in the adequate region $n_P(t,x) : [0,\infty) \times \Omega_P \mapsto \mathbb{R}_0^+$ and the positions of the $N(t)$ agents $x^{(i)}(t) \in \Omega_B$, $i = 1,\ldots,N(t)$. The total bacterial mass in the system given through

$$N_{\mathrm{tot}} = \int_{\Omega_P} n_P(t,x)\,\mathrm{d}x \;+\; N(t)\,,$$

is conserved throughout the simulation.

Example 4.1: Constant extracellular signal

In the first simulation we concentrate on the PBD part of the hybrid system, by keeping the extracellular signal constant. We use the travelling wave solution (2.19) for $S(t,x)$ and only change bacteria concentrations. We simulate $N_{\mathrm{tot}} = 10^4$ bacteria in the domain $\Omega = [-1,1]$ with $\Omega_P = [-0.15,1]$ and $\Omega_B = [-1,-0.05]$. We initialise all bacteria at $x = 0$, meaning the initial condition is $n_P(0,x) = N_{\mathrm{tot}}\,\delta(x)$, $N(0) = 0$; we apply no-flow (reflective) boundary conditions on both ends of the domain. The numerical solution is performed using an explicit Euler scheme with $\Delta t = 10^{-5}$ and $\Delta x = 2.5 \times 10^{-3}$. The system parameters are $\mu = 1/30$ and $\kappa = 2$. The result at $t = 1$ can be seen in Figure 3.14, where we use kernel density estimation with the

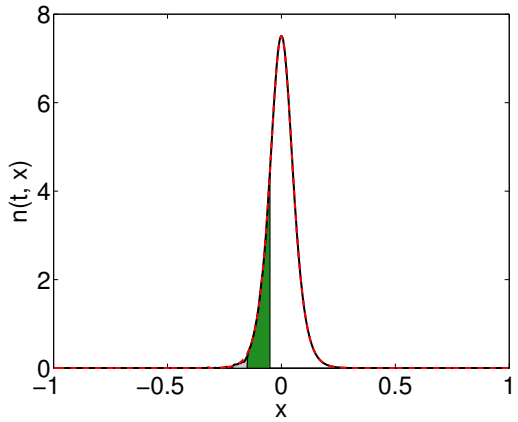


Figure 3.14: *Example* 4.1. *Bacterial chemotaxis with constant extracellular signal using PBD algorithm with* $\Omega_B = [-1, -0.05)$, $\Omega_P = (-0.15, 1]$ *and* $O = (-0.15, -0.05)$. *Solid (black) line: solution of the PBD algorithm; dashed (red) line: travelling wave solution* (2.19). *Filling indicates the different zones* $\Omega_B \setminus O$ *(grey),* O *(green) and* $\Omega_P \setminus O$ *(white).*

Gaussian kernel given in (2.21) and $\sigma = 5 \times 10^{-3}$ to obtain $n(1, x)$ inside Ω_B. The plot of $n(1, x)$ is superimposed onto the travelling wave solution (2.19) and we see that an excellent match is achieved.

Example 4.2: Changing extracellular signal

In the second part of this numerical example we use the double hybrid algorithm (Ch1)–(Ch5) given in Table 3.3 that combines the PBD method with the hybrid approach for bacterial chemotaxis presented in Chapter 2. The update of $S(t, x)$ in (Ch1) is performed using an estimate of $n(t, x)$ that is a combination of the PDE part and the particle-based description according to the PBD method. We again use kernel density estimation with the Gaussian kernel given in (2.21) and $\sigma = 5 \times 10^{-3}$ for the particle-based component. We apply the Dirichlet conditions

$$S(t, -1) = 0\,, \qquad S(t, 1) = 1\,, \qquad t > 0\,,$$

for the extracellular signal. All simulation parameters were chosen as in Example 4.1 and the results can be seen in Figure 3.15 compared to the travelling wave solutions of Keller and Segel (2.19). We again observe an excellent match between the hybrid method and the travelling wave solution indicating that this double hybrid method can be applied to bacterial chemotaxis in the standing-wave form. Additionally, closer observation of the bacteria concentration behind the wave shows the presence of particles that have dropped out from the wave, similar to the dropout discussed in

(Ch1) Calculate $\widetilde{n}_P(t+\Delta t, x)$ and $S(t+\Delta t, x)$ by numerically solving

$$\begin{aligned}\frac{\partial \widetilde{n}_P}{\partial t} - \frac{\partial \widetilde{n}_P}{\partial x} &= \mu \frac{\partial}{\partial x}\left(\frac{\partial \widetilde{n}_P}{\partial x} - \kappa \widetilde{n}_P \frac{1}{S}\frac{\partial S}{\partial x}\right), \\ \frac{\partial S}{\partial t} - \frac{\partial S}{\partial x} &= -\frac{1}{N_{\text{tot}}}\left(\widetilde{n}_P + \sum_{i=1}^{N} K\left(x - x^{(i)}(t)\right)\right),\end{aligned}$$

over the time interval $[t, t+\Delta t)$ and over all $x \in \Omega$ and calculate $\alpha(t+\Delta t)$ according to

$$\alpha(t+\Delta t) = \int_{\Omega_B \setminus \Omega_P} \widetilde{n}_P(t+\Delta t, x)\,\mathrm{d}x\,.$$

(Ch2) Update new particle positions according to the Euler-Maruyama approximation of (3.28) given by

$$x^{(i)}(t+\Delta t) = x^{(i)}(t) + \left(\mu\kappa \frac{1}{S\big(t, x^{(i)}(t)\big)} \frac{\partial S\left(t, x^{(i)}(t)\right)}{\partial x} - 1\right)\Delta t + \sqrt{2\mu\Delta t}\,\xi\,,$$

where $\xi \sim \mathcal{N}(0;1)$. $S\left(t, x^{(i)}(t)\right)$ and its derivative are approximated using linear interpolation of the values of S at neighbouring grid points.

(Ch3) Create N_{new} new particles similar to (B2) according to $\alpha(t+\Delta t)$ and calculate the rescaling factor β similar to (3.15).

(Ch4) Compute new bacteria concentration in Ω_P by

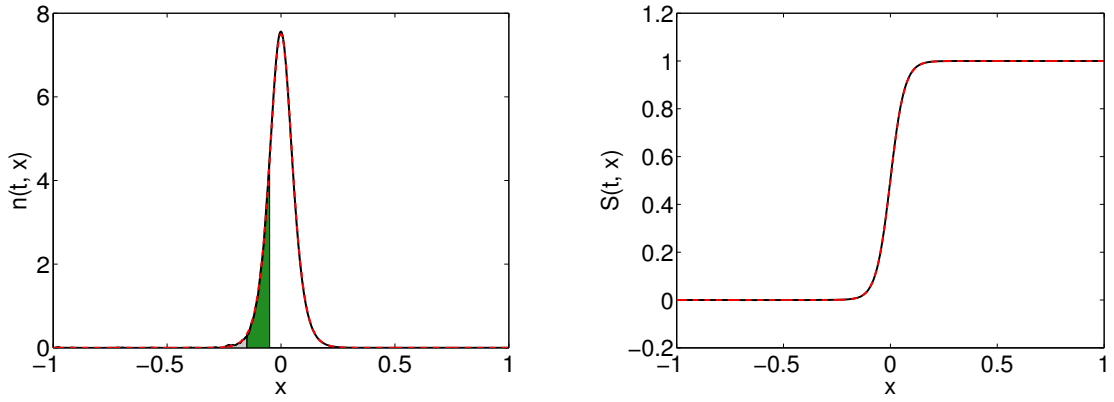
$$n_P(t+\Delta t, x) = \beta\,\widetilde{n}_P(t+\Delta t, x) + \sum_{x^{(i)}(t+\Delta t) \in \Omega_P \setminus \Omega_B} \delta\big(x - x^{(i)}(t+\Delta t)\big)\,, \quad x \in \Omega_P\,.$$

(Ch5) Update the number of BD particles by

$$N(t+\Delta t) = N(t) + N_{\text{new}} - \left|\left\{x^{(i)}(t+\Delta t) \in \Omega_P \setminus \Omega_B\,, i = 1, \ldots, N\right\}\right|\,.$$

Terminate computation of trajectories of BD molecules which landed in $\Omega_P \setminus \Omega_B$ (i.e. the BD particles which satisfy $x^{(i)}(t+\Delta t) \in \Omega_P \setminus \Omega_B$.)
Then continue with step (Ch1) for time $t + \Delta t$.

Table 3.3: *One time step of the double hybrid method for bacterial chemotaxis.*



(a) (b)

Figure 3.15: *Example* 4.2. *Bacterial chemotaxis using the double hybrid method* (Ch1)–(Ch5). *The PBD zones are* $\Omega_B = [-1, -0.05)$, $\Omega_P = (-0.15, 1]$ *and* $O = (-0.15, -0.05)$. *Solid (black) line: solution of the PBD algorithm; dashed (red) line: travelling wave solution* (2.19). *Filling indicates the different zones* $\Omega_B \setminus O$ *(grey),* O *(green) and* $\Omega_P \setminus O$ *(white).* (a) $n(t, x)$. (b) $S(t, x)$.

Chapter 2. One could therefore use the standing-wave form with this double hybrid method to improve understanding of the dropout effect.

3.6 Extension to higher space dimensions

In Sections 3.3–3.5 we developed PBD algorithms for a one-dimensional or a quasi one-dimensional domain. However, in most application areas, like ion channels [118] or filopodia [175] the three dimensional structure of the problem plays an important role for the behaviour of the system. We therefore aim to extend the algorithm (B1)–(B5) for two and three dimensional domains. Whilst in theory all of the steps (B1)–(B5) extend straightforwardly into higher dimensions, this is not necessarily the case in practice. One problem can be identified in step (B2) for case (i), where one has to sample from the d dimensional probability distribution $p_2(t + \Delta t, \mathbf{x})$ to find the position of new particles. One could use the values of $\widetilde{c}(t + \Delta t, \mathbf{x})$ on the gridpoints inside $\Omega_B \setminus O$ to sample from, but a large number of gridpoints in this region would be needed to maintain the required level of accuracy. Another problem with the algorithm (B1)–(B5) is that it calculates the new concentration $\widetilde{c}(t + \Delta t, \mathbf{x})$ for all points in Ω, thereby requiring one to solve the reaction-diffusion PDE (3.1) on the whole domain. Especially when considering three dimensional domains this can amount to a significant computational effort with little benefit, because only very little mass can be found inside Ω_B. In Section 3.6.1 we present a way to overcome

these two problems by reducing the PDE solver to the region Ω_P; the creation of new particles occurs in an event-based manner. We then apply this adjusted event-based algorithm to a diffusion problem in two dimensions in Section 3.6.2.

3.6.1 Event-based particle creation

The main idea to reduce the computational domain for the PDE solver stems from the proof to Theorem 3.2, where we see that in the limit $\Delta t \to 0$ the PBD algorithm (B1)–(B5) effectively enforces a homogeneous Dirichlet boundary condition on I_2 for the PDE part $c(t, \mathbf{x})$. If we were to change step (B1) by directly enforcing such a homogeneous boundary condition onto $c(t, \mathbf{x})$ and the solution $\widetilde{c}(t + \Delta t, \mathbf{x})$, we would only need a way to accurately measure the mass $\alpha(t + \Delta t)$ that crossed the boundary in this time step in order to keep the PBD algorithm accurate. Assuming we are aiming to solve the diffusion equation

$$\frac{\partial n}{\partial t} = D\Delta n\,, \tag{3.29}$$

we can then calculate the mass that left the region Ω_P in a given time step through

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega_P} c(t,\mathbf{x})\,\mathrm{d}\mathbf{x} = \int_{\Omega_P} \frac{\partial c(t,\mathbf{x})}{\partial t}\,\mathrm{d}\mathbf{x} = \int_{\Omega_P} D\Delta c(t,\mathbf{x})\,\mathrm{d}\mathbf{x} = \int_{\partial\Omega_P} D\,\mathbf{n}_\Omega \cdot \nabla c(t,\mathbf{x})\,\mathrm{d}\mathbf{x}\,,$$

where $\mathbf{n}_\Omega$ denotes the outwards pointing normal at the point $\mathbf{x} \in \partial\Omega_P$ and we have applied the divergence theorem in the last step. Assuming that we have no-flow boundary conditions along the outside of the domain Ω, this simplifies to

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega_P} c(t,\mathbf{x})\,\mathrm{d}\mathbf{x} = \int_{I_2} D\,\mathbf{n}_\Omega \cdot \nabla c(t,\mathbf{x})\,\mathrm{d}\mathbf{x}\,.$$

This implies that, if we have a method of measuring the gradient of $c(t, \mathbf{x})$ along the interface I_2, we can accurately calculate the mass $\alpha(t + \Delta t)$ that left the PDE regime in the time step $[t, t + \Delta t)$ through

$$\alpha(t+\Delta t) = -\int_t^{t+\Delta t}\int_{I_2} D\,\mathbf{n}_\Omega \cdot \nabla c(t,\mathbf{x})\,\mathrm{d}\mathbf{x}\,\mathrm{d}t\,.$$

In practice, $\alpha(t+\Delta t)$ can often be tracked as part of the numerical scheme that solves the PDE for the calculation of $\widetilde{c}(t + \Delta t, \mathbf{x})$. Moreover, using the definition above, one can define α as a function of position, with $\alpha(t + \Delta t, \mathbf{x})\,\mathrm{d}\mathbf{x}$ being the amount of mass that flowed across the area $\mathrm{d}S$ of the boundary in the vicinity of position $\mathbf{x} \in I_2$ in the time interval $[t, t + \Delta t)$ defined through

$$\alpha(t+\Delta t,\mathbf{x}) = -\int_t^{t+\Delta t} D\,\mathbf{n}_\Omega \cdot \nabla c(t,\mathbf{x})\,\mathrm{d}t\,, \qquad \mathbf{x} \in I_2\,.$$

This means that one can accurately calculate when the next particle will leave the domain Ω_P through the region $\Psi \subset I_2$ using Gillespie's algorithm [76] as follows

$$t_{\text{next}}(\Psi) = t - \frac{\log r}{\int_\Psi \alpha(t+\Delta t, \mathbf{x})\,\mathrm{d}\mathbf{x}}\,,$$

where $r \in [0,1]$ is a uniformly distributed random variable. For every gridpoint $\mathbf{r}^{(l)} \in I_2$ on the boundary, one can now define its associated region $\Psi^{(l)}$ depending on the choice of grid. It is then possible to use the numerical PDE solver to keep track of $\alpha\big(t+\Delta t, \mathbf{r}^{(l)}\big)$ and approximate $t_{\text{next}}\big(\Psi^{(l)}\big)$ using

$$t_{\text{next}}\big(\Psi^{(l)}\big) \approx t - \frac{\log r}{\big|\Psi^{(l)}\big|\alpha\big(t+\Delta t, \mathbf{r}^{(l)}\big)}\,,$$

for all boundary nodes $\mathbf{r}^{(l)}$. If this value is in the interval $[t, t+\Delta t)$, a new particle is created at point $\mathbf{r}^{(l)}$ at time $t_{\text{next}}\big(\Psi^{(l)}\big)$ and the rescaling of the system is done accordingly. Similarly, if no new particle is created we rescale the concentration $c(t,\mathbf{x})$ at the end of the time step or can use more advanced techniques to directly incorporate the redistribution of mass into the PDE solution. Because, we know exactly where a new particle was created at what time, we call this method an *event-based* implementation of the mass exchange. Note that this method can also be applied for the one-dimensional case, but that it presents a big improvement once we go into higher dimensions where the reduced number of gridpoints due to the smaller region considered in the PDE solver has a big influence on the computational effort required. Note also that the presented algorithm only works for $O \neq \emptyset$ and that its accuracy is strongly reduced if the overlap region is not significantly wider than the mean square displacement of a particle per time step, as discussed in the Remark to Theorem 3.2.

3.6.2 Example 5: Diffusion in two dimensions

In this example, we simulate the two-dimensional diffusion equation (3.29) on the domain $\Omega = [0,2] \times [0,1]$. We use $\Omega_P = [0,1] \times [0,1]$ and $\Omega_B = [0.9,2] \times [0,1]$ for the PDE and BD regions respectively. Hence the overlap region has the form $O = [0.9,1] \times [0,1]$. We simulate 100 particles started at $(x,y) = (0.5, 0.3) \in \Omega_P$, i.e. we use $c(0,[x,y]) = 100\,\delta(x-0.5)\,\delta(y-0.3)$ and $N(0) = 0$ as initial conditions. For the numerical solution of the diffusion equation in step (B1) we use a finite-element method with piecewise linear element functions on a regular triangular mesh with 201×201 gridpoints. This finite-element method was set up to automatically keep

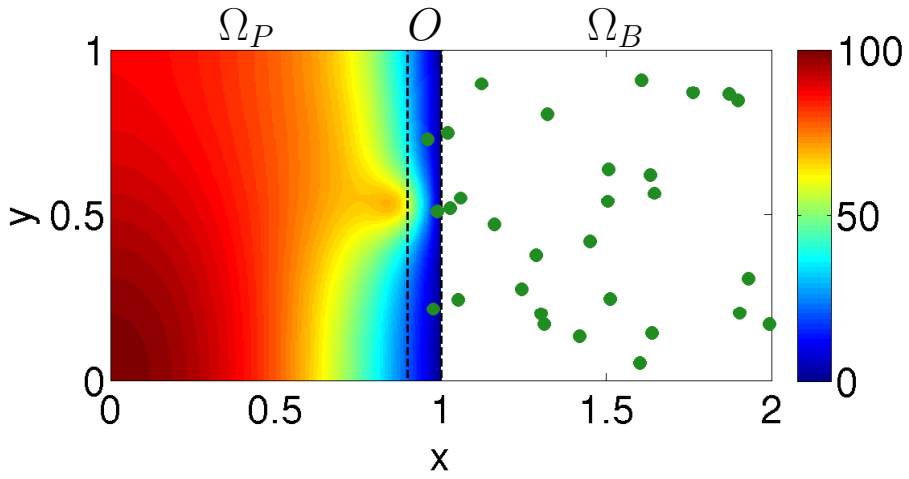
track of $\alpha\left(t+\Delta t, \mathbf{r}^{(l)}\right)$ on gridpoints that are situated on I_2. We then use the event-driven implementation described in Section 3.6.1 to implement the mass exchange.

We plot the results of one individual simulation in Figure 3.16(a). The (green) dots represent the positions of the particles existing at $t = 0.3$ and the colour gradient shows the concentration in the region Ω_P. With PBD algorithms being stochastic, a certain variation between realisations is to be expected and is clearly visible in the simulation results. In order to still get an idea of the accuracy of the algorithm, Figure 3.16(b) shows an ensemble average over 10^3 realisations and Figure 3.17 shows a slice through this distribution at $y = 0.4$. The outcome looks smooth and matches well with the expected result of a diffusion process. We can therefore conclude that the event-driven method works well and that the PBD algorithm (B1)–(B5) can in practice be applied to higher dimensional problems.

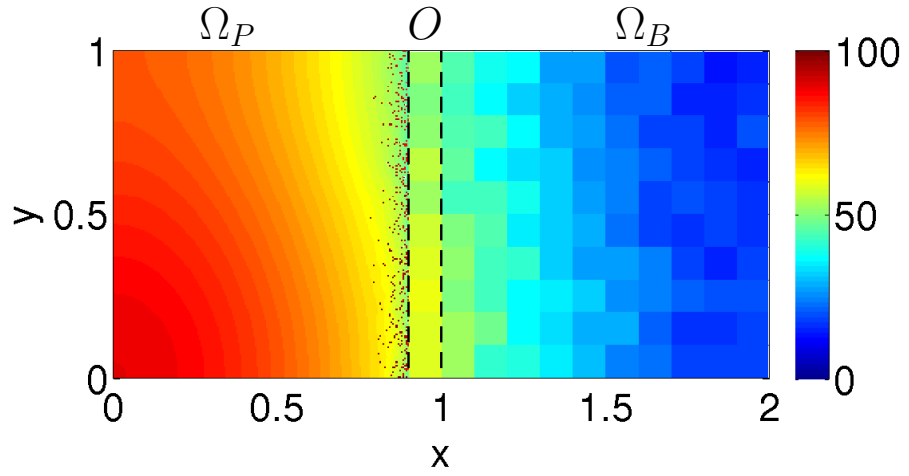
3.7 Discussion

In this chapter we have presented two PBD algorithms that combine Brownian dynamics with mean-field reaction-diffusion PDEs. These PBD methods produce exact Brownian dynamics simulations in one part of the domain and couple them with mean field approximations in another part of the domain. An algorithm of this type is useful for various application areas in computational biology and beyond, for example, when a detailed description of individual molecules is required near a receptor or ion channel, but becomes impractical in the bulk of a cell [36, 63]; or when a detailed stochastic simulation of actin dynamics is required inside filopodia, but becomes impractical in the bulk of a cell [175]. Another application area, chemisorption, was discussed in Section 3.5.3. By using this approach, it would also be possible to use finite-sized particles in the BD simulation and couple these with the corresponding mean field results presented in [22].

In the literature several hybrid models have been developed in the context of fluid dynamics, but they do not discuss issues that arise from the incorporation of chemical reactions. Alexander et al. [5] present a hybrid model that uses virtual particles at the grid point nearest to the interface to calculate fluxes across the boundary and to generate accurate density fluctuations inside the particle region. Reference [164] extends this approach by the introduction of an overlap region similar to O introduced in Section 3.4. An identical flux exchange with particles confined to a grid is presented in [64]. The principal disadvantage of these flux exchange schemes is that the resolution used for the numerical solution of the PDE regime needs to



(a)



(b)

Figure 3.16: *Example* 5. *Diffusion in two dimensions. Black (dashed) lines: interfaces between the regimes.*
(a) *Individual simulation result. (Green) dots: particles in* Ω_B*; colour gradient: concentration* $c(0.3, [x, y])$ *in* Ω_P.
(b) *Average over* 10^3 *realisations. Coloured boxes: particle counts in* $\Omega_B \cup O$*; colour gradient: concentration* $c(0.3, [x, y])$ *in* $\Omega_P \setminus O$*. Note that the red dots visible in* Ω_P *correspond to* δ *functions of particles that recently crossed the interface.*
Parameters as shown in the text.

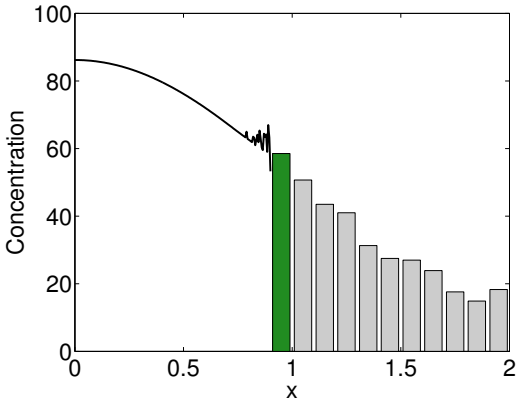


Figure 3.17: *Example* 5. *Diffusion in two dimensions. Slice through distribution seen in Figure 3.16(b) at* $y = 0.4$. *Solid (black) line:* $c(t, x)$ *in* $\Omega_P \setminus O$*; (grey) histograms: spatial concentration of particles in* $\Omega_B \setminus O$*; (green) bar: spatial concentration in* O.

be larger than the microscopic length-scale of the particles [164], whereas the PBD algorithms presented in this chapter work independently of the numerical scheme used in Ω_P. Additionally, the PBD algorithms guarantee exact conservation of mass and non-negativity of concentrations even for very small numbers of particles.

Chemical reactions in solution were considered in [48]. This model was discussed in Section 3.5.3. It couples the Brownian dynamics of molecules in solution with a more detailed description of the adsorbing boundary. In [48] a hybrid (RSA-PDE) model has been developed which replaces BD in the solution by solving the PDE (3.27) with a suitable stochastic boundary condition. The PBD algorithms are able to replace the stochastic boundary condition by a (small) BD region close to the surface. Although the hybrid RSA-PDE model introduced in [48] was sufficient in the case of (irreversible) adsorption, the situation is becoming more challenging whenever the binding reaction is reversible [109]. In this case, a molecule which is released by the surface will initially stay close to the surface and can rebind to the same receptor (binding site). This geminate recombination problem can be captured by the PBD approach. Reversible reactions are common in biological applications [63, 109].

Hybrid approaches for reaction-diffusion processes which couple different modelling approaches have also been introduced in the literature [42, 58, 62, 117]. A mesoscopic lattice-based description coupled with the macroscopic Fisher-Kolmogorov-Petrovsky-Piscounov PDE was used in [117] to study front propagation in a lattice-based reaction-diffusion model. A hybrid model for reaction-diffusion systems in porous media that combines pore-scale models with Darcy-scale models is presented in [156]. Flegg et al. [62, 142] introduced the so-called Two Regime Method which

couples a lattice-based (compartment-based) reaction-diffusion model with BD simulations. One advantage of the PBD algorithms over the Two Regime Method is that there are more efficient tools for PDE simulations than for compartment-based reaction-diffusion models. On the other hand, compartment-based models provide more details (including fluctuations) and hybrid models which couple BD simulations with compartment-based models do not require the overlap region [58, 62]. Since it is possible to couple the macroscopic PDE description with (mesoscopic) compartment-based models [42] and compartment-based models with (microscopic) BD simulations [62], then an alternative approach to PBD algorithms would be to use compartment-based models in the overlap region. That is, the computational domain would be divided into three regions where the PDE, compartment-based and BD descriptions would be used. These three regimes would be coupled using the results from the literature [42, 62]. Compartment-based models and macroscopic PDEs can also be coupled through another intermediate regime using a tau-leap method [58]. In this chapter, we showed that PDE models and BD simulations can be coupled without using intermediate compartment-based models.

We have seen in Section 3.6 that the PBD algorithm (B1)–(B5) extends into higher dimensions, as far as pure diffusion is concerned. However, the introduction of higher order reactions is not necessarily straightforward. Erban and Chapman [49] give an overview of how to implement bimolecular reactions in BD simulations, whilst the consideration of higher order reactions is inherent in reaction diffusion PDEs of the form (3.1). However, inside the overlap region $O = \Omega_B \cap \Omega_P$, one needs to consider reactions between particles and PDEs, a situation that is not well explored. For certain types of second order reactions, possible ways of dealing with reactions between particles and PDEs were presented in Chapter 2 as the 'reactions' between bacteria (particles) and extracellular signals (PDEs). Therefore, a combination of the two algorithms presented in Chapters 2 and 3 can display a significant advance in the direction of extremely flexible hybrid models in three dimensions for reaction diffusion systems. Additionally, the simulations shown in Section 3.5.4, where the two approaches were combined into one simulation of bacterial chemotaxis, show that the scope of PBD algorithms can be beyond pure diffusion and that they can account for more complex forms of motion. As such, one might be able to apply this algorithm to the ion channel application mentioned in the motivation of this chapter. In this case, electrical charges result in forcing terms that complicate the motion of ions away from pure diffusion. On the other hand, one might have to deal with particles applying

forces onto the PDE part of the PBD system, something that is not discussed in this thesis.

In the following chapter, we discuss swarm robotics experiments for target finding problems. One could apply PBD methods to this type of problem by interpreting the target area as a reaction zone and only considering an individual resolution in the area close to the target. Chapter 5 considers hard-sphere interactions in velocity jump processes. PBD methods can be a useful tool for simulations of systems with volume exclusions or other types of direct interactions: One could use individual-based descriptions in regions with high particle densities, where PDE models cannot guarantee a high accuracy. In areas with low densities, however, PDE models can provide good approximations of the real density and we do not need to simulate individual behaviour.

Chapter 4

Swarm robotic experiments: Introducing time delays into velocity jump processes

So far we have discussed hybrid methods that combine PDE descriptions with agent-based models in reaction-diffusion systems in general and for the application of bacterial chemotaxis in particular. In this chapter, we introduce a third framework for the study of biological systems of dispersal in the form of robot swarms. We perform experiments for a target finding task and use the outcomes of those experiments to motivate ways to improve the classical PDE models for velocity jump processes.

Most of the results presented in this chapter were submitted in the article [157] to *IMA Journal of Applied Mathematics*.

4.1 Motivation and problem formulation

Swarm robotic systems are gaining in importance as an experimental tool to investigate biological systems of dispersal [18]. The major advantage of the use of swarm robotic systems over purely simulation-based investigations is that physically realistic noise is inherently present in those experiments in the form of inaccuracies in the information perceived by the individual. One can roughly split the literature related to robot swarms in two categories [18]: mimicking of biological behaviour in the quest to improve understanding of biological systems and engineering a desired behaviour for human application areas. Intersections between these two areas are not uncommon, e.g. when biological behaviour is mimicked in applications [32].

One of the most famous examples of the first category is a group of cockroach-like robots [72] that was designed to mimic and eventually influence the aggregation

behaviour of cockroaches. A second important example was the introduction of a 'Robofish' into groups of *three-spined sticklebacks* [57]. The Robofish was able to act as leader in small groups of fish and significantly influence the group behaviour. In a similar flavour Garnier et al. [71] developed a group of robots along with a lighting system that is capable of mimicking the pheromone deposition and trail creation observed in ants. In [32] an algorithm mimicking the synchronisation of fireflies was used to detect faults in a robot swarm. Hamann et al. [80] study symmetry breaking in the nest choice of honeybees and Turgut et al. [160] apply a biologically motivated flocking behaviour to their robot swarm.

On the other hand, robot swarms are often directly designed for certain application areas, important examples being containment of oil spills [92] and mine detection [100]. This area also includes the study of group coordination [150], group decision making [116] and task-partitioning [111, 137].

In this chapter, we are aiming to study a target finding problem related to foraging in ants [111] and motivated by the process of bacterial chemotaxis presented in Chapter 2. In [40] a swarm robotic study related to the chemotaxis process is shown, but we are more interested in developing mathematical tools for the study of swarm robotic systems using a combination of agent-based models and PDE descriptions. Due to the nature of the robotic system studied, experiments will be performed in a two-dimensional environment, but analysis will be kept general for $d = 2, 3$ where possible.

A schematic of the general target finding problem studied in this chapter can be seen in Figure 4.1. Similar to before, we consider a finite domain $\Omega \subset \mathbb{R}^d$, with N identical particles of diameter ε signified through the positions of their centres $\mathbf{x}^{(i)} \in \Omega$ and their current velocities $\mathbf{v}^{(i)} \in V$, $i = 1, \dots, N$; V denotes the set of possible velocities that is in robotics applications defined through physical limitations of the robots. We sometimes call Ω the *effective domain* because it regulates the positions of the particle centres, meaning that parts of the particles can be situated outside Ω at any given time and it therefore does not exactly correspond to the physical domain used in experiments. Additionally, a *target area* is defined $\mathcal{T} \subset \mathbb{R}^d$ in such a way that $\Omega \cap \mathcal{T} = \emptyset$, but $\overline{\Omega} \cap \overline{\mathcal{T}} \neq \emptyset$. This naturally defines a split of the boundary $\partial\Omega$ into two parts through

$$\partial\Omega_{\mathcal{T}} = \overline{\Omega} \cap \overline{\mathcal{T}} = \partial\Omega \cap \partial\mathcal{T}\,, \qquad \partial\Omega_{\mathcal{R}} = \partial\Omega \backslash \partial\Omega_{\mathcal{T}}\,,$$

where we call $\partial\Omega_{\mathcal{T}}$ the *target boundary* and $\partial\Omega_{\mathcal{R}}$ the *reflective boundary*. Robots that hit the target boundary (with their centres) have achieved the target finding task and

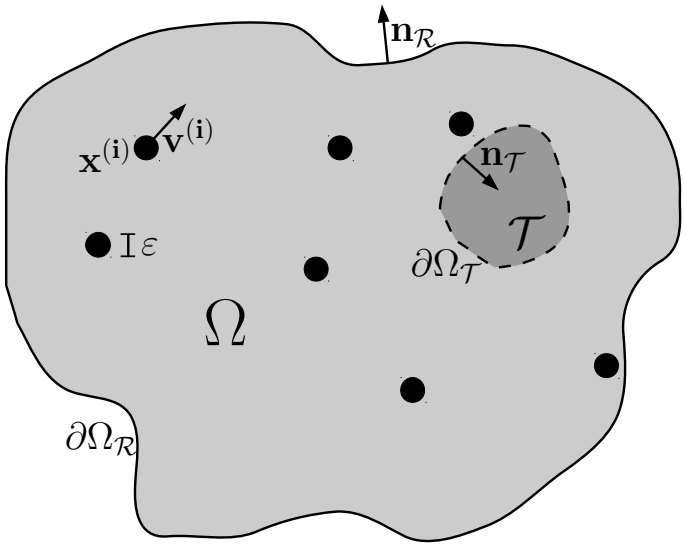


Figure 4.1: *Schematics of arena Ω along with notation used throughout this chapter: target area $\mathcal{T}$, target boundary $\partial\Omega_{\mathcal{T}}$, reflective boundary $\partial\Omega_{\mathcal{R}}$ and the respective outwards pointing normals $\mathbf{n}_{\mathcal{T}}$ and $\mathbf{n}_{\mathcal{R}}$.*

are removed from the system. The reflective boundary is implemented through walls of the arena, and robots that hit this boundary get reflected back into the arena. Additionally, $\mathbf{n} : \partial\Omega \mapsto \mathbb{S}^{d-1}$ denotes the outward pointing normal at the boundary of Ω, where the sphere $\mathbb{S}^{d-1}$ is defined through

$$\mathbb{S}^{d-1} = \{\mathbf{n} \in \mathbb{R}^d \ : \ \|\mathbf{n}\| = 1\}\,. \tag{4.1}$$

To clarify notation we will also use $\mathbf{n}_{\mathcal{T}} : \partial\Omega_{\mathcal{T}} \mapsto \mathbb{S}^{d-1}$ and $\mathbf{n}_{\mathcal{R}} : \partial\Omega_{\mathcal{R}} \mapsto \mathbb{S}^{d-1}$ for the outward pointing normal on the target and reflective parts of the boundary respectively. The interaction of a robot with the reflective boundary is defined through its reflected velocity $\mathbf{v}' \in V$ that takes the form

$$\mathbf{v}' = \mathbf{v} - 2(\mathbf{v} \cdot \mathbf{n}_{\mathcal{R}})\mathbf{n}_{\mathcal{R}}\,. \tag{4.2}$$

Note, that $(\mathbf{v}')' = \mathbf{v}$ and $\mathbf{v}' \cdot \mathbf{n}_{\mathcal{R}} = -\mathbf{v} \cdot \mathbf{n}_{\mathcal{R}}$ and that $(\cdot)'$ therefore defines a bijection between the two sets

$$V^+(\mathbf{n}) = \{\mathbf{v} \in V : \mathbf{v} \cdot \mathbf{n} > 0\}\,, \qquad \text{and} \qquad V^-(\mathbf{n}) = \{\mathbf{v} \in V : \mathbf{v} \cdot \mathbf{n} < 0\}\,. \tag{4.3}$$

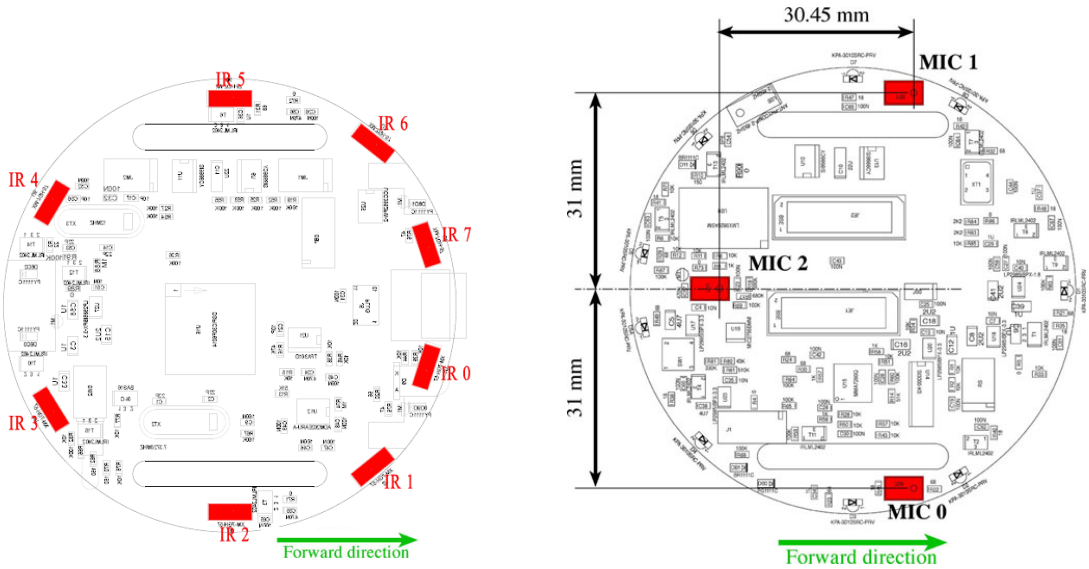
As has been the case throughout this thesis, we are looking for a combination of individual-based and continuum approaches to formulate and solve various problems related to the swarm experiments. Section 4.2 introduces the experimental system

that will be used for all experiments and as motivation for the mathematical analysis in the remainder of this chapter. We start by comparing experiments of random motion of a group of robots with different models of velocity jump processes in Section 4.3 and show how classical models can be extended to account for some of the intricacies present in the real-world system. In Section 4.4 we introduce an external signal in the form of a colour gradient into the experiments, hoping that this external clue can improve the target finding capabilities of the robots. Some of the discrepancies between experimental data and velocity jump models can be explained through interactions between robots that we discuss in great detail in Chapter 5.

4.2 Experimental set up and background

We used an experimental system based on 16 *E-Puck* robots [115] that will be described in detail in Section 4.2.1. Details of the programs we ran on those robots are given in Section 4.2.2 and in Section 4.2.3 we discuss the experiments we performed along with the shape and size of the arena.

4.2.1 *E-Puck* robots specifications



(a) (b)

Figure 4.2: E-Puck *sketches indicating the positioning of* (a) *the eight proximity sensors and* (b) *the three microphones.*
Source: www.e-puck.org (reprinted with permission).

i. Diameter: 75 mm,
ii. Height: 50 mm,
iii. Weight: 200 g,
iv. Speed throughout experiments: $0.058\,\mathrm{m\,sec^{-1}}$, (max speed: $0.13\,\mathrm{m\,sec^{-1}}$),
v. Angular turning speed throughout experiments: $4.65\,\mathrm{rad\,sec^{-1}}$,
vi. Autonomy: 2 hours moving,
vii. Processor: dsPIC 30 CPU @ 30 MHz (15 MIPS),
viii. RAM: 8 KB,
ix. Memory: 144 KB Flash,

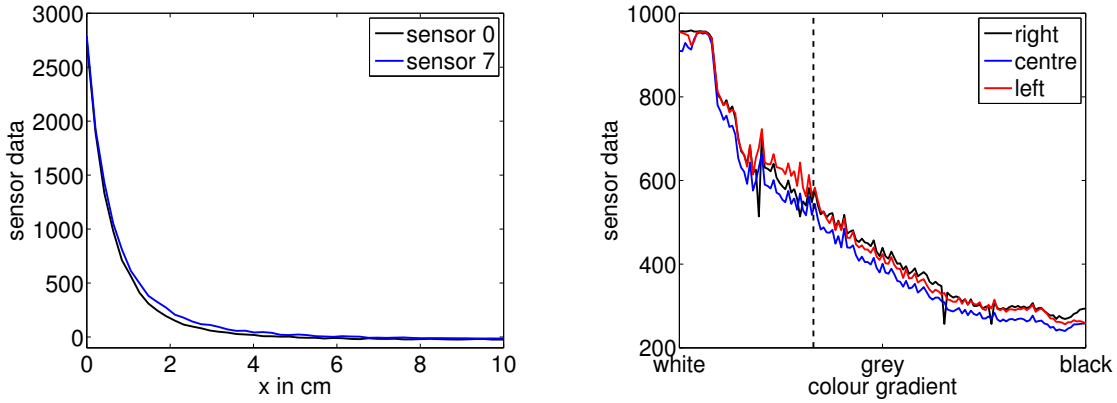
Other features:

x. 2 step motors,
xi. 8 infrared proximity sensors,
xii. 3 light sensors on the bottom,
xiii. Colour camera (max resolution 640×480),
xiv. 8 LEDs on outer ring, one body LED and one front LED,
xv. 3D accelerometers,
xvi. 3 microphones,
xvii. 1 speaker.

Table 4.1: *Full specifications for* E-Puck *robots*

The *E-Puck* robotic system was first introduced in [115] and is now widely used throughout the swarm robotic research area [150]. The robots are differential wheeled robots, meaning that the wheels can be programmed to move at velocities independently of each other. In particular, this implies that *E-Puck* robots can turn on the spot into any desired direction. A fact sheet of the *E-Puck* and its sensors can be seen in Table 4.1. The robot's microchip is a dsPic 30 CPU and can be programmed using C and a corresponding cross compiling platform, before the firmware is uploaded via Bluetooth.

Let us now draw attention to some of the features that will be used throughout the experiments in the remainder of this chapter. All the plots in Figures 4.3–4.5 were measured with the robot with unique manufacturer number 3138.



(a) (b)

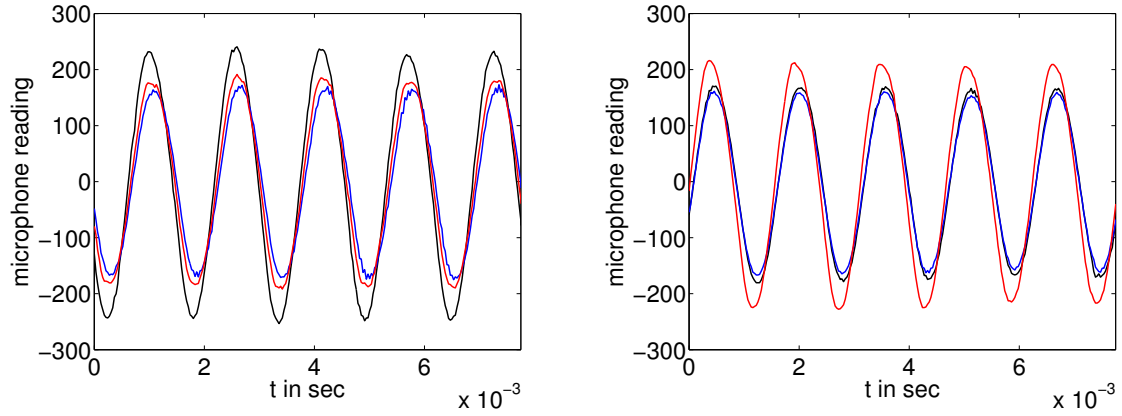
Figure 4.3: *Data from proximity and light sensors for* E-Puck *robot* 3138.
(a) *Proximity sensor as* E-Puck *approaches a wall. x denotes the distance to the wall.*
(b) *Colour sensors as* E-Puck *advances along a white-black gradient. The dashed (black) line indicates the level of grey used in experiments as shown in Figure* 4.6(b).

Proximity sensors

The *E-Puck* is equipped with eight proximity sensors that are placed as shown in Figure 4.2(a). These sensors constantly send out infra red signals and can detect obstacles through an analysis of the reflected signals. An example of the data delivered by the proximity sensors can be seen in Figure 4.3(a), where we show the characteristic curves of the two front sensors as the robot approaches a wall. We can see that without obstacles a neutral reading of around 0 is returned and that the signal starts increasing exponentially when the robot is about 4 cm away from the wall, returning readings of more than 1000 when the robot is closer than a centimetre to the obstacle. The main use of these proximity sensors in the experiments presented in this chapter is collision avoidance. This includes collisions with walls surrounding the arena as well as collisions with other robots.

Floor sensors

The three floor sensors are placed in the front of the *E-Puck* and detect the brightness of the floor. In Figure 4.3(b) we plot characteristic curves for each of the floor sensors of an *E-Puck* robot that were acquired by slowly advancing the robot along a white-black colour gradient. We can see that the measured value for white is around 900 and that the sensor readings decrease to somewhere about 300 as the robot enters the black region. Two effects are clearly visible in this characteristic curve that can be



(a) (b)

Figure 4.4: *Microphone recordings for sinusoidal signal coming from different directions for* E-Puck *robot* 3138. *Black line: right microphone (MIC0), blue line: left microphone (MIC1), red line: back microphone (MIC2).*
(a) *Sound emitted from the right of the robot.*
(b) *Sound emitted from the back of the robot.*

seen as motivations to using swarm robots in the first place: (a) the sensor readings are noisy and one can even sometimes observe readings that are invalid altogether and (b) the reading from each of the sensors differs slightly, but the difference between different robots can be rather drastic. This effect should be seen as motivation to not assume that all individuals have equal abilities in picking up a given signal. We will use the robots' ability to pick up colour gradients in the experimental results presented in Section 4.4.

Microphones

The robots possess three microphones that are located in a triangle as can be seen in Figure 4.2(b). These microphones can record incoming sound signals with a sampling frequency of up to 32 kHz and the internal memory allows storage of up to 3×256 data samples. The measured data is represented by positive integers that are typically in the region $[1000, 1600]$. The advantage of having three microphones is that one can use triangulation in order to approximate the direction the sound came from. An illustration of this idea can be seen in Figure 4.4, where we plot the recording of a 640 Hz sinusoidal signal coming from two different directions. Note, that the plotted signals are shifted to a mean of zero for better illustration. In Figure 4.4(a) we see the recording for a signal coming from the right of the robot. It is clearly visible that the recording of the right microphone (MIC0) has a higher amplitude and is slightly

advanced compared to the other signals, with the microphone in the back (MIC2) picking up the signal second. Similarly, in Figure 4.4(b) one can see that the signal emitted from the back of the robot is first detected by the microphone in the back (MIC2) and then by the other two.

Speaker

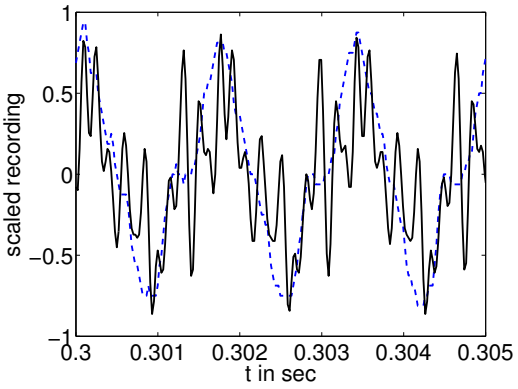


Figure 4.5: *Signal emitted by the* E-Puck*'s speaker for different volumes. Note that the plotted data are scaled versions of the actual recordings such that they fit in* $[-1, 1]$. *Solid (black) line: high volume, dashed (blue) line: low volume.*

Each *E-Puck* is equipped with a speaker placed on top of the robot close to its centre. The speaker can be used to emit audio signals and works with a sampling frequency of 3.6 kHz. In Figure 4.5 we show a 600 Hz beep sent from the *E-Puck* speaker at two different volumes that we have rescaled for better illustration. For the low-volume case plotted as the dashed (blue) line, the signal does not show a high content of noise and a regular 600 Hz sinusoidal signal can be seen. However, as the volume of the signal is increased, one can see that noise effects increase strongly. In particular, we repeatedly observed a strong noise at 5 kHz dominating the signal, which might stem from a resonance effect of the robot body. This very high noise level makes the communication from robot to robot using sound signals difficult at high volumes, because the direction approximation works best with a clear signal. On the other hand low volumes repeatedly lead to robots missing incoming sound signals altogether as the signal is barely significant compared to normal microphone noise.

4.2.2 Programming of the *E-Puck* robots

As mentioned before, the *E-Puck* robots are programmed in C and the compiled firmware can be uploaded using Bluetooth. In this section we will go into more detail

(C1) Initialise the motors and all the sensors. Choose an initial direction $\theta \in [-\pi, \pi]$ uniformly at random. If θ is positive, turn clockwise, otherwise anti-clockwise by $|\theta|$ and start running with speed s into the chosen direction.

(C2) If `timerCnt` has changed call `RandomWalk`.

(C3) Call `ObstacleAvoidance` to check for obstacles.

(C4) Go to (C2).

Table 4.2: *Main loop of* E-Puck *program.*

about the actual programs that we use throughout the experiments. We will make use of two sets of sensors described in Section 4.2.1 – the proximity sensors and the floor sensors. The proximity sensors play an important role in that they can always overrule the current behaviour in order to avoid collisions with obstacles and thereby avoid physical damage. As is common in microcontroller programming, the main program runs in an infinite loop that can be interrupted by timer interrupts that fire in regular intervals of $\Delta t = 0.1\,\text{sec}$. The main program can be seen in Table 4.2. After an initialisation process, the robot is set running into a random direction $\theta \in [-\pi, \pi]$ with speed s. We maintain the speed s constant throughout the whole program. After that the infinite loop is started, where we use the variable `timerCnt` to signify the number of timer interrupts that have happened throughout the lifetime of the program. The program reacts to changes in this variable in order to go into the `RandomWalk` function in regular intervals. We call this part the *synchronous* part of the program, whilst the reaction to proximity and microphone signals is *asynchronous* as it does not wait for a timer interrupt.

Random walk

The random walk part of the program is shown in Table 4.3. This function is called upon in regular intervals of Δt through the timer interrupts. The idea for this implementation of a biased random walk stems from Model III in Section 2.4.1 and is therefore motivated by the biological process of bacterial chemotaxis. In (D1) the data from the floor sensors is read into the signal variable S and rescaled onto $[0, 1]$. The internal variable z_2 is then updated using a forward Euler implementation for

(D1) Read data from the three floor sensors, calculate the average and divide by 1000 in order to generate the signal variable $S \in [0, 1]$. Set $S = 1 - S$.

(D2) Update the internal variable z_2 using

$$z_2 = z_2 + (S - z_2)\frac{\Delta t}{t_a}\,.$$

(D3) Generate a uniformly distributed random variable $r \in [0, 1]$.

(D4) If $r < \Delta t\, \lambda_0\, (1 - \alpha(S - z_2))$, generate a new random direction $\theta_* \in [-\pi, \pi]$ from the uniform distribution. If $\theta_* > 0$, turn clockwise, otherwise anti-clockwise by $|\theta_*|$ and start running with speed s into the chosen direction.

Table 4.3: `RandomWalk` *function of* E-Puck *program.*

the ODE (cf. (2.28))

$$\frac{\mathrm{d}z_2}{\mathrm{d}t} = \frac{S - z_2}{t_a}\,. \tag{4.4}$$

This internal variable is then used to calculate the biased turning frequency (cf. (2.30))

$$\lambda = \lambda_0(1 - \alpha(S - z_2))\,, \tag{4.5}$$

where λ_0, $\alpha \in \mathbb{R}^+$ are positive constants. The robot now turns with the probability $\Delta t\, \lambda$ into a new direction $\theta_* \in [-\pi, \pi]$ that is chosen uniformly at random. The robot is then set running again with the constant speed s. The chosen parameter values will be explained in Section 4.2.3.

Obstacle avoidance

The obstacle avoidance is performed using the data collected from the proximity sensors as seen in Figure 4.2(a) and the algorithm is shown in Table 4.4. Note, that in the current form of the algorithm, robots cannot distinguish between different types of obstacles, e.g. other robots or walls. In the future this could potentially be achieved using the integrated camera. The code shown in Figure 4.4 shows a primitive form of reflective behaviour, where depending on which sensor crosses a threshold of 600 first, the reflection occurs. The threshold of 600 corresponds to roughly a distance of 1 cm as can be seen in Figure 4.2(a). The angles were chosen in a way that the observed behaviour is close to a proper reflection, but could be improved by using

(E1) Measure the data for all eight proximity sensors into `IR[i]`, $i = 0, \ldots, 7$, according to the numbering given in Figure 4.2(a).

(E2) If `IR[0]` > 600, turn anti-clockwise by $0.9\,\pi$ and start a run with speed s into the new direction.

(E3) If `IR[7]` > 600, turn clockwise by $0.9\,\pi$ and start a run with speed s into the new direction.

(E4) If `IR[1]` > 600, turn anti-clockwise by $0.6\,\pi$ and start a run with speed s into the new direction.

(E5) If `IR[6]` > 600, turn clockwise by $0.6\,\pi$ and start a run with speed s into the new direction.

Table 4.4: `ObstacleAvoidance` *function of* E-Puck *program.*

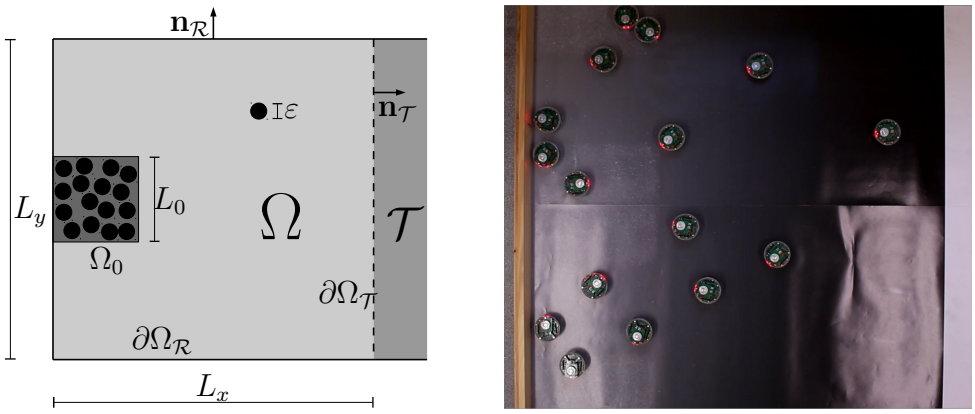
more than the reading of one of the sensors at a time. For all mathematical analysis, we will assume a perfect reflection into the new direction $\mathbf{v}'$ as defined in (4.2).

4.2.3 Experimental test problems and parameter values

As mentioned in the introduction to this chapter, we are looking at target finding problems and have defined a test scenario sketched out in Figure 4.6(a) for the experiments. In each experimental run 16 robots are initially placed in a removable pen Ω_0 at one end of the arena. The robots can move inside this pen before the release and it thereby contributes to the randomisation of their positions and velocities. There are also practical reasons that required such an initialisation period, as it is an uncomplicated way of releasing all robots at the same time without having to provide a start signal that could potentially not be picked up or picked up too early.

After releasing the robots from the pen Ω_0, they can move freely inside the whole arena Ω. When a robot crosses over into the target area $\mathcal{T}$, the leaving time is taken and the robot is removed instantly in order to avoid influencing the remaining robots. Because of this instant removal, the boundary $\partial\Omega_{\mathcal{T}}$ can be seen as an adsorbing boundary. We continue measuring until either all robots have left the arena Ω or until 300 sec have passed at which time we note the number of robots still remaining in the arena.

The test-problem as shown in Figure 4.6(a) comprises a rectangular arena that has



(a) (b)

Figure 4.6: (a)*Schematics for experimental test problem.* 16 *robots are initialised inside* Ω_0 *and then released into the arena. The parameters used in experiments are:* $L_x = 1.1825\,m$, $L_y = 1.145\,m$, $\varepsilon = 0.075\,m$, $L_0 = 0.305\,m$.
(b) *Photo from an experimental run with colour gradient as external clue. The gradient runs from the left (grey) to the right (black) part of the domain. Note that the light areas in the bottom half of the photo are due to reflections and cannot be observed by the robots.*

one open edge leading into the target area. The advantage of this test-problem is that robots will find the target relatively quickly even without a signal and that therefore a high number of repetitions can be performed in a reasonable time. Figure 4.6(b) shows a photo taken during one of the experiments including an external signal. We use a poster containing a grey-black gradient with the black edge leading into the target area seen as the white strip in the right of the photo. We did not use a full white-black gradient, because most of the robots cannot detect a difference between white and light grey. The grey used has about 33% intensity and is indicated by the dashed line in Figure 4.3(b). The results of the experiments containing the external signal are presented in Section 4.4.

Unless otherwise stated, the parameter values we use in all experiments are: turning frequency $\lambda = 0.25\,\mathrm{sec}^{-1}$; run speed $s = 0.58\,\mathrm{m\,sec}^{-1}$; angular turning speed $\omega = 4.65\,\mathrm{rad\,sec}^{-1}$; robot diameter $\varepsilon = 75\,\mathrm{mm}$. The value for λ was programmed into the robots and its accuracy depends on the accuracy of the robot's internal clock. The values for s and ω were averaged over a number of measurements and the variation between different robots and different repetitions was too small to measure by hand. Note, that for ω we use the maximum value possible for the *E-Puck* robots, whilst for s we only use about 50% of the physical maximum in order to avoid damage in case of collisions with obstacles. In practice, one would consider those parameters as

given due to the physical restrictions of the robots, whilst the programmer can decide on the turning frequency λ. The decision on an optimal λ is not trivial and depends strongly on the considered scenario.

4.3 Results without signals: introduction of finite turning times

In this section we concentrate on experimental, numerical and analytical results for systems without external signal, i.e. we consider unbiased velocity jump processes. We will start by reviewing the classical transport equation for this simple case. We also formulate a mean first passage time problem in order to analyse experimental data we gathered. We will then see that the classical models do not adequately describe this data and Section 4.3.2 extends the classical model by incorporating the non-negligible turning times seen in the robotic system.

4.3.1 Transport equation and mean first passage times

As was mentioned above, the random walk implemented in Table 4.3 is the discrete version of a velocity jump process in $d = 2$ space dimensions. The general velocity jump process was introduced in Section 1.3.2. We showed that one can formulate a transport equation for the mesoscopic particle density $p(t, \mathbf{x}, \mathbf{v})$ that formally describes the number of particles at position $\mathbf{x}$ with velocity $\mathbf{v}$. The transport equation takes the form

$$\frac{\partial p}{\partial t} + \mathbf{v} \cdot \nabla_{\mathbf{x}} p = -\lambda p + \lambda \int_V T(\mathbf{v}, \mathbf{v}_*)\, p(t, \mathbf{x}, \mathbf{v}_*)\, \mathrm{d}\mathbf{v}_* \,, \tag{4.6}$$

where λ is the turning frequency and $T(\mathbf{v}, \mathbf{v}_*)$ is the turning kernel. For the no-signal case and the implementation in Table 4.3, we have $\lambda = \lambda_0$ and the uniform turning kernel

$$T(\mathbf{v}, \mathbf{v}_*) = \frac{\delta\left(\|\mathbf{v}\| - s\right)}{2\pi s} \,.$$

In order to use the transport equation (4.6) for the target finding problem, we need to apply the following boundary conditions

$$\begin{aligned} &p(t, \mathbf{x}, \mathbf{v}) = 0 \,, && \mathbf{x} \in \partial\Omega_{\mathcal{T}},\ \mathbf{v} \cdot \mathbf{n}_{\mathcal{T}} < 0 \,, \\ &p(t, \mathbf{x}, \mathbf{v}) = p(t, \mathbf{x}, \mathbf{v}') \,, && \mathbf{x} \in \partial\Omega_{\mathcal{R}} \,, \end{aligned} \tag{4.7}$$

where $\mathbf{v}'$ is the reflected velocity given in (4.2). In Section 4.3.2 we show that one can derive a backwards problem for the mean first passage time $\tau(\mathbf{x}, \mathbf{v})$ as follows

$$-1 = \mathbf{v} \cdot \nabla_{\mathbf{x}} \tau - \lambda \tau + \lambda \int_V T(\mathbf{v}_*, \mathbf{v})\, \tau(\mathbf{x}, \mathbf{v}_*)\, \mathrm{d}\mathbf{v}_* \,. \tag{4.8}$$

Note that in (4.8) the arguments of the turning kernel T are switched in comparison to the transport equation (4.6). The boundary conditions take the form

$$\begin{aligned} \tau(\mathbf{x}, \mathbf{v}) &= 0\,, & \mathbf{x} &\in \partial\Omega_{\mathcal{T}},\ \mathbf{v}\cdot\mathbf{n}_{\mathcal{T}} > 0\,, \\ \tau(\mathbf{x}, \mathbf{v}) &= \tau(\mathbf{x}, \mathbf{v}')\,, & \mathbf{x} &\in \partial\Omega_{\mathcal{R}}\,. \end{aligned} \tag{4.9}$$

4.3.2 Finite turning times

The transport equation (4.6) is generally only valid under the assumptions of non-interacting particles and instant turning. For the robot application both of these assumptions are violated. In this section we concentrate on the second assumption and therefore aim to extend the classical transport equation (4.6) to include non-negligible turning times. The effect of hard sphere interactions in velocity jump models will be studied in great detail in Chapter 5. Note, that this effect is likely to lead to smaller discrepancies than finite turning times for the robotic system, due to the low number of robots in the group.

Let us begin by making two assumptions that are valid for the robotic system: **(a)** a new direction $\mathbf{v}_* \in V$ is chosen as soon as the particle enters the turning phase; **(b)** the time it takes for a particle to turn from velocity $\mathbf{v} \in V$ to $\mathbf{v}_* \in V$ is given through the function $K(\mathbf{v}_*, \mathbf{v}) : V \times V \mapsto \mathbb{R}^+$. Assumption **(b)** implies that the turning time is constant and equal for each particle and in particular does not depend on the particles' history.

For the robotic system studied here, we can additionally assume that the turning phase is equivalent to a directed rotation with a constant angular velocity $\omega \in \mathbb{R}^+$. Therefore, the turning time depends only on the angle between the current velocity $\mathbf{v} \in V$ and the new velocity $\mathbf{v}_* \in V$ and K takes the form

$$K(\mathbf{v}_*, \mathbf{v}) = K(\mathbf{v}, \mathbf{v}_*) = \frac{1}{\omega} \arccos\left(\frac{\mathbf{v}\cdot\mathbf{v}_*}{\|\mathbf{v}\|\,\|\mathbf{v}_*\|}\right). \tag{4.10}$$

We now extend the classical model (4.6) through the introduction of a *resting state* $r(t, \mathbf{x}, \mathbf{v}, \eta)$ that formally defines the number of particles currently turning towards their new chosen velocity $\mathbf{v}$ and with remaining turning time η. The density $p(t, \mathbf{x}, \mathbf{v})$ now only denotes the particles currently in the run phase. The update of the extended system is given through

$$\frac{\partial p}{\partial t} + \mathbf{v}\cdot\nabla_{\mathbf{x}} p = -\lambda p(t, \mathbf{x}, \mathbf{v}) + r(t, \mathbf{x}, \mathbf{v}, 0^+)\,, \tag{4.11}$$

$$\frac{\partial r}{\partial t} - \frac{\partial r}{\partial \eta} = \lambda \int_V p(t, \mathbf{x}, \mathbf{v}_*)\, T(\mathbf{v}, \mathbf{v}_*)\, \delta(\eta - K(\mathbf{v}, \mathbf{v}_*))\, \mathrm{d}\mathbf{v}_*\,. \tag{4.12}$$

In (4.11) we can see that running particles will enter a turning phase with a probability of $\lambda\,\mathrm{d}t$ and particles that have finished their turn signified through $\eta = 0$ re-enter the run-phase. Equation (4.12) represents the linear relation between t and η and shows that particles enter the turning phase depending on their newly chosen velocity direction. In order to guarantee conservation of mass throughout the system, we introduce the non-negativity condition for η through

$$r(t, \mathbf{x}, \mathbf{v}, \eta) = 0, \quad \eta < 0,\ \mathbf{x} \in \Omega,\ \mathbf{v} \in V,\ t > 0\,.$$

Additionally, the boundary conditions for the system (4.11)–(4.12) are given through

$$\begin{aligned}
p(t, \mathbf{x}, \mathbf{v}) &= 0\,, && \mathbf{x} \in \partial\Omega_{\mathcal{T}}\,, \mathbf{v} \cdot \mathbf{n}_{\mathcal{T}} < 0\,,\\
p(t, \mathbf{x}, \mathbf{v}) &= -r(t, \mathbf{x}, \mathbf{v}, 0^+)/(\mathbf{v} \cdot \mathbf{n}_{\mathcal{R}})\,, && \mathbf{x} \in \partial\Omega_{\mathcal{R}}\,, \mathbf{v} \cdot \mathbf{n}_{\mathcal{R}} < 0\,,\\
\frac{\partial r}{\partial t} - \frac{\partial r}{\partial \eta} &= \delta(\eta - K(\mathbf{v}, \mathbf{v}'))\,(\mathbf{v}' \cdot \mathbf{n}_{\mathcal{R}})\,p(t, \mathbf{x}, \mathbf{v}')\,, && \mathbf{x} \in \partial\Omega_{\mathcal{R}}\,, \mathbf{v} \cdot \mathbf{n}_{\mathcal{R}} < 0\,,\\
r(t, \mathbf{x}, \mathbf{v}, \eta) &= 0\,, && \mathbf{x} \in \partial\Omega_{\mathcal{R}}\,, \mathbf{v} \cdot \mathbf{n}_{\mathcal{R}} > 0\,,
\end{aligned} \tag{4.13}$$

where $\mathbf{v}'$ is the reflected velocity as given in (4.2). In order to show that the system (4.11)–(4.12) is actually consistent, the following theorem shows that mass in the system is conserved if no target is present.

Theorem 4.1. *The total mass in system* (4.11)–(4.12) *with the boundary conditions given in* (4.13) *in the case of reflective boundaries everywhere* ($\partial\Omega_{\mathcal{R}} = \partial\Omega\,, \partial\Omega_{\mathcal{T}} = \emptyset$) *given through*[1]

$$M(t) = \int_\Omega \int_V p(t, \mathbf{x}, \mathbf{v})\,\mathrm{d}\mathbf{v}\,\mathrm{d}\mathbf{x} + \int_{\overline{\Omega}} \int_V \int_0^\infty r(t, \mathbf{x}, \mathbf{v}, \eta)\,\mathrm{d}\eta\,\mathrm{d}\mathbf{v}\,\mathrm{d}\mathbf{x}\,,$$

is conserved.

Proof. In this proof we will use the definitions of the sets $V^+, V^- \subset V$ as defined in (4.3). Let us additionally define

$$R(t, \mathbf{x}, \mathbf{v}) = \int_0^\infty r(t, \mathbf{x}, \mathbf{v}, \eta)\,\mathrm{d}\eta\,.$$

Integrating (4.12) with respect to $\eta \in [0, \infty)$, we obtain after reordering for $\mathbf{x} \in \Omega \backslash \partial\Omega$:

$$\frac{\partial R}{\partial t} = -r(t, \mathbf{x}, \mathbf{v}, 0^+) + \lambda \int_V p(t, \mathbf{x}, \mathbf{v}_*) T(\mathbf{v}, \mathbf{v}_*)\,\mathrm{d}\mathbf{v}_*\,.$$

[1]Note that we use Ω for $p(t, \mathbf{x}, \mathbf{v})$, but $\overline{\Omega}$ for $r(t, \mathbf{x}, \mathbf{v}, \eta)$, because resting particles can be 'stuck' on the boundary, when they were forced to turn due to the boundary conditions. One could unify this by using $\overline{\Omega}$ for both quantities, because moving robots along the boundary have measure 0.

Hence, for every point $\mathbf{x} \in \Omega \setminus \partial\Omega$ we obtain

$$\frac{\partial}{\partial t}\left[p + R\right] = -\lambda p + \lambda \int_V p(t, \mathbf{x}, \mathbf{v}_*) T(\mathbf{v}, \mathbf{v}_*)\, \mathrm{d}\mathbf{v}_* - \mathbf{v} \cdot \nabla_{\mathbf{x}} p\,.$$

Integrating this with respect to $\mathbf{x} \in \Omega$ and $\mathbf{v} \in V$ gives

$$\int_\Omega \int_V \frac{\partial}{\partial t}\left[p(t, \mathbf{x}, \mathbf{v}) + R(t, \mathbf{x}, \mathbf{v})\right]\, \mathrm{d}\mathbf{v}\, \mathrm{d}\mathbf{x} = -\int_\Omega \int_V \mathbf{v} \cdot \nabla_{\mathbf{x}} p(t, \mathbf{x}, \mathbf{v})\, \mathrm{d}\mathbf{v}\, \mathrm{d}\mathbf{x}\,. \tag{4.14}$$

Using the divergence theorem, we can evaluate the integral on the right-hand side to

$$\begin{aligned}
\int_\Omega \int_V \mathbf{v} \cdot \nabla_{\mathbf{x}} p\, \mathrm{d}\mathbf{v}\, \mathrm{d}\mathbf{x} &= \int_{\partial\Omega} \int_V (\mathbf{v} \cdot \mathbf{n}_{\mathcal{R}})\, p(t, \mathbf{x}, \mathbf{v})\, \mathrm{d}\mathbf{v}\, \mathrm{d}\mathbf{x} \\
&= \int_{\partial\Omega} \int_{V^+(\mathbf{x})} (\mathbf{v} \cdot \mathbf{n}_{\mathcal{R}})\, p(t, \mathbf{x}, \mathbf{v})\, \mathrm{d}\mathbf{v}\, \mathrm{d}\mathbf{x} + \int_{\partial\Omega} \int_{V^-(\mathbf{x})} (\mathbf{v} \cdot \mathbf{n}_{\mathcal{R}})\, p(t, \mathbf{x}, \mathbf{v})\, \mathrm{d}\mathbf{v}\, \mathrm{d}\mathbf{x} \\
&= \int_{\partial\Omega} \int_{V^+(\mathbf{x})} (\mathbf{v} \cdot \mathbf{n}_{\mathcal{R}})\, p(t, \mathbf{x}, \mathbf{v})\, \mathrm{d}\mathbf{v}\, \mathrm{d}\mathbf{x} - \int_{\partial\Omega} \int_{V^-(\mathbf{x})} r(t, \mathbf{x}, \mathbf{v}, 0^+)\, \mathrm{d}\mathbf{v}\, \mathrm{d}\mathbf{x}\,,
\end{aligned}$$

where we use the second boundary condition in (4.13) in the last step. Additionally, for $\mathbf{x} \in \partial\Omega$ and $\mathbf{v} \in V^-(\mathbf{x})$, we obtain by integrating the third boundary condition in (4.13) with respect to $\eta \in [0, \infty)$

$$\frac{\partial R}{\partial t} = -r(t, \mathbf{x}, \mathbf{v}, 0^+) + (\mathbf{v}' \cdot \mathbf{n}_{\mathcal{R}})\, p(t, \mathbf{x}, \mathbf{v}')\,.$$

Integrating this with respect to $\mathbf{x} \in \partial\Omega$ and $\mathbf{v} \in V$ and using the last boundary condition in (4.13) we obtain

$$\begin{aligned}
\int_{\partial\Omega} \int_V \frac{\partial R}{\partial t}\, \mathrm{d}\mathbf{v}\, \mathrm{d}\mathbf{x} &= \int_{\partial\Omega} \int_{V^-(\mathbf{x})} \frac{\partial R}{\partial t}\, \mathrm{d}\mathbf{v}\, \mathrm{d}\mathbf{x} \\
&= -\int_{\partial\Omega} \int_{V^-(\mathbf{x})} r(t, \mathbf{x}, \mathbf{v}, 0^+)\, \mathrm{d}\mathbf{v}\, \mathrm{d}\mathbf{x} + \int_{\partial\Omega} \int_{V^-(\mathbf{x})} (\mathbf{v}' \cdot \mathbf{n}_{\mathcal{R}})\, p(t, \mathbf{x}, \mathbf{v}')\, \mathrm{d}\mathbf{v}\, \mathrm{d}\mathbf{x} \\
&= -\int_{\partial\Omega} \int_{V^-(\mathbf{x})} r(t, \mathbf{x}, \mathbf{v}, 0^+)\, \mathrm{d}\mathbf{v}\, \mathrm{d}\mathbf{x} + \int_{\partial\Omega} \int_{V^+(\mathbf{x})} (\mathbf{v} \cdot \mathbf{n}_{\mathcal{R}})\, p(t, \mathbf{x}, \mathbf{v})\, \mathrm{d}\mathbf{v}\, \mathrm{d}\mathbf{x}\,.
\end{aligned} \tag{4.15}$$

Summing up the results from (4.14) and (4.15), we obtain

$$\frac{\mathrm{d}M}{\mathrm{d}t} = 0\,,$$

and hence the total mass $M(t)$ in the system is conserved.

□

Transport equation with turning delays

We now aim to reformulate the system (4.11)–(4.12) into a revised transport equation similar to (4.6), by elimination of the resting state using the following theorem.

Theorem 4.2. *Assuming that* $r(0, \mathbf{x}, \mathbf{v}, \eta) = 0$ *for all* $\mathbf{x} \in \Omega$, $\mathbf{v} \in V$, $\eta \in [0, \infty)$, *the solution of* (4.12) *is given through*

$$r(t, \mathbf{x}, \mathbf{v}, \eta) = \lambda \int_{V'(t,\mathbf{v},\eta)} p(t - K(\mathbf{v}, \mathbf{v}_*) + \eta, \mathbf{x}, \mathbf{v}_*)\, T(\mathbf{v}, \mathbf{v}_*)\, \mathrm{d}\mathbf{v}_* \,, \tag{4.16}$$

where $V'(t, \mathbf{v}, \eta) \subset V$ *is given through*

$$V'(t, \mathbf{v}, \eta) = \{\mathbf{v}_* \in V \ : \ \eta \leq K(\mathbf{v}, \mathbf{v}_*) \leq \eta + t\} \,. \tag{4.17}$$

Proof. One can solve (4.12) using the method of characteristics. Let us therefore introduce the variable $z(t, \eta)$ and apply the standard ansatz

$$\frac{\mathrm{d}t}{\mathrm{d}z} = 1 \,, \qquad \frac{\mathrm{d}\eta}{\mathrm{d}z} = -1 \,,$$

which yields $t = z$ and $\eta = -z + \eta_0 = -t + \eta_0$. For $r(z, \mathbf{x}, \mathbf{v}) = r(t(z), \mathbf{x}, \mathbf{y}, \eta(z))$, the characteristic equation takes the form

$$\frac{\partial r}{\partial z} = \lambda \int_V p(z, \mathbf{x}, \mathbf{v}_*)\, T(\mathbf{v}, \mathbf{v}_*)\, \delta(\eta_0 - z - K(\mathbf{v}, \mathbf{v}_*))\, \mathrm{d}\mathbf{v}_* \,.$$

Integrating over the interval $[0, z]$ and using the fact that $r(t = 0) \equiv 0$, we obtain

$$r(z, \mathbf{x}, \mathbf{v}) = \lambda \int_0^z \int_V p(t, \mathbf{x}, \mathbf{v}_*)\, T(\mathbf{v}, \mathbf{v}_*)\, \delta(\eta_0 - t - K(\mathbf{v}, \mathbf{v}_*))\, \mathrm{d}\mathbf{v}_* \, \mathrm{d}t \,.$$

Switching integrals and evaluating the δ function for the cases where $\eta_0 - K(\mathbf{v}, \mathbf{v}_*) \in [0, z]$, we obtain

$$r(z, \mathbf{x}, \mathbf{v}) = \lambda \int_{V''(z,\mathbf{v},\eta_0)} p(\eta_0 - K(\mathbf{v}, \mathbf{v}_*), \mathbf{x}, \mathbf{v}_*)\, T(\mathbf{v}, \mathbf{v}_*)\, \mathrm{d}\mathbf{v}_* \,,$$

where the set $V''(z, \mathbf{v}, \eta_0) \subset V$ is defined through

$$V''(z, \mathbf{v}, \eta_0) = \{\mathbf{v}_* \in V \ : \ \eta_0 - K(\mathbf{v}, \mathbf{v}_*) \in [0, z]\} \,.$$

We can now substitute $\eta_0 = \eta + t$ and evaluate $r(t, \mathbf{x}, \mathbf{v}, \eta)$:

$$r(t, \mathbf{x}, \mathbf{v}, \eta) = \lambda \int_{V''(t,\mathbf{v},\eta+t)} p(t - K(\mathbf{v}, \mathbf{v}_*) + \eta, \mathbf{x}, \mathbf{v}_*)\, T(\mathbf{v}, \mathbf{v}_*)\, \mathrm{d}\mathbf{v}_* \,.$$

Additionally, the set $V''(t, \mathbf{v}, \eta + t)$ can be evaluated to

$$V''(t, \mathbf{v}, \eta + t) = \{\mathbf{v}_* \in V \ : \ t + \eta - K(\mathbf{v}, \mathbf{v}_*) \in [0, t]\} = V'(t, \mathbf{v}, \eta) \,,$$

and we have shown that the solution to (4.12) is given through (4.16). □

Using the above theorem, we can easily evaluate $r(t, \mathbf{x}, \mathbf{v}, 0^+)$ to be

$$r(t, \mathbf{x}, \mathbf{v}, 0^+) = \lambda \int_{V'(t,\mathbf{v},0)} p(t - K(\mathbf{v}, \mathbf{v}_*), \mathbf{x}, \mathbf{v}_*)\, T(\mathbf{v}, \mathbf{v}_*)\, \mathrm{d}\mathbf{v}_* \,.$$

Note that by evaluating $V'(t, \mathbf{v}, 0)$, we can alternatively write this as

$$r(t, \mathbf{x}, \mathbf{v}, 0^+) = \lambda \int_{V} p(t - K(\mathbf{v}, \mathbf{v}_*), \mathbf{x}, \mathbf{v}_*)\, T(\mathbf{v}, \mathbf{v}_*) H\left(t - K(\mathbf{v}, \mathbf{v}_*)\right)\, \mathrm{d}\mathbf{v}_* \,,$$

where $H : \mathbb{R} \mapsto \{0, 1\}$ denotes the Heaviside function. We can now substitute this expression into (4.11) to obtain

$$\frac{\partial p}{\partial t} + \mathbf{v} \cdot \nabla_{\mathbf{x}} p = -\lambda p + \lambda \int_{V'(t,\mathbf{v},0)} T(\mathbf{v}, \mathbf{v}_*)\, p(t - K(\mathbf{v}, \mathbf{v}_*), \mathbf{x}, \mathbf{v}_*)\, \mathrm{d}\mathbf{v}_* \,. \tag{4.18}$$

Note that (4.18) only considers particles in the run phase and hence does not strictly conserve mass. We can use a similar technique as before to derive the boundary conditions for the new transport equation

$$\begin{aligned} &p(t, \mathbf{x}, \mathbf{v}) = 0 \,, && && \mathbf{x} \in \partial\Omega_{\mathcal{T}} \,, \mathbf{v} \cdot \mathbf{n}_{\mathcal{T}} < 0 \,, \\ &p(t, \mathbf{x}, \mathbf{v}) = \begin{cases} p\left(t - K(\mathbf{v}, \mathbf{v}'), \mathbf{x}, \mathbf{v}'\right) \,, & K(\mathbf{v}, \mathbf{v}') \le t \,, \\ 0 \,, & K(\mathbf{v}, \mathbf{v}') > t \,, \end{cases} && && \mathbf{x} \in \partial\Omega_{\mathcal{R}} \,, \mathbf{v} \cdot \mathbf{n}_{\mathcal{R}} < 0 \,. \end{aligned} \tag{4.19}$$

Derivation of a mean first passage time problem

We now derive the mean exit time problem directly from the forward problem using the adjoint operator along with its main property [105]: For a forward problem specified by

$$\mathcal{M} p = 0 \,,$$

coupled with initial and boundary conditions, the backward problem is given by the adjoint operator

$$\mathcal{M}^* q = 0 \,,$$

with respective final conditions and adjoint boundary conditions. The adjoint operator itself is defined through an inner product and its effect on suitably smooth test-functions $p, q : [0, \infty) \times \Omega \times V \mapsto \mathbb{R}_0^+$ through [105]

$$\langle p, \mathcal{M}^* q \rangle = \langle \mathcal{M} p, q \rangle \,, \qquad \langle p, q \rangle = \int_0^\infty \int_\Omega \int_V p(t, \mathbf{x}, \mathbf{v})\, q(t, \mathbf{x}, \mathbf{v})\, \mathrm{d}\mathbf{v}\, \mathrm{d}\mathbf{x}\, \mathrm{d}t \,.$$

We aim to construct the backward problem for (4.18)–(4.19) using this inner product relation. The forward operator $\mathcal{M}$ as given in (4.18) takes the form

$$\mathcal{M}p = -\underbrace{\frac{\partial p}{\partial t}}_{\textbf{(i)}} - \underbrace{\mathbf{v}\cdot\nabla_{\mathbf{x}}p}_{\textbf{(ii)}} - \underbrace{\lambda p}_{\textbf{(iii)}} + \underbrace{\lambda\int_{V'(t,\mathbf{v},0)} T(\mathbf{v},\mathbf{v}_*)\,p(t-K(\mathbf{v},\mathbf{v}_*),\mathbf{x},\mathbf{v}_*)\,\mathrm{d}\mathbf{v}_*}_{\textbf{(iv)}}\,.$$

Due to the nature of the inner product and the additive relation between the parts **(i)**–**(iv)**, we can derive adjoint terms individually.

(i): We integrate by parts to obtain

$$\begin{aligned}\langle\mathcal{M}_{(i)}p,q\rangle &= \int_0^\infty\int_\Omega\int_V q\frac{\partial p}{\partial t}\,\mathrm{d}\mathbf{v}\,\mathrm{d}\mathbf{x}\,\mathrm{d}t\\ &= -\int_0^\infty\int_\Omega\int_V p\frac{\partial q}{\partial t}\,\mathrm{d}\mathbf{v}\,\mathrm{d}\mathbf{x}\,\mathrm{d}t = \langle p,\mathcal{M}^*_{(i)}q\rangle\,,\end{aligned}\tag{4.20}$$

where we assume that the term $p\,q$ vanishes as $t\to\{0,\infty\}$. This assumption is satisfied, as $p\to 0$ as $t\to\infty$ and $q\to 0$ as $t\to 0$.

(ii): Using the divergence theorem, we obtain

$$\begin{aligned}\langle\mathcal{M}_{(ii)}p,q\rangle &= \int_0^\infty\int_\Omega\int_V q\,\mathbf{v}\cdot\nabla_{\mathbf{x}}p\,\mathrm{d}\mathbf{v}\,\mathrm{d}\mathbf{x}\,\mathrm{d}t\\ &= -\int_0^\infty\int_\Omega\int_V p\,\mathbf{v}\cdot\nabla_{\mathbf{x}}q\,\mathrm{d}\mathbf{v}\,\mathrm{d}\mathbf{x}\,\mathrm{d}t = \langle p,\mathcal{M}^*_{(ii)}q\rangle\,,\end{aligned}$$

where the adjoint boundary conditions are chosen such that the boundary term vanishes. These boundary conditions are derived in Appendix A and take the form

$$\begin{aligned}&q(t,\mathbf{x},\mathbf{v}) = 0\,, && \mathbf{x}\in\partial\Omega_{\mathcal{T}},\ \mathbf{v}\cdot\mathbf{n}_{\mathcal{T}}>0\,,\\ &q(t,\mathbf{x},\mathbf{v}) = q(t+K(\mathbf{v}',\mathbf{v}),\mathbf{x},\mathbf{v}')\,, && \mathbf{x}\in\partial\Omega_{\mathcal{R}},\ \mathbf{v}\cdot\mathbf{n}_{\mathcal{R}}>0\,.\end{aligned}\tag{4.21}$$

(iii): The adjoint operator for this term can be seen directly

$$\langle\mathcal{M}_{(iii)}p,q\rangle = \int_0^\infty\int_\Omega\int_V \lambda\,p\,q\,\mathrm{d}\mathbf{v}\,\mathrm{d}\mathbf{x}\,\mathrm{d}t = \langle p,\mathcal{M}^*_{(iii)}q\rangle\,.$$

(iv): Let us first write down the forward operator $\mathcal{M}_{(iv)}$

$$\langle\mathcal{M}_{(iv)}p,q\rangle = \int_0^\infty\int_\Omega\int_V q\left(\int_{V'(t,\mathbf{v},0)} T(\mathbf{v},\mathbf{v}_*)\,p(t-K(\mathbf{v},\mathbf{v}_*),\mathbf{x},\mathbf{v}_*)\,\mathrm{d}\mathbf{v}_*\right)\mathrm{d}\mathbf{v}\,\mathrm{d}\mathbf{x}\,\mathrm{d}t\,.$$

Noting that $V'(t,\mathbf{v},0)=\{\mathbf{v}_*\in V: 0\le K(\mathbf{v},\mathbf{v}_*)\le t\}$, we can change the order of integration to obtain

$$\langle\mathcal{M}_{(iv)}p,q\rangle = \int_\Omega\int_V\int_V\int_{K(\mathbf{v},\mathbf{v}_*)}^\infty T(\mathbf{v},\mathbf{v}_*)\,p(t-K(\mathbf{v},\mathbf{v}_*),\mathbf{x},\mathbf{v}_*)q(t,\mathbf{x},\mathbf{v})\,\mathrm{d}t\,\mathrm{d}\mathbf{v}_*\,\mathrm{d}\mathbf{v}\,\mathrm{d}\mathbf{x}\,.$$

Using the transformation $\hat{t} = t - K(\mathbf{v}, \mathbf{v}_*)$, we get

$$\begin{aligned}
\langle \mathcal{M}_{(iv)} p, q \rangle &= \int_\Omega \int_V \int_V \int_0^\infty q\left(\hat{t} + K(\mathbf{v}, \mathbf{v}_*), \mathbf{x}, \mathbf{v}\right) T(\mathbf{v}, \mathbf{v}_*)\, p\left(\hat{t}, \mathbf{x}, \mathbf{v}_*\right)\, \mathrm{d}\hat{t}\, \mathrm{d}\mathbf{v}_*\, \mathrm{d}\mathbf{v}\, \mathrm{d}\mathbf{x} \\
&= \int_0^\infty \int_\Omega \int_V p(t, \mathbf{x}, \mathbf{v}_*) \int_V q(t + K(\mathbf{v}, \mathbf{v}_*), \mathbf{x}, \mathbf{v})\, T(\mathbf{v}, \mathbf{v}_*)\, \mathrm{d}\mathbf{v}\, \mathrm{d}\mathbf{v}_*\, \mathrm{d}\mathbf{x}\, \mathrm{d}t \\
&= \int_0^\infty \int_\Omega \int_V p \left(\int_V T(\mathbf{v}_*, \mathbf{v})\, q(t + K(\mathbf{v}_*, \mathbf{v}), \mathbf{x}, \mathbf{v}_*)\, \mathrm{d}\mathbf{v}_* \right) \mathrm{d}\mathbf{v}\, \mathrm{d}\mathbf{x}\, \mathrm{d}t \\
&= \langle p, \mathcal{M}^*_{(iv)} q \rangle \,,
\end{aligned}$$

where we change the order of integration, drop the hat on t for clarity and switch the integration variables $\mathbf{v}_*$, $\mathbf{v}$.

Putting the results **(i)**–**(iv)** back together, the backwards equation for $q(t_0, \mathbf{x_0}, \mathbf{v_0})$ takes the form

$$-\frac{\partial q}{\partial t_0} - \mathbf{v_0} \cdot \nabla_{\mathbf{x}} q = -\lambda q + \lambda \int_V T(\mathbf{v}_*, \mathbf{v_0})\, q(t + K(\mathbf{v}_*, \mathbf{v_0}), \mathbf{x_0}, \mathbf{v}_*)\, \mathrm{d}\mathbf{v}_* \tag{4.22}$$

Equation (4.22) forms the adjoint problem to (4.6)–(4.7) in combination with the adjoint boundary conditions (4.21). We can understand the backwards problem, by noting that the forward probability p is actually $p(t, \mathbf{x}, \mathbf{v} | t_0, \mathbf{x_0}, \mathbf{v_0})$, the probability of being at $(t, \mathbf{x}, \mathbf{v})$ after starting from $(t_0, \mathbf{x_0}, \mathbf{v_0})$, where $t_0 < t$. Similarly, the probability q can be written as $q(t_0, \mathbf{x_0}, \mathbf{v_0}) = p(t, \mathbf{x}, \mathbf{v} | t_0, \mathbf{x_0}, \mathbf{v_0})$ thereby explaining the name backwards problem.

In order to derive the mean first passage time, we define the mass left in the system at time t, given that the system is initialised with a Dirac of intensity one at $(\mathbf{x_0}, \mathbf{v_0})$ at time $t = 0$ through

$$\begin{aligned}
\rho(t, \mathbf{x_0}, \mathbf{v_0}) &= \int_\Omega \int_V p(t, \mathbf{x}, \mathbf{v} |\, 0, \mathbf{x_0}, \mathbf{v_0})\, \mathrm{d}\mathbf{v}\, \mathrm{d}\mathbf{x} \\
&= \int_\Omega \int_V p(0, \mathbf{x}, \mathbf{v} | -t, \mathbf{x_0}, \mathbf{v_0})\, \mathrm{d}\mathbf{v}\, \mathrm{d}\mathbf{x} \,,
\end{aligned}$$

where we make use of the fact that the operator $\mathcal{M}$ is time invariant and we can therefore shift the system in time. Integrating the backwards problem with respect to $\mathbf{x} \in \Omega$ and $\mathbf{v} \in V$ and using negative time, we can now obtain an equation for ρ

$$\frac{\partial \rho}{\partial t} - \mathbf{v_0} \cdot \nabla_{\mathbf{x_0}} \rho = -\lambda \rho + \lambda \int_V T(\mathbf{v}_*, \mathbf{v_0})\, \rho(t - K(\mathbf{v}_*, \mathbf{v_0}), \mathbf{x_0}, \mathbf{v}_*)\, \mathrm{d}\mathbf{v}_* \,. \tag{4.23}$$

The boundary conditions for ρ can be derived from (4.21) and take the form

$$\begin{aligned}
&\rho(t, \mathbf{x}, \mathbf{v}) = 0 \,, && \mathbf{x} \in \partial\Omega_{\mathcal{T}},\ \mathbf{v} \cdot \mathbf{n}_{\mathcal{T}} > 0 \,, \\
&\rho(t, \mathbf{x}, \mathbf{v}) = \rho(t - K(\mathbf{v}', \mathbf{v}), \mathbf{x}, \mathbf{v}') \,, && \mathbf{x} \in \partial\Omega_{\mathcal{R}},\ \mathbf{v} \cdot \mathbf{n}_{\mathcal{R}} > 0 \,.
\end{aligned} \tag{4.24}$$

We Taylor expand (4.23) to get

$$\begin{aligned}\frac{\partial \rho}{\partial t} - \mathbf{v_0} \cdot \nabla_{\mathbf{x_0}} \rho(t, \mathbf{x_0}, \mathbf{v_0}) &= -\lambda \rho(t, \mathbf{x_0}, \mathbf{v_0}) \\ &+ \lambda \int_V T(\mathbf{v}_*, \mathbf{v_0}) \left\{ \rho(t, \mathbf{x_0}, \mathbf{v}_*) - K(\mathbf{v}_*, \mathbf{v_0}) \frac{\partial \rho}{\partial t} \pm ... \right\} \mathrm{d}\mathbf{v}_* \,.\end{aligned} \tag{4.25}$$

As ρ denotes the mass left in the system at time t, it is clear that the mass leaving the system in the interval $[t, t + \mathrm{d}t)$ is

$$\rho(t, \mathbf{x_0}, \mathbf{v_0}) - \rho(t + \mathrm{d}t, \mathbf{x_0}, \mathbf{v_0}) \approx -\frac{\partial \rho}{\partial t} \mathrm{d}t \,.$$

Hence, the mean first passage time can be calculated using

$$\tau(\mathbf{x_0}, \mathbf{v_0}) = -\int_0^\infty t \, \frac{\partial \rho}{\partial t} \, \mathrm{d}t = \int_0^\infty \rho \, \mathrm{d}t \,,$$

where we use the fact that $\rho \to 0$ as $t \to \infty$. We can derive the mean first passage time problem by integrating (4.25) with respect to $t \in [0, \infty)$

$$\begin{aligned}\mathbf{v_0} \cdot \nabla_{\mathbf{x_0}} \tau - \lambda \tau + \lambda \int_V T(\mathbf{v}_*, \mathbf{v_0}) \, \tau(\mathbf{x_0}, \mathbf{v}_*) \, \mathrm{d}\mathbf{v}_* \\ = -\left(1 + \lambda \int_V T(\mathbf{v}_*, \mathbf{v_0}) K(\mathbf{v}_*, \mathbf{v_0}) \, \mathrm{d}\mathbf{v}_* \right) .\end{aligned} \tag{4.26}$$

Note that all the other terms of the Taylor expansion in (4.25) vanish, because all derivatives of ρ are zero at $t = \{0, \infty\}$.[2] By integrating the boundary conditions in (4.24), we can obtain the corresponding boundary conditions on τ that take the form

$$\begin{aligned}\tau(\mathbf{x_0}, \mathbf{v_0}) &= 0 \,, & \mathbf{x_0} \in \partial\Omega_{\mathcal{T}} \,, \mathbf{v_0} \cdot \mathbf{n}_{\mathcal{T}} > 0 \,, \\ \tau(\mathbf{x_0}, \mathbf{v_0}') &= \tau(\mathbf{x_0}, \mathbf{v_0}) + K(\mathbf{v_0}', \mathbf{v_0}) \,, & \mathbf{x_0} \in \partial\Omega_{\mathcal{R}} \,, \mathbf{v_0} \cdot \mathbf{n}_{\mathcal{R}} > 0 \,.\end{aligned} \tag{4.27}$$

Simplification for experimental situation

We can make several simplifications to obtain a mean-first passage time equation that corresponds to the robotic experiments. In particular, we can use the fact that $V = s\,\mathbb{S}^1$ and therefore that all velocities $v \in V$ can be written in the form

$$\mathbf{v} = s \begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix} , \qquad \theta \in [-\pi, \pi] \,.$$

This fact allows one to formulate the mean exit time problem in three variables such that $\tau = \tau(x, y, \theta)$, where $\mathbf{x} = [x, y]^T \in \Omega = [0, L_x] \times [0, L_y]$ according to the definitions

[2] For $t = 0$ this is the case, because from every position $\mathbf{x} \in \Omega \setminus \partial\Omega$ it takes a minimum time $t_0(\mathbf{x}) > 0$ to leave the arena and therefore $\rho(t, \mathbf{x}) \equiv 1$ for $t \in [0, t_0)$.

made in Figure 4.1. Additionally, the uniform turning kernel can be written in the form

$$T(\theta_*, \theta) = \frac{1}{2\pi}\,, \qquad \theta, \theta_* \in [-\pi, \pi]\,.$$

The exit time problem takes the form

$$s\cos\theta\,\frac{\partial\tau}{\partial x} + s\sin\theta\,\frac{\partial\tau}{\partial y} - \lambda\tau + \frac{\lambda}{2\pi}\int_{-\pi}^{\pi}\tau(x, y, \theta_*)\,\mathrm{d}\theta_* = -\left(1 + \frac{\lambda}{2\pi}\int_{-\pi}^{\pi}K(\theta_*, \theta)\,\mathrm{d}\theta_*\right)\,.$$

Using the turning times given in (4.10), we can evaluate the integral

$$\int_{-\pi}^{\pi}K(\theta_*, \theta)\,\mathrm{d}\theta_* = \int_{-\pi}^{\pi}K(\theta_*, 0)\,\mathrm{d}\theta_* = 2\int_0^{\pi}\frac{\theta_*}{\omega}\,\mathrm{d}\theta_* = \frac{\pi^2}{\omega}\,.$$

From the nature of the problem, we can additionally expect only small differences in exit-times along the y-coordinate, which leads to the idea of defining the exit-time purely along the x-coordinate as follows

$$\bar{\tau}(x, \theta) = \frac{1}{L_y}\int_0^{L_y}\tau(x, y, \theta)\,\mathrm{d}y\,.$$

Integrating the intermediate exit-time equation with respect to y and dividing by L_y, we can obtain an equation for the newly defined $\bar{\tau}(x, \theta)$:

$$s\cos\theta\,\frac{\partial\bar{\tau}}{\partial x} + \frac{s\sin\theta}{L_y}\,\tau(x, y, \theta)\Big|_{y=0}^{y=L_y} - \lambda\bar{\tau} + \frac{\lambda}{2\pi}\int_{-\pi}^{\pi}\bar{\tau}(x, \theta_*)\,\mathrm{d}\theta_* = -\left(1 + \frac{\lambda\pi}{2\omega}\right)\,.$$

The only term posing an additional difficulty is the second term. To simplify this term, we can employ a symmetry argument of the form

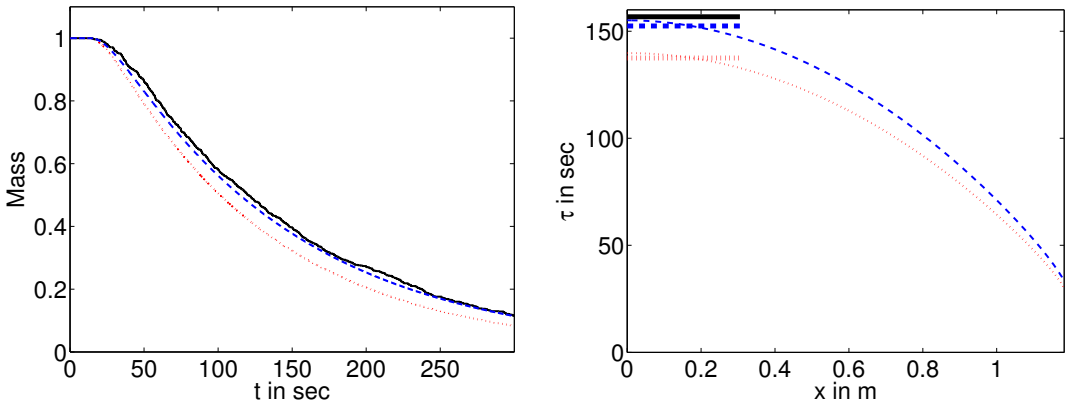
$$\tau(x, L_y, \theta) = \tau(x, 0, \theta') = \tau(x, 0, \theta) + K(\theta, \theta')\,, \qquad \theta \in [0, \pi]\,,$$

where $\theta' = -\theta$ is the result of reflecting the velocity on the x-axis. We apply the boundary condition (4.27) to obtain the second part of the equation. We can additionally make use of the fact that θ and $-\theta$ result in the exact same equation in this system and reduce the system to $\theta \in [0, \pi]$. Additionally, $K(\theta, -\theta)$ is given through

$$K(\theta, -\theta) = \frac{2}{\omega}\min(\theta, \pi - \theta)\,, \qquad \theta \in [0, \pi]\,.$$

Therefore, the full one-dimensional first-exit time equation takes the form

$$s\cos\theta\,\frac{\partial\bar{\tau}}{\partial x} + \frac{2s\sin\theta}{L_y\omega}\min(\theta, \pi - \theta) - \lambda\bar{\tau} + \frac{\lambda}{\pi}\int_0^{\pi}\bar{\tau}(x, \theta_*)\,\mathrm{d}\theta_* = -\left(1 + \frac{\lambda\pi}{2\omega}\right)\,, \quad (4.28)$$



(a) (b)

Figure 4.7: (a) *Mean mass in system over time. Solid (black) line: experimental data; dotted (red) line: numerical solution of* (4.6) *with boundary conditions* (4.7)*; dashed (blue) line: numerical solution of* (4.11)–(4.12) *with boundary conditions* (4.13)
(b) *Mean exit time averaged over all velocities. Solid (black) line: experimental data; dotted (red) line: numerical solution of* (4.8) *with boundary conditions* (4.9)*; bold dotted (red) line: average of dotted line over* L_0*; dashed (blue) line: numerical solution of* (4.26) *with boundary conditions* (4.27)*; bold dashed (blue) line: average of dashed line over* L_0*.*
For both plots parameters and numerical methods are given in the text.

for all $x \in [0, L_x]$ and $\theta \in [0, \pi]$. The boundary conditions for this equation take the form

$$\begin{aligned} \bar{\tau}(L_x, \theta) &= 0\,, \\ \bar{\tau}(0, \theta) &= \bar{\tau}(0, \pi - \theta) - \frac{\pi - 2\theta}{\omega}\,, \qquad \theta \in \left[0, \frac{\pi}{2}\right]\,, \end{aligned} \tag{4.29}$$

where the second boundary condition is obtained from the second condition in (4.27) for the situation where $x = 0$ by considering the reflection of the angle θ on the y-axis. We will compare the adjusted system (4.28)–(4.29) to the measured exit times of the robots. Note that we can consider the same system with instant turning by letting $\omega \to \infty$ and therefore eliminating all terms containing ω. Another interesting thing about (4.28) is that the right-hand side represents a scaling factor of $1 + \lambda\pi/2\omega$. One could also arrive at this scaling through the naive approach of multiplying the average time of a tumble ($\pi/2\omega$) with the frequency of tumbles λ. Whilst this scaling approach might present a reasonable approximation for this particular system, it is not valid for arbitrary turning kernels and turning times. The scaling approach also does not account for finite turning times that occur when a robot collides with a wall.

4.3.3 Comparison between experimental data and theory

In this section we compare the classical theory introduced in Section 1.3.2 and reviewed in Section 4.3.1 and the extended theory developed in Section 4.3.2 to experimental data. The experimental data was collected using 50 repetitions of the test problem without signal ($\alpha = 0$) described in Section 4.2.3 with all experimental parameters as given in Section 4.2.3. We let each of those experiments run until $t = 300\,\text{sec}$ and measure the exit times for the robots. Naturally there were a varying number of robots left in the arena at the end of the experiments. In total 92 out of the $50 \times 16 = 800$ robots did not leave the arena in the allotted time. In the following, we present two different ways of interpreting the experimental data.

Mass over time

The first way of interpreting the experimental data is through the proportion of mass left in the system at a given time. We thereby count the number of robots left in the arena at a given time and compare them to continuum implementations of the corresponding systems in Figure 4.7(a). The solid (black) line indicates the experimental data, which is the percentage of robots that have not left the arena at that point in time.

The result of a numerical solution of the classical theory given through (4.6) and the boundary conditions (4.7) is shown as the dotted (red) line. For the numerical solution we used a first order explicit finite volume scheme with the grid size $\Delta x = 1.1825\,\text{m}/200$, time step $\Delta t = 10^{-3}\,\text{sec}$ and a total of 40 velocity directions.

The dashed (blue) line shows a numerical solution of the adjusted system (4.11)–(4.12) with boundary conditions given in (4.13). The numerical solution was achieved using a first order finite volume method as before, paired with an upwind scheme for (4.12). For (4.11) we used $\Delta x = 1.1825\,\text{m}/200$, $\Delta t = 10^{-3}\,\text{sec}$ and a total of 40 velocity directions. For (4.12) we used the same $\Delta t = 10^{-3}\,\text{sec}$ and a discretisation of $\Delta\eta = 3.38 \times 10^{-2}\,\text{sec}$ corresponding to the time it takes to turn from one velocity direction to the next.

Figure 4.7(a) shows that the classical model corresponds qualitatively to the experimental data, but shows a significant quantitative difference. The model that incorporates finite turning times, however, shows a much improved match.

Mean exit time problem

The second way of interpreting the experimental data is through the use of mean exit times. The problem we are facing here is that we only know exit times for 708 out of the 800 robots involved in the experiments. The mean exit time for those 708 robots is 121.92 sec. In order to get an estimate for the remaining robots, we use the fact that the mass over time plot for the experimental data as seen in Figure 4.7(a) takes the form of a decaying exponential for $t \in [100, 300]$ sec. Using a best exponential fit, we can estimate the decay rate to $8.02 \times 10^{-3}\,\mathrm{sec}^{-1}$. We can therefore approximate the average leaving time of robots remaining in the arena at $t = 300$ sec to be

$$300\,\mathrm{sec} + \frac{1}{8.02 \times 10^{-3}\,\mathrm{sec}^{-1}} = 424.69\,\mathrm{sec}\,.$$

Using this, we can now estimate the total mean exit time of all robots to

$$121.92\,\mathrm{sec} \cdot \frac{708}{800} + 424.69\,\mathrm{sec} \cdot \frac{92}{800} = 156.74\,\mathrm{sec}\,.$$

We compare this experimental leaving time to the solutions of the simplified mean first passage time system given in (4.28)–(4.29) for the case with instant turning ($\omega = \infty$) and the case with finite turning times ($\omega = 4.65\,\mathrm{rad\,sec}^{-1}$). In both cases we achieve a numerical solution of the system using a first order combined up-down-winding approach on a regular (x, θ) grid with $\Delta\theta = \pi/40$ rad and $\Delta x = 1.1825\,\mathrm{m}/200$.

The comparison can be seen in Figure 4.7(b), where the solid (black) line indicates the experimentally approximated exit time. The dotted (red) line shows the case with instant turning and the dashed (blue) line shows the adjusted result for finite turning times. In both cases the plotted curves represent $\bar{\tau}$ averaged over all values of θ. Additionally, averages over the initial pen $[0, L_0]$ are shown as bold horizontal lines. For the classical model ($\omega = \infty$) the mean-exit time from the initial pen is 137.49 sec (error of 12.3% compared to experimental data) and for the adjusted model the value is 152.43 sec (error of 2.7%). Again, the adjusted model presented in Section 4.3.2 gives a much improved match compared to the classical models. Hence, the inclusion of finite turning times is vital in explaining the experimental data and we have in this section presented a consistent way of incorporating those into classical velocity jump models.

4.4 Introduction of signals

Let us now return to the experimental test problem defined in Section 4.2.3 and incorporate the grey-black gradient poster into the system as was shown in Figure 4.6(b).

In Figure 4.3(b) we have already seen the characteristic curve returned by one of the robots as it moves through a gradient. The implementation of the reaction to the gradient can be seen in Table 4.3 and is motivated by Model III in Section 2.4.1. As the gradient improves the target finding of the robots, the exit times are much reduced compared to the experiment without signals and we can actually wait until all robots have left the arena. We again completed 50 runs of 16 robots and observed mean exit times of the robots of 78.77 sec compared to the estimated 156.74 sec without signal.

4.4.1 Velocity jump models with varying turning frequencies

In this section, we are aiming to formulate velocity jump models that incorporate changing turning frequencies λ. In particular, we are interested in turning frequencies that depend on the current velocity of the robot as well as its position in the domain, i.e. $\lambda = \lambda(\mathbf{x}, \mathbf{v})$. The general velocity jump model for this case can be formulated as (cf. (4.6))

$$\frac{\partial p}{\partial t} + \mathbf{v} \cdot \nabla_{\mathbf{x}} p = -\lambda(\mathbf{x}, \mathbf{v})\, p + \int_V \lambda(\mathbf{x}, \mathbf{v}_*)\, T(\mathbf{v}, \mathbf{v}_*)\, p(t, \mathbf{x}, \mathbf{v}_*)\, \mathrm{d}\mathbf{v}_*\,, \tag{4.30}$$

with the boundary conditions given in (4.7). Similarly, we can formulate this system by incorporating the resting period (cf. (4.11)–(4.12))

$$\begin{aligned} \frac{\partial p}{\partial t} + \mathbf{v} \cdot \nabla_{\mathbf{x}} p &= -\lambda(\mathbf{x}, \mathbf{v})\, p(t, \mathbf{x}, \mathbf{v}) + r(t, \mathbf{x}, \mathbf{v}, 0^+)\,, \\ \frac{\partial r}{\partial t} - \frac{\partial r}{\partial \eta} &= \int_V \lambda(\mathbf{x}, \mathbf{v}_*)\, p(t, \mathbf{x}, \mathbf{v}_*)\, T(\mathbf{v}, \mathbf{v}_*)\, \delta(\eta - K(\mathbf{v}, \mathbf{v}_*))\, \mathrm{d}\mathbf{v}_*\,, \end{aligned} \tag{4.31}$$

with boundary conditions (4.13). The system (4.31) can again be formulated in the form of a delay differential equation (cf. (4.18))

$$\frac{\partial p}{\partial t} + \mathbf{v} \cdot \nabla_{\mathbf{x}} p = -\lambda(\mathbf{x}, \mathbf{v})\, p + \int_{V'(t,\mathbf{v},0)} \lambda(\mathbf{x}, \mathbf{v}_*)\, T(\mathbf{v}, \mathbf{v}_*)\, p(t - K(\mathbf{v}, \mathbf{v}_*), \mathbf{x}, \mathbf{v}_*)\, \mathrm{d}\mathbf{v}_*\,, \tag{4.32}$$

where the boundary conditions take the form (4.19). Similarly to the derivation in Section 4.3.2, one can derive the backwards problem, with the mean first passage time equation taking the form (cf. (4.26))

$$\begin{aligned} \mathbf{v_0} \cdot \nabla_{\mathbf{x_0}} \tau - \lambda(\mathbf{x_0}, \mathbf{v_0})\tau + \lambda(\mathbf{x_0}, \mathbf{v_0}) \int_V T(\mathbf{v}_*, \mathbf{v_0})\, \tau(\mathbf{x_0}, \mathbf{v}_*)\, \mathrm{d}\mathbf{v}_* \\ = -\left(1 + \lambda(\mathbf{x_0}, \mathbf{v_0}) \int_V T(\mathbf{v}_*, \mathbf{v_0})\, K(\mathbf{v}_*, \mathbf{v_0})\, \mathrm{d}\mathbf{v}_*\right)\,, \end{aligned} \tag{4.33}$$

with boundary conditions as given in (4.27).

4.4.2 Turning frequency in experiments with gradient

In Section 4.4.1 we showed the general velocity jump models under the inclusion of changing turning frequencies. In order to be able to compare those models to experimental results, we have to approximate the form of $\lambda(\mathbf{x}, \mathbf{v})$ that corresponds to the behaviour of robots. The reaction of the robots to the colour gradient is implemented in the random walk (D1)–(D4) shown in Table 4.3. The governing equations for the internal dynamics and the change in turning frequencies are given in (4.4)–(4.5) and are motivated by bacterial chemotaxis, in particular by Model III in Section 2.4.1.

According to results from [56], a macroscopic density formulation for the robotic system is given through the hyperbolic chemotaxis equation

$$\frac{1}{\lambda_0}\frac{\partial^2 n}{\partial t^2} + \frac{\partial n}{\partial t} = \frac{s^2}{d\lambda_0}\Delta n - \nabla \cdot \left(n \frac{\alpha \lambda_0 s^2 t_a}{d\lambda_0(1+\lambda_0 t_a)} \nabla S \right) , \tag{4.34}$$

where $S : \Omega \mapsto \mathbb{R}$ indicates the colour gradient and $n(t, \mathbf{x})$ describes the concentration of robots in Ω. Equation (4.34) can be approximated by the velocity jump process (4.30) with the form for the turning frequency given by

$$\lambda(\mathbf{x}, \mathbf{v}) = \lambda_0 - \gamma\, \mathbf{v} \cdot \nabla S(\mathbf{x}) , \qquad \gamma = \frac{\alpha t_a \lambda_0}{1 + \lambda_0 t_a} . \tag{4.35}$$

Because the gradient of the colour signal S is parallel to the x-axis in the test problem described in Section 4.2.3, we can again simplify the formulation of the exit time problem (4.33) by averaging along the y-axis. The resulting equation takes the form

$$s\cos\theta \frac{\partial \bar{\tau}}{\partial x} + \frac{2s\sin\theta}{L_y \omega} \min(\theta, \pi - \theta) - \lambda(x,\theta)\,\bar{\tau} + \frac{\lambda(x,\theta)}{\pi} \int_0^\pi \bar{\tau}(x, \theta_*)\, \mathrm{d}\theta_* = -1 - \lambda(x,\theta)\frac{\pi}{2\omega} , \tag{4.36}$$

where $\lambda(x, \theta)$ is given through

$$\lambda(x,\theta) = \lambda_0 - \gamma\, s\cos\theta \frac{\partial S(x)}{\partial x} . \tag{4.37}$$

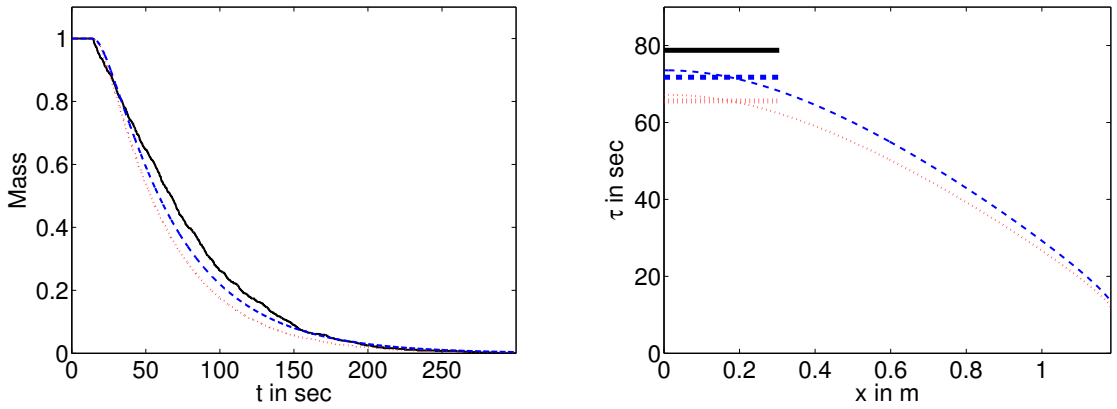
One problem that still needs to be addressed is how one approximates the signal $S(x)$. In Figure 4.3(b) we saw the characteristic curve of one of the 16 robots used during the experiments. However, when repeating the simple gradient measure with a number of robots we saw a strong difference between individuals. This difference occurs in the shape of the measured curve as well as its height. In order to find a good average for all robots, we approximate the signal by a linear function. We measured the values at both end points of the domain for each robot and averaged over the group. The mean value measured at the left (grey) end of the domain is 768.1, whilst

the right (black) end bordering the target area was measured at 377.9. Using the transformation of those measurements into the signal S as given in (D1) in Table 4.3, the linear approximation of $S(x)$ takes the form

$$S(x) = 1.0 - \frac{768.1 - (768.1 - 377.9)x/L_x}{1000}\,, \qquad \frac{\partial S}{\partial x} = \frac{768.1 - 377.9}{1000 L_x} \approx 0.33\,\mathrm{m}^{-1}\,.$$

We will use this linear form of $S(x)$ for all comparisons between experimental data and the derived models.

4.4.3 Comparison between models and experimental results



(a) (b)

Figure 4.8: *Comparison between velocity jump process and experimental data for experiment including colour gradient.* (a) *Mean mass in system over time. Solid (black) line: experimental data; dotted (red) line: numerical solution of* (4.30)*; dashed (blue) line: numerical solution of* (4.31)*. Turning frequency* $\lambda(\mathbf{x}, \mathbf{v})$ *as given in* (4.35)*.*
(b) *Mean exit time averaged over all velocities. Solid (black) line: experimental data; dotted (red) line: numerical solution of* (4.36) *for* $\omega = \infty$*; bold dotted (red) line: average of dotted line over* L_0*; dashed (blue) line: numerical solution of* (4.36) *for* $\omega = 4.65\,\mathrm{rad\,sec}^{-1}$*; bold dashed (blue) line: average of dashed line over* L_0*. Turning frequency* $\lambda(x, \theta)$ *as given in* (4.37)*.*
For both plots parameters and numerical methods are given in the text.

We now want to compare the experimental data with the velocity jump models described in Section 4.4.1. The numerical solutions were achieved using the exact same methods and parameters as in Section 4.3.2 and the results can be seen in Figure 4.8. The parameter values used for the robots are $\lambda_0 = 0.25\,\mathrm{sec}^{-1}$, $\alpha = 8$, $t_a = 10\,\mathrm{sec}$ and $s = 0.58\,\mathrm{m\,sec}^{-1}$.

In Figure 4.8(a) we plot the mass left in the system over time. The solid (black) line again represents the percentage of robots still in the arena at that point in time.

The dotted (red) line is a numerical solution of the velocity jump equation (4.30) with the corresponding boundary conditions (4.7). The dashed (blue) line is a numerical solution of the velocity jump system with resting state given in (4.31) and boundary conditions as in (4.13).

In Figure 4.8(b) we plot the dependence of the mean exit time upon the position along the x-axis. The horizontal solid (black) line again indicates the experimentally measured exit time of 78.77 sec. The dotted (red) line shows a numerical solution of (4.36) with instant turning, i.e. $\omega = \infty$. The dashed (blue) line shows a numerical solution of (4.36) with $\omega = 4.65\,\mathrm{rad\,sec^{-1}}$. For both of these solutions the boundary conditions are given in (4.27). The bold horizontal lines again indicate the average over the initial pen of length L_0.

In both plots in Figure 4.8, we see a good agreement between models and experimental data. The models including finite turning delays represented through the dashed (blue) lines give an improved match compared to the models without this delay. The numerically estimated exit time for the model with instant turning ($\omega = \infty$) is 65.59 sec (error of 16.7% compared to experimental data); with finite turning times it is 71.76 sec (error of 8.9%). The remaining difference between the models and the experimental data can be explained by noisy measurement of the signal $S(\mathbf{x})$ as well as the fact that we used a linear approximation averaged over all robots to obtain the numerical results. We can conclude from this brief study of robot experiments including a colour gradient signal that this signal indeed improves the target finding capacity of the robots and that the models developed in Section 4.3 can be easily generalised to incorporate turning frequencies that change according to external signals.

4.5 Discussion

In this chapter, we introduced swarm robotic experiments as an additional tool for the study of biological systems of dispersal. We studied a target finding task that could in practice be applied to mine detection [100] or oil spill confrontation [92]. We use an experimental system based on 16 *E-Puck* robots [115] placed in a rectangular arena with one edge leading into the target area. Robots are removed as soon as they have achieved the target finding task. We use an algorithm that is inspired by the signal sensing in *E. coli* bacteria [55] and that we studied extensively in Chapter 2.

The main result from this chapter is the adaptation of classical velocity jump processes for a robotic system. When comparing classical models with the experimental results, one observes a significant quantitative difference. We identified the

finite turning times required for a robot to change its direction as the main reason for this difference. In Section 4.3.2 we show that one can incorporate a resting state into the classical velocity jump equation to account for those turning times. Figure 4.7 shows that the adjusted models greatly improve the match with the experimental data. The remaining difference can be explained by experimental noise and by interactions between robots that will be discussed in Chapter 5. Note that the models derived in this chapter can be applied to arbitrary turning kernels and turning time distributions and therefore go beyond the experiments performed here.

The way we incorporate the resting state into the velocity jump process is different from tumbling in bacterial chemotaxis that has been repeatedly studied in the literature [55, 128, 129]. In these models, the tumbling times are exponentially distributed and can therefore be directly incorporated into transport equations that take into account probabilistic changes from resting to moving states [128]. Othmer et al. [128] additionally introduce that the length of the resting phase has an influence on the velocity distribution after the tumble. In the case of robots, the new direction is chosen before entering the resting state and the turning times depend linearly on the turning angle, as formulated in (4.10).

In the second set of experiments, we introduced an external signal in the form of a grey-black gradient poster into the robotic system. The robots can observe this gradient using light sensors on their bottom. As seen in Figure 4.3(b), these measurements are noisy and a strong difference between individual robots can be observed. We showed how the velocity jump models derived in Section 4.3.2 can be extended to incorporate changing turning frequencies and use those changing frequencies as a reaction to the signal. The experiments clearly show that the gradient improves the target finding capabilities of the robots, as the average time to reach the target is almost halved compared to the experiments without signal. We again show that the derived models match well with the experimental results. There is, however, a more significant difference in the signal case. We can account this difference to the fact that we only approximate the signal as a linear function and that every robot observes the signal differently. It would therefore be interesting to repeat the experiments with individual robots, in order to measure whether certain individuals are better at following the gradient than others. Unfortunately, I did not have time to perform these additional experiments during the research leading to this thesis, but I believe further studies of individual performances could lead to interesting new results. Another suggestion for future work would be the use of different turning kernels to incorporate effects like correlations between old and new velocity during the turning phase.

In biological systems, signals are rarely constant, but usually follow a reaction-diffusion PDE similar to the ones seen in Chapter 3. Therefore, an implementation of the full run-and-tumble chemotactic model in the robotic system requires either special sensors for detecting chemical signals, e.g. robots for odour detection [144], or replacing chemical signals by suitable caricatures of them, e.g. using a glowing floor for *E-Puck* robots [113]. Especially in the latter case, one could consider systems where robots are able to change the signal themselves [71], thereby acquiring a system that can be used to implement hybrid models similar to the ones studied in Chapter 2 for the coupling between individual-based species and a changing environment.

Chapter 5

Velocity jump processes with hard-sphere interactions

So far in this thesis, we have mainly considered indirect interactions between particles, e.g. through chemical signals in Chapter 2 or in the form of reactions in Chapter 3. In this chapter, we are interested in one type of direct interaction between particles caused by volume exclusion, an effect that we have already encountered during the swarm robotic experiments in Chapter 4. However, the effect of volume exclusion is not restricted to swarm robots. It can be found in almost every biological system of dispersal, from *E. coli* bacteria to fish, birds and bigger animals and has been incorporated into many existing biological models. For example, in the classical Couzin-type models [38] one could interpret the repulsion effect as one form of volume exclusion. In this chapter, we study hard-sphere volume exclusion, i.e. the considered particles do not change shape to accommodate other particles, in a velocity jump system.

The results presented in this chapter have been accepted for publication in *Physical Review E* [68].

5.1 Types of collisions

As mentioned in the motivation to this chapter, one can consider different types of volume exclusion effects. Even when deciding on hard-sphere interactions, one can distinguish between a number of different collision types. One classical type of collisions can be observed in the kinetic behaviour of ideal gases [26], where interactions occur in the form of fully elastic collisions, i.e. momentum is conserved during a collision. However, for biological applications this type of collision might be less relevant, as animals do not usually transfer momentum when interacting with each other.

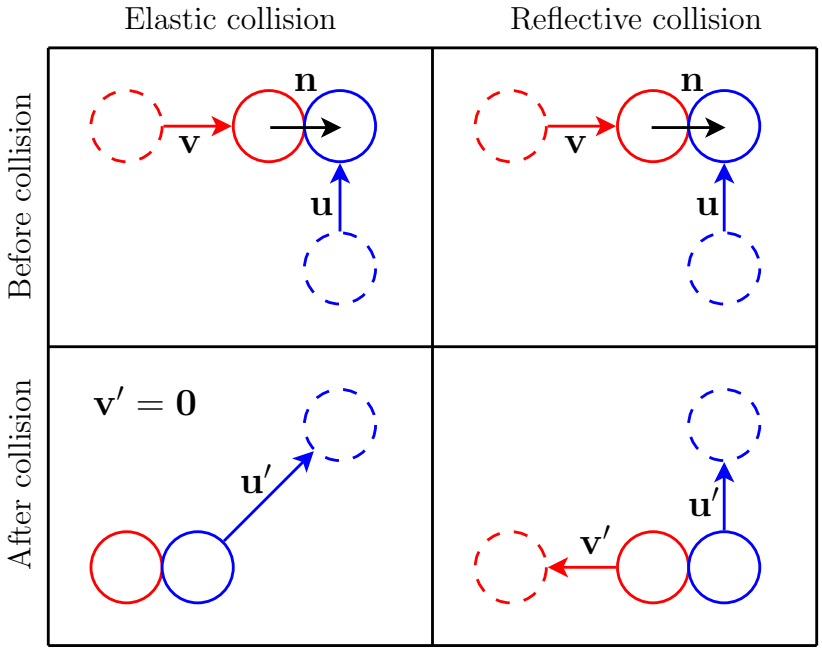


Figure 5.1: Comparison between reflective collisions and fully elastic collisions.

For this reason, we are in this chapter more interested in systems where all particles always move with the speed $s \in \mathbb{R}^+$. Therefore we consider the so-called reflective (speed-preserving) collisions similar to the ones observed during the swarm-robotic experiments in Chapter 4. In this type of interaction particles get directly reflected off each other with the individual speed of each particle being conserved. This type of collision has more biological relevance than fully elastic collisions. For example, in [38], the formation of fish swarms is studied and reflective collisions play an important part in this model in the form of repulsion. Other examples can be found in [153].

The two types of collisions are illustrated in Figure 5.1. In both cases, a particle at position $\mathbf{x}$ and with velocity $\mathbf{v}$ collides with a second particle at $\mathbf{x} + \varepsilon\mathbf{n}$ that has velocity $\mathbf{u}$, where $\mathbf{n} \in \mathbb{S}^{d-1}$ is a unit vector. Similarly to Chapter 4, ε describes the (identical) diameter of each of the particles. We denote the velocities after the collision took place by $\mathbf{v}'$ and $\mathbf{u}'$ respectively. For the reflective collisions, these are given through

$$\mathbf{v}' = \mathbf{v} - 2(\mathbf{v} \cdot \mathbf{n})\,\mathbf{n}\,, \qquad \mathbf{u}' = \mathbf{u} - 2(\mathbf{u} \cdot \mathbf{n})\,\mathbf{n}\,. \tag{5.1}$$

In the case of fully elastic collisions, the new velocities take the form

$$\mathbf{v}' = \mathbf{v} - ((\mathbf{v} - \mathbf{u}) \cdot \mathbf{n})\;\mathbf{n}\,, \qquad \mathbf{u}' = \mathbf{u} + ((\mathbf{v} - \mathbf{u}) \cdot \mathbf{n})\;\mathbf{n}\,. \tag{5.2}$$

The main differences between those two types of collisions are that reflective collisions preserve speed, i.e. particles travel at the same speed before and after the collision, whilst speeds typically change during fully elastic collisions; on the other hand, fully

elastic collisions preserve total momentum in the system, whilst this is not the case for reflective collisions.

We begin this chapter by introducing a numerical simulation framework for those hard-sphere interactions. Using this simulation framework we perform a numerical study regarding the number of collisions in a given system. The first steps towards an analytic way of understanding the effect of collisions is performed in Section 5.4. In Section 5.5 we introduce the so-called *Boltzmann integral* as one way of dealing with collisions of very small particles and we extend this method beyond that limit in Section 5.6. In Section 5.7, we numerically compare the different hard-sphere models and discuss the results in Section 5.8.

5.2 Kinetic Monte-Carlo algorithm

In the later sections of this chapter, we will perform a number of simulations of velocity jump particles with hard-sphere interactions. Therefore, it is necessary to develop an efficient algorithm that allows for fast and accurate simulations. The algorithm we are presenting fits into the general class of Kinetic Monte-Carlo (KMC) methods that was first introduced by Alder and Wainwright [4]. The KMC algorithm we are using is an event-driven algorithm, i.e. it jumps directly from event to event and does not use a fixed time step. We use a similar notation to that introduced in Chapter 1: we assume a system with N particles that are described by their positions $\mathbf{x}^{(i)} = [x_1^{(i)}, x_2^{(i)}]^T \in \Omega$ and their velocities $\mathbf{v}^{(i)} = [v_1^{(i)}, v_2^{(i)}]^T \in V$. Each particle is a circular object with diameter ε and no two particles can overlap at any time, which can be formulated through the restriction

$$\left\|\mathbf{x}^{(i)} - \mathbf{x}^{(j)}\right\| \geq \varepsilon\,, \qquad i,j = 1,\ldots,N,\, i \neq j\,.$$

We can distinguish between three different types of events that need to be considered: (i) a particle goes into a random turning phase, (ii) particle-wall collisions, (iii) particle-particle collisions. The main observation is that after an event has taken place, we can calculate the times of all possible next events and find the minimum in order to directly jump forward to this event. This results in an algorithm that is exact in that all events are considered at exactly the time they happen and no events are missed, and optimal in the sense that one cannot choose a coarser time step without missing events.

In order to calculate the time of the next random turn for each particle, we make use of Gillespie's algorithm for chemical reaction networks [76] that predicts the next

turning time to be

$$T_{\text{turn}}^{(i)} = t - \frac{\log \xi}{\lambda_i}, \qquad i = 1, \ldots, N,$$

where t is the current time, ξ is a uniformly distributed random variable in the interval $[0, 1]$ and λ_i is the turning frequency of particle $i = 1, \ldots, N$. In the simulations used throughout this chapter, we assume an identical (constant) turning frequency λ for all particles. When a turn event happens the particle rotates into the new direction $\theta_* \in [-\pi, \pi]$ that is chosen uniformly at random. We do not include finite turning times as discussed in Chapter 4 in this consideration as they would vastly complicate the analysis.

For the collisions of a particle with the wall, the time until the next event depends on the arena shape. Assuming a rectangular arena $\Omega = [0, L_x] \times [0, L_y]$, the collision time can be calculated as

$$T_{\text{wall}}^{(i)} = t + \min\left(\max\left(\frac{x_1^{(i)}}{v_1^{(i)}}, \frac{L_x - x_1^{(i)}}{v_1^{(i)}}\right), \max\left(\frac{x_2^{(i)}}{v_2^{(i)}}, \frac{L_y - x_2^{(i)}}{v_2^{(i)}}\right)\right),$$

where the use of max could be replaced by a check of the sign of $v_1^{(i)}$ and $v_2^{(i)}$ in order to find the relevant wall. During such a wall collision, the considered particle changes its current direction from $\mathbf{v}^{(i)}$ to $\left(\mathbf{v}^{(i)}\right)'$ as given in (4.2).

The third possible set of events are particle-particle collisions. As particles are considered to be circular objects of diameter ε, we can simply reduce the question to the collision between a point object positioned at $\mathbf{x}$ and moving with velocity $\mathbf{v}$ with a circle situated at the origin with radius ε. Assuming that $\|\mathbf{x}\| > \varepsilon$, i.e. that a collision has not already happened, this can be calculated to

$$T_{\text{coll}}(\mathbf{x}, \mathbf{v}) = -\frac{1}{\|\mathbf{v}\|}\left[\mathbf{v} \cdot \mathbf{x} + \sqrt{\Delta}\right],$$

where Δ is given through

$$\Delta = (\mathbf{v} \cdot \mathbf{x})^2 - \|\mathbf{v}\| \left(\|\mathbf{x}\| - \varepsilon\right).$$

A collision does not occur at all if $\mathbf{v} \cdot \mathbf{x} > 0$, $\|\mathbf{v}\| = 0$ or if $\Delta < 0$; in this case T_{coll} is set to infinity. For two particles $i, j = 1, \ldots, N$, $i \neq j$, the collision time can now be calculated through

$$T_{\text{coll}}^{(i,j)} = t + T_{\text{coll}}\left(\mathbf{x}^{(i)} - \mathbf{x}^{(j)}, \mathbf{v}^{(i)} - \mathbf{v}^{(j)}, \varepsilon\right).$$

In the case that such a collision happens, each of the particles' velocities is updated according to the reflective collision update given in (5.1) (or a different type of collisions, e.g. fully elastic collisions given in (5.2)).

(F1) Initialise the non-overlapping particle positions $\mathbf{x}^{(i)} \in \Omega$ and the velocities $\mathbf{v}^{(i)} \in V$, $i = 1, \ldots, N$. Set $t = 0$.

(F2) Calculate $T_{\text{wall}}^{(i)}$, $T_{\text{turn}}^{(i)}$, $T_{\text{coll}}^{(i,j)}$ for $i, j = 1, \ldots, N$ and find the minimum T_{event}.

(F3) Update all particle positions according to

$$\mathbf{x}^{(i)} = \mathbf{x}^{(i)} + (T_{\text{event}} - t)\mathbf{v}^{(i)}\,, \quad i = 1, \ldots, N\,.$$

Set $t = T_{\text{event}}$.

(F4) Change the velocities of particle(s) that are involved in the current event.

(F5) Go to (F2).

Table 5.1: Basic KMC algorithm

Using these three possible events, it is straightforward to choose the next event as the minimum of $T_{\text{turn}}^{(i)}$, $T_{\text{wall}}^{(i)}$ and $T_{\text{coll}}^{(i,j)}$ over all $i, j = 1, \ldots, N$, where $T_{\text{coll}}^{(i,j)}$ is set to infinity if no collision occurs. The basic algorithm can therefore be implemented as given in Table 5.1.

This algorithm has a complexity of $\mathcal{O}(E \cdot N^2)$, where E denotes the number of events that occurred throughout the simulation. The reason for this complexity is that step (F2) requires the calculation of all $T_{\text{coll}}^{(i,j)}$, $i, j = 1, \ldots, N$. However, one thing we note is that step (F4) changes the velocities of no more than two particles, which means that all event times of particles that are not involved in the current event stay unchanged. Therefore, instead of calculating all the event times in (F2), we can store them throughout the simulation and use (F4) to update the stored data according to the new velocities of the involved particles. As events can only include a maximum of two particles, and the number of events to update per particle is $N + 1$ ($N - 1$ collisions, 1 wall event, 1 turn event), the update step takes a complexity of $\mathcal{O}(N)$.

Because (F2) still requires to find the minimum event time, this change does not change the complexity of the algorithm, but still has a big effect on the run time, as the number of collision times to be calculated reduces drastically. One can now reduce the complexity of the whole algorithm through the use of a priority queue for the event times that reduces the complexity of finding T_{event} to $\mathcal{O}(1)$, but increases the update complexity to $\mathcal{O}(N \log N)$ because events have to be shifted in the priority queue.

Further optimisations are possible, e.g. using the knowledge that a collision that is scheduled for after the next turn or wall event of one of the particles will never happen in practice and its value can therefore be set to infinity. The high number of infinity values in the collision time table can also be exploited directly in the construction of the priority queues. Using all these techniques, we end up with a simulation tool that allows the quick simulation of a high number of particles over many repetitions.

Unless explicitly stated otherwise, the simulations performed are based on reflective collisions as given in (5.1) and each particle therefore always moves at speed s. As parameters, we use $s = 20$ and $\lambda = 200$, which results in $D_0 = s^2/2\lambda = 1$ as discussed in the introduction of the Cattaneo approximation in Section 1.3.2. Figure 5.2(a) shows the first simulation result obtained using the KMC algorithm. We use point particles ($\varepsilon = 0$) inside the unit square domain $\Omega = [0, 1] \times [0, 1]$ in this simulation, i.e. particle-particle collisions are not present in this simulation. The particles are initialised uniformly at random inside the circle with centre at $(0.5, 0.5)$ and radius 0.25. In each run we use $N = 10^3$ particles and stop the simulation at $t = 0.05$. We repeat the experiment 10^4 times and plot a distribution of the 10^7 particle positions using a regular 50×50 grid. We compare the outcome with a numerical solution of the classical velocity jump model presented in (4.6) along with the boundary conditions in (4.7) for a unit sphere arena without target set in Figure 5.2(b). In order to provide matching initial conditions, we use

$$p(0, \mathbf{x}, \mathbf{v}) = \frac{\chi_{B_{0.25}((0.5,0.5))}}{|V|\pi 0.25^2}\,, \qquad \mathbf{x} \in \Omega\,, \mathbf{v} \in V\,, \tag{5.3}$$

where χ denotes the characteristic function and $B_r(\mathbf{x})$ the ball of radius r around $\mathbf{x}$ as defined through

$$B_r(\mathbf{x}) = \{\mathbf{y} \in \mathbb{R}^d : \|\mathbf{x} - \mathbf{y}\| < r\}\,. \tag{5.4}$$

Figure 5.2 shows an excellent match between the classical model (4.6) and the KMC simulations, which was to be expected for the case of point-particles. The discrepancy that is visible when looking at a slice at $x_2 = 0.5$ in Figure 5.2(c) can be attributed to additional numerical diffusion in the PDE solution. We use this simulation result as a proof of concept of our KMC algorithm and we will use this algorithm for the study of finite-sized particles throughout the remainder of this chapter.

5.3 Numerical study of collision frequency

In this first numerical study, we use the algorithm explained in Section 5.2 to perform numerical experiments that count the frequency of collisions from an individual per-

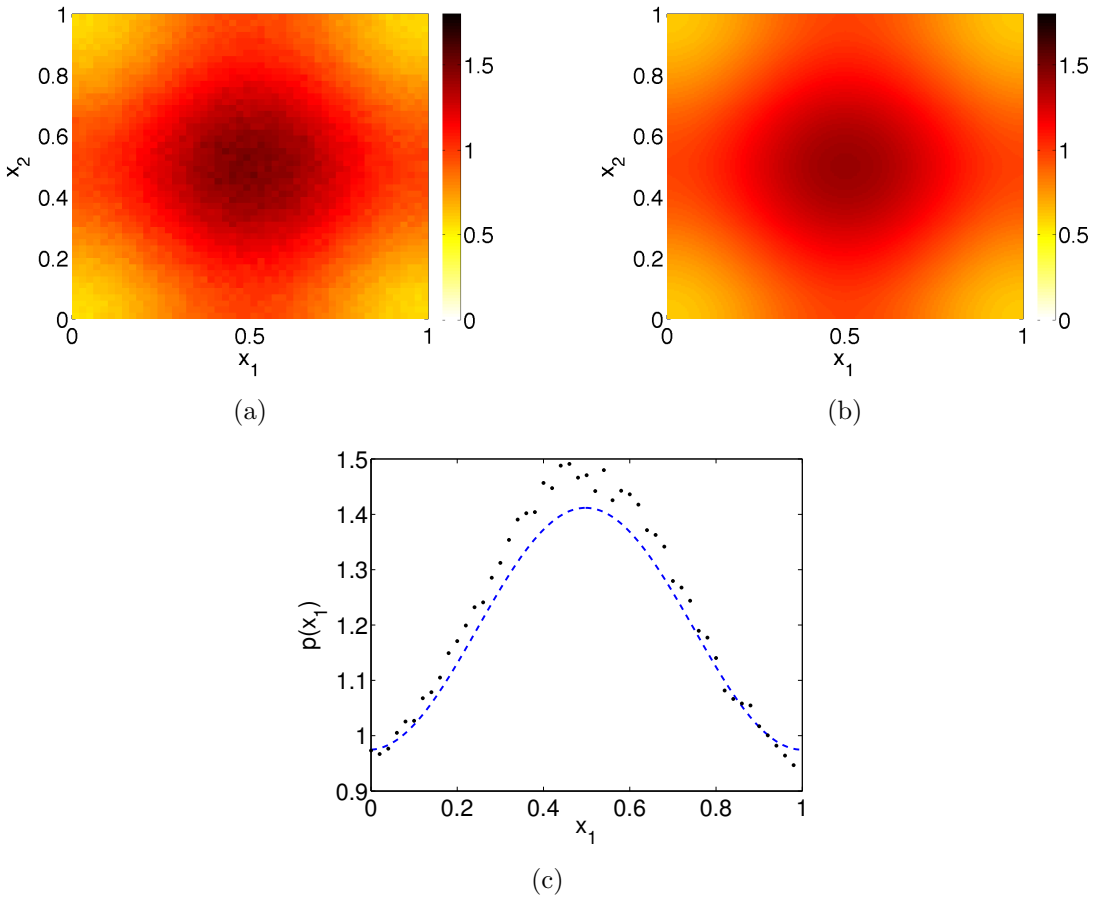
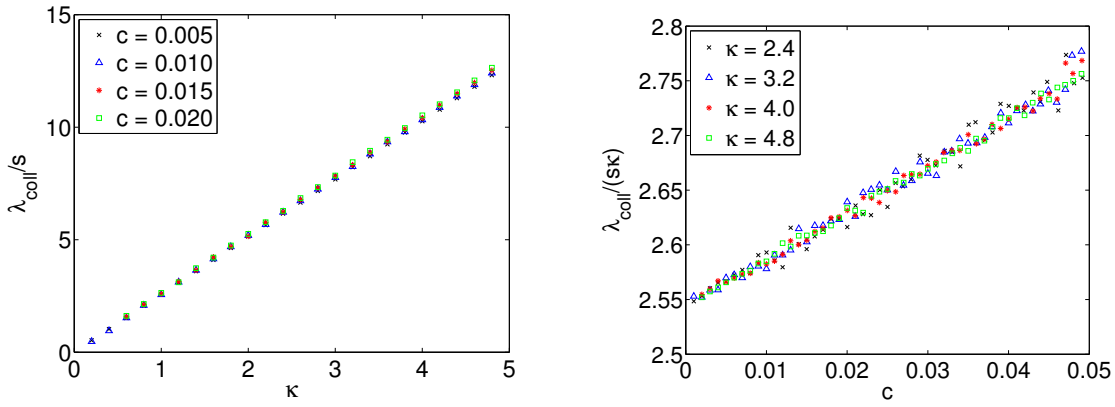


Figure 5.2: *Comparison between KMC simulation for point particles and numerical solution of* (4.6).
(a)*KMC simulation for point particles* ($\varepsilon = 0$).
(b)*Numerical solution of* (4.6). *Numerical parameters explained in the text.*
(c)*Slice through the distributions at* $x_2 = 0.5$. *Dashed (blue) line: distribution (b); (black) dots: KMC simulation.*
We use the initial condition given in (5.3) *and run the system until* $t = 0.05$.

spective. We use a unit square with periodic boundary conditions in order to avoid boundary influences. In those experiments the number of direction changes due to collisions in the system is counted for a certain amount of time and then divided by the number of particles and by the run-time. We use two main parameters, κ and c defined as

$$\kappa = (N-1)\varepsilon\,, \qquad c = \frac{1}{4}N\pi\varepsilon^2\,, \tag{5.5}$$

where κ plays a special role in the Boltzmann collision term discussed later and c represents the volume filled by all the particles and especially in the case of a unit square the fraction of volume filled or the volume concentration of particles. For a



(a) (b)

Figure 5.3: (a) *Frequency of collisions per particle as a function of κ for different values of c.*
(b) *Frequency of collisions per particle divided by κ as a function of c.*
For both plots parameters and numerical methods are given in the text.

given pair (κ, c), the nearest integer value N and an adequate value of ε is found and an experiment is performed. In Figure 5.3(a), we can see how the collision frequency λ_{coll} depends on the value of κ and is on a leading order scale independent of the concentration c. We have a linear connection, which can be estimated to

$$\lambda_{\text{coll}} \approx 2.55 s\kappa\,.$$

The linear dependence on s is necessary, seeing that an increase in particle speed is equivalent to decreasing the run time of the system and vice-versa. Using results from kinetic theory [158], we can predict the frequency of collisions to be

$$\lambda_{\text{coll}} = \frac{\mathbb{E}\left[\left\|\mathbf{v}^{(i)} - \mathbf{v}^{(j)}\right\|\right]}{l_{\text{coll}}}\,,$$

where l_{coll} is the mean free path that can be calculated using the fact that the mean free area $2\,\varepsilon\,l_{\text{coll}} = 1/N$. We can evaluate the mean relative velocity by solving the integral

$$\mathbb{E}\left[\left\|\mathbf{v}^{(i)} - \mathbf{v}^{(j)}\right\|\right] = \frac{1}{|V|^2}\int_V\int_V \|\mathbf{v}-\mathbf{u}\|\,\mathrm{d}\mathbf{v}\,\mathrm{d}\mathbf{u} = \frac{4s}{\pi}\,,$$

and hence obtain for λ_{coll}

$$\lambda_{\text{coll}} = \frac{8s}{\pi}\kappa \approx 2.55 s\kappa\,,$$

which provides an excellent match with the numerical results. We then use this information to get additional insight into the influence of the concentration c, by plotting the value of $\lambda_{\text{coll}}/s\kappa$ as a function of c for different values of κ in Figure 5.3(b). Interestingly, for small concentrations ($c < 0.05$) this dependence does not change with κ and forms a monotonically increasing function, such that

$$\lambda_{\text{coll}} = \frac{8s}{\pi}\kappa f(c)\,.$$

For the range of concentrations plotted in Figure 5.3(b), we can approximate $f(c)$ to be

$$f(c) \approx 1 + 1.73c\,.$$

This additional dependence on c could be caused by grouping effects when more than two particles are close together and bump into each other repeatedly before they break up.

5.4 The BBGKY hierarchy

In this section we derive transport equations for the N-particle system and later for the special case of a two-particle system. These equations can be interpreted as the first equation of the BBGKY hierarchy [16, 98], a hierarchical system of transport equations that models the general kinetics of gases and liquids. Let us begin by defining the N-particle group properties $\vec{\mathbf{x}} = \left(\mathbf{x}^{(1)},\ldots,\mathbf{x}^{(N)}\right)$ and $\vec{\mathbf{v}} = \left(\mathbf{v}^{(1)},\ldots,\mathbf{v}^{(N)}\right)$ and write down an N-particle transport equation

$$\begin{aligned}\frac{\partial}{\partial t}P\left(t,\vec{\mathbf{x}},\vec{\mathbf{v}}\right) + \sum_{i=1}^{N}\mathbf{v}^{(i)}\cdot\nabla_{\mathbf{x}^{(i)}}P\left(t,\vec{\mathbf{x}},\vec{\mathbf{v}}\right)\\ = -N\lambda P\left(t,\vec{\mathbf{x}},\vec{\mathbf{v}}\right) + \lambda\sum_{i=1}^{N}\int_V T(\mathbf{v},\mathbf{v}_*)\,P\left(t,\vec{\mathbf{x}},\mathbf{v}^{(i)}=\mathbf{v}_*\right)\,\mathrm{d}\mathbf{v}_*\,.\end{aligned} \tag{5.6}$$

This transport equation is valid in the region $\vec{\mathbf{x}} \in \Omega_\varepsilon^N$ defined through

$$\Omega_\varepsilon^N = \left\{\left(\mathbf{x}^{(1)},\ldots,\mathbf{x}^{(N)}\right)\in\Omega^N \;:\; \left\|\mathbf{x}^{(i)}-\mathbf{x}^{(j)}\right\| \geq \varepsilon\,,\forall i,j = 1\ldots,N\,,i\neq j\right\}\,.$$

Assuming that the volume fraction c (in two dimensions we have $c = \pi N \varepsilon^2/4$) is small compared to the area of the domain Ω (in the case of a unit square $c \ll 1$), collisions between two particles happen with a probability $\mathcal{O}(c)$, whilst collisions between three or more particles occur with probability $\mathcal{O}(c^2)$. Hence, two-particle collisions present the leading order and we can neglect interactions of more than two particles. We will therefore concentrate on the two-particle case of (5.6) that takes the form[1]

$$\begin{aligned}\underbrace{\frac{\partial P}{\partial t}}_{\text{(i)}} + \underbrace{\mathbf{v}\cdot\nabla_{\mathbf{x}} P}_{\text{(ii)}} + \underbrace{\mathbf{u}\cdot\nabla_{\mathbf{y}} P}_{\text{(iii)}} = \underbrace{-2\lambda P}_{\text{(iv)}} + \underbrace{\lambda\int_V T(\mathbf{v},\mathbf{v}_*)\,P\left(t,\mathbf{x},\mathbf{y},\mathbf{v}_*,\mathbf{u}\right)\,\mathrm{d}\mathbf{v}_*}_{\text{(v)}} \\ + \underbrace{\lambda\int_V T(\mathbf{u},\mathbf{u}_*)\,P\left(t,\mathbf{x},\mathbf{y},\mathbf{v},\mathbf{u}_*\right)\,\mathrm{d}\mathbf{u}_*}_{\text{(vi)}}\,.\end{aligned} \tag{5.7}$$

The two-particle density function is subject to the external boundary conditions

$$\begin{aligned} P(t,\mathbf{x},\mathbf{y},\mathbf{v},\mathbf{u}) &= P(t,\mathbf{x},\mathbf{y},\mathbf{v}',\mathbf{u})\,, \qquad \mathbf{x}\in\partial\Omega\,,\\ P(t,\mathbf{x},\mathbf{y},\mathbf{v},\mathbf{u}) &= P(t,\mathbf{x},\mathbf{y},\mathbf{v},\mathbf{u}')\,, \qquad \mathbf{y}\in\partial\Omega\,,\end{aligned} \tag{5.8}$$

where $\mathbf{v}'$ and $\mathbf{u}'$ are the reflected velocities for wall collisions given in (4.2). Additionally, we impose the following collision condition

$$P(t,\mathbf{x},\mathbf{y},\mathbf{v},\mathbf{u}) = P(t,\mathbf{x},\mathbf{y},\mathbf{v}',\mathbf{u}')\,, \qquad \mathbf{x},\mathbf{y}\in\Omega, \|\mathbf{x}-\mathbf{y}\| = \varepsilon\,, \tag{5.9}$$

where the velocities after collision $\mathbf{v}', \mathbf{u}'$ are defined in (5.1). In order to derive a one-particle transport equation similar to the one in (4.6), we need to integrate over the coordinates of the second particle. In particular, we integrate with respect to $\mathbf{u}\in V$ and $\mathbf{y}\in\Omega^{(2)}(\mathbf{x})$ given through

$$\Omega^{(2)}(\mathbf{x}) = \left\{\mathbf{y}\in\Omega \;:\; \|\mathbf{x}-\mathbf{y}\|\geq\varepsilon\right\}\,.$$

We then define the one-particle density as follows

$$p\left(t,\mathbf{x},\mathbf{v}\right) = \int_{\Omega^{(2)}(\mathbf{x})}\int_V P\left(t,\mathbf{x},\mathbf{y},\mathbf{v},\mathbf{u}\right)\,\mathrm{d}\mathbf{u}\,\mathrm{d}\mathbf{y}\,.$$

Integrating each component **(i)**–**(vi)** in (5.7) individually, we can derive the one-particle transport equation.

(i): We can switch the integration and the derivative to obtain

$$\int_{\Omega^{(2)}(\mathbf{x})}\int_V \frac{\partial P}{\partial t}\,\mathrm{d}\mathbf{u}\,\mathrm{d}\mathbf{y} = \frac{\partial p}{\partial t}\,.$$

[1]For improved readability, we change the notation to $\mathbf{x}=\mathbf{x}^{(1)}$, $\mathbf{y}=\mathbf{x}^{(2)}$, $\mathbf{v}=\mathbf{v}^{(1)}$, $\mathbf{u}=\mathbf{v}^{(2)}$.

(ii): We use Reynolds' transport theorem in space to obtain

$$\int_{\Omega^{(2)}(\mathbf{x})} \int_V \mathbf{v} \cdot \nabla_{\mathbf{x}} P \, \mathrm{d}\mathbf{u} \, \mathrm{d}\mathbf{y} = \mathbf{v} \cdot \nabla_{\mathbf{x}} p - \int_{\partial B_\varepsilon(\mathbf{x})} \int_V (\mathbf{v} \cdot \mathbf{n}) \, P(t, \mathbf{x}, \mathbf{y}, \mathbf{v}, \mathbf{u}) \, \mathrm{d}\mathbf{u} \, \mathrm{d}\mathbf{y} \,,$$

where $B_\varepsilon(\mathbf{x})$ denotes the ball around $\mathbf{x}$ with radius ε as defined in (5.4) and $\mathbf{n}$ is the outwards pointing normal vector. Note that in this case outwards means out of $\Omega^{(2)}(\mathbf{x})$, hence $\mathbf{n}$ in fact points into the ball $B_\varepsilon(\mathbf{x})$, i.e. it can be written as

$$\mathbf{n} = \frac{\mathbf{x} - \mathbf{y}}{\|\mathbf{x} - \mathbf{y}\|} \,. \tag{5.10}$$

(iii): Using the divergence theorem, we obtain

$$\int_{\Omega^{(2)}(\mathbf{x})} \int_V \mathbf{u} \cdot \nabla_{\mathbf{x}} P \, (t, \mathbf{x}, \mathbf{y}, \mathbf{v}, \mathbf{u}) \, \mathrm{d}\mathbf{u} \, \mathrm{d}\mathbf{y} = \int_{\partial\Omega \cup \partial B_\varepsilon(\mathbf{x})} \int_V (\mathbf{u} \cdot \mathbf{n}) P(t, \mathbf{x}, \mathbf{y}, \mathbf{v}, \mathbf{u}) \, \mathrm{d}\mathbf{u} \, \mathrm{d}\mathbf{y} \,,$$

where $\mathbf{n}$ is again the outwards pointing normal vector that for the boundary part $\partial B_\varepsilon(\mathbf{x})$ is given in (5.10). Using the boundary conditions along the wall $\partial\Omega$ given in (5.8) we can show that

$$\int_{\partial\Omega} \int_V (\mathbf{u} \cdot \mathbf{n}) P(t, \mathbf{x}, \mathbf{y}, \mathbf{v}, \mathbf{u}) \, \mathrm{d}\mathbf{u} \, \mathrm{d}\mathbf{y} = 0 \,.$$

(iv): One can simply integrate to obtain

$$-2\lambda \int_{\Omega^{(2)}(\mathbf{x})} \int_V P \, (t, \mathbf{x}, \mathbf{y}, \mathbf{v}, \mathbf{u}) \, \mathrm{d}\mathbf{u} \, \mathrm{d}\mathbf{y} = -2\lambda p(t, \mathbf{x}, \mathbf{v}) \,.$$

(v): Switching the order of integration, we obtain

$$\int_{\Omega^{(2)}(\mathbf{x})} \int_V \lambda \int_V T(\mathbf{v}, \mathbf{v}_*) \, P \, (t, \mathbf{x}, \mathbf{y}, \mathbf{v}_*, \mathbf{u}) \, \mathrm{d}\mathbf{v}_* \, \mathrm{d}\mathbf{u} \, \mathrm{d}\mathbf{y} = \lambda \int_V T(\mathbf{v}, \mathbf{v}_*) \, p \, (t, \mathbf{x}, \mathbf{v}_*) \, \mathrm{d}\mathbf{v}_* \,.$$

(vi): We can again switch the order of integration and use that $\int_V T(\mathbf{v}, \mathbf{u}) \, \mathrm{d}\mathbf{v} = 1$:

$$\begin{aligned}
&\int_{\Omega^{(2)}(\mathbf{x})} \int_V \lambda \int_V T(\mathbf{u}, \mathbf{u}_*) \, P \, (t, \mathbf{x}, \mathbf{y}, \mathbf{v}, \mathbf{u}_*) \, \mathrm{d}\mathbf{u}_* \, \mathrm{d}\mathbf{u} \, \mathrm{d}\mathbf{y} \\
&\qquad = \lambda \int_{\Omega^{(2)}(\mathbf{x})} \int_V \int_V T(\mathbf{u}, \mathbf{u}_*) \, \mathrm{d}\mathbf{u} \, P \, (t, \mathbf{x}, \mathbf{y}, \mathbf{v}, \mathbf{u}_*) \, \mathrm{d}\mathbf{u}_* \, \mathrm{d}\mathbf{y} \\
&\qquad = \lambda \int_{\Omega^{(2)}(\mathbf{x})} \int_V P \, (t, \mathbf{x}, \mathbf{y}, \mathbf{v}, \mathbf{u}_*) \, \mathrm{d}\mathbf{u}_* \, \mathrm{d}\mathbf{y} = \lambda p(t, \mathbf{x}, \mathbf{v}) \,.
\end{aligned}$$

Summing up the results in **(i)**–**(vi)**, the one-particle transport equation takes the form

$$\frac{\partial p}{\partial t} + \mathbf{v}\cdot\nabla_{\mathbf{x}} p = -\lambda p + \lambda \int_V T(\mathbf{v},\mathbf{v}_*)\, p(t,\mathbf{x},\mathbf{v}_*)\,\mathrm{d}\mathbf{v}_* \\ + \int_{\partial B_\varepsilon(\mathbf{x})} \int_V P(t,\mathbf{x},\mathbf{y},\mathbf{v},\mathbf{u})\,[\mathbf{n}\cdot(\mathbf{v}-\mathbf{u})]\,\mathrm{d}\mathbf{u}\,\mathrm{d}\mathbf{y}\,,$$

where $\mathbf{n} = (\mathbf{x}-\mathbf{y})/\varepsilon$ is a normal vector. Inverting from $\mathbf{n}$ to $-\mathbf{n}$, $\mathbf{y}$ can be written as $\mathbf{y} = \mathbf{x} + \varepsilon\mathbf{n}$ and we can transform the integral over the surface of the ball $B_\varepsilon(\mathbf{x})$ into an integral over the unit sphere in d dimensions

$$\frac{\partial p}{\partial t} + \mathbf{v}\cdot\nabla_{\mathbf{x}} p = -\lambda p + \lambda \int_V T(\mathbf{v},\mathbf{v}_*)\, p(t,\mathbf{x},\mathbf{v}_*)\,\mathrm{d}\mathbf{v}_* \\ - \varepsilon^{d-1} \int_{\mathbb{S}^{d-1}} \int_V P(t,\mathbf{x},\mathbf{x}+\varepsilon\mathbf{n},\mathbf{v},\mathbf{u})\,[\mathbf{n}\cdot(\mathbf{v}-\mathbf{u})]\,\mathrm{d}\mathbf{u}\,\mathrm{d}\mathbf{n}\,,$$

where the sign of the correction term changes because of the flip from $\mathbf{n}$ to $-\mathbf{n}$. Because the influence of collisions of more than two particles is negligible as discussed, we can generalise this equation for N particles by simply adding up the influences of each of the other $(N-1)$ particles and we obtain

$$\frac{\partial p}{\partial t} + \mathbf{v}\cdot\nabla_{\mathbf{x}} p = -\lambda p + \lambda \int_V T(\mathbf{v},\mathbf{v}_*)\, p(t,\mathbf{x},\mathbf{v}_*)\,\mathrm{d}\mathbf{v}_* \\ - \varepsilon^{d-1}(N-1) \int_{\mathbb{S}^{d-1}} \int_V P(t,\mathbf{x},\mathbf{x}+\varepsilon\mathbf{n},\mathbf{v},\mathbf{u})\,[\mathbf{n}\cdot(\mathbf{v}-\mathbf{u})]\,\mathrm{d}\mathbf{u}\,\mathrm{d}\mathbf{n}\,. \tag{5.11}$$

We find again the parameter κ as defined in (5.5) that we had already seen in the number of collisions. In order to analyse this equation further, we define the two subsets of $\mathbb{S}$

$$\mathbb{S}^{d-1}_+(\mathbf{a}) = \{\mathbf{n}\in\mathbb{S} \;:\; \mathbf{n}\cdot\mathbf{a} > 0\}\,, \qquad \mathbb{S}^{d-1}_-(\mathbf{a}) = \{\mathbf{n}\in\mathbb{S} \;:\; \mathbf{n}\cdot\mathbf{a} < 0\}\,,$$

and we will for the remainder of this chapter write $\mathbb{S}^{d-1}_+$ and $\mathbb{S}^{d-1}_-$ as always meaning $\mathbb{S}^{d-1}_+(\mathbf{v}-\mathbf{u})$ and $\mathbb{S}^{d-1}_-(\mathbf{v}-\mathbf{u})$. We can now split the integral in the transport equation (5.11) and apply the boundary conditions given in (5.9). We obtain

$$\begin{aligned} &-\kappa \int_{\mathbb{S}^{d-1}} \int_V P(t,\mathbf{x},\mathbf{x}+\varepsilon\mathbf{n},\mathbf{v},\mathbf{u})\,[\mathbf{n}\cdot(\mathbf{v}-\mathbf{u})]\,\mathrm{d}\mathbf{u}\,\mathrm{d}\mathbf{n} \\ &\quad = -\kappa \int_{\mathbb{S}^{d-1}_+} \int_V P(t,\mathbf{x},\mathbf{x}+\varepsilon\mathbf{n},\mathbf{v},\mathbf{u})\,[\mathbf{n}\cdot(\mathbf{v}-\mathbf{u})]\,\mathrm{d}\mathbf{u}\,\mathrm{d}\mathbf{n} \\ &\qquad -\kappa \int_{\mathbb{S}^{d-1}_-} \int_V P(t,\mathbf{x},\mathbf{x}+\varepsilon\mathbf{n},\mathbf{v},\mathbf{u})\,[\mathbf{n}\cdot(\mathbf{v}-\mathbf{u})]\,\mathrm{d}\mathbf{u}\,\mathrm{d}\mathbf{n} \\ &\quad = \kappa \int_{\mathbb{S}^{d-1}_+} \int_V \left[P(t,\mathbf{x},\mathbf{x}-\varepsilon\mathbf{n},\mathbf{v}',\mathbf{u}') - P(t,\mathbf{x},\mathbf{x}+\varepsilon\mathbf{n},\mathbf{v},\mathbf{u})\right][\mathbf{n}\cdot(\mathbf{v}-\mathbf{u})]\,\mathrm{d}\mathbf{u}\,\mathrm{d}\mathbf{n}\,. \end{aligned}$$

Plugging this into (5.11), we end up with

$$\frac{\partial p}{\partial t} + \mathbf{x} \cdot \nabla_{\mathbf{x}} p = -\lambda p + \lambda \int_V T(\mathbf{v}, \mathbf{u})\, p(t, \mathbf{x}, \mathbf{u})\, \mathrm{d}\mathbf{u} \\ + \kappa \int_{\mathbb{S}_+^{d-1}} \int_V \left[P(t, \mathbf{x}, \mathbf{x} - \varepsilon \mathbf{n}, \mathbf{v}', \mathbf{u}') - P(t, \mathbf{x}, \mathbf{x} + \varepsilon \mathbf{n}, \mathbf{v}, \mathbf{u}) \right] [\mathbf{n} \cdot (\mathbf{v} - \mathbf{u})]\, \mathrm{d}\mathbf{u}\, \mathrm{d}\mathbf{n} \,. \tag{5.12}$$

The problem one is facing at this point is that this equation still contains the two-particle density function P, which is unknown. We will use the following two sections to discuss different approaches to resolve this issue.

5.5 The Boltzmann collision integral

The first approach developed by Boltzmann [26] uses the molecular chaos assumption and considers the system in the dilute gas limit given through

$$N \to \infty \,, \qquad \varepsilon \to 0 \,, \qquad N \varepsilon^{d-1} = \kappa \,.$$

Note that in this limit the gas is dilute in the sense that the volume fraction c vanishes [27]. The additional molecular chaos assumption states that velocities are locally independent of each other, and we can write [26]

$$P(t, \mathbf{x}, \mathbf{y}, \mathbf{v}, \mathbf{u}) = p(t, \mathbf{x}, \mathbf{v})\, p(t, \mathbf{y}, \mathbf{u}) \,, \qquad \mathbf{x}, \mathbf{y} \in \Omega \,, \mathbf{v}, \mathbf{u} \in V \,.$$

Plugging this into (5.12), we obtain an equation that contains the so-called *Boltzmann integral* as last term [26]

$$\frac{\partial p}{\partial t} + \mathbf{v} \cdot \nabla_{\mathbf{x}} p = -\lambda p + \lambda \int_V T(\mathbf{v}, \mathbf{v}_*)\, p(t, \mathbf{x}, \mathbf{v}_*)\, \mathrm{d}\mathbf{v}_* \\ + \kappa \int_V \int_{\mathbb{S}_+^{d-1}} \left(p(t, \mathbf{x}, \mathbf{v}')\, p(t, \mathbf{x}, \mathbf{u}') - p(t, \mathbf{x}, \mathbf{v})\, p(t, \mathbf{x}, \mathbf{u}) \right) [\mathbf{n} \cdot (\mathbf{v} - \mathbf{u})]\, \mathrm{d}\mathbf{n}\, \mathrm{d}\mathbf{u} \,. \tag{5.13}$$

For the remainder of this chapter we will concentrate on a two-dimensional environment, which helps evaluating many of the integral terms present. The general ideas could be applied for $d = 3$, but the evaluation of the integrals might prove significantly more difficult.

5.5.1 Numerical solution

Let us now derive a way of achieving a numerical solution for the velocity jump equation with a Boltzmann-like term. In the first step, we reformulate (5.13) for

angular probability distributions $p(t, \mathbf{x}, \theta) = s\, p\left(t, \mathbf{x}, s \begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix}\right)$. Plugging this into (5.13) and accounting for the changes in integrals from V to $\mathbb{S}^1$, we get [2]

$$\begin{aligned} \underbrace{\frac{\partial p(\theta)}{\partial t}}_{(i)} + \underbrace{s \begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix} \cdot \nabla_{\mathbf{x}} p(\theta)}_{(ii)} = \underbrace{-\lambda p(\theta)}_{(iii)} + \underbrace{\lambda \int_0^{2\pi} T(\theta, \phi) p(\phi)\, \mathrm{d}\phi}_{(iv)} \\ + \kappa s \int_0^{2\pi} \int_{\mathbb{S}^1_+} \underbrace{(p(\theta')\, p(\phi')}_{(v)} \underbrace{- p(\theta)\, p(\phi))}_{(vi)} \mathbf{n} \cdot \left(\begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix} - \begin{pmatrix} \cos\phi \\ \sin\phi \end{pmatrix} \right) \mathrm{d}\mathbf{n}\, \mathrm{d}\phi\,. \end{aligned} \tag{5.14}$$

Let us now discretise the angle space into M intervals $[\frac{2i-1}{2}\Delta\theta, \frac{2i+1}{2}\Delta\theta]$ around the central velocity directions $\theta_i = i\Delta\theta$, $i = 0, \ldots, M-1$, where $\Delta\theta = 2\pi/M$. We use this to discretise the probability distribution using the definition

$$p_i(t, \mathbf{x}) = \int_{\frac{2i-1}{2}\Delta\theta}^{\frac{2i+1}{2}\Delta\theta} p(t, \mathbf{x}, \theta)\, \mathrm{d}\theta\,. \tag{5.15}$$

Additionally, for the evaluation of the Boltzmann integral we approximate $p(\theta)$ by the piecewise constant function

$$p(\theta) = \frac{p_i}{\Delta\theta}\,, \qquad \text{for } \theta \in \left[\frac{2i-1}{2}\Delta\theta, \frac{2i+1}{2}\Delta\theta\right], \tag{5.16}$$

where we also imply the periodicity of $p(\theta)$. We integrate (5.14) with respect to $\theta \in \left[\frac{2i-1}{2}\Delta\theta, \frac{2i+1}{2}\Delta\theta\right]$ and individually evaluate the results.

(i): Switching the integral and the derivative results in

$$\int_{\frac{2i-1}{2}\Delta\theta}^{\frac{2i+1}{2}\Delta\theta} \frac{\partial p(\theta)}{\partial t}\, \mathrm{d}\theta = \frac{\partial p_i}{\partial t}\,.$$

(ii): We, again, switch the integral and the derivative and apply the piecewise constant approximation of $p(\theta)$ given in (5.16) to obtain

$$\begin{aligned} \int_{\frac{2i-1}{2}\Delta\theta}^{\frac{2i+1}{2}\Delta\theta} s \begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix} \cdot \nabla p(\theta)\, \mathrm{d}\theta &= s \int_{\frac{2i-1}{2}\Delta\theta}^{\frac{2i+1}{2}\Delta\theta} \begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix} \mathrm{d}\theta \cdot \nabla_{\mathbf{x}} p_i \\ &= s \begin{pmatrix} \cos\theta_i \\ \sin\theta_i \end{pmatrix} \cdot \nabla_{\mathbf{x}} p_i \frac{\sin(\Delta\theta/2)}{\Delta\theta/2}\,. \end{aligned}$$

(iii): Via the definition of p_i given in (5.15), we can directly integrate and obtain

$$\int_{\frac{2i-1}{2}\Delta\theta}^{\frac{2i+1}{2}\Delta\theta} \lambda\, p(\theta)\, \mathrm{d}\theta = \lambda p_i\,.$$

[2] To improve readability, we drop the t and $\mathbf{x}$ arguments from the one particle probability distribution $p(t, \mathbf{x}, \theta)$.

(iv): As we are considering an unbiased velocity jump process, we have $T(\theta,\phi) = 1/2\pi$. Using this along with the fact that $M\Delta\theta = 2\pi$ and the definition of p_i given in (5.15), we obtain

$$\int_{\frac{2i-1}{2}\Delta\theta}^{\frac{2i+1}{2}\Delta\theta} \lambda \int_0^{2\pi} T(\theta,\phi)\, p(\phi)\, \mathrm{d}\phi\, \mathrm{d}\theta = \frac{\lambda}{M} \sum_{i=0}^{M-1} p_i\,.$$

(vi): To keep notation consistent, we define

$$I_{(vi)} = -\kappa s \int_{\frac{2i-1}{2}\Delta\theta}^{\frac{2i+1}{2}\Delta\theta} \int_0^{2\pi} \int_{\mathbb{S}^1_+} p(\theta)\, p(\phi) \left(\begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix} - \begin{pmatrix} \cos\phi \\ \sin\phi \end{pmatrix} \right) \cdot \mathbf{n}\, \mathrm{d}\mathbf{n}\, \mathrm{d}\theta\, \mathrm{d}\phi\,.$$

We can directly evaluate the innermost integral $\int_{\mathbb{S}^1_+} \mathbf{n}\, \mathrm{d}\mathbf{n}$ using (B.1) from Lemma B.1 and get

$$I_{(vi)} = -2\kappa s \int_{\frac{2i-1}{2}\Delta\theta}^{\frac{2i+1}{2}\Delta\theta} \int_0^{2\pi} p(\theta)\, p(\phi) \left\| \begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix} - \begin{pmatrix} \cos\phi \\ \sin\phi \end{pmatrix} \right\| \mathrm{d}\theta\, \mathrm{d}\phi\,.$$

Using (5.16), we can write this as

$$\begin{aligned} I_{(vi)} &= -2\kappa s p_i \sum_{j=0}^{M-1} \frac{p_j}{(\Delta\theta)^2} \int_{\frac{2i-1}{2}\Delta\theta}^{\frac{2i+1}{2}\Delta\theta} \int_{\frac{2j-1}{2}\Delta\theta}^{\frac{2j+1}{2}\Delta\theta} \left\| \begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix} - \begin{pmatrix} \cos\phi \\ \sin\phi \end{pmatrix} \right\| \mathrm{d}\theta\, \mathrm{d}\phi \\ &= -\kappa s p_i \sum_{j=0}^{M-1} B_{i,j}\, p_j\,. \end{aligned}$$

We can evaluate $B_{i,j}$ by making the observation

$$\begin{aligned} \left\| \begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix} - \begin{pmatrix} \cos\phi \\ \sin\phi \end{pmatrix} \right\| &= \sqrt{2 - 2 \begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix} \cdot \begin{pmatrix} \cos\phi \\ \sin\phi \end{pmatrix}} \\ &= 2\sqrt{\frac{1-\cos(\theta-\phi)}{2}} = 2\left|\sin\left(\frac{\theta-\phi}{2}\right)\right|. \end{aligned}$$

For $i \neq j$ the integral can then be explicitly evaluated making use of the fact that $\theta - \phi$ does not change sign throughout the region of integration. We get

$$B_{i,j} = 4 \left(\frac{\sin(\Delta\theta/4)}{\Delta\theta/4} \right)^2 \left| \sin\left(\frac{\theta_i - \theta_j}{2} \right) \right|. \tag{5.17}$$

For $i = j$, one has to be aware that the sign of $\theta - \phi$ changes inside the region of integration. The result is

$$B_{i,i} = \frac{16}{\Delta\theta} \left[1 - \frac{\sin(\Delta\theta/2)}{\Delta\theta/2} \right]. \tag{5.18}$$

(v): The remaining part of the Boltzmann term is

$$I_{(v)} = \kappa s \int_{\frac{2i-1}{2}\Delta\theta}^{\frac{2i+1}{2}\Delta\theta} \int_0^{2\pi} \int_{\mathbb{S}^1_+} p(\theta')\, p(\phi') \left(\begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix} - \begin{pmatrix} \cos\phi \\ \sin\phi \end{pmatrix} \right) \cdot \mathbf{n}\, \mathrm{d}\mathbf{n}\, \mathrm{d}\phi\, \mathrm{d}\theta\,,$$

where the reflected direction θ' can be calculated as

$$\theta' = -\theta + 2\gamma - \pi\,,$$

with γ being the angle between the x-axis and $\mathbf{n}$. Using this, one can calculate γ for a given pair (θ, θ'):

$$\gamma_\pm = \frac{\theta' + \theta \pm \pi}{2}\,,$$

where two values for γ are possible, because using $-\mathbf{n}$ instead of $\mathbf{n}$ does not change its reflection property as can be seen in the definition of $\mathbf{v}'$ in (5.1). Because of this, it is also clear that for each pair (θ, θ') one of the two vectors $\mathbf{n}$ and $-\mathbf{n}$ is inside $\mathbb{S}^1_+$. We can therefore change the innermost integral to be with respect to θ' instead of $\mathbf{n}$ as follows

$$I_{(v)} = \frac{1}{2}\kappa s \int_{\frac{2i-1}{2}\Delta\theta}^{\frac{2i+1}{2}\Delta\theta} \int_0^{2\pi} \int_0^{2\pi} p(\theta')\, p(\phi') \left| \left(\begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix} - \begin{pmatrix} \cos\phi \\ \sin\phi \end{pmatrix} \right) \cdot \begin{pmatrix} \cos\gamma_+ \\ \sin\gamma_+ \end{pmatrix} \right| \mathrm{d}\theta'\, \mathrm{d}\phi'\, \mathrm{d}\theta\,,$$

where the choice of γ_+ over γ_- is arbitrary and does not change the outcome. We can also change the two angles θ and ϕ inside the integral to θ' and ϕ' without changing the value of the integral as $(\mathbf{v} - \mathbf{u}) \cdot \mathbf{n} = -(\mathbf{v}' - \mathbf{u}') \cdot \mathbf{n}$. We can then write $I_{(v)}$ as

$$I_{(v)} = \kappa s \sum_{i,j=0}^{M-1} A_{i,j,k}\, p_j\, p_k\,,$$

where

$$A_{i,j,k} = \frac{1}{2(\Delta\theta)^2} \int_{\frac{2i-1}{2}\Delta\theta}^{\frac{2i+1}{2}\Delta\theta} \int_{\frac{2j-1}{2}\Delta\theta}^{\frac{2j+1}{2}\Delta\theta} \int_{\frac{2k-1}{2}\Delta\theta}^{\frac{2k+1}{2}\Delta\theta} \left| \left(\begin{pmatrix} \cos\theta' \\ \sin\theta' \end{pmatrix} - \begin{pmatrix} \cos\phi' \\ \sin\phi' \end{pmatrix} \right) \cdot \begin{pmatrix} \cos\gamma_+ \\ \sin\gamma_+ \end{pmatrix} \right| \mathrm{d}\theta'\, \mathrm{d}\phi'\, \mathrm{d}\theta\,.$$

We can simplify this to

$$A_{i,j,k} = \frac{1}{(\Delta\theta)^2} \int_{\frac{2k-1}{2}\Delta\theta}^{\frac{2k+1}{2}\Delta\theta} \underbrace{\int_{\frac{2i-1}{2}\Delta\theta}^{\frac{2i+1}{2}\Delta\theta} \left| \cos\left(\frac{\phi' - \theta}{2} \right) \right| \mathrm{d}\theta}_{I_{(v,1)}} \underbrace{\int_{\frac{2j-1}{2}\Delta\theta}^{\frac{2j+1}{2}\Delta\theta} \left| \sin\left(\frac{\phi' - \theta'}{2} \right) \right| \mathrm{d}\theta'}_{I_{(v,2)}}\, \mathrm{d}\phi'\,. \tag{5.19}$$

We can evaluate $I_{(v,1)}$ and $I_{(v,2)}$ further:

$$I_{(v,1)} = \begin{cases} 4\sin(\Delta\theta/4)\left|\cos\left(\frac{\phi'-\theta_i}{2}\right)\right|, & 2|i-k| \neq M, \\ 4\left(1-\cos(\Delta\theta/4)\cos\left(\frac{\phi'-\theta_k}{2}\right)\right), & 2|i-k| = M, \end{cases}$$

$$I_{(v,2)} = \begin{cases} 4\sin(\Delta\theta/4)\left|\sin\left(\frac{\phi'-\theta_j}{2}\right)\right|, & j \neq k, \\ 4\left(1-\cos(\Delta\theta/4)\cos\left(\frac{\phi'-\theta_k}{2}\right)\right), & j = k. \end{cases}$$

One could possibly evaluate the remaining integrals further, but the additional value of this is reduced seeing that a numerical evaluation of $A_{i,j,k}$ at the beginning of the simulation can easily be performed using the current form. The equation for the numerical solution therefore takes the final form

$$\begin{aligned} &\frac{\partial p_i}{\partial t} + s\begin{pmatrix}\cos\theta_i \\ \sin\theta_i\end{pmatrix}\cdot\nabla_{\mathbf{x}} p_i \,\frac{\sin(\Delta\theta/2)}{\Delta\theta/2} \\ &\qquad = -\lambda p_i + \frac{\lambda}{M}\sum_{j=0}^{M-1} p_j + \kappa s\left(\sum_{j,k=0}^{M-1} A_{i,j,k}\, p_j\, p_k - p_i \sum_{j=0}^{M-1} B_{i,j}\, p_j\right), \end{aligned} \tag{5.20}$$

where $B_{i,j}$ is defined in (5.17)–(5.18) and $A_{i,j,k}$ in (5.19). In Section 5.7, we will compare this numerical solution along with other models to KMC simulations.

5.5.2 Cattaneo approximation

In order to study the effect the Boltzmann collision term has on the effective diffusion, we derive a Cattaneo approximation [86] of equation (5.13). The concept of Cattaneo approximations and the definitions of the velocity moments $m^{(0)}$ and $\mathbf{m}^{(1)}$ were introduced in Section 1.3.2 and will be used throughout the reminder of this and the following section. We can derive the equation for the zeroth moment by integrating (5.13) with respect to $\mathbf{v} \in V$. Due to symmetry in $\mathbf{u}$ and $\mathbf{v}$ the Boltzmann collision term vanishes in this equation and we obtain the same conservation of mass property as in Section 1.3.2:

$$\frac{\partial m^{(0)}}{\partial t} + \nabla_{\mathbf{x}}\cdot\mathbf{m}^{(1)} = 0.$$

Multiplying (5.13) with $\mathbf{v}$ and then integrating with respect to $\mathbf{v} \in V$, the only potential difference to the classical Cattaneo approximation presented in Section 1.3.2 is the Boltzmann collision term that does not necessarily vanish. Let us therefore investigate the integral

$$I = \int_V\int_V\int_{\mathbb{S}^1_+} \mathbf{v}\,\underbrace{[p(\mathbf{v}')\,p(\mathbf{u}')}_{I_1} - \underbrace{p(\mathbf{v})\,p(\mathbf{u})]}_{I_2}\,(\mathbf{v}-\mathbf{u})\cdot\mathbf{n}\,\mathrm{d}\mathbf{n}\,\mathrm{d}\mathbf{u}\,\mathrm{d}\mathbf{v}.$$

We will investigate the part I_1 a bit more closely. We can first perform a change of variables from $(\mathbf{v}, \mathbf{u}, \mathbf{n})$ to $(\mathbf{v}', \mathbf{u}', \mathbf{n})$ using the fact that $(\cdot)' : V \mapsto V$ presents a bijection from V onto itself as well as the fact that $\mathbf{v}' \cdot \mathbf{n} = -\mathbf{v} \cdot \mathbf{n}$:

$$\begin{aligned} I_1 &= \int_V \int_V \int_{\mathbb{S}^1_+} \mathbf{v}\, p(\mathbf{v}')\, p(\mathbf{u}')\, (\mathbf{v} - \mathbf{u}) \cdot \mathbf{n} \, \mathrm{d}\mathbf{n} \, \mathrm{d}\mathbf{u} \, \mathrm{d}\mathbf{v} \\ &= -\int_V \int_V \int_{\mathbb{S}^1_+} (\mathbf{v}')' p(\mathbf{v}')\, p(\mathbf{u}')\, (\mathbf{v}' - \mathbf{u}') \cdot \mathbf{n} \, \mathrm{d}\mathbf{n} \, \mathrm{d}\mathbf{u} \, \mathrm{d}\mathbf{v} \\ &= -\int_V \int_V \int_{\mathbb{S}^1_-} (\mathbf{v}')' p(\mathbf{v}')\, p(\mathbf{u}')\, (\mathbf{v}' - \mathbf{u}') \cdot \mathbf{n} \, \mathrm{d}\mathbf{n} \, \mathrm{d}\mathbf{u}' \, \mathrm{d}\mathbf{v}' \\ &= \int_V \int_V \int_{\mathbb{S}^1_+} \mathbf{v}' p(\mathbf{v})\, p(\mathbf{u})\, (\mathbf{v} - \mathbf{u}) \cdot \mathbf{n} \, \mathrm{d}\mathbf{n} \, \mathrm{d}\mathbf{u} \, \mathrm{d}\mathbf{v} \,, \end{aligned} \tag{5.21}$$

where we drop the primes in the last step for convenience. We also change the sign by integrating over $\mathbb{S}^1_+$ instead of $\mathbb{S}^1_-$, which is possible because for the reflective collisions defined in (5.1) (and indeed for perfectly elastic collisions given in (5.2)) $\mathbf{v}'$ is invariant under the change $\mathbf{n} \to -\mathbf{n}$. Integral I now takes the form

$$I = \int_V \int_V \int_{\mathbb{S}^1_+} (\mathbf{v}' - \mathbf{v})\, p(\mathbf{v})\, p(\mathbf{u})\, (\mathbf{v} - \mathbf{u}) \cdot \mathbf{n} \, \mathrm{d}\mathbf{n} \, \mathrm{d}\mathbf{u} \, \mathrm{d}\mathbf{v} \,.$$

For the reflective collisions defined in (5.1) we have $\mathbf{v}' - \mathbf{v} = -2(\mathbf{v} \cdot \mathbf{n})\mathbf{n}$. Using Lemma B.2, we obtain for I

$$I = -\frac{4}{3} \int_V \int_V \left(\frac{(\mathbf{v} - \mathbf{u}) \cdot \mathbf{v}}{\|\mathbf{v} - \mathbf{u}\|} (\mathbf{v} - \mathbf{u}) + \|\mathbf{v} - \mathbf{u}\| \, \mathbf{v} \right) p(\mathbf{v}) p(\mathbf{u}) \, \mathrm{d}\mathbf{u} \, \mathrm{d}\mathbf{v} \,.$$

Using the definition of the Euclidean norm of a vector, we obtain

$$\|\mathbf{v} - \mathbf{u}\|^2 = (\mathbf{v} - \mathbf{u}) \cdot (\mathbf{v} - \mathbf{u}) = 2(s^2 - \mathbf{v} \cdot \mathbf{u}) = 2(\mathbf{v} - \mathbf{u}) \cdot \mathbf{v} \,.$$

Hence,

$$\begin{aligned} I &= -\frac{4}{3} \int_V \int_V \left(\frac{3}{2} \|\mathbf{v} - \mathbf{u}\| \, \mathbf{v} - \frac{1}{2} \|\mathbf{v} - \mathbf{u}\| \, \mathbf{u} \right) p(\mathbf{v})\, p(\mathbf{u}) \, \mathrm{d}\mathbf{u} \, \mathrm{d}\mathbf{v} \\ &= -\frac{4}{3} \int_V \int_V \|\mathbf{v} - \mathbf{u}\| \, \mathbf{v}\, p(\mathbf{v})\, p(\mathbf{u}) \, \mathrm{d}\mathbf{u} \, \mathrm{d}\mathbf{v} \,, \end{aligned}$$

where we use symmetry between $\mathbf{v}$ and $\mathbf{u}$ in the second step. In order to evaluate this integral, we assume that $p(\mathbf{v})$ is close to an equilibrium, i.e. that we can write

$$p(\mathbf{v}) = \frac{m^{(0)}}{|V|} + \delta g(\mathbf{v}) \,, \text{with } \delta \ll 1 \text{ and } g(\mathbf{v}) \sim \mathcal{O}(1) \,. \tag{5.22}$$

We can plug this into the equation for I to obtain

$$
\begin{aligned}
I &= -\frac{4}{3}\int_V\int_V \|\mathbf{v}-\mathbf{u}\|\,\mathbf{v}\left(\frac{m^{(0)}}{|V|}+\delta g(\mathbf{v})\right)\left(\frac{m^{(0)}}{|V|}+\delta g(\mathbf{u})\right)\,\mathrm{d}\mathbf{u}\,\mathrm{d}\mathbf{v}\\
&\approx -\frac{4}{3}\left[\left(\frac{m^{(0)}}{|V|}\right)^2\int_V\int_V \|\mathbf{v}-\mathbf{u}\|\,\mathbf{v}\,\mathrm{d}\mathbf{u}\,\mathrm{d}\mathbf{v}+\delta\frac{m^{(0)}}{|V|}\int_V\int_V \|\mathbf{v}-\mathbf{u}\|\,\mathbf{v}\,(g(\mathbf{v})+g(\mathbf{u}))\,\mathrm{d}\mathbf{u}\,\mathrm{d}\mathbf{v}\right]\\
&= -\frac{4m_0}{3|V|}\delta\int_V\int_V \|\mathbf{v}-\mathbf{u}\|\,\mathbf{v}\,(g(\mathbf{v})+g(\mathbf{u}))\,\mathrm{d}\mathbf{u}\,\mathrm{d}\mathbf{v}\,,
\end{aligned}
$$

where the first integral vanishes due to the fact that $\int_V \mathbf{v}\,\mathrm{d}\mathbf{v}=\mathbf{0}$. Note that in the first step we have dropped terms of order δ^2, because they are negligible compared to the other terms. Using the following two integral equalities for all $\mathbf{v}\in V$:

$$
\int_V \|\mathbf{v}-\mathbf{u}\|\,\mathrm{d}\mathbf{u}=8s^2\,,\qquad \int_V \|\mathbf{v}-\mathbf{u}\|\,\mathbf{u}\,\mathrm{d}\mathbf{u}=-\frac{8s^2}{3}\mathbf{v}\,,
$$

we obtain for I (using $|V|=2\pi s$)

$$
\begin{aligned}
I &\approx \frac{4m_0}{3|V|}\left(8s^2-\frac{8s^2}{3}\right)\delta\int_V \mathbf{v}g(\mathbf{v})\,\mathrm{d}\mathbf{v}\\
&= \frac{4}{6\pi s}\frac{16s^2}{3}m^{(0)}\mathbf{m}^{(1)}\\
&= \frac{32s}{9\pi}m^{(0)}\mathbf{m}^{(1)}\,,
\end{aligned}
$$

where we use the fact that $\mathbf{m}^{(1)}=\int_V \mathbf{v}p(\mathbf{v})\,\mathrm{d}\mathbf{v}=\delta\int_V \mathbf{v}g(\mathbf{v})\,\mathrm{d}\mathbf{v}$. Hence, the second equation of the Cattaneo approximation takes the form

$$
\frac{\partial\mathbf{m}^{(1)}}{\partial t}+\frac{s^2}{d}\nabla_{\mathbf{x}}m^{(0)}=-\mathbf{m}^{(1)}\left(\lambda+s\kappa\frac{32}{9\pi}m^{(0)}\right)\,.
$$

We can apply parabolic scaling limits as described in [55] in order to obtain the effective diffusivity of the system to be

$$
D_{\text{eff}}\left(m^{(0)}\right)=\frac{s^2}{2\left(\lambda+s\kappa\frac{32}{9\pi}m^{(0)}\right)}<\frac{s^2}{2\lambda}=D_0\,.
$$

The way we will use this Cattaneo approximation is by formulating an alternative transport equation as follows

$$
\begin{aligned}
\frac{\partial p}{\partial t}-\mathbf{v}\cdot\nabla_{\mathbf{x}}p &= -\lambda_1 p+\lambda_1\int_V T(\mathbf{v},\mathbf{u})\,p(\mathbf{u})\,\mathrm{d}\mathbf{u}\,,\\
\lambda_1 &= \lambda+s\kappa\frac{32}{9\pi}\int_V p(\mathbf{v})\,\mathrm{d}\mathbf{v}\,.
\end{aligned}
\tag{5.23}
$$

This adjusted transport equation yields the same Cattaneo approximation as above and is therefore an ideal candidate to numerically study the derived correction terms in comparison with KMC simulations. We perform those numerical comparisons in Section 5.7.

5.6 Matched asymptotic expansion

The next step is to introduce a matched asymptotic expansion similar to what can be seen in [22]. Therefore, we extend the Boltzmann assumption to

$$P(t, \mathbf{x}, \mathbf{x} + \varepsilon\mathbf{n}, \mathbf{v}, \mathbf{u}) \sim p(t, \mathbf{x}, \mathbf{v}) \left[p(t, \mathbf{x}, \mathbf{u}) + \varepsilon\mathbf{n} \cdot \nabla_{\mathbf{x}} p(t, \mathbf{x}, \mathbf{u})\right] .$$

Starting from equation (5.12), we end up with a Boltzmann equation that has an additional correction term of $\mathcal{O}(\varepsilon)$:

$$-\kappa\varepsilon \int_V \int_{\mathbb{S}^1_+} \left[p(\mathbf{v}') \left(\mathbf{n} \cdot \nabla_{\mathbf{x}} p(\mathbf{u}')\right) + p(\mathbf{v}) \left(\mathbf{n} \cdot \nabla_{\mathbf{x}} p(\mathbf{u})\right)\right] (\mathbf{v} - \mathbf{u}) \cdot \mathbf{n} \, \mathrm{d}\mathbf{n} \, \mathrm{d}\mathbf{u} \, .$$

Again, multiplying by $\mathbf{v}$ and integrating with respect to $\mathbf{v} \in V$, we can derive the influence of this correction term on the Cattaneo approximation:

$$J = -\kappa\varepsilon \int_V \int_V \int_{\mathbb{S}^1_+} \mathbf{v} \left[p(\mathbf{v}') \left(\mathbf{n} \cdot \nabla_{\mathbf{x}} p(\mathbf{u}')\right) + p(\mathbf{v}) \left(\mathbf{n} \cdot \nabla_{\mathbf{x}} p(\mathbf{u})\right)\right] (\mathbf{v} - \mathbf{u}) \cdot \mathbf{n} \, \mathrm{d}\mathbf{n} \, \mathrm{d}\mathbf{u} \, \mathrm{d}\mathbf{v} \, .$$

Repeating the steps from (5.21) in Section 5.5, we arrive at

$$J = \kappa\varepsilon \int_V \int_V \int_{\mathbb{S}^1_+} (\mathbf{v}' - \mathbf{v}) \, p(\mathbf{v}) \left(\mathbf{n} \cdot \nabla_{\mathbf{x}} p(\mathbf{u})\right) (\mathbf{v} - \mathbf{u}) \cdot \mathbf{n} \, \mathrm{d}\mathbf{n} \, \mathrm{d}\mathbf{u} \, \mathrm{d}\mathbf{v} \, .$$

Substituting in the definition of $\mathbf{v}'$ in (5.1), we get

$$J = -2\kappa\varepsilon \int_V \int_V \int_{\mathbb{S}^1_+} \mathbf{n}(\mathbf{n} \cdot \mathbf{v}) \, p(\mathbf{v}) \left(\mathbf{n} \cdot \nabla_{\mathbf{x}} p(\mathbf{u})\right) (\mathbf{v} - \mathbf{u}) \cdot \mathbf{n} \, \mathrm{d}\mathbf{n} \, \mathrm{d}\mathbf{u} \, \mathrm{d}\mathbf{v} \, .$$

We can again address the integral with respect to $\mathbf{n} \in \mathbb{S}^1_+$ first using (B.4) from Lemma B.3 and obtain

$$\begin{aligned} J = &- \kappa\varepsilon \frac{\pi}{4} \int_V \int_V \; p(\mathbf{v}) \, (\mathbf{v} - \mathbf{u}) (\mathbf{v} \cdot \nabla_{\mathbf{x}} p(\mathbf{u})) \, \mathrm{d}\mathbf{u} \, \mathrm{d}\mathbf{v} \\ &- \kappa\varepsilon \frac{\pi}{4} \int_V \int_V \; p(\mathbf{v}) \, \nabla_{\mathbf{x}} p(\mathbf{u}) (\mathbf{v} \cdot (\mathbf{v} - \mathbf{u})) \, \mathrm{d}\mathbf{u} \, \mathrm{d}\mathbf{v} \\ &- \kappa\varepsilon \frac{\pi}{4} \int_V \int_V \; p(\mathbf{v}) \, \mathbf{v} \left((\mathbf{v} - \mathbf{u}) \cdot \nabla_{\mathbf{x}} p(\mathbf{u})\right) \mathrm{d}\mathbf{u} \, \mathrm{d}\mathbf{v} \, . \end{aligned}$$

This can be written in the form

$$J = -\kappa\varepsilon \frac{\pi}{4} \int_V \int_V p(\mathbf{v}) \; \left[\mathbf{v} \cdot (\mathbf{v} - \mathbf{u}) \, I + 2 \, \mathbf{v} \, \mathbf{v}^T - \mathbf{v} \, \mathbf{u}^T - \mathbf{u} \, \mathbf{v}^T\right] \nabla_{\mathbf{x}} p(\mathbf{u}) \, \mathrm{d}\mathbf{u} \, \mathrm{d}\mathbf{v} \, ,$$

where $I \in \mathbb{R}^{d \times d}$ denotes the identity matrix. Using again the approximation (5.22), we get

$$J \approx -\kappa\varepsilon \frac{\pi}{2} \mathbf{M}^{(2)} \nabla_{\mathbf{x}} m^{(0)} - \kappa\varepsilon \frac{\pi s^2}{4} m^{(0)} \nabla_{\mathbf{x}} m^{(0)} \, ,$$

where we drop terms of $\mathcal{O}(\delta^2)$. Plugging in the result (1.12) from the Cattaneo approximation, this can be approximated to

$$J \approx -\kappa\varepsilon\frac{\pi s^2}{2} m^{(0)} \nabla_{\mathbf{x}} m^{(0)} \,.$$

Plugging all the corrections into the second equation of the Cattaneo approximation, we arrive at

$$\frac{\partial \mathbf{m}^{(1)}}{\partial t} + \frac{s^2}{2}\nabla_{\mathbf{x}} m^{(0)} \left(1 + \kappa\varepsilon\pi m^{(0)}\right) = -\mathbf{m}^{(1)} \left(\lambda + s\kappa\frac{32}{9\pi} m^{(0)}\right) , \tag{5.24}$$

and therefore, using again the parabolic scaling limit [55], we get for the effective diffusivity

$$D_{\text{eff}}\left(m^{(0)}\right) = \frac{s^2\left(1 + \kappa\varepsilon\pi m^{(0)}\right)}{2\left(\lambda + s\kappa\frac{32}{9\pi} m^{(0)}\right)} \,. \tag{5.25}$$

We can see that depending on parameter values D_{eff} can be higher or lower than $D_0 = s^2/2\lambda$ and we can therefore explain a variety of different behaviours using this approach. Similarly to (5.23), we can again formulate an adjusted velocity jump process according to

$$\begin{aligned} \frac{\partial p}{\partial t} - \mathbf{v}\cdot\nabla_{\mathbf{x}} p &= -\lambda_2 p + \lambda_2 \int_V T(\mathbf{v},\mathbf{u})\, p(\mathbf{u})\, \mathrm{d}\mathbf{u} \,, \\ \lambda_2 &= \frac{\lambda + s\kappa\frac{32}{9\pi}\int_V p(\mathbf{v})\,\mathrm{d}\mathbf{v}}{1 + \kappa\varepsilon\pi\int_V p(\mathbf{v})\,\mathrm{d}\mathbf{v}} \,. \end{aligned} \tag{5.26}$$

Though this adjusted system does not lead to exactly the same Cattaneo approximation as in (5.24), the results are very close and we will compare (5.26) to KMC simulations in Section 5.7.

5.7 Numerical comparison between KMC simulation and PDE descriptions

In Sections 5.5 and 5.6, we have presented a total of three different models that we want to compare to KMC simulation results. The three models are given by **(i)** the Boltzmann-like equation (5.13) with its numerical solution given in (5.20), **(ii)** the first adjusted velocity jump model (5.23) that approximates the Boltzmann term, and **(iii)** the second adjusted velocity jump model (5.26) that was derived from matched asymptotic expansions. Models **(i)** and **(ii)** are valid only in the dilute gas limit, i.e. we can only expect those to compare well to KMC simulations for very small values

#	N	ε	κ	c
(A)	1001	4×10^{-3}	4	1.26×10^{-2}
(B)	201	2×10^{-2}	4	6.31×10^{-2}

Table 5.2: *Parameters for example simulations.*

of ε. Model **(iii)** on the other hand should give good comparisons even for larger values of ε.

The numerical solutions of all three models were achieved using a first order explicit finite volume scheme. In all simulations, we use a unit square domain $\Omega = [0,1] \times [0,1]$ with no-flow (reflective) boundary conditions everywhere. We discretise the velocity space into 40 velocity directions and use a gridsize of $\Delta x = 0.005$ and a time step of $\Delta t = 10^{-4}$. For KMC simulations, we start all particles uniformly distributed inside the circle $B_{0.25}(0.5, 0.5)$, i.e. inside the circle with radius 0.25 and centre in the middle of the domain. When initialising the particles, one has to ensure that they do not overlap, which can be achieved by resampling in case of an overlap. The corresponding initial condition for the PDE models is given in (5.3). In all simulations, we run the system until $t = 0.05$ and use the parameter values $\lambda = 200$, $s = 20$.

5.7.1 Distributions for two example simulations

In this section, we compare the three models with KMC simulations for the two test cases (A) and (B) as shown in Table 5.2. Notably, in both of these test cases we have $\kappa = 4$. As Model **(i)** as given in (5.13) only depends on the vale of κ and not otherwise on s or ε, this model will give the same result for both test cases (A) and (B) and we therefore only plot this result once. The same argument holds for Model **(ii)**. The distributions can be seen in Figure 5.4 for problem (A) and Figure 5.5 for problem (B). In Figures 5.4(e) and 5.5(c), we show horizontal slices through the relevant distributions at $x_2 = 0.5$.

For case (A), we can see that all four plotted distributions look very similar and in particular all three Models **(i)**–**(iii)** seem to give a good approximation to the KMC results. One can attribute this similarity to the fact that example (A) contains a very small particle diameter ε and therefore a small volume fraction, i.e. it is close to the Boltzmann limit, where Models **(i)** and **(ii)** are accurate. However, when looking at the slice in Figure 5.4(e), we can already see that Model **(iii)** shown as the dash-dotted (green) line gives a much better approximation to the KMC simulations than the other two models. Additionally, we can see that the results of Models **(i)** and **(ii)**

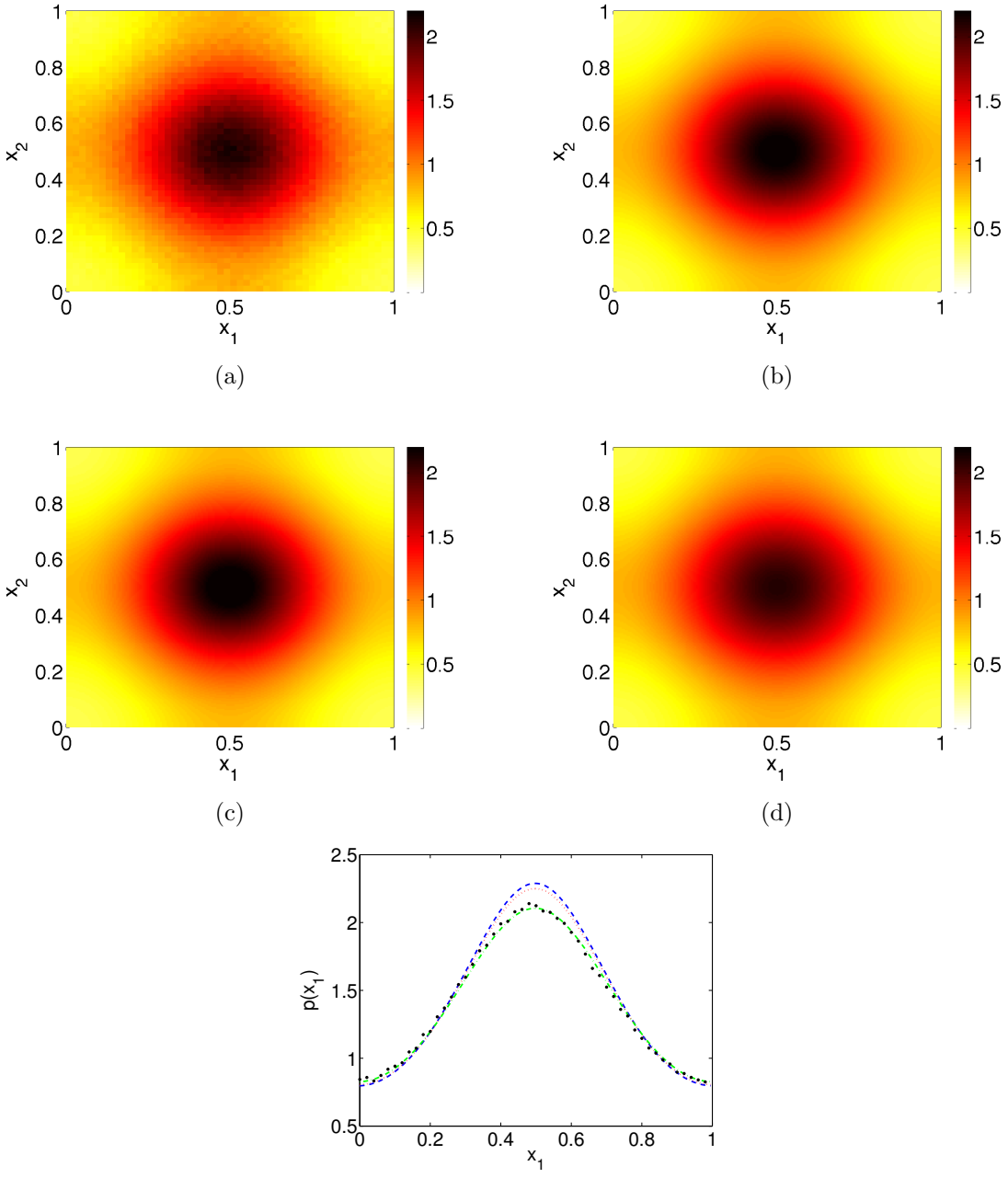


Figure 5.4: *Comparison between KMC simulation and numerical solutions of continuum approximations for the parameters* $N = 1001$, $\varepsilon = 0.004$ *and consequently* $\kappa = 4$. *We use the initial condition given in* (5.3), *zero-flux boundary conditions and plot distributions at time* $t = 0.05$. (a) *KMC simulation for* 1001 *particles of diameter* $\varepsilon = 0.004$. (b) *Numerical solution of Model* **(i)** *given by* (5.13). (c) *Numerical solution of Model* **(ii)** *given by* (5.23). (d) *Numerical solution of Model* **(iii)** *given by* (5.26). (e) *Slice through the distributions at* $x_2 = 0.5$. *Dashed (blue) line: Model* **(i)**; *dotted (red) line: Model* **(ii)**; *dash-dotted (green) line: Model* **(iii)**; *black circles: KMC simulation.*

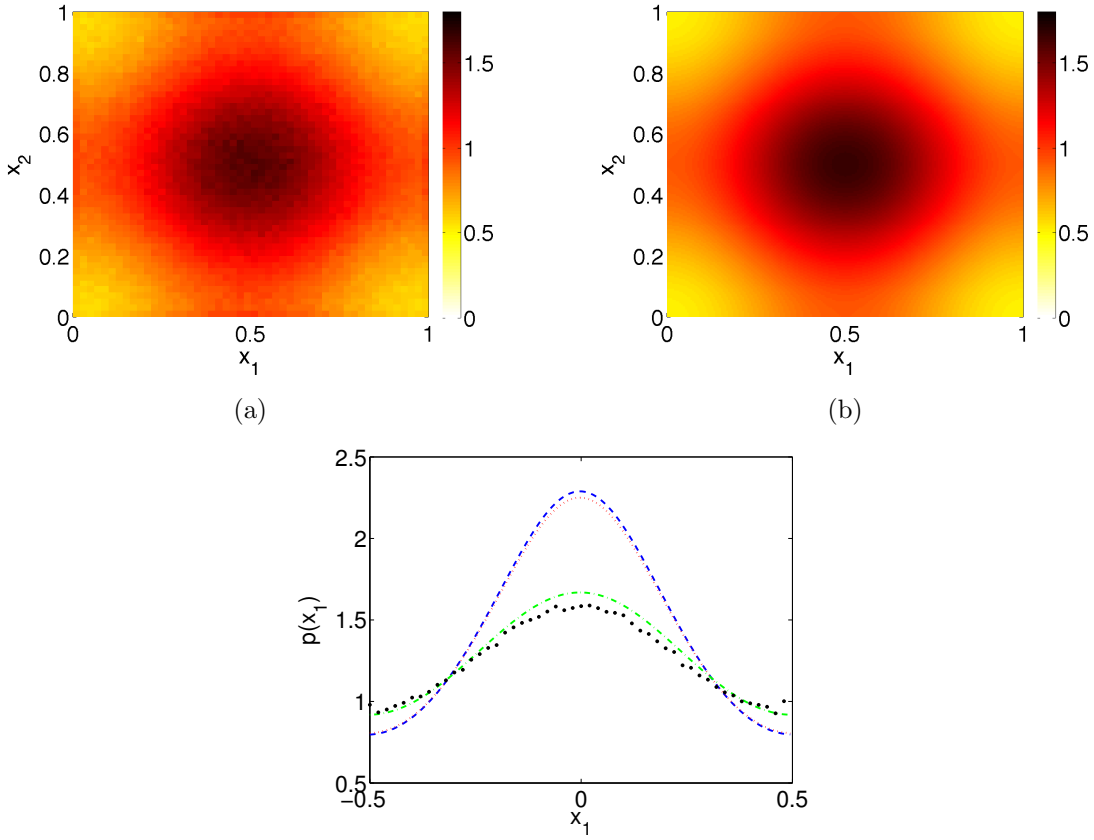


Figure 5.5: *Comparison between KMC simulation and numerical solutions of continuum approximations for the parameters* $N = 201$, $\varepsilon = 0.02$ *and consequently* $\kappa = 4$. *We use the initial condition given in* (5.3), *zero-flux boundary conditions and plot distributions at time* $t = 0.05$.
(a) *KMC simulation for* 201 *particles of diameter* $\varepsilon = 0.02$.
(b) *Numerical solution of Model* **(iii)** *given by* (5.26).
(c) *Slice through the distributions at* $x_2 = 0.5$. *Dashed (blue) line: Model* **(i)** (*distribution given in Figure* 5.4(b)); *dotted (red) line: Model* **(ii)** (*distribution given in Figure* 5.4(c)); *dash-dotted (green) line: Model* **(iii)**; *black circles: KMC simulation.*

#	N	ε
(a)	50	$0, \ldots, 4 \times 10^{-2}$
(b)	$1, \ldots, 250$	2×10^{-2}
(c)	$100, \ldots, 2000$	$3 \times 10^{-2}, \ldots, 1.5 \times 10^{-3}$
(d)	$6, \ldots, 400$	$0.11, \ldots, 1.26 \times 10^{-2}$
	κ	c
(a)	$0, \ldots, 1.96$	$0, \ldots, 6.28 \times 10^{-2}$
(b)	$0, \ldots, 4.98$	$3.14 \times 10^{-4}, \ldots, 7.82 \times 10^{-2}$
(c)	3	$7.21 \times 10^{-2}, \ldots, 3.5 \times 10^{-3}$
(d)	$0.56, \ldots, 5.04$	5×10^{-2}

Table 5.3: *Parameter ranges for simulations in Figure* 5.6.

match each other well, as expected. Diffusion in the KMC simulations seems to be enhanced compared to the Boltzmann limit, as predicted by (5.25).

For case (B), the results shown in Figure 5.5 indicate that the particles have spread considerably further than in case (A). As mentioned above, the corresponding simulations for Models **(i)** and **(ii)** were already shown in Panels (b) and (c) of Figure 5.4, respectively, and seem to differ greatly from the KMC results. This is confirmed in the slice plots in Figure 5.5(c), where neither Model **(i)** nor Model **(ii)** match well with the KMC results. The reason for this discrepancy is that the volume fraction in test problem (B) is not negligible and this system is therefore far from the dilute gas limit. Model **(iii)** shown as dash-dotted (green) line in Figure 5.5(c), on the other hand, shows a good match with the KMC simulations. This result confirms the validity of the adjusted system (5.26) as an approximation for particles undergoing a velocity jump process with reflective hard-sphere interactions in the considered parameter region.

5.7.2 Numerical comparison for changing parameter values

In order to further investigate the parameter regions in which each of the adjusted models gives a good match to the KMC simulations, we now perform a numerical investigation for varying parameter values. The condition that particles do not overlap during the initialisation process, presents a limit to the parameter regime we can investigate. The parameter values are shown in Table 5.3.

In order to compare the distributions at the end of the simulation, we define the mean distance from the centre (MDC) for KMC simulations through

$$\langle \|\mathbf{x}_i - (0.5, 0.5)\| \rangle = \frac{1}{N} \sum_{i=1}^{N} \|\mathbf{x}_i - (0.5, 0.5)\| \ .$$

During the simulations, we choose a number of runs such that N multiplied by the number of runs is at least 10^6 and take the average MDC over all those runs. The MDC for the PDE description takes the form

$$\frac{\int_\Omega \|\mathbf{x} - (0.5, 0.5)\| \int_V p(t, \mathbf{x}, \mathbf{v}) \, \mathrm{d}\mathbf{v} \, \mathrm{d}\mathbf{x}}{\int_\Omega \int_V p(t, \mathbf{x}, \mathbf{v}) \, \mathrm{d}\mathbf{v} \, \mathrm{d}\mathbf{x}} \, .$$

Note that we explicitly only use this measure to compare the various distributions. We do not use this measure to derive diffusion constants and this measure does not correspond to the mean square displacement of particles during the simulation. This is important to note, because Bruna and Chapman [21] show that the mean square displacement is not an adequate measure for the collective diffusion constant, but for the self diffusion constant. However, because we are only using the MDC as a measure of the width of the distributions at the end of the simulations, it is a valid measure for the comparison between PDE Models **(i)**–**(iii)** and KMC simulations.

The results of this comparison can be seen in Figure 5.6. In all four plots, the dotted (red) line indicates the uncorrected velocity jump equation (1.6) that does not consider collisions at all. The dashed (blue) line indicates the first correction given in (5.23) (Model **(ii)**) and the dash-dotted (green) line shows the second correction given in (5.26) (Model **(iii)**). The (black) solid line shows the results obtained from KMC simulations. Note that we do not include Model **(i)** in this consideration, because the results are expected to be very similar to those of Model **(ii)**.

In Figure 5.6(a) we plot the results for simulation runs with $N = 50$ and varying $\varepsilon \in [0, 0.04]$. We can see that the MDC in KMC simulations, as well as in Model **(iii)**, undergoes a non-monotonic behaviour with a minimum close to $\varepsilon = 0.02$. Model **(ii)** does not show such a behaviour, as κ is monotonically increasing with ε and diffusion is therefore increasingly slowed down. This model matches the KMC results well for very small values of ε, whilst Model **(iii)** provides a good match for values up to $\varepsilon \sim 0.02$. Above this value the KMC simulations and the second correction (5.26) start to diverge and one would need to consider further correction terms to achieve an accurate approximation in this regime. Interestingly for values of ε greater than about 0.034 the hard-sphere particles actually spread faster than point particles.

The second experiment shown in Figure 5.6(b) plots the dependence of MDC on N as we keep $\varepsilon = 0.02$ constant. We can see that the MDC decreases monotonically in the KMC simulations as well as in the PDE models. The first correction (5.23) does not provide a good match for N bigger than about 5, whilst Model **(iii)** improves this match up to intermediate values of N. We see that for large values of $N > 100$

the KMC simulation spreads faster than both approximations, but slower than point particles.

Figure 5.6(c) presents the results for a constant value of κ. As is clear from the formulation of Model **(ii)** in (5.23), the first correction solely depends on κ and therefore provides a horizontal line in this case. The KMC simulations show higher values of MDC for lower values of N, i.e. in a regime far away from the Boltzmann limit. As we approach the Boltzmann limit when $N \to \infty$, the KMC simulations converge towards the value provided by Model **(ii)**. As should be clear from the definition of Model **(iii)** in (5.26), the second approximation undergoes a similar behaviour and provides a very good match to the KMC simulations throughout.

In the last experiment we keep the area fraction of particles in the simulation constant, i.e. $c = \pi N\varepsilon^2/4 = 0.05$ and vary N and ε. The KMC simulations, as well as the PDE models, show monotonically decreasing values for the MDC throughout the considered parameter regime. Investigating the forms of the first and second corrections in (5.23) and (5.26) respectively, it becomes clear that the diffusion vanishes in the limit $N \to \infty$ when keeping the volume fraction constant. The reason for this is that κ goes to infinity in this limit. Therefore, we should expect the KMC results to converge towards the MDC of the initial condition for large values of N. In Figure 5.6(d), we can see that Model **(ii)** provides significantly different results to the KMC simulations in this regime that is far from the dilute gas limit. Model **(iii)** does not provide a perfect match to the simulation results either, but provides a significant improvement over Model **(ii)**.

We conclude from this numerical study that the first approximation (Model **(ii)**) provides a good match to KMC simulations when a system close to the Boltzmann limit is considered. As one moves away from this limit and the area fraction becomes non-negligible, the second correction term (Model **(iii)**) provides an improved match. However, even this correction is only valid up to certain limits in area fraction c. One would have to consider additional terms of the Taylor expansion of the two-particle probability distribution to derive more accurate results for larger values of c.

5.8 Discussion

In this chapter we studied the effect of reflective collisions on the general behaviour of a group of particles. These reflective collisions differ greatly from the fully elastic collisions observed in gas molecules [158]. For biological applications, reflective collisions might be more relevant than the elastic collisions as can be seen in [38],

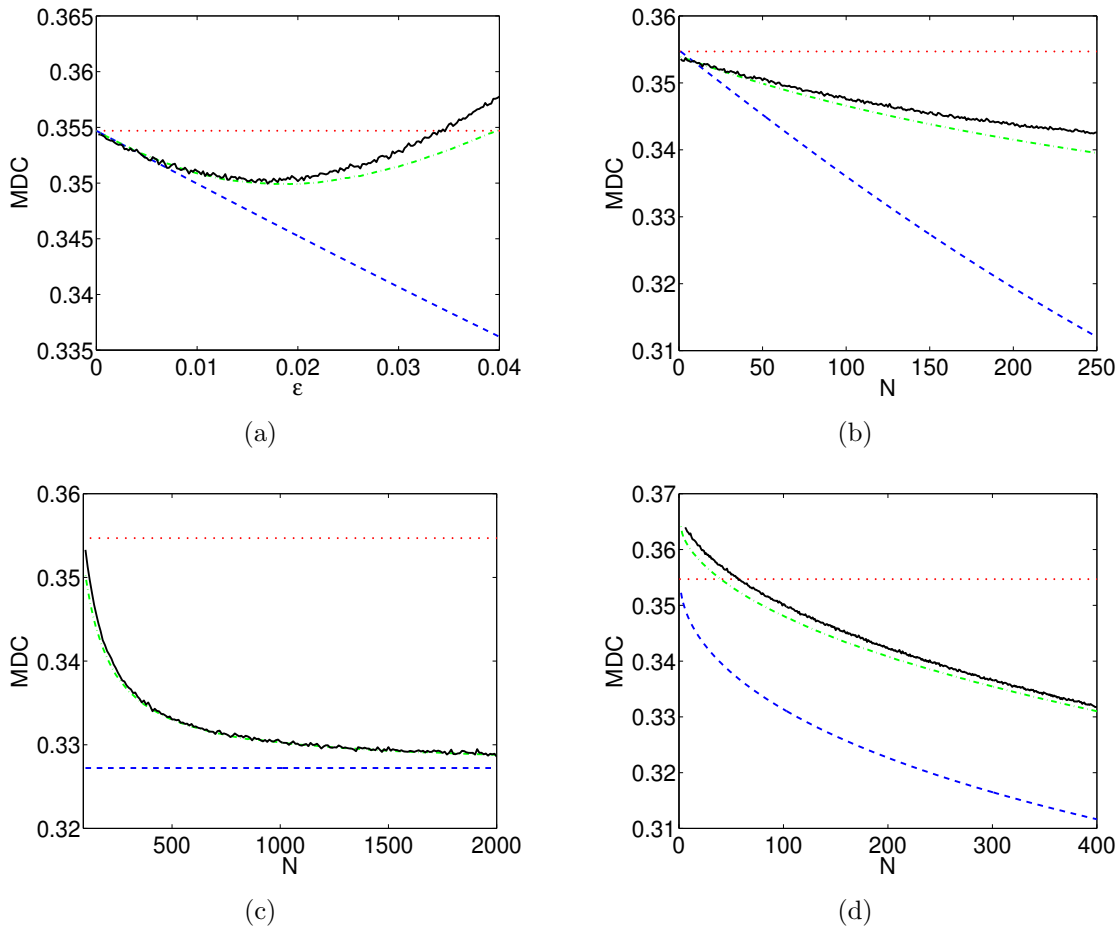


Figure 5.6: *Comparison between KMC simulation and numerical solutions of velocity jump processes with adjustments for collisions. Solid (black) line: KMC simulations; dotted (red) line: classical velocity jump equation* (1.6)*; dashed (blue) line: Model* **(ii)**; *dash-dotted (green) line: Model* **(iii)**. *The simulation parameters are given in Table* 5.3.

where animals aim to avoid each other but evidently cannot transfer momentum. Other biological examples of systems of dispersal with volume exclusion are discussed in [153].

We developed an efficient event-based KMC algorithm that is optimal in the sense that it jumps directly from event to event. Further improvement of the implementation, means that one can simulate groups of up to 2×10^3 particles in a matter of seconds. Starting from the BBGKY hierarchy [16, 98] we then developed a number of PDE descriptions that we compared to the results of KMC simulations. The first model we introduced stems from the Boltzmann equation [26] that is used in fluid flow simulations [30]. Using Cattaneo approximations [86] we then study the effect the additional collision term has on the diffusive behaviour of the group of particles. We show that in the dilute gas limit collisions are always slowing down the collective diffusion.

We then move away from the dilute gas limit and introduce finite sized particles, using a matched asymptotic expansion approach adapted from [22]. Again, using the Cattaneo approximation technique [86], we are able to derive a form for the collective diffusion constant. This diffusion constant is larger than the one in the dilute gas limit. One can compare the results for velocity jump processes obtained in this chapter to the excluded volume methods in BD simulations [22] by considering the limit $s, \lambda \to \infty$, keeping $s^2/2\lambda = D_0$ constant. In this limit, the adjusted diffusion constant given in (5.25) takes the form

$$D_{\text{eff}}\left(m^{(0)}\right) = D_0\left(1 + \kappa\varepsilon\pi m^{(0)}\right) ,$$

which is indeed the form given by Bruna and Chapman in [21, 22]. This indicates that the results shown in this chapter are consistent with those for Brownian particles. Motivated by Bruna and Chapman's work in [21], one could extend this theory to interactions between particles of two distinct species, or to the considerations of particles in confined domains [23].

Whilst this chapter conceptually differs slightly from preceding chapters, it fits well into the context of interacting particles in systems of dispersal and could thereby help addressing one of the major weaknesses of all models studied thus far, namely the lack of direct interactions in the form of volume exclusions. Using the adjusted models presented here, one could derive alternative models for bacterial chemotaxis that can be simulated using the hybrid approaches presented in Chapter 2. Similarly, one could imagine applying the techniques studied in Chapter 3 to systems with volume exclusion. In this case, the adjusted PDEs could be applied in areas of low

concentration, whilst KMC simulations would be necessary in regions where crowding effects play an important role. As for the robot experiments, the models derived in this section can be directly applied to the experiments studied in Chapter 4 and can be combined with the models incorporating finite turning times.

Chapter 6

Conclusion

In this thesis we studied biological systems of dispersal with a focus on different modelling and simulation techniques that are applicable to those systems. We started by reviewing two classical techniques: agent-based modelling and continuum PDE descriptions. These are two well-known techniques that have been and continue to be applied to a great number of biological systems. The disadvantage of those techniques is that one has very little flexibility and that inhomogeneities in a particular system cannot be exploited to a satisfying extent. Therefore the main idea of this thesis is to introduce a number of techniques that allow for a more flexible way of studying general systems of dispersal.

The first technique we introduced is hybrid modelling. Hybrid models allow an independent choice of modelling framework for each biological species. The test system that we repeatedly applied those techniques to is the process of bacterial chemotaxis, where bacteria adjust their motion according to an extracellular chemical. We saw that the most important and difficult part of hybrid models is the interaction between the various components, in this case the reactions between an agent-based species and a PDE species. Two particular matching problems can be identified and we investigated ways of solving them using interpolation techniques as well as kernel density estimations. We briefly discussed how the choice of kernel can influence the behaviour of the system and how a careless choice can lead to problematic convergence behaviour; however, a conclusive mathematical analysis has not been part of this thesis. The statistical physics literature could provide a good starting point for this doubtlessly interesting future study. Perturbation methods and closure approximations have been used for the analysis of those hybrid models [77, 124] as well as kinetic and hydrodynamic approaches [28]. One particularly interesting numerical study that could be performed with hybrid models and could help understand the

influence of different kernels is to repeat the experiment presented in [132] with hybrid models. Painter and Hillen [132] show a number of patterns that can emerge in continuum models of bacterial chemotaxis for various parameter values. It would be very interesting to examine whether similar patterns can be produced using hybrid models and how the bifurcations between various patterns change depending on the kernel.

In the next step we extended this hybrid technique by incorporating growth and death processes into the agent-based species, with the intensity of those processes potentially depending on all the species in the system. This idea is motivated by the presence of proliferation and starvation in most biologically relevant systems and could also be used to model decay and bimolecular creation reactions in chemical applications. We discuss two approaches to incorporating those processes into the hybrid system, one biologically motivated and one based on mass action kinetics. Both lead to mathematically well-posed concepts that can be matched using PDE descriptions. Throughout Chapter 2 the hybrid models do not only provide a great computational improvement compared to purely agent-based models, but also yield interesting effects that cannot be explained with mean-field PDEs.

In Chapter 3 we explored a different hybrid method, where PDEs and agent-based models in the form of BD simulations can be applied in different spatial subdomains. Those subdomains can be chosen to achieve more detailed models in interesting parts of a biological system, e.g. near reactive surfaces in virus coating [59] and the nucleus as subdomain inside a cell, or to attribute to the fact that certain areas have much reduced copy numbers compared to the rest of the studied domain, e.g. inside ion-channels [31] or filopodia [52]. Interestingly, we showed that a simple rescaling argument is sufficient to consistently predict the expected number of particles in the BD domain. This result, though counterintuitive at first, is equivalent to assuming that all particles in the PDE regime follow an identical probability distribution. However, the loss of information when moving from individual-based to group-level descriptions can lead to errors in the variances inside the BD regime. We introduced an overlap region to overcome this problem and to recover correct particle variances. All those ideas can be formulated easily in two and three dimensional regimes, but there are some implementation details that are not trivial. The introduction of zero and first order reactions does not present a great difficulty, as these act on the individual parts independently. One problem that still needs to be explored in more detail is the consideration of bimolecular reactions. It is well established how to implement those reactions in each regime individually [49]. However, the implementation inside the

overlap region might present some difficulties, as here a continuum species can react with agents. The interesting link to the work in Chapter 2 is that the first type of hybrid models is concerned exactly with those reactions and could be applied inside the overlap region.

A problem one regularly observes in the development of techniques in mathematical biology is that they are often only approachable to those with considerable experience in mathematics and programming. In order to improve this, an important next step is to provide an easy way for anybody to apply those hybrid techniques to their particular system of interest. One can easily imagine incorporating those two hybrid techniques into an extremely flexible modelling toolbox that allows combinations of them with high numbers of species and various subdomains. In Section 3.5.4 we provided a simple proof of concept idea that links the two techniques into one model. Additionally, Dr. Martin Robinson, a member of Dr. Radek Erban's research group, is currently working on an implementation of the hybrid methods presented in this thesis as well as methods seen in [62] and others into a plug-in for Smoldyn [7], a simulation tool broadly used in cell biology. Once this implementation is finished, biological researchers will be able to adjust existing models to incorporate the hybrid techniques presented in this thesis. In a further extension one could also imagine intelligent methods that automatically split the domain and assign modelling techniques to certain species in certain subdomains. In particular for the analysis of travelling wave behaviour, moving boundaries could also be an interesting extension [142].

In Chapter 4 we introduced swarm robotic experiments as a novel way to study biological systems of dispersal. The main advantage those swarms provide is that one can perform controlled experiments in a physically realistic environment with inherently noisy signals. Additionally, robots, just like animals, are not completely identical, but have individual properties that make them better at certain tasks but worse at others. We saw that even very simple experiments can motivate interesting mathematical studies that could be applied to a variety of biological systems. In this chapter, we provided a way of incorporating turning delays into classical models of velocity jump processes. The turning delays studied in this chapter assume directed turning, i.e. the turning time depends solely on the velocity before and after the turn. We show that classical velocity jump models can be extended to include this type of finite turning times. Whilst the results were obtained for and compared to swarm robotic experiments, one could imagine incorporating similar finite turning times into more complex models for animal behaviour. However, one needs to be

aware of the limitation of the models presented in this chapter, in that turning times cannot depend on any other factor but the two velocities involved.

In a more general context, the concept of swarm robotics is already actively being deployed in areas like oil spill confrontation [92] and mine detection [100]. One application area of swarm robotic experiments in mathematical biology could be the testing of animal behaviour models in physical environments, thereby providing a platform for the 'reverse engineering' of biological systems. Real signals like sound and vision can be studied using combinations of various sensors and actors, and might provide a new approach to understanding animal behaviour. One idea that would be great to explore in future research is the emergence of hierarchies in animal groups [148], e.g. as seen in a pecking order amongst hens [108] and in hierarchical following observed in homing pigeons [123]. The existing experimental system would, however, need to be greatly extended to incorporate these more complex interactions between individuals. Another interesting area for future research is the combination of the results obtained in this chapter with the hybrid models, for example by incorporating turning delays into hybrid models of bacterial chemotaxis. On the other hand hybrid robotic systems where robots can change external signals are physically difficult to implement, but could be simulated through pseudo-signals like glowing floors [71, 113] that change colour according to the behaviour of the robots.

Finally, we studied the effect of volume exclusion in the form of hard-sphere interactions in velocity jump models in Chapter 5. We derived adjusted velocity jump equations that incorporate reflective collisions and compared those to simulation results as well as to the existing literature. Whilst the model system studied in this chapter is very simplistic, it presents a first step towards including more complex volume exclusion effects that are inherently present in almost every biological system. In many of those systems though, hard-sphere interactions are not directly applicable, as some biological individuals can change their shape to accommodate others, e.g. in groups of amoeba [130]. Volume exclusion effects are often implicitly represented in existing models through repulsion effects [38].

Combining the models presented in Chapters 4 and 5, it is possible to obtain adequate agent-based and PDE representations to model primitive robotic systems. Those descriptions can be used to predict the behaviour of robots for cases that are difficult to simulate and for the investigation of scaling effects. On the other hand, these models can be seen as a first step towards improved models of fish swarms or bird flocks in the mould of [38]. Similarly, one could also study crowding behaviour by combining the KMC simulations seen in Chapter 5 with the corresponding PDE

models using PBD approaches. Whilst areas of low densities can be adequately approximated using the presented approximations, areas with high concentrations of particles still require individual-based simulations for an adequate representation.

In this thesis we generally studied a number of approaches to improve existing simulation and model frameworks for systems of dispersal. These approaches present a great opportunity to help understand the intricacies in certain types of interactions between individuals. However, in real systems interactions are often much more complex than the types of interaction studied here. We studied direct interactions through collisions and reactions, as well as indirect interactions via a chemotactic medium, but neglected effects like direct communication or individual recognition that can play an important role in biological systems. In particular the swarm robotic experiments could provide a framework to study those types of effects, but the existing implementations and models would need to be vastly extended.

The advantage of all the techniques we presented throughout this thesis is that they can in theory be applied to a vast range of systems on a great number of different scales. In practice, the modelling techniques and model extensions presented in this thesis provide a first step towards an understanding of systems of interacting particles. The smallest such systems occur inside the cell on a nanometre scale [33], whilst the biggest are seen in animal groups that roam over areas covering kilometres [20]. Despite their huge variety and differences, the common factor in all those systems is that group-level dynamics emerge even though they are often not intended or even understood by the individuals.

Appendix A

Derivation of the adjoint boundary conditions

In this appendix, the derivation of the adjoint boundary conditions (4.21) is presented. Let us therefore state again the boundary conditions related to the forward problem as given in (4.19)

$$\begin{aligned} &p(t,\mathbf{x},\mathbf{v}) = 0\,, && \mathbf{x}\in\partial\Omega_{\mathcal{T}}\,,\mathbf{v}\cdot\mathbf{n}_{\mathcal{T}}<0\,,\\ &p(t,\mathbf{x},\mathbf{v}) = \begin{cases} p\left(t-K(\mathbf{v},\mathbf{v}'),\mathbf{x},\mathbf{v}'\right)\,, & K(\mathbf{v},\mathbf{v}')\le t\,,\\ 0\,, & K(\mathbf{v},\mathbf{v}')>t\,, \end{cases} && \mathbf{x}\in\partial\Omega_{\mathcal{R}}\,,\mathbf{v}\cdot\mathbf{n}_{\mathcal{R}}<0\,. \end{aligned} \tag{A.1}$$

The adjoint boundary conditions are derived in step **(ii)** of the calculation of the adjoint operator $\mathcal{M}^*$ in Section 4.3.2. Let us add some more detail to this step. We apply the divergence theorem to obtain

$$\begin{aligned} \langle \mathcal{M}_{(ii)}p, q\rangle &= \int_0^\infty\int_\Omega\int_V q\,\mathbf{v}\cdot\nabla_{\mathbf{x}}p\,\mathrm{d}\mathbf{v}\,\mathrm{d}\mathbf{x}\,\mathrm{d}t\\ &= \int_0^\infty\int_{\partial\Omega}\int_V p\,q\,(\mathbf{v}\cdot\mathbf{n})\,\mathrm{d}\mathbf{v}\,\mathrm{d}\mathbf{x}\,\mathrm{d}t - \int_0^\infty\int_\Omega\int_V p\,\mathbf{v}\cdot\nabla_{\mathbf{x}}q\,\mathrm{d}\mathbf{v}\,\mathrm{d}\mathbf{x}\,\mathrm{d}t\,. \end{aligned} \tag{A.2}$$

From this, the adjoint operator $\mathcal{M}^*_{(ii)}$ can be obtained as

$$\langle p, \mathcal{M}^*_{(ii)}q\rangle = -\int_0^\infty\int_\Omega\int_V p\,\mathbf{v}\cdot\nabla_{\mathbf{x}}q\,\mathrm{d}\mathbf{v}\,\mathrm{d}\mathbf{x}\,\mathrm{d}t\,.$$

The adjoint boundary conditions can be derived by setting the boundary term in (A.2) to zero:

$$\begin{aligned} 0 &= \int_0^\infty\int_{\partial\Omega}\int_V p\,q\,(\mathbf{v}\cdot\mathbf{n})\,\mathrm{d}\mathbf{v}\,\mathrm{d}\mathbf{x}\,\mathrm{d}t\\ &= \underbrace{\int_0^\infty\int_{\partial\Omega_{\mathcal{R}}}\int_V p\,q\,(\mathbf{v}\cdot\mathbf{n}_{\mathcal{R}})\,\mathrm{d}\mathbf{v}\,\mathrm{d}\mathbf{x}\,\mathrm{d}t}_{I_1} + \underbrace{\int_0^\infty\int_{\partial\Omega_{\mathcal{T}}}\int_V p\,q\,(\mathbf{v}\cdot\mathbf{n}_{\mathcal{T}})\,\mathrm{d}\mathbf{v}\,\mathrm{d}\mathbf{x}\,\mathrm{d}t}_{I_2}\,. \end{aligned}$$

We can now evaluate the integrals $I_1 = I_2 = 0$ independently to obtain the conditions for the corresponding parts of the boundary. For I_1, we can split the velocity set V into the parts V^+ and V^- as defined in (4.3)

$$I_1 = \underbrace{\int_0^\infty \int_{\partial\Omega_\mathcal{R}} \int_{V^+} p\, q\, (\mathbf{v}\cdot\mathbf{n}_\mathcal{R})\, \mathrm{d}\mathbf{v}\, \mathrm{d}\mathbf{x}\, \mathrm{d}t}_{I_{1,1}} + \underbrace{\int_0^\infty \int_{\partial\Omega_\mathcal{R}} \int_{V^-} p\, q\, (\mathbf{v}\cdot\mathbf{n}_\mathcal{R})\, \mathrm{d}\mathbf{v}\, \mathrm{d}\mathbf{x}\, \mathrm{d}t}_{I_{1,2}} .$$

For $I_{1,2}$ we can apply the second boundary condition in (A.1) and obtain

$$\begin{aligned} I_{1,2} &= \int_0^\infty \int_{\partial\Omega_\mathcal{R}} \int_{V^-} p(t,\mathbf{x},\mathbf{v})\, q(t,\mathbf{x},\mathbf{v})\, (\mathbf{v}\cdot\mathbf{n}_\mathcal{R})\, \mathrm{d}\mathbf{v}\, \mathrm{d}\mathbf{x}\, \mathrm{d}t \\ &= \int_{K(\mathbf{v},\mathbf{v}')}^\infty \int_{\partial\Omega_\mathcal{R}} \int_{V^-} p(t-K(\mathbf{v},\mathbf{v}'),\mathbf{x},\mathbf{v}')\, q(t,\mathbf{x},\mathbf{v})\, (\mathbf{v}\cdot\mathbf{n}_\mathcal{R})\, \mathrm{d}\mathbf{v}\, \mathrm{d}\mathbf{x}\, \mathrm{d}t \\ &= \int_0^\infty \int_{\partial\Omega_\mathcal{R}} \int_{V^-} p(t,\mathbf{x},\mathbf{v}')\, q(t+K(\mathbf{v},\mathbf{v}'),\mathbf{x},\mathbf{v})\, (\mathbf{v}\cdot\mathbf{n}_\mathcal{R})\, \mathrm{d}\mathbf{v}\, \mathrm{d}\mathbf{x}\, \mathrm{d}t\,. \end{aligned}$$

We now switch the integration variable from $\mathbf{v}$ to $\mathbf{v}'$. The sign of the whole term changes, because $\mathbf{v}'\cdot\mathbf{n}_\mathcal{R} = -\mathbf{v}\cdot\mathbf{n}_\mathcal{R}$:

$$I_{1,2} = -\int_0^\infty \int_{\partial\Omega_\mathcal{R}} \int_{V^+} p(t,\mathbf{x},\mathbf{v})\, q(t+K(\mathbf{v}',\mathbf{v}),\mathbf{x},\mathbf{v}')\, (\mathbf{v}\cdot\mathbf{n}_\mathcal{R})\, \mathrm{d}\mathbf{v}\, \mathrm{d}\mathbf{x}\, \mathrm{d}t\,.$$

We obtain the boundary condition on the reflective boundary $\partial\Omega_\mathcal{R}$ for q by adding $I_{1,1}$ and $I_{1,2}$:

$$q(t,\mathbf{x},\mathbf{v}) = q(t+K(\mathbf{v}',\mathbf{v}),\mathbf{x},\mathbf{v}')\,, \qquad \mathbf{x}\in\partial\Omega_\mathcal{R}\,, \quad \mathbf{v}\cdot\mathbf{n}_\mathcal{R} > 0\,. \tag{A.3}$$

On the target boundary, the integral I_2 simplifies to

$$\begin{aligned} I_2 &= \int_0^\infty \int_{\partial\Omega_\mathcal{T}} \int_{V^+} p\, q\, (\mathbf{v}\cdot\mathbf{n}_\mathcal{T})\, \mathrm{d}\mathbf{v}\, \mathrm{d}\mathbf{x}\, \mathrm{d}t + \int_0^\infty \int_{\partial\Omega_\mathcal{T}} \int_{V^-} p\, q\, (\mathbf{v}\cdot\mathbf{n}_\mathcal{T})\, \mathrm{d}\mathbf{v}\, \mathrm{d}\mathbf{x}\, \mathrm{d}t \\ &= \int_0^\infty \int_{\partial\Omega_\mathcal{T}} \int_{V^+} p\, q\, (\mathbf{v}\cdot\mathbf{n}_\mathcal{T})\, \mathrm{d}\mathbf{v}\, \mathrm{d}\mathbf{x}\, \mathrm{d}t\,, \end{aligned}$$

because p vanishes on V^- according to (A.1). Therefore the second adjoint boundary condition takes the form

$$q(t,\mathbf{x},\mathbf{v}) = 0\,, \qquad \mathbf{x}\in\partial\Omega_\mathcal{T}\,, \quad \mathbf{v}\cdot\mathbf{n}_\mathcal{T} > 0\,. \tag{A.4}$$

The two boundary conditions (A.3)–(A.4) form the adjoint boundary conditions given in (4.21).

Appendix B

Integrals over the unit sphere in 2 dimensions

In this appendix we present some lemmas regarding the integration over parts of the unit sphere $\mathbb{S}^1$ in two dimensions. In particular, we define the positive component of the unit sphere with respect to the vector $\mathbf{a}$ through

$$\mathbb{S}^1_+(\mathbf{a}) = \{\mathbf{n} \in \mathbb{S}^1 \ : \ \mathbf{n} \cdot \mathbf{a} > 0\}\,.$$

Lemma B.1. *The equality*

$$\int_{\mathbb{S}^1_+(\mathbf{a})} \mathbf{n}\,\mathrm{d}\mathbf{n} = 2\frac{\mathbf{a}}{\|\mathbf{a}\|}\,, \tag{B.1}$$

holds for all vectors $\mathbf{a} \in \mathbb{R}^2\backslash\{\mathbf{0}\}$.

Proof. Let us rewrite the vectors $\mathbf{a}$ and $\mathbf{n}$ to the following

$$\mathbf{a} = \|\mathbf{a}\| \begin{pmatrix} \cos\alpha \\ \sin\alpha \end{pmatrix}, \qquad \mathbf{n} = \begin{pmatrix} \cos\gamma \\ \sin\gamma \end{pmatrix}.$$

We can now directly evaluate the integral on the left hand side of (B.1).

$$\begin{aligned}\int_{\mathbb{S}^1_+(\mathbf{a})} \mathbf{n}\,\mathrm{d}\mathbf{n} &= \int_{\alpha-\pi/2}^{\alpha+\pi/2} \begin{pmatrix} \cos\gamma \\ \sin\gamma \end{pmatrix} \mathrm{d}\gamma = \int_{-\pi/2}^{\pi/2} \begin{pmatrix} \cos(\gamma+\alpha) \\ \sin(\gamma+\alpha) \end{pmatrix} \mathrm{d}\gamma \\ &= 2\begin{pmatrix} \cos\alpha \\ \sin\alpha \end{pmatrix} = 2\frac{\mathbf{a}}{\|\mathbf{a}\|}\,.\end{aligned}$$

□

Lemma B.2. *The equality*

$$\int_{\mathbb{S}^1_+(\mathbf{a})} (\mathbf{n}\cdot\mathbf{a})(\mathbf{n}\cdot\mathbf{b})\mathbf{n}\,\mathrm{d}\mathbf{n} = \frac{2}{3}\left(\frac{\mathbf{a}}{\|\mathbf{a}\|} + \|\mathbf{a}\|\,\mathbf{b}\right), \tag{B.2}$$

holds for all vectors $\mathbf{a}, \mathbf{b} \in \mathbb{R}^2\backslash\{\mathbf{0}\}$.

Proof. Let us rewrite the vectors $\mathbf{a}, \mathbf{b}$ and $\mathbf{n}$ to the following

$$\mathbf{a} = \|\mathbf{a}\| \begin{pmatrix} \cos\alpha \\ \sin\alpha \end{pmatrix}, \qquad \mathbf{b} = \|\mathbf{b}\| \begin{pmatrix} \cos\beta \\ \sin\beta \end{pmatrix}, \qquad \mathbf{n} = \begin{pmatrix} \cos\gamma \\ \sin\gamma \end{pmatrix}.$$

We can now rewrite the integral on the left hand side of (B.2) to

$$\begin{aligned}
&\|\mathbf{a}\| \|\mathbf{b}\| \int_{\alpha-\pi/2}^{\alpha+\pi/2} \cos(\gamma-\alpha)\cos(\gamma-\beta) \begin{pmatrix} \cos\gamma \\ \sin\gamma \end{pmatrix} \mathrm{d}\mathbf{n} \\
&\quad = \|\mathbf{a}\| \|\mathbf{b}\| \int_{-\pi/2}^{\pi/2} \cos(\gamma)\cos(\gamma+\alpha-\beta) \begin{pmatrix} \cos(\gamma+\alpha) \\ \sin(\gamma+\alpha) \end{pmatrix} \mathrm{d}\mathbf{n} \\
&\quad = \|\mathbf{a}\| \|\mathbf{b}\| \int_{-\pi/2}^{\pi/2} \cos^3(\gamma)\cos(\alpha-\beta) \begin{pmatrix} \cos\alpha \\ \sin\alpha \end{pmatrix} \mathrm{d}\mathbf{n} \\
&\qquad + \|\mathbf{a}\| \|\mathbf{b}\| \int_{-\pi/2}^{\pi/2} \cos(\gamma)\sin^2(\gamma)\sin(\alpha-\beta) \begin{pmatrix} \sin\alpha \\ -\cos\alpha \end{pmatrix} \mathrm{d}\mathbf{n} \\
&\quad = \frac{2}{3} \|\mathbf{a}\| \|\mathbf{b}\| \left(2\cos(\alpha-\beta) \begin{pmatrix} \cos\alpha \\ \sin\alpha \end{pmatrix} + \sin(\alpha-\beta) \begin{pmatrix} \sin\alpha \\ -\cos\alpha \end{pmatrix} \right).
\end{aligned}$$

Standard addition formulae show that

$$\cos(\alpha-\beta) \begin{pmatrix} \cos\alpha \\ \sin\alpha \end{pmatrix} + \sin(\alpha-\beta) \begin{pmatrix} \sin\alpha \\ -\cos\alpha \end{pmatrix} = \begin{pmatrix} \cos\beta \\ \sin\beta \end{pmatrix}, \tag{B.3}$$

Hence, we can rewrite the integral as

$$\begin{aligned}
&\|\mathbf{a}\| \|\mathbf{b}\| \int_{\alpha-\pi/2}^{\alpha+\pi/2} \cos(\gamma-\alpha)\cos(\gamma-\beta) \begin{pmatrix} \cos\gamma \\ \sin\gamma \end{pmatrix} \mathrm{d}\mathbf{n} \\
&\quad = \frac{2}{3} \|\mathbf{a}\| \|\mathbf{b}\| \left(\cos(\alpha-\beta) \begin{pmatrix} \cos\alpha \\ \sin\alpha \end{pmatrix} + \begin{pmatrix} \cos\beta \\ \sin\beta \end{pmatrix} \right) \\
&\quad = \frac{2}{3} \|\mathbf{a}\| \|\mathbf{b}\| \left(\frac{\mathbf{a}\cdot\mathbf{b}}{\|\mathbf{a}\| \|\mathbf{b}\|} \frac{\mathbf{a}}{\|\mathbf{a}\|} + \frac{\mathbf{b}}{\|\mathbf{b}\|} \right) \\
&\quad = \frac{2}{3} \left(\frac{\mathbf{a}}{\|\mathbf{a}\|} + \|\mathbf{a}\| \mathbf{b} \right).
\end{aligned}$$

□

Lemma B.3. *The equality*

$$\int_{\mathbb{S}^1_+(\mathbf{a})} (\mathbf{n}\cdot\mathbf{a})(\mathbf{n}\cdot\mathbf{b})(\mathbf{n}\cdot\mathbf{c})\mathbf{n}\,\mathrm{d}\mathbf{n} = \frac{1}{8}\pi \left(\mathbf{a}(\mathbf{b}\cdot\mathbf{c}) + \mathbf{b}(\mathbf{a}\cdot\mathbf{c}) + \mathbf{c}(\mathbf{a}\cdot\mathbf{b}) \right), \tag{B.4}$$

holds for all vectors $\mathbf{a}, \mathbf{b}, \mathbf{c} \in \mathbb{R}^2$.

Proof. Let us rewrite the vectors $\mathbf{a}, \mathbf{b}$, $\mathbf{c}$ and $\mathbf{n}$ to the following

$$\begin{aligned}
\mathbf{a} &= \|\mathbf{a}\| \begin{pmatrix} \cos\alpha \\ \sin\alpha \end{pmatrix}, & \mathbf{b} &= \|\mathbf{b}\| \begin{pmatrix} \cos\beta \\ \sin\beta \end{pmatrix}, \\
\mathbf{c} &= \|\mathbf{c}\| \begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix}, & \mathbf{n} &= \begin{pmatrix} \cos\gamma \\ \sin\gamma \end{pmatrix}.
\end{aligned}$$

Now we can write the integral and use similar techniques as before

$$\begin{aligned}
&\frac{1}{\|\mathbf{a}\|\,\|\mathbf{b}\|\,\|\mathbf{c}\|}\int_{\mathbb{S}^1_+(\mathbf{a})}(\mathbf{n}\cdot\mathbf{a})(\mathbf{n}\cdot\mathbf{b})(\mathbf{n}\cdot\mathbf{c})\mathbf{n}\,\mathrm{d}\mathbf{n}\\
&\qquad=\int_{\alpha-\pi/2}^{\alpha+\pi/2}\cos(\gamma-\alpha)\cos(\gamma-\beta)\cos(\gamma-\theta)\begin{pmatrix}\cos\gamma\\ \sin\gamma\end{pmatrix}\mathrm{d}\gamma\\
&\qquad=\int_{-\pi/2}^{\pi/2}\cos\gamma\cos(\gamma+\alpha-\beta)\cos(\gamma+\alpha-\theta)\begin{pmatrix}\cos(\gamma+\alpha)\\ \sin(\gamma+\alpha)\end{pmatrix}\mathrm{d}\gamma\,.
\end{aligned}$$

We can now simplify this integral using the fact that

$$\int_{-\pi/2}^{\pi/2}\sin(\gamma)\cos^3(\gamma)\,\mathrm{d}\gamma=\int_{-\pi/2}^{\pi/2}\sin^3(\gamma)\cos(\gamma)=0\,.$$

We obtain

$$\begin{aligned}
&\frac{1}{\|\mathbf{a}\|\,\|\mathbf{b}\|\,\|\mathbf{c}\|}\int_{\mathbb{S}^1_+(\mathbf{a})}(\mathbf{n}\cdot\mathbf{a})(\mathbf{n}\cdot\mathbf{b})(\mathbf{n}\cdot\mathbf{c})\mathbf{n}\,\mathrm{d}\mathbf{n}\\
&\qquad=\int_{-\pi/2}^{\pi/2}\cos^4(\gamma)\cos(\alpha-\beta)\cos(\alpha-\theta)\begin{pmatrix}\cos\alpha\\ \sin\alpha\end{pmatrix}\mathrm{d}\gamma\\
&\qquad\quad+\int_{-\pi/2}^{\pi/2}\cos^2(\gamma)\sin^2(\gamma)\sin(\alpha-\beta)\sin(\alpha-\theta)\begin{pmatrix}\cos\alpha\\ \sin\alpha\end{pmatrix}\mathrm{d}\gamma\\
&\qquad\quad+\int_{-\pi/2}^{\pi/2}\cos^2(\gamma)\sin^2(\gamma)\cos(\alpha-\beta)\sin(\alpha-\theta)\begin{pmatrix}\sin\alpha\\ -\cos\alpha\end{pmatrix}\mathrm{d}\gamma\\
&\qquad\quad+\int_{-\pi/2}^{\pi/2}\cos^2(\gamma)\sin^2(\gamma)\sin(\alpha-\beta)\cos(\alpha-\theta)\begin{pmatrix}\sin\alpha\\ -\cos\alpha\end{pmatrix}\mathrm{d}\gamma\,.
\end{aligned}$$

This integral can be evaluated using

$$\int_{-\pi/2}^{\pi/2}\cos^4(\gamma)\,\mathrm{d}\gamma=\frac{3\pi}{8}\,,\qquad\int_{-\pi/2}^{\pi/2}\sin^2(\gamma)\cos^2(\gamma)\,\mathrm{d}\gamma=\frac{\pi}{8}\,.$$

Plugging this in and using (B.3), we get

$$\begin{aligned}
&\frac{1}{\|\mathbf{a}\|\,\|\mathbf{b}\|\,\|\mathbf{c}\|}\int_{\mathbb{S}^1_+(\mathbf{a})}(\mathbf{n}\cdot\mathbf{a})(\mathbf{n}\cdot\mathbf{b})(\mathbf{n}\cdot\mathbf{c})\mathbf{n}\,\mathrm{d}\mathbf{n}\\
&\qquad=\int_{\alpha-\pi/2}^{\alpha+\pi/2}\cos(\gamma-\alpha)\cos(\gamma-\beta)\cos(\gamma-\theta)\begin{pmatrix}\cos\gamma\\ \sin\gamma\end{pmatrix}\mathrm{d}\gamma\\
&\qquad=\frac{\pi}{8}\left[\cos(\alpha-\beta)\begin{pmatrix}\cos\theta\\ \sin\theta\end{pmatrix}+\cos(\alpha-\theta)\begin{pmatrix}\cos\beta\\ \sin\beta\end{pmatrix}+\cos(\beta-\theta)\begin{pmatrix}\cos\alpha\\ \sin\alpha\end{pmatrix}\right]\\
&\qquad=\frac{\pi}{8}\left[\frac{\mathbf{a}\cdot\mathbf{b}}{\|\mathbf{a}\|\,\|\mathbf{b}\|}\frac{\mathbf{c}}{\|\mathbf{c}\|}+\frac{\mathbf{a}\cdot\mathbf{c}}{\|\mathbf{a}\|\,\|\mathbf{c}\|}\frac{\mathbf{b}}{\|\mathbf{b}\|}+\frac{\mathbf{b}\cdot\mathbf{c}}{\|\mathbf{b}\|\,\|\mathbf{c}\|}\frac{\mathbf{a}}{\|\mathbf{a}\|}\right].
\end{aligned}$$

Multiplying both sides by $\|\mathbf{a}\|\,\|\mathbf{b}\|\,\|\mathbf{c}\|$, we obtain (B.4). □

Bibliography

[1] J. Adler. Chemotaxis in bacteria. *Science*, 153:708–716, 1966.

[2] J. Adler. Chemoreceptors in bacteria. *Science*, 166:1588–1597, 1969.

[3] T. Alarcón, H. M. Byrne, and P. K. Maini. A cellular automaton model for tumour growth in inhomogeneous environment. *Journal of Theoretical Biology*, 225:257–274, 2003.

[4] B. J. Alder and T. E. Wainwright. Studies in molecular dynamics. I. General method. *The Journal of Chemical Physics*, 31(2):459–466, 1959.

[5] F. J. Alexander, A. L. Garcia, and D. M. Tartakovsky. Algorithm refinement for stochastic partial differential equations. *Journal of Computational Physics*, 182:47–66, 2002.

[6] A. R. A. Anderson. A hybrid mathematical model of solid tumour invasion: the importance of cell adhesion. *Mathematical Medicine and Biology: a Journal of the IMA*, 22:163–186, 2005.

[7] S. S. Andrews, N. J. Addy, R. Brent, and A. P. Arkin. Detailed simulations of cell biology with Smoldyn 2.1. *PLoS Computational Biology*, 6(3):e1000705, 2010.

[8] S. S. Andrews and D. Bray. Stochastic simulation of chemical reactions with spatial resolution and single molecular detail. *Physical Biology*, 1:137–151, 2004.

[9] L. Arnold. *Stochastic differential equations, theory and applications.* Wiley-Interscience Publication, 1974.

[10] N Barkai and S Leibler. Bacterial chemotaxis - united we sense ... *Nature*, 393:18–21, 1998.

[11] H. C. Berg. How bacteria swim. *Scientific American*, 233:36–44, 1975.

[12] H. C. Berg. *Random walks in biology*. Princeton University Press, 1983.

[13] H. C. Berg. Motile behaviour of bacteria. *Physics Today*, 53(1):24, 2000.

[14] H. C. Berg and D. A. Brown. Chemotaxis in *Escherichia coli* analysed by three-dimensional tracking. *Nature*, 239:500–504, 1972.

[15] G. V. Bobashev, D. M. Goedecke, F. Yu, and J. M. Epstein. A hybrid epidemic model: Combining the advantages of agent-based and equation-based approaches. In S. Henderson, editor, *Proceedings of the 2007 Winter Simulation Conference*, pages 1532–1537. IEEE, 2007.

[16] M. Born and H. S. Green. A general kinetic theory of liquids. I. The molecular distribution functions. *Proceedings of the Royal Society London A*, 188(1012):10–18, 1946.

[17] R. B. Bourret, K. A. Borkovich, and M. I. Simon. Signal transduction pathways involving protein phosphorylation in prokaryotes. *Annual Review of Biochemistry*, 60:401–441, 1991.

[18] M. Brambilla, E. Ferrante, M. Birattari, and M. Dorigo. Swarm robotics: a review from the swarm engineering perspective. *Swarm Intelligence*, 7:1–41, 2013.

[19] M. P. Brenner, L. S. Levitov, and E. O. Budrene. Physical mechanisms for chemotactic pattern formation by bacteria. *Biophysical Journal*, 74(4):1677–1693, 1998.

[20] B. K. Briscoe, M. A. Lewis, and S. E. Parrish. Home range formation in wolves due to scent marking. *Bulletin of Mathematical Biology*, 64(2):261–284, 2002.

[21] M. Bruna and S. J. Chapman. Diffusion of multiple species with excluded-volume effects. *The Journal of Chemical Physics*, 137:204116, 2012.

[22] M. Bruna and S. J. Chapman. Excluded-volume effects in the diffusion of hard spheres. *Physical Review E*, 85:011103, 2012.

[23] M. Bruna and S. J. Chapman. Diffusion of finite-size particles in confined geometries. *Bulletin of Mathematical Biology*, 76:947–982, 2014.

[24] E. O. Budrene and H. C. Berg. Complex patterns formed by motile cells of *Escherichia coli*. *Nature*, 349:630–633, 1991.

[25] E. O. Budrene and H. C. Berg. Dynamics of formation of symmetrical patterns by chemotactic bacteria. *Nature*, 376:49–53, 1995.

[26] C. Cercignani. *The Boltzmann equation and its applications*. Springer, 1988.

[27] C. Cercignani, R. Illner, and M. Pulvirenti. *The mathematical theory of dilute gases*. Applied Mathematical Sciences, 106, Springer, 1994.

[28] P. H. Chavanis. Nonlinear mean field Fokker-Planck equations. application to the chemotaxis of biological populations. *European Physical Journal B*, 62(2):179–208, 2008.

[29] P. H. Chavanis. A stochastic Keller-Segel model of chemotaxis. *Communications in nonlinear science and numerical simulations*, 15(1):60–70, 2010.

[30] S. Chen and G. D. Doolen. Lattice Boltzmann method for fluid flows. *Annual Review of Fluid Mechanics*, 30:329–364.

[31] W. Chen, R. Erban, and S. J. Chapman. From Brownian dynamics to Markov chain: an ion channel example. *SIAM Journal on Applied Mathematics*, 74(1):208–235, 2013.

[32] A. L. Christensen, R. O'Grady, and M. Dorigo. From fireflies to fault tolerant swarms of robots. *IEEE Transactions on Evolutionary Computation*, 13(4):754–766, 2009.

[33] S. H. Chung, T. W. Allen, M. Hoyles, and S. Kuyucak. Permeation of ions across the potassium channel: Brownian dynamics studies. *Biophysical Journal*, 77(5):2517–2533, 1999.

[34] E. A. Codling and N. A. Hill. Calculating spatial statistics for velocity jump processes with experimentally observed reorientation parameters. 2005.

[35] E. A. Codling, M. J. Plank, and S. Benhamou.

[36] B. Corry, S. Kuyucak, and S. Chung. Tests of continuum theories as models of ion channels. II. Poisson-Nernst-Planck theory versus Brownian dynamics. *Biophysical Journal*, 78:2364–2381, 2000.

[37] I. Couzin, J. Krause, N. Franks, and S. Levin. Effective leadership and decision-making in animal groups on the move. *Nature*, 433:513–516, 2005.

[38] I. Couzin, J. Krause, R. James, G. Ruxton, and N. Franks. Collective memory and spatial sorting in animal groups. *Journal of Theoretical Biology*, 218(1):1–11, 2002.

[39] J. Dallon and H. G. Othmer. A discrete cell model with adaptive signalling for aggregation of *Dictyostelium discoideum*. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 352:391–417, 1997.

[40] A. Dhariwal, G. S. Sukhatme, and A. A. G. Requicha. Bacterium-inspired robots for environmental monitoring. In *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, volume 2, pages 1436–1443, 2004.

[41] H. Elman, D. Silvester, and A. Wathen. *Finite elements and fast iterative solvers*. Oxford Science Publisher, 2005.

[42] S. Engblom, L. Ferm, A. Hellander, and P. Lötstedt. Simulation of stochastic reaction-diffusion processes on unstructured meshes. *SIAM Journal on Scientific Computing*, 31:1774–1797, 2009.

[43] T. W. Engelmann. Zur Biologie der Schizomyceten. *Archiv für die gesamte Physiologie des Menschen und der Tiere*, 26(1):537–545, 1881.

[44] R. Erban. *From individual to collective behaviour in biological systems*. PhD thesis, University of Minnesota, 2005.

[45] R. Erban. Stochastic modelling of biological processes. Lecture notes, 2012.

[46] R. Erban and S. J. Chapman. On chemisorption of polymers to solid surfaces. *Journal of Statistical Physics*, 127:1255–1277, 2007.

[47] R. Erban and S. J. Chapman. Reactive boundary conditions for stochastic simulations of reaction-diffusion processes. *Physical Biology*, 4:16–28, 2007.

[48] R. Erban and S. J. Chapman. Time scale of random sequential adsorption. *Physical Review E*, 75:041116, 2007.

[49] R. Erban and S. J. Chapman. Stochastic modelling of reaction-diffusion processes: algorithms for bimolecular reactions. *Physical Biology*, 6:046001, 2009.

[50] R. Erban, S. J. Chapman, K. Fisher, I. Kevrekidis, and L. Seymour. Dynamics of polydisperse irreversible adsorption: a pharmacological example. *Mathematical Models and Methods in Applied Sciences (M3AS)*, 17(5):759–781, 2007.

[51] R. Erban, S. J. Chapman, and P. K. Maini. A practical guide to stochastic simulations of reaction-diffusion processes. 35 pages, available at http://arxiv.org/abs/0704.1908, 2007.

[52] R. Erban, M. B. Flegg, and G. A. Papoian. Multiscale stochastic reaction-diffusion modelling: application to actin dynamics in filopodia. *Bulletin of Mathematical Biology*, 76:799–818, 2014.

[53] R. Erban and J. Haskovec. From individual to collective behaviour of coupled velocity jump processes: A locust example. *Kinetic and Related Models*, 5(4):817–842, 2012.

[54] R. Erban, I. Kevrekidis, and H. G. Othmer. An equation-free computational approach for extracting population-level behavior from individual-based models of biological dispersal. *Physica D*, 215:1–24, 2006.

[55] R. Erban and H. G. Othmer. From individual to collective behaviour in bacterial chemotaxis. *SIAM Journal on Applied Mathematics*, 65:361–391, 2004.

[56] R. Erban and H. G. Othmer. From signal transduction to spatial pattern formation in *E. coli*: A paradigm for multi-scale modeling in biology. *Multiscale Modeling and Simulation*, 3:362–394, 2005.

[57] J. J. Faria, J. R. G. Dyer, R. O. Clment, I. D. Couzin, N. Holt, A. J. W. Ward, D. Waters, and J. Krause. A novel method for investigating the collective behaviour of fish: introducing robofish. *Behavioral Ecology and Sociobiology*, 64:1211–1218, 2010.

[58] L. Ferm, A. Hellander, and P. Lötstedt. An adaptive algorithm for simulation of stochastic reaction-diffusion processes. *Journal of Computational Physics*, 229:343–360, 2010.

[59] K. Fisher, Y. Stallwood, N. Green, K. Ulbrich, V. Mautner, and Seymour L. Polymer-coated adenovirus permits efficient retargeting and evades neutralising antibodies. *Gene therapy*, 8(5):341–348, 2001.

[60] R. Fisher. The wave of advance of advantageous genes. *Annals of Eugenics*, 7:355–369, 1937.

[61] R. FitzHugh. Impulses and physiological states in theoretical models of nerve membrane. *Biophysical Journal*, 1:445–466, 1961.

[62] M. B. Flegg, S. J. Chapman, and R. Erban. The Two Regime Method for optimizing stochastic reaction-diffusion simulations. *Royal Society Interface*, 9:859–868, 2012.

[63] M. B. Flegg, S. Ruediger, and R. Erban. Diffusive spatio-temporal noise in a first-passage time model for intracellular calcium release. 2013.

[64] E. Flekkøy, J. Feder, and G. Wagner. Coupling particles and fields in a diffusive hybrid model. *Physical Review E*, 64:066302, 2001.

[65] B. Franz. Synchronisation properties of an agent-based animal behaviour model. Master's thesis, University of Oxford, 2009.

[66] B. Franz and R. Erban. Hybrid modelling of individual movement and collective behaviour. In M. Lewis, P. K. Maini, and S. Petrovskii, editors, *Dispersal, individual movement and spatial ecology: A mathematical perspective*. Springer, 2013.

[67] B. Franz, M. B. Flegg, S. J. Chapman, and R. Erban. Multiscale reaction-diffusion algorithms: PDE-assisted Brownian dynamics. *SIAM Journal on Applied Mathematics*, 73(3):1224–1247, 2013.

[68] B. Franz, J. P. Taylor-King, C. A. Yates, and R. Erban. Hard-sphere interactions in velocity jump models.

[69] B. Franz, C. Xue, K. J. Painter, and R. Erban. Travelling waves of hybrid chemotaxis models. *Bulletin of Mathematical Biology*, 76(2):377–400, 2014.

[70] G. Gardiner. *Handbook of stochastic processes for physics, chemistry and natural sciences.* Springer, 2nd edition, 1985.

[71] S. Garnier, M. Combe, C. Jost, and G. Theraulaz. Do ants need to estimate the geometrical properties of trail bifurcations to find an efficient route? A swarm robotics test bed. *PLoS Computational Biology*, 9(3):e1002903, 2013.

[72] S. Garnier, J. Gautrais, M. Asadpour, C. Jost, and G. Theraulaz. Self-organized aggregation triggers collective decision making in a group of cockroach-like robots. *Adaptive Behaviour*, 17(2):109–133, 2009.

[73] V. Geraldes, V. Semião, and M. N. Pinho. Numerical modelling of mass transfer in slits with semi-permeable membrane wall. *Engineering Computations*, 17(3):192–218, 2000.

[74] A. Gerisch and K. J. Painter. Mathematical modelling of cell adhesion and its applications to developmental biology and cancer invasion. In A. Chauviere and L. Preziosi, editors, *Cell Mechanics: From Single Scale-Based Models to Multiscale Modeling*, chapter 12, pages 319–350. CRC Press, 2010.

[75] P. Gerlee and A. R. A. Anderson. An evolutionary hybrid cellular automaton model of solid tumour growth. *Journal of Theoretical Biology*, 246:583–603, 2007.

[76] D. Gillespie. Exact stochastic simulations of coupled chemical reactions. *Journal of Physical Chemistry*, 81:2340–2361, 1977.

[77] R. Grima. Multiscale modelling of biological pattern formation. *Current Topics in Developmental Biology*, 81:435–460, 2008.

[78] G. Grimmet and D. Stirzaker. *Probability and random processes*. Oxford, 2001.

[79] Z. Guo, P. Sloot, and J. Tay. A hybrid agent-based approach for modeling microbiological systems. *Journal of Theoretical Biology*, 255:163–175, 2008.

[80] H. Hamann, T. Schmickl, H. Wörn, and K. Crailsheim. Analysis of emergent symmetry breaking in collective decision making. *Neural Computation and Applications*, 21:207–218, 2013.

[81] J. Hattne, D. Fange, and J. Elf. Stochastic reaction-diffusion simulation with MesoRD. *Bioinformatics*, 21(12):2923–2924, 2005.

[82] D. Helbing, I. Farkas, and T. Vicsek. Simulating dynamical features of escape panic. *Nature*, 407:487–490, 2000.

[83] A. Heppenstall, A. Evans, and M. Birkin. A hybrid multi-agent / spatial interaction model system for petrol price setting. *Transactions in GIS*, 9:35–51, 2005.

[84] A. Heppenstall, A. Evans, and M. Birkin. Using hybrid agent-based systems to model spatially-influenced retail markets. *Journal of Artificial Societies and Social Simulation*, 9:2, 2006.

[85] A. Heppenstall, A. Evans, and M. Birkin. Genetic algorithm optimisation of an agent-based model for simulating a retail market. *Environment and Planning B: Planning and Design*, 34:1051–1070, 2007.

[86] T. Hillen. On the L^2 closure of transport equations: the Cattaneo approximation. *Discrete and Continuous Dynamical Systems B*, 4(4):961–982, 2004.

[87] T. Hillen and K. J. Painter. A user's guide to PDE models for chemotaxis. *Journal of Mathematical Biology*, 58:183–217, 2009.

[88] D. Horstmann. From 1970 until present: the Keller-Segel model in chemotaxis and its consequences. II. *Jahresbericht des Deutschen Mathematiker Vereins*, 106:51–69, 2004.

[89] M. Howard. How to build a robust intracellular concentration gradient. *Trends in Cell Biology*, 22(6):311–317, 2012.

[90] M. Jones. The performance of kernel density functions in kernel distribution function estimation. *Statistics & Probability Letters*, 9(2):129 – 132, 1990.

[91] M. Jones, J. Marron, and S. Sheather. A brief survey of bandwidth selection for density estimation. *Journal of the American Statistical Association*, 91:401–407, 1996.

[92] N. M. P. Kakalis and Y. Ventikos. Robotic swarm concept for efficient oil spill confrontation. *Journal of Hazardous Materials*, 154:880–887, 2008.

[93] G. Karch. Selfsimilar profiles in large time asymptotics of solutions to damped wave equations. *Studia Mathematica*, 143:175–197, 2000.

[94] M.E. Kavousanakis, R. Erban, A. G. Boudouvis, C.W. Gear, and I.G. Kevrekidis. Projective and coarse projective integration for problems with continuous symmetries. *Journal of Computational Physics*, 225(1):382–407, 2007.

[95] E. Keller and L. Segel. Model for chemotaxis. *Journal of Theoretical Biology*, 30:225–234, 1971.

[96] E. Keller and L. Segel. Traveling bands of chemotactic bacteria: A theoretical analysis. *Journal of Theoretical Biology*, 30:235–248, 1971.

[97] C. Kennedy and R. Aris. Traveling waves in a simple population model involving growth and death. *Bulletin of Mathematical Biology*, 42:397–429, 1980.

[98] J. G. Kirkwood. The statistical mechanical theory of transport processes I. General theory. *The Journal of Chemical Physics*, 14(3):180–201, 1946.

[99] P. Kloeden and E. Platen. *Numerical solution of stochastic differential equations*. Springer, 1999.

[100] V. Kumar and F. Şahin. Cognitive maps in swarm robots for the mine detection application. In *Systems, Man and Cybernetics, 2003. IEEE International Conference on*, volume 4, pages 3364–3369, 2003.

[101] K. Landman, G. Petter, and D. Newgreen. Chemotactic cellular migration: smooth and discontinuous travelling wave solutions. *SIAM Journal on Applied Mathematics*, 63(5):1666–1681, 2003.

[102] K. A. Landman, M. J. Simpson, and D. F. Newgreen. Mathematical and experimental insights into the development of the enteric nervous system and Hirschsprung's disease. *Development, Growth & Differentiation*, 49:277–286, 2007.

[103] K. A. Landman, M. J. Simpson, J. Slater, and D. F. Newgreen. Diffusive and chemotactic cellular migration: Smooth and discontinuous travelling wave solutions. *SIAM Journal on Applied Mathematics*, 65:1420–1442, 2005.

[104] J. Landsberg and R. Waring. A generalised model of forest productivity using simplified concepts of radiation-use efficiency, carbon balance and partitioning. *Forest Ecology and Management*, 95:209–228, 1997.

[105] J. Lewins. *Importance, the adjoint function: The physical basis of variational and perturbation theory in transport and diffusion problems*. Pergamon Press, 1965.

[106] T. Li and Z. Wang. Asymptotic nonlinear stability of traveling waves to conservation laws arising from chemotaxis. *Journal of Differential Equations*, 250:1310–1333, 2011.

[107] T. Li and Z. Wang. Steadily propagating waves of a chemotaxis model. *Mathematical Biosciences*, 240:161–168, 2012.

[108] W. B. Lindquist and I. D. Chase. Data-based analysis of winner-loser models of hierarchy formation in animals. *Bulletin of Mathematical Biology*, 71(3):556–584, 2009.

[109] J. Lipkova, K. Zygalakis, S. J. Chapman, and R. Erban. Analysis of Brownian dynamics simulations of reversible bimolecular reactions. *SIAM Journal on Applied Mathematics*, 71:714–730, 2011.

[110] K. Lipkow, S. Andrews, and D. Bray. Simulated diffusion of phosphorylated CHeY through the cytoplasm of *Escherichia coli*. *Journal of Bacteriology*, 187:45–53, 2005.

[111] W. Liu, A. F. T. Winfield, J. Sa, J. Chen, and L. Dou. Towards energy optimization: Emergent task allocation in a swarm of foraging robots. *Adaptive Behaviour*, 15:289–305, 2007.

[112] R. Lui and Z. Wang. Traveling wave solutions from microscopic to macroscopic chemotaxis models. *Journal of Mathematical Biology*, 61:739–761, 2010.

[113] R. Mayet, J. Roberz, T. Schmickl, and K. Crailsheim. Antbots: A feasible visual emulation of pheromone trails for swarm robots. In M. Dorigo et al., editor, *ANTS 2010, LNCS 6234*, pages 84–94. Springer, 2010.

[114] M. Metcalf, J. Merkin, and S. Scott. Oscillating wave fronts in isothermal chemical systems with arbitrary powers of autocatalysis. *Proceedings of the Royal Society London A*, 447:155–174, 1994.

[115] F. Mondada, M. Bonani, X. Raemy, J. Pugh, C. Cianci, A. Klaptocz, J.-C. Magnenat, S.and Zufferey, D. Floreano, and A. Martinoli. The e-puck, a robot designed for education in engineering. *Proceedings of the 9th Conference on Autonomous Robot Systems and Competitions*, 1:59–65, 2009.

[116] M. A. Montes de Oca, E. Ferrante, A. Scheidler, C. Pinciroli, M. Birattari, and M. Dorigo. Majority-rule opinion dynamics with differential latency: a mechanism for self-organized collective decision-making. *Swarm Intelligence*, 5:305–327, 2011.

[117] E. Moro. Hybrid method for simulating front propagation in reaction-diffusion systems. *Physical Review E*, 69:060101, 2004.

[118] G. Moy, B. Corry, S. Kuyucak, and S. Chung. Tests of continuum theories as models of ion channels. I. Poisson-Boltzmann theory versus Brownian dynamics. *Biophysical Journal*, 78:2349–2363, 2000.

[119] J. Murray. *Mathematical biology I: An introduction*. Springer, 3rd edition, 2002.

[120] J. Murray. *Mathematical biology II: Spatial models and biomedical applications*. Springer, 3rd edition, 2003.

[121] G. Nadin, B. Perthame, and L. Ryzhik. Traveling waves for the Keller-Segel system with Fisher birth term. *Interfaces and Free Boundaries*, 10:517–538, 2008.

[122] J. Nagumo, S. Arimoto, and S. Yoshizawa. An active pulse transmission line simulating nerve axon. *Proceedings of the IRE*, 50:2061–2070, 1962.

[123] M. Nagy, Z. Akos, D. Biro, and T. Vicsek. Hierarchical group dynamics in pigeon flocks. *Nature*, 464:890–893, 2010.

[124] T. J. Newman and R. Grima. Many-body theory of chemotactic cell-cell interactions. *Physical Review E*, 70:051916, 2004.

[125] J. Ockendon, S. Howison, A. Lacey, and A. Movchan. *Applied partial differential equations*. Oxford University Press, revised edition, 2003.

[126] P. O'Rourke and J. Brackbill. On particle-grid interpolation and calculating chemistry in particle-in-cell methods. *Journal of Computational Physics*, 103:37–52, 1993.

[127] J. Osborne, A. Walter, S. Kershaw, G. Mirams, A. Fletcher, P. Pathmanathan, D. Gavaghan, O. Jensen, P. K. Maini, and H. M. Byrne. A hybrid approach to multi-scale modelling of cancer. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 368:5013–5028, 2010.

[128] H. G. Othmer, S. Dunbar, and W. Alt. Models of dispersal in biological systems. *Journal of Mathematical Biology*, 26:263–298, 1988.

[129] H. G. Othmer and T. Hillen. The diffusion limit of transport equations II: Chemotaxis equations. *SIAM Journal on Applied Mathematics*, 62(4):1222–1250, 2002.

[130] H. G. Othmer and P. Schaap. Oscillatory cAMP signaling in the development of *Dictyostelium discoideum*. *Comments on Theoretical Biology*, 5:175–282, 1998.

[131] H. G. Othmer, X. Xin, and C. Xue. Excitation and adaptation in bacteria – a model signal transduction system that controls taxis and spatial pattern formation. *International Journal of Molecular Sciences*, 14:9205–9248, 2013.

[132] K. J. Painter and T. Hillen. Spatio-temporal chaos in a chemotaxis model. *Physica D: Nonlinear Phenomena*, 240(4-5):363 – 375, 2011.

[133] E. Parzen. *Annals of Mathematical Statistics*, 33:1065–1076, 1962.

[134] A. Patel, E. Gawlinski, S. Lemieux, and R. Gatenby. A cellular automaton model of early tumor growth and invasion. *Journal of Theoretical Biology*, 213:315–331, 2001.

[135] N. Pelechano and N. Badler. Modeling crowd and trained leader behaviour during building evacuation. *IEEE Computer Graphics and Applications*, 26:80–86, 2006.

[136] W Pfeffer. Über chemotaktische Bewegungen von Bakterien, Flagellaten und Volvocineen. *Untersuchungen aus dem Botanischen Institut in Tübingen*, 2:582–681, 1888.

[137] G. Pini, A. Brutschy, M. Frison, A. Roli, M. Dorigo, and M. Birattari. Task partitioning in swarms of robots: an adaptive method for strategy selection. *Swarm Intelligence*, 5:283–304, 2011.

[138] S. C. Pratt, D. J. T. Sumpter, E. B. Mallon, and N. R. Franks. An agent-based model of collective nest choice by the ant *Temnothorax albipennis*. *Animal Behaviour*, 70(5):1023–1036, 2005.

[139] R.D. Reitz. A study of numerical methods for reaction-diffusion equations. *SIAM Journal on Scientific and Statistical Computing*, 2:95–106, 1981.

[140] C. W. Reynolds.

[141] B. Ribba, T. Alarcón, K. Marron, P. K. Maini, and Z. Agur. The use of hybrid cellular automaton models for improving cancer therapy. In P. Sloot, B. Chopard, and A. Hoekstra, editors, *ACRI 2004, Lecture Notes in Computer Science*, volume 3305, pages 444–453. Springer, 2004.

[142] M. Robinson, M. B. Flegg, and R. Erban. Adaptive two-regime method: application to front propagation. *The Journal of Chemical Physics*, 140:124109, 2014.

[143] B. Rosenberg, L. Van Camp, and T. Krigas. Inhibition of cell division in *Escherichia coli* by electrolysis products from a platinum electrode. *Nature*, 205(4972):698–699, 1965.

[144] R. Russell. Survey of robotic applications for odor-sensing technology. *International Journal of Robotics Research*, 20(2):144–162, 2001.

[145] J. Saragosti, V. Calvez, N. Bournaveas, A. Buguin, P. Silberzan, and B. Perthame. Mathematical description of bacterial traveling pulses. *PLoS Computational Biology*, 6:e1000890, 2010.

[146] J. Saragosti, V. Calvez, N. Bournaveas, B. Perthame, A. Buguin, and P. Silberzan. Directional persistence of chemotactic bacteria in a traveling concentration wave. *Proceedings of the National Academy of Sciences*, 108(39):16235–16240, 2011.

[147] R. Satnoianu, P. K. Maini, F. Garduno, and J. Armitage. Travelling waves in a nonlinear degenerate diffusion model for bacterial pattern formation. *Discrete and Continuous Dynamical Systems B*, 1:339–362, 2001.

[148] J. Shen. Cucker-Smale flocking under hierarchical leadership. *SIAM Journal on Applied Mathematics*, 68(3):694–719, 2008.

[149] K. Smallbone, R. Gatenby, R. Gillies, P. K. Maini, and D. Gavaghan. Metabolic changes during carcinogenesis: Potential impact on invasiveness. *Journal of Theoretical Biology*, 244:703–713, 2007.

[150] V. Sperati, V. Trianni, and S. Nolfi. Evolving coordinated group behaviours through maximisation of mean mutual information. *Swarm Intelligence*, 2:73–95, 2008.

[151] P. Spiro, J. Parkinson, and H. G. Othmer. A model of excitation and adaptation in bacterial chemotaxis. *Proceedings of the National Academy of Sciences*, 94:7263–7268, 1997.

[152] A. Stevens. The derivation of chemotaxis equations as limit dynamics of moderately interacting stochastic many-particle systems. *SIAM Journal on Applied Mathematics*, 61:183–212, 2000.

[153] D. J. T. Sumpter. *Collective animal behavior*. Princeton University Press, 2010.

[154] J. Sun, E. M. Bollt, M. A. Porter, and M. S. Dawkins. A mathematical model for the dynamics and synchronization of cows. *Physica D*, 240(19):1497–1509, 2011.

[155] K. Takahashi, S. Tanase-Nicola, and P. ten Wolde. Spatio-temporal correlations can drastically change the response of a MAPK pathway. *Proceedings of the National Academy of Sciences*, 107:19820–19825, 2010.

[156] A. Tartakovsky, D. Tartakovsky, T. Scheibe, and P. Meakin. Hybrid simulations of reaction-diffusion systems in porous media. *SIAM Journal on Scientific Computing*, 30:2799–2816, 2008.

[157] J. P. Taylor-King, B. Franz, C. A. Yates, and R. Erban. Mathematical modelling of turning delays in swarm robotics. *IMA Journal of Applied Mathematics*, 2013. Accepted, available at: http://arxiv.org/abs/1401.7612.

[158] D. Tong. Kinetic theory. Course notes for the Tripos Part III course Kinetic Theory, URL: http://www.damtp.cam.ac.uk/user/tong/kintheory/kt.pdf, 2012.

[159] F. Tostevin, P. ten Wolde, and M. Howard. Fundamental limits to position determination by concentration gradients. *PLOS Computational Biology*, 3:763–771, 2007.

[160] A. E. Turgut, H. Çelikkanat, F. Gökçe, and E. Şahin. Self-organized flocking in mobile robot swarms. *Swarm Intelligence*, 2:97–120, 2008.

[161] N. van Kampen. *Stochastic processes in physics and chemistry*. North-Holland, Amsterdam, 3rd edition, 2007.

[162] J. van Zon and ten Wolde P. Simulating biochemical networks at the particle level and in time and space: Green's function reaction dynamics. *Physical Review Letters*, 94:128103, 2005.

[163] J. von Neumann. *Mathematical foundations of quantum mechanics*. Princeton University Press, 1955.

[164] G. Wagner and E. Flekkøy. Hybrid computations with flux exchange. *Philosophical Transactions of the Royal Society A: Mathematical, Physical & Engineering Sciences*, 362:1655–1665, 2004.

[165] M. Wand and M. Jones. *Kernel smoothing*. Chapman & Hall/CRC, 1994.

[166] Z.A. Wang. Mathematics of traveling waves in chemotaxis – review paper. *Discrete and Continuous Dynamical Systems Series B*, 13:601–641, 2013.

[167] N. Wiener. *Extrapolation, interpolation, and smoothing of stationary time series*. MIT Press, 1964.

[168] M. B. Witte and A. Barbul. General principles of wound healing. *Surgical Clinics of North America*, 77:509–528, 1997.

[169] L. Wolpert, R. Beddington, T. Jessel, P. Lawrence, E. Meyerowitz, and J. Smith. *Principles of development*. Oxford University Press, 2002.

[170] M. Wooldridge. *An introduction to multi-agent systems*. Wiley, 2002.

[171] C. Xue. Macroscopic equations for bacterial chemotaxis: integration of detailed biochemistry of cell signaling. *Journal of Mathematical Biology*, 70:1–44, 2015.

[172] C. Xue, E. Budrene, and H. G. Othmer. Radial and spiral streams in *Proteus mirabilis* colonies. *PLoS Computational Biology*, 7(12):e1002332, 2011.

[173] C. Xue, H. Hwang, K. J. Painter, and R. Erban. Travelling waves in hyperbolic chemotaxis equations. *Bulletin of Mathematical Biology*, 73(8):1695–1733, 2011.

[174] C. Xue and H. G. Othmer. Multiscale models of taxis-driven patterning in bacterial populations. *SIAM Journal on Applied Mathematics*, 70:133–167, 2009.

[175] P. Zhuravlev and G. Papoian. Molecular noise of capping protein binding induces macroscopic instability in filopodial dynamics. *Proceedings of the National Academy of Sciences*, 106(28):11570–11575, 2009.