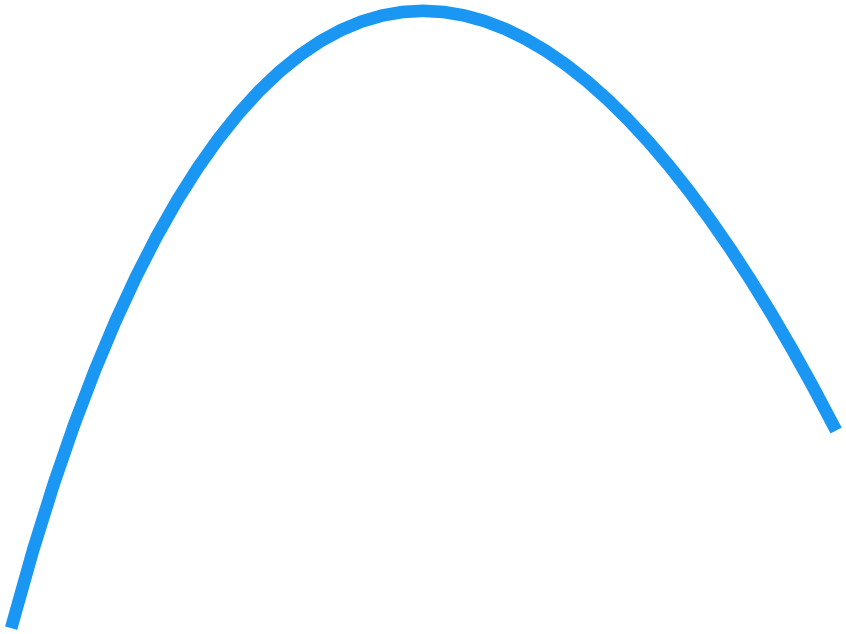

Rare Events in Reaction-Diffusion Systems

Field-theoretical Approximations and Monte Carlo Simulations

Master's thesis in Complex Adaptive Systems

EDOARDO M. MANONI

DEPARTMENT OF PHYSICS

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2025
www.chalmers.se

Master's thesis

Rare Events in Reaction-Diffusion Systems

Field-theoretical Approximations and Monte Carlo Simulations

EDOARDO M. MANONI

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Physics
Chalmers University of Technology
Gothenburg, Sweden 2025

Rare Events in Reaction-Diffusion Systems
Field-theoretical Approximations and Monte Carlo Simulations
EDOARDO M. MANONI

Supervisor: Johannes Hofmann, Department of Physics, University of Gothenburg
Examiner: Johannes Hofmann, Department of Physics, University of Gothenburg

Master's Thesis 2025
Department of Physics
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: the probability distribution for the particle number in a binary annihilation system.

Published by Chalmers Open Digital Repository
Gothenburg, Sweden 2025

Rare Events in Reaction-Diffusion Systems
Field-theoretical Approximations and Monte Carlo Simulations
EDOARDO M. MANONI
Department of Physics
Chalmers University of Technology

Abstract

Rare events are events that have near-zero probability of occurring. Despite their apparent irrelevance, when they do occur, they can have substantial, and even catastrophic, repercussions. In this thesis, we study rare events in reaction-diffusion systems, a class of mathematical models that finds various applications in physics and life sciences. We employ both theoretical and computational methods to determine the tails of the probability distribution describing the state of system.

Firstly, we follow existing literature to derive a quantum-mechanical description for entirely classical reaction-diffusion systems, called the Doi-Peliti formalism. We express the time evolution of the systems as a Feynman path integral, which we then evaluate at the saddle point to obtain a semiclassical approximation for the probability distribution, and a closed-form leading-order expression for the tails.

Secondly, we tailor a lesser-known Monte Carlo algorithm for rare probability estimation, called adaptive multilevel splitting, to compute the probability distribution of reaction-diffusion processes. We derive some theoretical results regarding its efficiency, discuss practical implementation choices, and benchmark its performance against well-understood examples.

Lastly, we compare the semiclassical approximation to the computational results, determining under which conditions the former succeeds or fails.

Keywords: rare events, reaction-diffusion, Doi-Peliti, semiclassical approximation, Monte Carlo method, adaptive multilevel splitting.

Acknowledgements

This thesis is the fruit of one year of independent work, but it would not have been possible without the help of several other people.
First and foremost, I would like to express my sincere gratitude to my supervisor, Prof Johannes Hofmann, for suggesting this topic and for his invaluable guidance throughout this endeavour, without which I would surely have been lost.
I would like to thank Enrique Rozas Garcia for always being willing to discuss my doubts and for truly going out of his way to help me.
Finally, I would like to acknowledge that none of this would have been possible without my parents' unconditional support from afar. Thank you.

Edoardo M. Manoni, Gothenburg, December 2024

Contents

1

Introduction

Rare events are events that have near-zero probability of occurring. They exist in every real-world system and, despite their apparent irrelevance, they must not be underestimated because they can, in some cases, bring about disproportionally large, and even catastrophic, repercussions, as in natural disasters (Figure 1.1), stock market crashes, mechanical failures, and so on. We would like to predict these risks so that strategies to avoid them may be developed. Moreover, some common macroscopic phenomena, such as the formation of raindrops or chemical reactions, are a consequence of events that are rare on the microscopic scale. For these reasons, it is important to be able to estimate very small probabilities in a reasonable time frame and with good accuracy.

In this thesis, we will focus on rare events that occur in so-called *reaction-diffusion systems*, a vast class of mathematical models used to describe seemingly different natural phenomena: from their original application to chemical reactions, reaction-diffusion systems have been successfully used to model population dynamics in ecology, animal coat patterns in biology, excitons recombination in physics, infectious disease spreading in medicine, and more (Figure 1.2) [4, 2, 7]. Their universality makes them an important subject of study.

In more detail, reaction-diffusion systems involve particles (or other entities with no relevant spatial extent) that move in space and also react locally with each other, potentially generating new particles or destroying old ones. Consider, as an example, a simple chemical reaction, such as the synthesis of water from hydrogen and oxygen:

$$2H_2 + O_2 \xrightarrow{\lambda} 2H_2O, \tag{1.1}$$

where λ is the reaction rate. Because the reactions are governed by the rate λ, i.e., the *average* number of reactions that occur in a unit of time, they are non-deterministic, and subject to fluctuations. Furthermore, diffusion, the motion of particles in space driven by their thermal energy, is also stochastic and characterised by another rate D, called the *diffusion constant*. It appears in the stochastic differential equation known as Langevin equation, which defines a displacement caused by diffusion as

$$dx = \sqrt{D}dW, \tag{1.2}$$

where dW is the Wiener process, whose details will not be covered here.

For these reasons, reaction-diffusion systems give rise to potentially different behaviours every time. Some of these behaviours can be expected, as they do not significantly deviate from the *average* behaviour, while others are *rare*. For example, we could ask what is the probability that the synthesis of a certain number of water molecules will take ten times the average time.

To better illustrate this point, consider the application of reaction-diffusion systems to disease-spreading dynamics. Instead of "particles", we then talk about "individuals" and we could have a system of reactions such as the following:

$$S+I \xrightarrow{\lambda_1} 2I \qquad I \xrightarrow{\lambda_2} H \qquad I \xrightarrow{\lambda_3} D. \tag{1.3}$$

Figure 1.1: A rogue wave, an unusually tall wave compared to surrounding waves, in the North Atlantic. Rogue waves are rare events capable of causing significant damage to ships and offshore structures. Due to their rarity, they were long considered to be myths until photographic evidence confirmed their existence. They have since been successfully modelled mathematically [1].

These reactions tell us that, first, infected individuals I that come into contact with susceptible individuals S may infect them with rate λ_1; second, infected individuals heal (H) with rate λ_2; and, third, they die (D) with rate λ_3. More realistic and, therefore, more complex descriptions can be made. For instance, one could take into account the category of individuals that have already been infected but are not infectious yet. In this context, a rare event to analyse is the risk of an epidemic: a sudden infection of a large part of the population, which we are all too familiar with. In the field of ecology, instead, we would be interested in the probability of a species going extinct in the future.

Notice how, formally, the water synthesis reaction (1.1) and the infection reaction (1.3) are very similar, even though they represent totally unrelated natural processes. In addition, both processes depend on the position (and movement) in space of their constituents: hydrogen and oxygen molecules need to be close to react, just like individuals need to be close to infect one another. This exemplifies the property of universality of reaction-diffusion systems: every result obtained for one case can be extended to all the others. It should be pointed out that, when considering non-physical "particles", such as humans or animals, diffusion, as a thermodynamic process, is not the exact description of their movement, but it can be taken as a relatively accurate approximation because it is very well understood theoretically.

On the topic of rare events in reaction-diffusion systems, Elgart and Kamenev [8] proposed a theoretical approximation for the probability distribution of the number of particles for the simplest kind of systems: single-species and zero-dimensional reaction processes, i.e., neglecting the effects of diffusion. However, for more complex cases, no significant progress has been made since, and the only approximation available is the mean-field approximation [9], which, by definition, is unable to capture rare events. Recently, there have been studies that show how to efficiently simulate reaction-diffusion systems using artificial neural networks, such as [10, 11], but the problem of rare events has not been addressed yet.

The objective of this thesis will be to study reaction-diffusion systems by, firstly, reviewing previous theoretical literature in Chapter 2 and, secondly, by designing an original computational approach to the problem in Chapter 3. Indeed, rare events present challenges from a computational point of view as well, since simply simulating a system until the rare event we are interested in occurs is not feasible: the rarer the event, the longer it will take. One needs to develop appropriate strategies to "guide" the

Figure 1.2: Applications of reaction-diffusion models to real-world systems. Top left: Belusov-Zhabotinsky reactions are a fascinating class of chemical reactions that give rise to complex and dynamical spatial patterns [2]. Top right: Turing instabilities explain the spontaneous formation of characteristic patterns on animal coats starting from an uniform distribution of chemical reactants [3]. Bottom left: Lotka-Volterra equations explain oscillations observed in predator-prey populations. The most notorious example is arguably the one of Canada lynxes and snowshoe hares [4]. Bottom right: excitons are electron-hole pairs generated when a semiconductor absorbs a photon. They briefly diffuse before the electron and the hole meet and react, returning at rest [5]. The picture shows the experimentally-measured wave function squared of an exciton [6].

system towards such events. In particular, we will adapt an already-existing, but lesser-known, Monte Carlo algorithm called *adaptive multilevel splitting* (AMS for short) [12, 13], which, to the best of our knowledge, has not yet been employed to analyse this type of systems yet. The obtained results will be tested on a range of problems and compared to the theoretical approximation in Chapter 4. Our proposed solution will be easily extendable to more complex systems than those we have the chance to present in this thesis.

2

Field-theoretical Approximations

In this chapter we derive an analytical approximation for the probability distribution of the number of particles in a single-species zero-dimensional reaction system. In more detail, in Section 2.1, we present the most natural description of any discrete-state stochastic process, the master equation. After pointing out its shortcomings, we introduce, in Section 2.2, the Doi-Peliti formalism, which maps the classical reaction system to a quantum-mechanical equivalent, described by an imaginary-time Schrödinger equation. Starting from it, in Section 2.3, we replicate Feynman's derivation of a path-integral description of the system. Finally, in Section 2.4, we follow [8] to evaluate the path integral at the saddle-point, obtaining a semiclassical approximation for the generating function and, consequently, for the probability distribution. This procedure yields a leading-order closed-form expression for the probability tails. In Section 2.5, we introduce a spatial dependency and present the generalisation of previously-derived equations.

2.1 Master Equation

Consider a system of particles A that react with each other according to

$$kA \xrightarrow{\lambda} lA, \tag{2.1}$$

where $k, l \in \mathbb{N}$ and $\lambda \in \mathbb{R}^+$ is the *reaction rate*. Some of the simplest reactions of this form are: $A \to 2A$ (birth or branching), $A \to \varnothing$ (death), $2A \to \varnothing$ (binary annihilation) and $2A \to A$ (coagulation). Notice that, while these reactions might not seem physically realistic, although valid in the context of a toy model, they imply the coexistence of other chemical species whose effects are time independent and already incorporated in the reaction rate. Hence, for example, the symbol $\varnothing$ does not signify a proper annihilation, but rather the transformation into a particle species that is not of interest to our model. Initially, we will also assume no spatial dependency, meaning that every particle can react with every other particle at all times; this is known as the *well-mixing hypothesis*.

We are interested in the probability $P_n(t)$ of finding $n \in \mathbb{N}$ particles in the system at time $t \in \mathbb{R}^+$, especially when $P_n(t) \ll 1$. The time evolution of the system is fully described by the *master equation*, which expresses the derivative of the probability distribution function $P_n(t)$ as the difference between a gain term and a loss term:

$$\frac{d}{dt}P_n(t) = \lambda \left[\binom{n+k-l}{k} P_{n+k-l}(t) - \binom{n}{k} P_n(t) \right], \quad n > 0,\ n+k-l > 0. \tag{2.2}$$

The gain term is proportional to the probability of reaching the state with n particles from another state, namely the one with $n+k-l$ particles, while the loss term is proportional to the probability of leaving the former state. The initial condition is generally a Poisson distribution with mean n_0, $P_n(0) = e^{n_0} n_0^n / n!$, or a fixed particle number n_0, $P_n(0) = \delta_{n,n_0}$.

Unfortunately, except for a few special cases, the master equation is unsolvable directly. However, one can often solve the corresponding *mean-field* equation obtained from (2.2) by multiplying both sides by $\sum_n n$, in the limit of small fluctuations around the mean. For example, in the case of binary annihilation,

$$\begin{aligned}\frac{d}{dt}\langle n\rangle &= \lambda \sum_n n\left[\binom{n+2}{2} P_{n+2}(t) - \binom{n}{2} P_n(t)\right] \\ &= \frac{\lambda}{2}\sum_n \left[n(n-1)(n-2) - n^2(n-1)\right] P_n(t) \\ &= \lambda\left\langle -n^2 + n\right\rangle \\ &\approx -\lambda\langle n\rangle^2,\end{aligned}$$

where, in the last line, we assumed small fluctuations, i.e, $\langle n^2\rangle \approx \langle n\rangle^2$, and also $\langle n\rangle \gg 1$. This leads to

$$\langle n\rangle(t) = \frac{1}{n_0^{-1} + \lambda t} \to \frac{1}{\lambda t}. \tag{2.3}$$

The mean-field behaviour is deterministic and does not take fluctuations into account, and thus it is entirely unable to capture rare events. For the distribution tails, a better description is needed. In the following sections we will show how to derive one making use of the quantum-mechanical formalism.

2.2 Schrödinger Equation

Many theoretical results for reaction-diffusion systems make use of the quantum-mechanical formalism, even though the systems themselves are wholly classical in nature. The following theory was developed by Doi in 1976 [14] and later expanded by Peliti in 1985 [15].

Let $|n\rangle$, $n \in \mathbb{N}$, be the *microscopic* state representing a system with n particles. Introduce the *creation* and *annihilation* operators as, respectively,

$$a^\dagger|n\rangle = |n+1\rangle, \tag{2.4a}$$
$$a|n\rangle = n|n-1\rangle. \tag{2.4b}$$

It follows that $|n\rangle = (a^\dagger)^n|0\rangle$, where $|0\rangle$ is also called *vacuum*, which we define to be normalised, $\langle 0|0\rangle = 1$. The two operators are adjoints of each other and do not commute,

$$[a, a^\dagger] = 1. \tag{2.5}$$

These operators are analogous to the quantum *bosonic* or *ladder* operators but, unlike them, they implicitly define the microstates $|n\rangle$ as a non-normalised basis of the Fock space. In fact, taking the adjoint of both sides of (2.4), we obtain their effect on bra vector states:

$$\langle n|a = \langle n+1|,$$
$$\langle n|a^\dagger = n\langle n-1|.$$

Then,

$$\begin{aligned}\langle m|n\rangle &= \langle m-1|a|n\rangle \\ &= n\langle m-1|n-1\rangle\end{aligned}$$

and so, recursively,

$$\begin{aligned}\langle m|n\rangle &= n(n-1)...(n-m+1)\langle 0|n-m\rangle \\ &= n!\delta_{m,n}.\end{aligned}$$

Moreover, their combination is the *particle number* operator and the $|n\rangle$'s are its eigenstates:

$$a^\dagger a|n\rangle = n|n\rangle. \tag{2.6}$$

Now let us define the *superposition* state of the system

$$|\Psi(t)\rangle = \sum_{n=0}^{\infty} P_n(t)|n\rangle. \tag{2.7}$$

Carrying on with binary annihilation as an example, we can then write

$$\begin{aligned}\frac{d}{dt}|\Psi(t)\rangle &= \frac{d}{dt}\sum_n P_n(t)|n\rangle \\ &= \frac{\lambda}{2}\sum_n \left[\binom{n+2}{2}P_{n+2}(t) - \binom{n}{2}P_n(t)\right]|n\rangle \\ &= \frac{\lambda}{2}\sum_n \left[n(n-1)P_n(t)|n-2\rangle - n(n-1)P_n(t)|n\rangle\right] \\ &= \frac{\lambda}{2}\sum_n P_n(t)\left[a^2 - (a^\dagger)^2a^2\right]|n\rangle \\ &= -H|\Psi(t)\rangle\end{aligned} \tag{2.8}$$

with the quasi-Hamiltonian operator

$$H[a^\dagger, a] = \lambda/2[(a^\dagger)^2 - 1]a^2. \tag{2.9}$$

For a general reaction (2.1), $H[a^\dagger, a] = \lambda/k![(a^\dagger)^k - (a^\dagger)^l]a^k$. Moreover, for a system with mutiple concurring reactions, the Hamiltonian is simply the sum of the independent Hamiltonians of each reaction, with their own rates. If the system also presents multiple particle species, then the operators must carry a species-specific index [16]. For example, for a Lotka-Volterra system (4.7),

$$H[a^\dagger, a] = \sigma[a^\dagger_{N_1} - 1]a_{N_1} + \mu a^\dagger_{N_2}[1 - a^\dagger_{N_2}]a_{N_2} + \lambda a^\dagger_{N_1}[a^\dagger_{N_2} - a^\dagger_{N_1}]a_{N_1}a_{N_2}.$$

The differential equation (2.8) is formally equivalent to a Schrödinger equation and is solved by

$$|\Psi(t)\rangle = e^{-Ht}|\Psi(0)\rangle, \tag{2.10}$$

where $\exp(-Ht)$ is the *propagator*, and the initial condition for a Poisson with mean n_0 distribution is $|\Psi(0)\rangle = \exp[n_0(a^\dagger - 1)]|0\rangle$, or $|\Psi(0)\rangle = |n_0\rangle$, for a fixed number of particles n_0 instead. Equation (2.10) allows us to compare the stochasticity of classical non-equilibrium system to that of quantum systems, in order to make use of the powerful tools already developed for the latter.

We can now use (2.10) to write the expectation value of an observable quantity $A(n)$ of the system [16]. First, notice the identities

$$\langle 0|e^a|n\rangle = \langle 0|\sum_{m=0}^{\infty}\frac{a^m}{m!}|n\rangle = \langle 0|0\rangle = 1$$

and

$$e^a f[a^\dagger, a] = f[a^\dagger + 1, a]e^a,$$

which follows from (2.5). Thus,

$$\begin{aligned}\langle A\rangle(t) &= \sum_n A(n)P_n(t) \\ &= \sum_n \langle 0|e^a|n\rangle A(n)P_n(t) \\ &= \langle 0|e^a A[a^\dagger, a]|\Psi(t)\rangle,\end{aligned}$$

where $A[a^\dagger, a]$ is an appropriate operator-valued function when applied to a superposition state. Then,

$$\begin{aligned}\langle A\rangle(t) &= \langle 0|e^a A[a^\dagger, a]e^{-tH[a^\dagger,a]}|\Psi(0)\rangle \\ &= \langle 0|A[a^\dagger + 1, a]e^{-tH[a^\dagger+1,a]}e^a|\Psi(0)\rangle,\end{aligned} \tag{2.11}$$

where the last operation is usually called *Doi-shift*.

The observable we are interested in is $A(z, n) = z^n$, so that $\langle A\rangle(t)$ is the probability generating function $G(z, t)$. In this case, the corresponding operator-valued function is $A[a^\dagger, a] = \exp[(z - 1)a]$:

$$\begin{aligned}\langle 0|e^a e^{(z-1)a}|\Psi(t)\rangle &= \sum_m \frac{z^m a^m}{m!} \sum_n P_n(t)|n\rangle \\ &= \sum_m \frac{z^m}{m!} \sum_{n\geq m} \frac{n!}{(n-m)!} P_n(t)\langle 0|n - m\rangle \\ &= \sum_m z^m P_m(t).\end{aligned}$$

We are not aware of any general rule to obtain $A[a^\dagger, a]$ from $A(n)$.

Our objective will be to approximate $G(z, t)$ and subsequently extract $P_n(t)$ from it using the inverse formula

$$P_n(t) = \frac{1}{2\pi i} \oint G(z, t)z^{-(n+1)}dz, \tag{2.12}$$

where the integration is performed over a closed contour encircling $z = 0$ in the region of analycity of G in the complex plane. Equation (2.12) is an immediate consequence of Cauchy's *residue theorem*, since $P_n(t)$ is the residue of $\sum_m P_m(t)(z - z_0)^{m-n-1}$ at $z_0 = 0$.

2.3 Path Integral

In this section, we will follow the standard path integral derivation from (2.11) to express the same information therein contained in another form, which more easily lends itself to approximation. We start by defining the *coherent states* $|\phi\rangle$, $\phi \in \mathbb{C}$, as

$$|\phi\rangle = e^{\phi a^\dagger}|0\rangle = \sum_{n=0}^{\infty} \frac{\phi^n (a^\dagger)^n}{n!}|0\rangle = \sum_{n=0}^{\infty} \frac{\phi^n}{n!}|n\rangle. \tag{2.13}$$

The last equality shows they are similar to Poisson-distributed superposition states, although they lack the normalisation constant $e^{-\phi}$ and their "mean" can be complex. They are the eigenstates of

the annihilation operator:

$$a|\phi\rangle = a\sum_{n=0}^{\infty}\frac{\phi^n}{n!}|n\rangle = \sum_{n=0}^{\infty}\frac{\phi^n}{(n-1)!}|n-1\rangle = \phi\sum_{n=0}^{\infty}\frac{\phi^n}{n!}|n\rangle = \phi|\phi\rangle.$$

Likewise,

$$\langle\phi|a^\dagger = \phi^*\langle\phi|.$$

They form an overcomplete basis:

$$\begin{aligned}\langle\phi_1|\phi_2\rangle &= \left(\sum_n\langle 0|\frac{(\phi_1^*)^n a^n}{n!}\right)\left(\sum_m\frac{(\phi_2)^m(a^\dagger)^m}{m!}|0\rangle\right)\\ &= \sum_{n,m}\frac{(\phi_1^*)^n\phi_2^m}{n!m!}\langle n|m\rangle\\ &= e^{\phi_1^*\phi_2}.\end{aligned}$$

Consider the following intermediate result:

$$\begin{aligned}\int_{\mathbb{C}} d\phi^* d\phi e^{-\phi^*\phi}(\phi^*)^m\phi^n &= 2i\int_{\mathbb{R}}dx\int_{\mathbb{R}}dy\ e^{-(x-iy)(x+iy)}(x-iy)^m(x+iy)^n\\ &= 2i\int_0^\infty dr\ e^{-r^2}r^{n+m+1}\int_0^{2\pi}d\theta e^{i\theta(m-n)}\\ &= 4\pi i\int_0^\infty dr\ e^{-r^2}r^{n+m+1}\delta_{n,m}\\ &= 2\pi i\delta_{n,m}\Gamma(1+n)\\ &= 2\pi i n!\delta_{n,m}.\end{aligned}$$

Now we can write the identity operator in terms of coherent states, also called *closure relation*, as

$$\begin{aligned}I &= \sum_n\frac{1}{n!}|n\rangle\langle n|\\ &= \sum_{n,m}\frac{1}{n!}|n\rangle\langle m|\delta_{n,m}\\ &= \sum_{n,m}\frac{1}{n!}|n\rangle\langle m|\frac{1}{2\pi i m!}\int d\phi^* d\phi\ e^{-\phi^*\phi}(\phi^*)^m\phi^n\\ &= \int\frac{d\phi^* d\phi}{2\pi i}e^{-\phi^*\phi}|\phi\rangle\langle\phi|.\end{aligned}\tag{2.14}$$

Finally, we return to Equation (2.11) and apply Trotter's decomposition formula

$$e^{-Ht} = \lim_{N\to\infty}(e^{\frac{t}{N}H})^N = \lim_{N\to\infty}(e^{\Delta tH})^N$$

to write the propagator as an infinite product of infinitesimal propagators, each representing the system evolution in an infinitesimal time interval Δt:

$$\langle A\rangle(t) = \langle 0|A[a^\dagger+1,a]e^{-\Delta tH}\ ...\ e^{-\Delta tH}e^a|\Psi(0)\rangle.$$

Now we insert the identity (2.14) around every propagator, marking every coherent state with an unique index t_j, starting with $t_0 = 0$ and ending with $t_N = t$:

$$\begin{aligned}\langle A\rangle(t) &= \langle 0|A[a^\dagger+1,a]Ie^{-\Delta tH}I\ ...\ Ie^{-\Delta tH}Ie^a|\Psi(0)\rangle \\ &= \langle 0|A[a^\dagger+1,a]\int\frac{d\phi^*_{t_N}d\phi_{t_N}}{2\pi i}e^{-\phi^*_{t_N}\phi_{t_N}}|\phi_{t_N}\rangle\langle\phi_{t_N}|e^{-\Delta tH}\ ... \\ &\quad ...\ e^{-\Delta tH}\int\frac{d\phi^*_{t_0}d\phi_{t_0}}{2\pi i}e^{-\phi^*_{t_0}\phi_{t_0}}|\phi_{t_0}\rangle\langle\phi_{t_0}|e^a|\Psi(0)\rangle \\ &= \int ...\int\left(\prod_{j=0}^{N}\frac{d\phi^*_{t_j}d\phi_{t_j}}{2\pi i}\right)e^{-\phi^*_{t_N}\phi_{t_N}-...-\phi^*_{t_0}\phi_{t_0}}\langle 0|A[a^\dagger+1,a]|\phi_{t_N}\rangle \\ &\quad \langle\phi_{t_N}|e^{-\Delta tH}|\phi_{t_{N-1}}\rangle\ ...\ \langle\phi_{t_1}|e^{-\Delta tH}|\phi_{t_0}\rangle\langle\phi_{t_0}|e^a|\Psi(0)\rangle. \end{aligned} \tag{2.15}$$

In the limit $N\to\infty$, this expression simplifies to

$$\begin{aligned}\langle\phi_{t_j}|e^{-\Delta tH[a^\dagger+1,a]}|\phi_{t_{j-1}}\rangle &= \langle\phi_{t_j}|I-\Delta tH[a^\dagger+1,a]|\phi_{t_{j-1}}\rangle \\ &= \langle\phi_{t_j}|\phi_{t_{j-1}}\rangle-\langle\phi_{t_j}|\Delta tH[a^\dagger+1,a]|\phi_{t_{j-1}}\rangle.\end{aligned}$$

Assuming a normal-ordered Hamiltonian, meaning that all $a^\dagger$'s are to the left and all a's are to the right, which is always obtainable using (2.5),

$$\begin{aligned}\langle\phi_{t_j}|\phi_{t_{j-1}}\rangle-\langle\phi_{t_j}|\Delta tH[a^\dagger+1,a]|\phi_{t_{j-1}}\rangle &= \langle\phi_{t_j}|\phi_{t_{j-1}}\rangle-\langle\phi_{t_j}|\Delta tH[\phi^*_{t_j}+1,\phi_{t_{j-1}}]|\phi_{t_{j-1}}\rangle \\ &= e^{\phi_{t_j}\phi_{t_{j-1}}}(I-\Delta tH[\phi^*_{t_j}+1,\phi_{t_{j-1}}]) \\ &= e^{\phi_{t_j}\phi_{t_{j-1}}}e^{-\Delta tH[\phi^*_{t_j}+1,\phi_{t_{j-1}}]}.\end{aligned}$$

Moreover,

$$\langle 0|A[a^\dagger+1,a]|\phi_{t_N}\rangle = A[1,\phi_{t_N}]\langle 0|\phi_{t_N}\rangle = A[1,\phi_{t_N}] = A[\phi(t)], \tag{2.16}$$

where $A[\phi(t)]$ is simply obtained by replacing n with $\phi(t)$ in $A(n)$.

Combining all together in (2.15), we obtain the *path integral* expression for the expectation value:

$$\langle A\rangle(t) = \int\mathcal{D}[\phi^*,\phi]A[\phi(t)]e^{-S[\phi^*,\phi]}, \tag{2.17}$$

where

$$\mathcal{D}[\phi^*,\phi] = \prod_{j=0}^{N}\frac{d\phi^*_{t_j}d\phi_{t_j}}{2\pi i}$$

is the integration measure, and

$$S[\phi^*,\phi] = \sum_{j=0}^{N}\left(\phi^*_{t_j}(\phi_{t_j}-\phi_{t_{j-1}})+\Delta tH[\phi^*_{t_j}+1,\phi_{t_{j-1}}]\right)+\phi^*_{t_0}\phi_{t_0}-\ln\langle\phi_{t_0}|e^a|\Psi(0)\rangle \tag{2.18}$$

is the *Doi-Peliti action*. The last term $\ln\langle\phi_{t_0}|e^a|\Psi(0)\rangle$ is equal to $n_0\phi^*_{t_0}$ for an initial Poisson distribution and equal to $n_0\ln(1+\phi^*_{t_0})$ for an initial fixed number of particles. In the first case, which is the one we will consider, it cancels with the only other term outside the sum, $\phi^*_{t_0}\phi_{t_0}$, by recalling that $\phi(t) = n$ follows from (2.16). The boundary condition $\phi(0) = n_0$ also follows from (2.16). In continuous notation,

$$S[\phi^*,\phi] = \int_0^t dt'\ \left(\phi^*(t')\frac{d}{dt'}\phi(t')+H[\phi^*(t')+1,\phi(t')]\right). \tag{2.19}$$

In words, Equation (2.17) defines the expectation value of an observable as its average over all possible states the system can be in after having evolved as described by the path, or trajectory, $\phi(t')$ from $t' = 0$ to $t' = t$. Each value is weighted according to an analogue of the classical action along the corresponding path.

Notice that the master equation (2.2), the Schrödinger equation (2.8) and the path integral (2.17) are all equivalent to one another, and carry the same amount of information. Until now, no approximation has been made.

2.4 Semiclassical approximation

Since the "weights" decay exponentially, the integral (2.17) is dominated by the classical solution (ϕ^*_{cl}, ϕ_{cl}), i.e, the stationary path of the action (2.18), in which we also have to include the term $-\ln(A[\phi_{t_N}])$. In general, we can expand the action around the stationary path to get

$$S[\phi^*, \phi] = S[\phi^*_{cl}, \phi_{cl}] + O((\Delta\phi^*)^2, \Delta\phi^*\Delta\phi, \Delta\phi^2) \tag{2.20}$$

since, of course, the first functional derivative is zero by definition. Then, changing the integration variables from ϕ^*, ϕ to $\Delta\phi^* = \phi^* - \phi^*_{cl}$, $\Delta\phi = \phi - \phi_{cl}$ and taking out the first term, which does not depend on the latter pair, we can write

$$\langle A\rangle(t) = \mathcal{N} e^{-S[\phi^*_{cl}, \phi_{cl}]}, \tag{2.21}$$

where $\mathcal{N}$ is a proportionality constant, representing the remaining integral over the deviations from the classical path. When referring to distributions, $\mathcal{N}$ is the normalisation constant. This is known as the *semiclassical* or *saddle-point* approximation.

To obtain an expression for (2.21), we follow [8] and begin by determining the classical path. Solving $\partial S[\phi^*, \phi]/\partial\phi^*_{t_k} = 0$ gives

$$\begin{cases} \dfrac{\phi_{t_k} - \phi_{t_{k-1}}}{\Delta t} = -\dfrac{H[\phi^*_{t_k} + 1, \phi_{t_{k-1}}]}{\partial\phi^*_{t_k}} & 1 \le k \le N \\ \\ \phi_{t_0} = \dfrac{\partial \ln\langle\phi_{t_0}|e^a|\Psi(0)\rangle}{\partial\phi^*_{t_0}} & k = 0 \end{cases},$$

while $\partial S[\phi^*, \phi]/\partial\phi_{t_k} = 0$ gives

$$\begin{cases} \dfrac{\phi^*_{t_{k+1}} - \phi^*_{t_k}}{\Delta t} = \dfrac{H[\phi^*_{t_{k+1}} + 1, \phi_{t_k}]}{\partial\phi_{t_k}} & 0 \le k \le N-1 \\ \\ \phi^*_{t_N} = -\dfrac{\partial \ln(A[\phi_{t_N}])}{\partial\phi_{t_N}} & k = N \end{cases}.$$

In continuous notation, we obtain the classical equations of motion of the system:

$$\begin{cases} \dfrac{\partial\phi(t')}{\partial t'} = -\lambda(1 + \phi^*(t'))\phi^2(t') \\ \\ \dfrac{\partial\phi^*(t')}{\partial t'} = \lambda((1 + \phi^*(t'))^2 - 1)\phi(t') \end{cases} \tag{2.22}$$

with boundary conditions

$$\begin{cases} \phi(0) = n_0 \\ \\ \phi^*(t) = z - 1 \end{cases}. \tag{2.23}$$

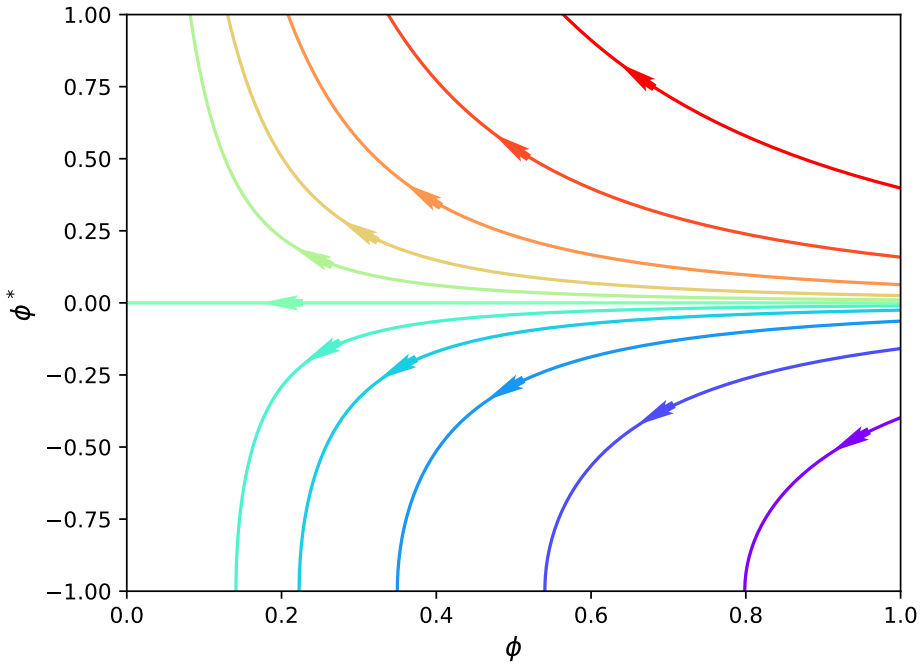

Figure 2.1: The phase portrait of the PDE system (2.22) around the zero-energy trajectory.

This is a conservative system with the energy

$$E = H[\phi^*(t') + 1, \phi(t')] \tag{2.24}$$

as integral of motion, where H is given by substituting $a^\dagger$ and a with ϕ^* and ϕ, respectively, in (2.9). To solve it, start by rewriting (2.24) as

$$\phi^*_{cl}(t') = \frac{\sqrt{(\phi^2_{cl}(t) + 2E/\lambda)}}{\phi_{cl}(t)} - 1,$$

where we have kept only the positive solution since trajectories with positive energy must have $\phi^*_{cl}(t') > 0\ \forall t'$, as inferred from (2.24), and also shown in Figure 2.1, which does not hold for the negative solution. Then, using $\phi_{cl}(0) = n_0 \gg 1$, $\phi^*_{cl}(0) \approx 0$ follows for trajectories with finite E, correctly approaching zero from the direction given by the sign of E. Now that we have initial and final conditions for $\phi^*_{cl}(t)$, we can compute the time it takes for a trajectory of energy E to reach $z - 1$:

$$\begin{aligned}
t &= \int_0^t dt' \\
&= \int_0^{z-1} d\phi^*_{cl}\ \frac{dt'}{d\phi^*_{cl}} \\
&= \int_0^{z-1} \frac{d\phi^*_{cl}}{\lambda((1+\phi^*_{cl}(t))^2 - 1)\phi_{cl}(t)} \\
&= \frac{1}{\sqrt{2\lambda E}} \int_0^{z-1} \frac{d\phi^*_{cl}}{\sqrt{(\phi^*_{cl})^2 + 2\phi^*_{cl}}} \\
&= \frac{\ln(z + \sqrt{z^2 - 1})}{\sqrt{2\lambda E}}.
\end{aligned}$$

Equivalently,

$$E = -\frac{\arccos^2(z)}{2\lambda t^2}. \tag{2.25}$$

Since the energy is constant along a classical trajectory, we can take it out of the integral in (2.19) and simplify the action further:

$$\begin{aligned}
S[\phi^*_{cl}, \phi_{cl}] &= Et + \int_0^t dt' \ \left(\phi^*_{cl}(t')\frac{d}{dt'}\right)\phi_{cl}(t') - (z-1)\phi(t) \\
&= Et + [\phi^*_{cl}(t')\phi_{cl}(t')]_0^t - \int_0^t dt' \ \left(\frac{d}{dt'}\phi^*_{cl}(t')\right)\phi_{cl}(t') - (z-1)\phi(t) \\
&= Et - \int_0^t dt' \ \frac{d}{dt'}\phi^*_{cl}(t')\phi_{cl}(t') \\
&= Et - \int_0^{z-1} d\phi^*_{cl}(t') \ \phi_{cl}(t') \\
&= Et - \int_0^{z-1} d\phi^*_{cl}(t') \ \sqrt{\frac{2E}{\lambda((1+\phi^*_{cl}(t'))^2 - 1)}} \\
&= -\sqrt{\frac{E}{2\lambda}}\ln(z + \sqrt{z^2-1}) \\
&= \frac{\arccos^2(z)}{2\lambda t}.
\end{aligned} \tag{2.26}$$

Right away, we can put the semiclassical approximation to the test by using one of the fundamental properties of generating functions:

$$\langle n \rangle = \lim_{z\to 1}\frac{\partial}{\partial z}G(z,t) = \mathcal{N}\lim_{z\to 1}\frac{\partial}{\partial z}e^{-S[\phi^*_{cl},\phi_{cl}]} = \frac{\mathcal{N}}{\lambda t}, \tag{2.27}$$

which is precisely the mean-field approximation (2.3) for $\mathcal{N} = 1$, i.e., under the mean-field assumption that the higher-order terms in (2.20) vanish.

Now that we have an approximate expression for $G(z,t)$, in order to solve (2.12) and obtain an expression for $P_n(t)$, we will perform a second saddle-point approximation, since the integrand is once again an exponential. We set the saddle-point condition:

$$\frac{d}{dz}\left(-\frac{\arccos^2(z)}{2\lambda t} - \ln(z)(n+1)\right) = 0.$$

Assuming $n \gg 1$,

$$\frac{n}{\langle n \rangle} = \frac{z\arccos(z)}{\sqrt{1-z^2}}. \tag{2.28}$$

This equation must be solved numerically for z, and then substituted in (2.26). The resulting numerical function of $n/\langle n \rangle$ is what we will refer to as the semiclassical approximation throughout the rest of the thesis.

To obtain a closed-form analytical result, we consider different values for the ratio $n/\langle n \rangle$ and approximate to the leading order. Consistently with (2.27), $n/\langle n \rangle = 1$ is solved by $z = 1$. Thus, rare events $n \ll \langle n \rangle$ and $n \gg \langle n \rangle$ are given by $0 < z \ll 1$ and $z \gg 1$, respectively. We expand the right-hand side

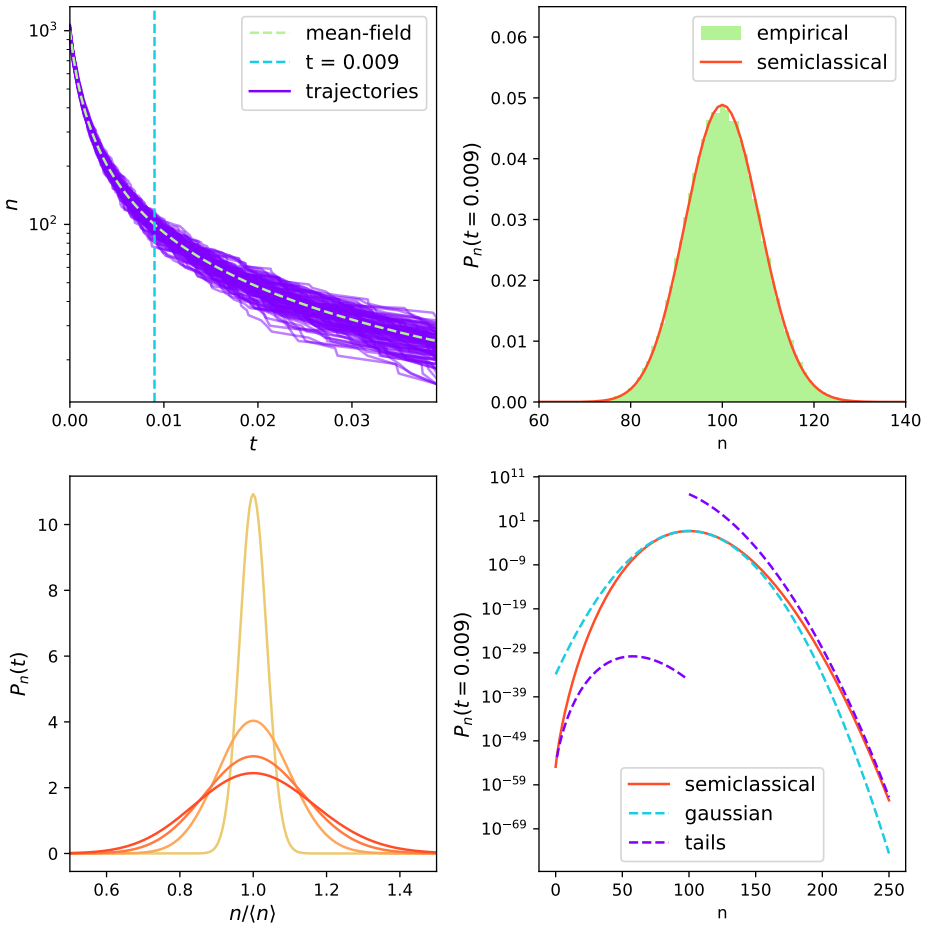

Figure 2.2: The binary annihilation process with $\lambda = 1$ and $n_0 = 10^3$. Top left: sample trajectories and mean-field behaviour. Top right: the empirical distribution for the number of particles n at $t = 0.009$, when $\langle n \rangle = 100$, compared to the semiclassical approximation. Bottom left: the semiclassical approximation at four evenly-spaced, subsequent times. Bottom right: the semiclassical approximation and its three limit cases (2.29).

of (2.28) in these three limiting cases:

$$\frac{z \arccos(z)}{\sqrt{1-z^2}} = \begin{cases} \dfrac{\pi z}{2} & z \ll 1 \\ \dfrac{2z+1}{3} & z \approx 1 \\ \ln(2z) & z \gg 1 \end{cases} .$$

Now, we invert these expressions to extract z as a function of $n/\langle n \rangle$ and substitute in (2.12), finally

obtaining expressions for the probability distribution function:

$$-\ln(P_n(t)) = \begin{cases} \dfrac{\pi^2\langle n\rangle}{8} + n\ln\left(\dfrac{2n}{\pi\langle n\rangle}\right) & n \ll \langle n\rangle \\ \dfrac{3}{4}\dfrac{(n-\langle n\rangle)^2}{\langle n\rangle} & n \approx \langle n\rangle \\ \dfrac{1}{2}\dfrac{n^2}{\langle n\rangle} - \ln(2)n & n \gg \langle n\rangle \end{cases} . \tag{2.29}$$

For $n \approx \langle n\rangle$, the distribution is Gaussian, entailing the normalisation constant

$$\mathcal{N} = \sqrt{\frac{3}{4\pi\langle n\rangle}}, \tag{2.30}$$

which we will use as the generating function coefficient in (2.21). In this case, $\mathcal{N}$ is obtained under a less-stringent assumption (Gaussian distribution) compared to the mean-field one (deterministic process) from (2.27).

In Figure 2.2 we first present the numerically-inverted solution compared to the empirically-observed distribution, demonstrating its accuracy for non-rare probabilities (down to $\sim 10^{-2}$), and its time evolution. We also show that the leading-order tail approximations are correct, but only at extremely low probabilities, while the Gaussian is to be preferred at more reasonable value ranges.

2.5 Diffusion

Now we would like to introduce a spatial dependency to our reaction system [16, 17]. We assume that the particles hop, with *hopping rate* D, on a finite-dimensional lattice L^d, with a total of N sites. The state of the system is uniquely determined by the list of the particles numbers at each site $\{n_i, 1 \le i \le N\} = n_1, n_2, ..., n_N$, called *occupation numbers*. Particles can only react locally, i.e, if they are on the same site. The Master equation (2.2) generalises to

$$\begin{aligned} \frac{d}{dt}P_{\{...,n_i,...\}}(t) = \lambda \sum_i \left[\binom{n_i+k-l}{k} P_{\{...,n_i+k-l,...\}}(t) - \binom{n_i}{k} P_{\{...,n_i,...\}}(t) \right] \\ + \frac{D}{2d}\sum_{(i,j)} [n_i P_{\{...,n_i+1,...,n_j-1,...\}} + n_j P_{\{...,n_i-1,...,n_j+1,...\}}(t) \\ - (n_i+n_j)P_{\{...,n_i,...,n_j,...\}}], \end{aligned} \tag{2.31}$$

where (i,j) are all the adjacent-sites couples. The term $1/2d$ will be included in the rate D henceforth for clarity. As one can observe, the reaction and diffusion components are independent of one another.

We can also expand the definition of the ladder operators (2.4) to work on one specific site as follows:

$$a_i^\dagger|...,n_i,...\rangle = |...,n_i+1,...\rangle \tag{2.32}$$

$$a_i|...,n_i,...\rangle = n_i|...,n_i-1,...\rangle. \tag{2.33}$$

All relationships discussed in Section 2.2 can be rewritten in the same manner. In particular, as a consequence of (2.31), the Hamiltonian of this system is also made up of two distinct components:

$$H = \frac{\lambda}{2}\sum_i ((a_i^\dagger)^2 - 1)a_i^2 + D\sum_{(i,j)}(a_i^\dagger - a_j^\dagger)(a_i - a_j). \tag{2.34}$$

Notice that the diffusive part remains unchanged after being Doi-shifted.

In the continuous path integral formulation, $\sum_i$ becomes $a_0^{-d}\int d^dx$, where $a_0 \to 0$ is the lattice constant. The fields $\phi^*(t')$ and $\phi(t')$ are now space-dependent and become, respectively, $\hat{\phi}(x,t')$ and $a_0^d\phi(x,t')$. Notice that the latter, which still represents the observable state of the system, is now defined as a *density of particles*, while the former remains dimensionless. The action (2.19), in turn , becomes

$$S[\hat{\phi}(x,t'),\phi(x,t')] = \int d^dx \left[\int_0^t dt' \left(\hat{\phi}(x,t')\frac{d}{dt'}\phi(x,t') + H[\hat{\phi}(x,t')+1,\phi(x,t')]\right) - (z(x)-1)\right], \tag{2.35}$$

where

$$H[\hat{\phi}(x,t')+1,\phi(x,t')] = \tilde{\lambda}[(\hat{\phi}(x,t')+1)^2-1] + \tilde{D}\nabla\hat{\phi}(x,t')\nabla\phi(x,t'), \tag{2.36}$$

$\tilde{D} = a_0^2 D$, $\tilde{\lambda} = a_0^d/2\lambda$, and $z(x)$ is the generating-function input field. In particular, the multivariate generating function is

$$G_{\{z_i\}}(t) = \sum_{\{n_i\}} z_1^{n_1}...z_N^{n_N} P_{\{n_i\}}(t). \tag{2.37}$$

The equations of motion become

$$\begin{cases} \dfrac{\partial\phi(x,t')}{\partial t'} = -\tilde{\lambda}(1+\hat{\phi}(x,t'))\phi^2(x,t') + \tilde{D}\nabla^2\phi(x,t') \\ \dfrac{\partial\hat{\phi}(x,t')}{\partial t'} = \tilde{\lambda}((1+\hat{\phi}(x,t'))^2-1)\phi(x,t') + \tilde{D}\nabla^2\hat{\phi}(x,t') \end{cases} \tag{2.38}$$

with boundary conditions

$$\begin{cases} \phi(x,0) = n_0(x) \\ \hat{\phi}(x,t) = z(x) - 1 \end{cases}. \tag{2.39}$$

The mean-field solution is again obtained by setting $z(x) = 1$:

$$\frac{\partial\phi(x,t)}{\partial t} = -\tilde{\lambda}\phi^2(x,t) + \tilde{D}\nabla^2\phi(x,t), \tag{2.40}$$

but in general system (2.38) is not analytically solvable. A numerical alternative is presented in [8], but it falls outside the scope of this thesis. Nonetheless, the computational methods presented in the next chapter offer an alternative also applicable to systems with a spatial component, as we will show.

3

Monte Carlo Simulations

In this chapter we present the main focus of this thesis, an algorithm for rare probability estimation known as importance splitting. We provide a theoretical description that we use to derive an expression for its accuracy. Then, we explain how to simulate stochastic reaction processes. In Section 3.1, we introduce the standard Monte Carlo method and show why it is unfit to estimate very small probabilities. In Section 3.2, we give an overview of importance sampling, the most common variance-reduction technique for direct sampling from distributions. In Section 3.3, we formulate the general idea behind importance splitting, which is particularly suited for Markov processes such as those we aim to study. We present two further refinements to the algorithm in Sections 3.4 and 3.5, the latter of which, AMS, we will use it throughout the rest of the thesis. In Sections 3.6 and 3.7, we discuss the optimal choices for the only two free parameters of the algorithm. In Section 3.9, we compare two different ways to simulate stochastic reaction processes, and, in Section 3.10, we show how diffusion can also be simulated using the same tools. Finally, in Section 3.11, we suggest how to reduce memory occupation when storing reaction-diffusion trajectories for the purposes of AMS.

3.1 Standard Monte Carlo

Appropriate methods have to be used to compute the probabilities of rare events, as direct sampling is not computationally efficient. To understand why this is the case, let us consider the *standard*, also known as *naive*, *Monte Carlo* method, or SMC for short.

Suppose the state of the system is represented by the random variable $X \in \mathcal{X} \subseteq \mathbb{R}^d$ and that we are interested in the event $X \in A \subset \mathcal{X}$, which we know to be rare, i.e., $P = P(X \in A) \ll 1$. A trivial estimator of P consists in applying the classical definition of probability by simply generating N different independent realisations of X and counting the number of times event A occurs:

$$\hat{P}_{\mathrm{SMC}}(X \in A) = \frac{1}{N} \sum_{i=1}^{N} I_{X_i \in A}, \tag{3.1}$$

where I is the indicator function, which is by definition Bernoulli-distributed with parameter P. This estimator is unbiased, meaning that its expectation value is the true value that is being estimated:

$$\mu = E[\hat{P}_{SMC}] = \frac{1}{N} \sum_{i=1}^{N} E[I_{X_i \in A}] = \frac{1}{N} \sum_{i=1}^{N} P = P.$$

The difference $\mu - P$ is called *bias*. Unbiasedness is an essential property for an estimator as it guarantees that, as long as its variance does not diverge, one can obtain an arbitrarily accurate estimate by averaging over an increasing number of independent runs of the algorithm (in this case, N). Of course, unbiasedness does not guarantee efficiency; that depends on the variance, which measures how

fast μ converges to P. In the case of our estimator it is equal to

$$\begin{aligned}
\sigma^2 = Var[\hat{P}_{SMC}] &= E[\hat{P}_{SMC}^2] - E[\hat{P}_{SMC}]^2 \\
&= \frac{1}{N^2} \sum_{i,j} E[I_{X_i \in A} I_{X_j \in A}] - P^2 \\
&= \frac{1}{N^2} \left(\sum_{i=j} E[I_{X_i \in A}] + \sum_{i \neq j} E[I_{X_i \in A} I_{X_j \in A}] \right) - P^2 \\
&= \frac{1}{N^2} \left(NP + \left(N^2 - N \right) P^2 - N^2 P^2 \right) \qquad (3.2) \\
&= \frac{P - P^2}{N}. \qquad (3.3)
\end{aligned}$$

According to the central limit theorem, $\hat{P}_{\mathrm{SMC}}$ is therefore normally distributed with mean μ and variance σ^2.

The relative error

$$\eta = \frac{\sigma}{\mu} = \frac{1}{\sqrt{N}} \sqrt{\frac{1}{P} - 1} \approx \frac{1}{\sqrt{NP}} \quad \text{for } P \ll 1 \qquad (3.4)$$

shows why this method is ineffective for rare events: large sample sizes N do not guarantee accurate results when P is very small. For example, if one wanted a 1% accuracy when estimating a probability of order 10^{-10}, one would need to generate around 10^{14} independent samples. Such a large amount of data would require an impossibly long computation time even for the simplest systems. A practical instance of this problem is discussed in Section 4.1 together with a proposed solution.

Historically, Monte Carlo simulations were invented in the 1940s by von Neumann and Ulam to numerically solve integro-differential equations describing neutron diffusion or transport through an isotropic medium [18]. In fact, the estimator (3.1) can be generalised as an approximation of the integral of a function f over a domain $\mathcal{X}$ with volume V by making use of the law of large numbers:

$$\lim_{N \to \infty} \frac{1}{N} \sum_{i=1}^{N} f(X_i) = \frac{1}{V} \int_{\mathcal{X}} f(x) dx, \qquad (3.5)$$

where the random variables X_i are uniformly distributed over $\mathcal{X}$.

Since $\sigma \sim O(\frac{1}{\sqrt{N}})$, as proved in (3.2), Monte Carlo methods are often used for multidimentional integration, as the accuracy does not depend on the dimension d itself and, because of this, they outperform the trapezoidal rule for $d > 4$, which instead converges as $O(N^{-2/d})$. Considering the pervasiveness of complex systems in nature that can be described by higher-dimensional systems of partial differential equations, it is not difficult to see why Monte Carlo integration is a fundamental part of computational mathematics and is extensively used in all fields of science. Numerous variants exist to tackle a wide range of problems and, in the following subsections, we will focus on those specific for small probabilities estimation.

3.2 Importance Sampling

Before describing the main algorithm that will be used throughout this thesis, let us introduce briefly another very popular variance reduction technique. *Importance sampling* was invented by Kahn and Harris quickly after the invention of Monte Carlo integration [19]. In fact, it can be applied to the general method (3.5), not only to (3.1).

The idea is to sample the points X_i not uniformly but to prioritise "interesting" points, which, in our case, means points in A. When sampling from another distribution p on $\mathcal{X}$, a bias is introduced. So, to keep the estimator unbiased, the contribution of each point must then be weighted accordingly. This leads to the following estimator:

$$\hat{P}_{\text{IS}} = \frac{1}{N}\sum_{i=1}^{N}\frac{I_{X_i\in A}}{p(X_i)} = \frac{1}{N}\sum_{i=1}^{N} w(X_i)I_{X_i\in A} \qquad X_i \sim p, \tag{3.6}$$

where $w = 1/p$ is called the *weight function*. It has to be well-defined on A, which means that $p(x) > 0\ \forall x \in A$ (outside of A we can consider the product to be zero anyway). The unbiasedness of the estimator can be expressed as

$$E_p[w(X_i)I_{X_i\in A}] = E[I_{X_i\in A}] = P,$$

where the notation $E_p[f(X)]$ represents the expectancy value of $f(x)$ when $X \sim p$. The notation without subscript implies uniform distribution instead. We make use of this relation to compute the variance of the estimator:

$$\begin{aligned}
Var(\hat{P}_{IS}) &= E_p[\hat{P}_{IS}^2] - E_p[\hat{P}_{IS}]^2 \\
&= \frac{1}{N^2}\sum_{i,j} E_p[w(X_i)w(X_j)I_{X_i}I_{X_j}] - P^2 \\
&= \frac{1}{N^2}\left(\sum_{i=j} E_p[w(X_i)^2 I_{X_i}^2] + \sum_{i\neq j} E_p[w(X_i)w(X_j)I_{X_i}I_{X_j}]\right) - P^2 \\
&= \frac{1}{N^2}\left(\sum_{i} E[w(X_i)I_{X_i}] + (N-1)\sum_{i} E_p[w(X_i)I_{X_i}]^2\right) - P^2 \\
&= \frac{1}{N^2}\left(NE[w(X_i)I_{X_i}] + (N^2-N)P^2\right) - P^2 \\
&= \frac{E[w(X_i)I_{X_i\in A}] - P^2}{N},
\end{aligned} \tag{3.7}$$

where we have shortened the notation of the indicator function for clarity. This variance is potentially smaller than (3.2) depending on our choice of w. To compare the two, suppose that $w(x) = W\ \forall x \in A$. Then, $E[w(X_i)I_{X_i\in A}] = WE[I_{X_i\in A}] = WP$. This term is smaller than its counterpart in (3.2), i.e., P, if $W < 1$, which, in turn, means that $P(X \in A, X \sim p) = \int_A p(x)dx = 1/W\int_A dx = P/W > P$. In other words, importance sampling increases the accuracy compared to SMC if the samples are generated using a probability distribution that favours points in A relatively to the rest of $\mathcal{X}$. In fact, if $w(x) = P\ \forall x \in A$ the variance vanishes completely! This, however, is the case where only points in A are generated ($P/W = 1$) and requires knowledge of A, and thus P, which is exactly what we were trying to compute in the first place. This means that, in practice, zero is only an unachievable lower bound. At the same time, we have to be careful because a bad choice of w leads to a decrease in accuracy and, potentially, an unbounded error. Indeed, in the limit case where $p(x) = 0\ \forall x \in A$, which we had previously excluded, W, and so (3.7) as a whole, diverges to infinity.

In conclusion, importance sampling is a powerful and well-established technique of variance reduction that can be used when we approximately know or we can guess "where" to find our target rare event A in the state space $\mathcal{X}$. Our educated guess is represented by the weight function w.

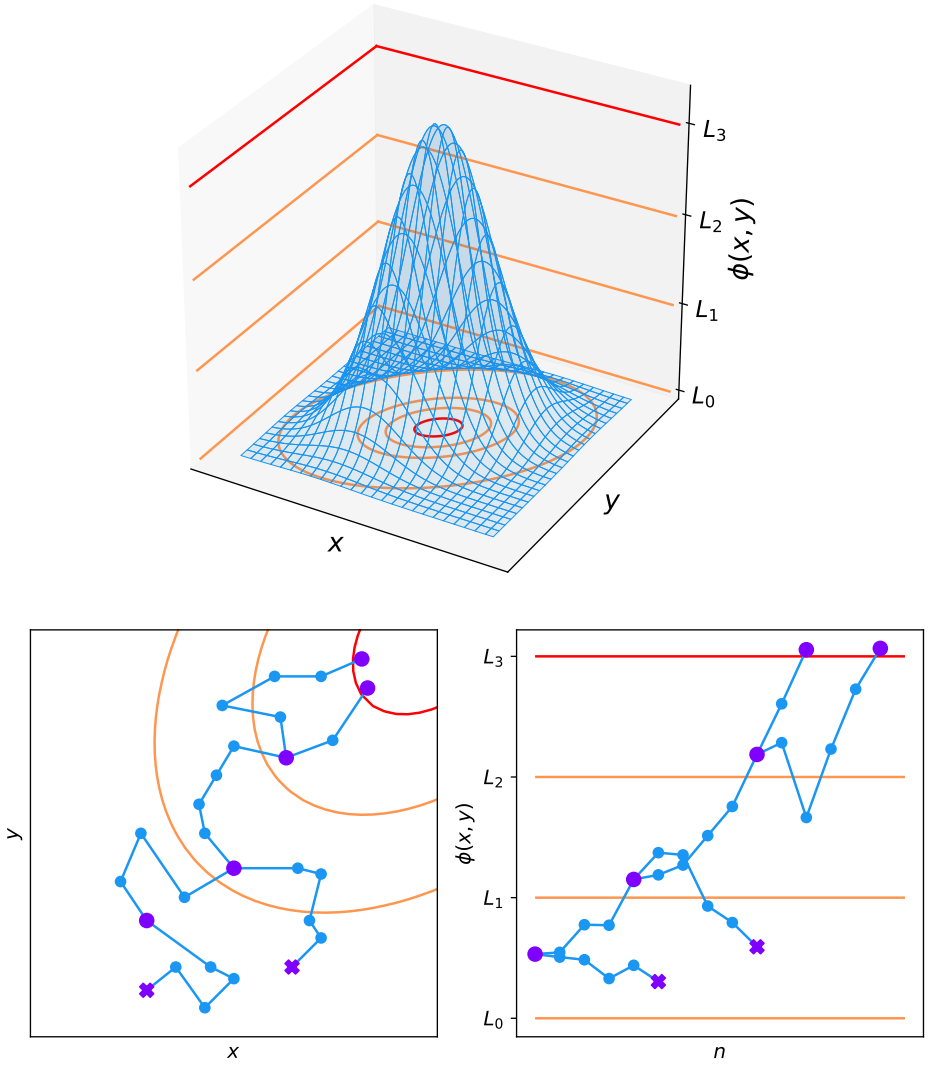

Figure 3.1: The importance function and the splitting algorithm. Top: the sets $\mathbb{R}^2 \supset A_0 \supset ... \supset A_3 = A$ (A is in red) in the xy-plane, and the corresponding levels $L_0 < ... < L_3$ (L3 is also in red) along the z-axis, as determined by an importance function $z = \phi(x, y)$ (in blue). Bottom left: a sample run of the importance splitting algorithm ($N_0 = k = 2$) shown on the phase space. Bottom right: the same run shown as mapped by ϕ on $\mathbb{R}$. The conditional probabilities in this trivial example are $P_1 = 1/2, P_2 = 1/2, P_3 = 1$, for a final estimate of $P = 1/4$.

3.3 Importance Splitting

Importance splitting is a technique of variance reduction for small probabilities estimation, also invented by von Neumann and Ulam according to Kahn [20]. It is the computational method that we will use throughout this thesis, as it is particularly suited to study rare events defined as states of Markov processes. This is generally referred to as the "dynamic case". The "static case", on the other hand, concerns situations in which one can sample from the probability distribution directly. This is the most common and natural formulation of importance sampling, that we too presented above. Despite this, with some adjustments, both methods can be applied to both cases (refer to Section 2.3 of [21] and

[22], respectively). Since the systems analysed in this thesis will be represented as Markov processes, we will use the former formulation from now on.

Let $X(t)$ be a Markov process over the state space $\mathcal{X} \subseteq \mathbb{R}^d$. We are interested in the rare probability that $X(t) \in A$ before it reaches another distinct set B, that is, the stopping condition. Usually, but not necessarily, it is defined by a stopping time T. The fundamental idea of importance splitting is to express the rare event as a chain of conditional events, thus dividing one difficult problem in a series of simpler sub-problems. To do so, we divide the domain $\mathcal{X}$ into a sequence of nested subsets:

$$\mathcal{X} = A_0 \supset A_1 \supset ... \supset A_M = A$$

so that we can write:

$$P = \prod_{j=1}^{M} P_j = \prod_{j=1}^{M} P(X(t_j) \in A_j \mid X(t_{j-1}) \in A_{j-1}),$$

with $t_j > t_{j-1}\ \forall j$. In order for this to be of any use, the events $\{X(t) \in A_j \mid X(t_{j-1}) \in A_{j-1}\}$ themselves must not be rare, of course.

Defining this sequence is not trivial, so one instead maps the domain to the real numbers through an *importance function*, also known as "score function" or "committor", $\phi : \mathcal{X} \to \mathbb{R}$ such that

$$\phi(x \in A_j) > \phi(x \in A_{j-1} \setminus A_j),$$

and then rewrite the chain of events as

$$P = \prod_{j=1}^{M} P(\phi(X(t_j)) > L_j \mid \phi(X(t_{j-1})) > L_{j-1}),$$

where the L_j are called "levels".

In practice, one usually picks the importance function and the levels without worrying about the underlying set representation given by the sequence A_j. The importance function helps to discriminate between "promising" and "unpromising" trajectories or, in other words, between those that have a good chance of reaching the rare event and those that do not. Of course, we want to focus our computational efforts on the former and neglect the latter. The contribution of the surviving ones to the final value will be weighted taking into consideration those that have been stopped prematurely. In this sense, the importance function in Importance Splitting has a similar role to the weight function in Importance Sampling.

Given these definitions, importance splitting is defined by Algorithm 1.

Algorithm 1 Importance Splitting: Fixed Splitting

Input: Set of states B; initial distribution p_0 or, equivalently, initial state x_0; initial number of trajectories N_0; splitting rate k; levels $L_1, ..., L_M$; importance function ϕ.
Output: Probability estimate $\hat{P}$.
  Initialise: sample $X_0(0), ..., X_{N_0}(0) \sim p_0$; $\hat{P} \leftarrow 1$.
  for $j = 1$ to M **do**
    for $i = 1$ to N_{j-1} **do**
      Iterate the dynamics of $X_i(t)$ until either $\phi(X_i(t)) \geq L_j$ or $X_i(t) \in B$.
    end for
    Discard trajectories that have reached B.
    Set R_j as the number of trajectories that reached L_j and create k copies of each. Set $N_j \leftarrow kR_j$.
    Set $\hat{P} \leftarrow \hat{P}R_j/N_{j-1}$.
  end for

This algorithm is also unbiased and it can be proven by induction [23]. Let $\hat{P}_j = \frac{R_j}{N_{j-1}}$, with $N_{j-1} = kR_{j-1}$ for $j > 1$, be the estimator for $P(\phi(X(t_j)) > L_j \mid \phi(X(t_{j-1}) > L_{j-1})$. Clearly $E[\hat{P}_1] = P_1$ since this first step is equivalent to a standard Monte Carlo algorithm. Then, assuming $E[\hat{P}_1 ... \hat{P}_{j-1}] = P_1 ... P_{j-1}$:

$$\begin{aligned} E[\hat{P}_1 ... \hat{P}_j] &= E[\hat{P}_1 ... \hat{P}_{j-1}] E[\hat{P}_j | \hat{P}_1 ... \hat{P}_{j-1}] \\ &= E[\hat{P}_1 ... \hat{P}_{j-1}] \frac{E[R_j]}{N_{j-1}} \\ &= E[\hat{P}_1 ... \hat{P}_{j-1}] \frac{P_j N_{j-1}}{N_{j-1}} \\ &= P_1 ... P_j. \end{aligned}$$

3.4 Fixed Effort

The previous algorithm is simple but impractical due to the number of parameters to tune, namely the initial number of trajectories N_0, the splitting rate k, the levels L_j, and the importance function ϕ. Luckily, improvements can be easily made.

First, one can consider different possible ways of splitting. In the case above *fixed splitting* is used, since the *splitting factor* (or *branching rate*) k is a constant. However, as with any parameter, choosing the correct value of k is tricky. In particular, we would like to have $\frac{N_j}{N_{j-1}} \approx 1$ for each j, so that the number of trajectories does not blow up (increasing the computational cost) but also not shrink (making it harder to reach the rare event). Both situations are exactly what we are trying to avoid by using importance splitting.

To solve this, one can use the *fixed effort* or *sequential Monte Carlo* variant, where the number N of trajectories is fixed instead. This way, after reaching a level, the unsuccessful trajectories are discarded and the successful ones are sampled, allowing repetition, until there are N trajectories again.

Algorithm 2 Importance Splitting: Fixed Effort

Input: Set of states B; initial distribution p_0 or, equivalently, initial state x_0; number of trajectories N; levels $L_1, ..., L_M$; importance function ϕ.
Output: Probability estimate $\hat{P}$.

```
Initialise: sample X_0(0), ..., X_N(0) ~ p_0; P̂ ← 1.
for j = 1 to M do
    for i = 1 to N do
        Iterate the dynamics of X_i(t) until either φ(X_i(t)) ≥ L_j or X_i(t) ∈ B.
    end for
    Discard trajectories that have reached B.
    Set R_j as the number of trajectories that reached L_j.
    Sample, with repetition, from the R_j successful trajectories until you have
      a total of N trajectories again.
    Set P̂ ← P̂ R_j / N.
end for
```

The variance of this new estimator is [24]:

$$\begin{aligned} Var(\hat{P}_{FE}) &= E[\hat{P}_{FE}^2] - E[\hat{P}_{FE}]^2 \\ &= E[\prod_{j=1}^{M} \hat{P}_j{}^2] - E[\prod_{j=1}^{M} \hat{P}_j]^2 \\ &= \prod_{j=1}^{M} E[\hat{P}_j{}^2] - P^2 \\ &= \prod_{j=1}^{M} \left(Var(\hat{P}_j) + E[\hat{P}_j]^2 \right) - P^2 \\ &= \prod_{j=1}^{M} \left(\frac{P_j(1-P_j)}{N} + P_j^2 \right) - P^2 \\ &= P^2 \left(\prod_{j=1}^{M} (\frac{1-P_j}{NP_j} + 1) - 1 \right), \end{aligned}$$

where we have used the fact that each estimator $\hat{P}_j$ is independent, to swap the expectancy operator with the product, and unbiased, to replace $E[\hat{P}_j]$ with $\hat{P}_j$. The former assumption is not entirely true in practice and we will get back to it later. For large N the variance simplifies to

$$Var(\hat{P}_{FE}) = P^2 \left(1 + \sum_{j=1}^{M} \frac{1-P_j}{NP_j} + o(\frac{1}{N}) - 1 \right) \approx \frac{P^2}{N} \sum_{j=1}^{M} \frac{1-P_j}{P_j}.$$

This expression cannot be yet compared to (3.2) because of the unclear value of the product: the conditional probabilities P_j depend on the choice of levels, which we will analyse next. Moreover, we have to be careful because N has a different meaning in the two expressions. In (3.2) the number of samples N also represents the total computational cost C of the algorithm. Making use of the notion of time complexity, we can say that the SMC is $O(N)$. In the case of importance splitting, it is not as straightforward because trajectories are (partially) restarted during the run. The complexity must be proportional to both the number of trajectories N and the number of restarts, i.e. the levels, M; so it is $O(NM)$. Of course, as the iterations increase, trajectories are restarted closer and closer to A and, thus, their contribute is smaller, but we can ignore proportionality coefficients using the big-O notation. In other words, we should write the expression above as

$$Var(\hat{P}_{FE}) \approx \frac{MP^2}{C} \sum_{j=1}^{M} \frac{1-P_j}{P_j}. \tag{3.8}$$

3.5 Adaptive Multilevel Splitting

The choice of levels also presents challenges. In general, one would like to avoid levels that are too close (increasing the computational effort in exchange of small variations in the value of $\hat{P}$) or too far apart (risking that none of the trajectories reaches the next level, causing the algorithm to fail) [12].

There are also theoretical reasons: one can prove that, under simplified assumptions, the variance of P is minimised when $P_j \approx p = \text{const.}\ \forall j$ [24]. To show this, we solve the optimization problem of minimising equation (3.8) with respect to the sequence P_j under the constraint $\prod_j P_j = P$ by introducing the Lagrange multiplier λ and taking the partial derivatives of $\mathcal{L}(P_j, \lambda) = \sum_{j=1}^{M} \frac{1-P_j}{P_j} +$

$\lambda(\prod_j P_j - P)$. The prefactor in (3.8) can be omitted as it does not affect the minimum point.

$$\frac{\partial}{\partial P_i}\mathcal{L}(P_j,\lambda) = -\frac{1}{P_i^2} + \lambda\frac{P}{P_i} = 0 \quad \forall i$$
$$\implies P_i = p = \frac{1}{\lambda P} \quad \forall i$$

as anticipated. Further,

$$\frac{\partial}{\partial \lambda}\mathcal{L}(P_j,\lambda) = \prod_{j=1}^{M} P_j - P = 0$$
$$\implies \prod_{j=1}^{M} \frac{1}{\lambda P} - P = \frac{1}{(\lambda P)^M} - P = 0$$
$$\implies \lambda = P^{-1-1/M}$$
$$\implies P_i = p = P^{1/M}.$$

By substituting P_j into (3.8) we obtain

$$Var(\hat{P}_{\text{AMS}}) = \frac{M^2P^2(P^{-1/M}-1)}{C}. \tag{3.9}$$

The reason for the subscript "AMS" will be clear soon. Now one can also minimise this last result with regard to M. For small P it is equivalent to minimising $M^2P^{-1/M}$ which, assuming for simplicity M to be continuous and not discrete, has a minimum in $M = -\ln(P)/2$. This means that we should choose the levels L_i such that

$$P_j = e^{-2} \quad \forall j. \tag{3.10}$$

Knowing an approximate distribution of the events A_j one could pick the L_j's so that (3.10) is approximately met. However, in the general case, without such knowledge, levels cannot be chosen a priori.

A solution is setting levels adaptively during the run so that a predefined percentage of the trajectories crosses each of them. To achieve this, at every iteration, we compute

$$\mu_i = \max_n \phi(X_{i,n}) \qquad i = 1, ..., N,$$

then we sort the sequence so that $\mu_1 \leq \mu_2 \leq ... \leq \mu_N$ and finally we pick

$$L_j = \mu_K, \tag{3.11}$$

effectively cutting off no more than $N - K$ trajectories (they are exactly $N - K$ when $\mu_{K-1} \neq \mu_K$). In this case, the number of levels M is variable instead. This method takes the name of *Adaptive Multilevel Splitting*, or AMS for short.

Then, by using AMS, (3.10) is approximately true and we can substitute $M = -\ln(P)/2$ into (3.9) to obtain

$$Var(\hat{P}_{AMS}) = \frac{(e^2-1)P^2\ln^2(P)}{4C}. \tag{3.12}$$

Finally we can compare the efficiency of importance splitting to that of standard Monte Carlo (3.2). Substituting (3.12) in the definition of relative error η and using the same hypotheses as in Section 3.1,

Algorithm 3 Importance Splitting: Adaptive Multilevel Splitting

Input: Set of states B; initial distribution p_0 or, equivalently, initial state x_0; number of trajectories N; minimum number of trajectories to keep each level K; importance function ϕ.
Output: Probability estimate $\hat{P}$.
Initialise: sample $X_0(0), ..., X_N(0) \sim p_0$; $\hat{P} \leftarrow 1$.
for $j = 1$ to M **do**
  for $i = 1$ to N **do**
    Iterate the dynamics of $X_i(t)$ until either $\phi(X_i(t)) \geq L_M$ or $X_i(t) \in B$.
    Set μ_i as the maximum of the importance function along X_i.
  end for
  Sort the array μ in ascending order.
  Set $L_j \leftarrow \mu_K$.
  Set R_j as the number of trajectories that reached L_j and cut them off at the first point t_i such that $X(t_i) \geq L_j$.
  Discard trajectories that have not reached L_j.
  Sample, with repetition, from the R_j successful trajectories until you have a total of N trajectories again.
  Set $\hat{P} \leftarrow \hat{P}R_j/N$.
end for

i.e. $\eta = 0.01$ and $P = 10^{-10}$, we get $C \approx 10^7$. An impressive improvement of 7 orders of magnitude! Notice that we can refine the definition of computational cost for AMS by taking into consideration the fact that the number of restarted trajectories at any level is (almost) constant and equal to $N-K$, so $C \approx (N-K)M$.

One can prove that the estimator is also unbiased by showing that is can be interpreted as a limit of sequential Monte Carlo [12].

AMS is the most commonly used version of the Importance Splitting algorithm thanks to its evident advantages. We will be using this setup in all the following simulations.

3.6 Optimal K

According to (3.10), when using AMS to estimate a probability $P \ll 1$, given a certain precision $Var(\hat{P})$, the optimal choice is to set $K = (1-e^{-2})N \approx 0.865N$ to minimise the required computational cost C. However, in practice, P is unknown and, therefore, C cannot be computed a priori. Instead, one guesses a reasonable value for C and runs the algorithm. If the algorithm fails, by returning $\hat{P} = 0$, then C must be increased (since M can no longer be chosen when using AMS, the only parameter to increase is N). In this situation, we would like to know what is the optimal choice of K given a computational cost C sufficiently large to estimate P.

To test it, we have empirically compared the execution times for different values of $K \in]0, N[$. As a benchmark case, we have considered the distribution right-hand tail for a binary annihilation system, as described in depth in Section 4.4. In particular, all results presented in Figure 3.2 are averaged over 1000 independent runs.

As one can see, when C is sufficiently large, varying the value of K does not affect the precision of the algorithm in the least: the outputs perfectly overlap. However, it does have a significant effect on the number of splits M, which decreases as K increases. As mentioned at the beginning of Section 3.5, M has, in turn, a direct effect of the performance of the algorithm, which decreases linearly with K. From $K = 0.1N$ to $K = 0.9N$ the execution time is halved. Significantly, we did not observe a minimum in $K = (1 - e^{-2})N \approx 0.865N$ for either of the two quantities, even though their values

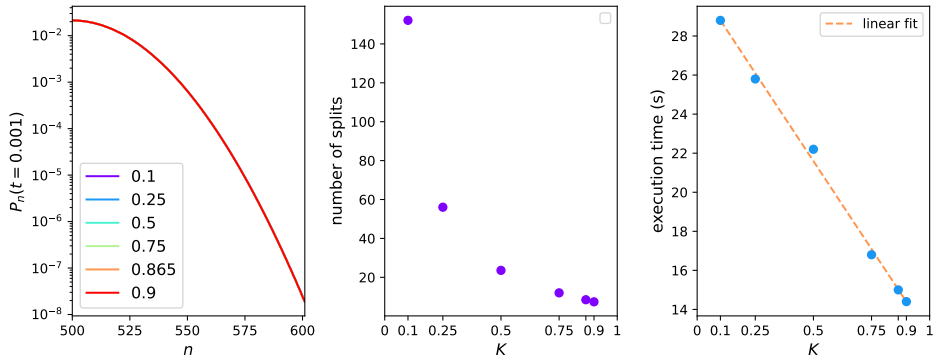

Figure 3.2: Benchmarking the parameter K using a binary annihilation system as a case study (more details in Section 4.4). Left: the output of the AMS algorithm for different values of K as fractions of N. Centre: the average number of splits. Right: the execution time.

are very close to the respective empirical optima. We tested higher values of K than those shown in Figure 3.2, although they were omitted to avoid visual clutter. In particular, we found that, when $N = 10^4$, values up to $K = 0.99N$ lead to perfectly accurate results and an average of $M = 4$, but when $K \geq 0.999N$ the algorithm inevitably fails.

Consequently, we conclude that the best choice is to pick K as large as possible, keeping in mind that some variance in the set of successful trajectories is necessary to avoid getting stuck in local maxima of the importance function. What happens in practice when K is close to N is that one very high importance value is found early on and, because the sample pool is so small, many trajectories are restarted from that particular point. However, as it happens for all optimisation problems, the local maximum could be very far from the global maximum. For example, the former could be very close to the stopping time T, meaning that trajectories restarted there do not have any chance to significantly move away from it. In this case, at every split, more and more trajectories will be restarted in the neighbourhood of the point, stalling the algorithm in a state it can no longer recover from, leading it to fail. The condition to detect this deadlock and interrupt the run is

$$\mu_K = \mu_N, \tag{3.13}$$

where μ is sorted in ascending order.

3.7 Optimal Importance Function

Notice that if both fixed effort and adaptive levels are used, the importance function becomes the only parameter of the algorithm that is non-trivial to tune. However, it does not appear as a parameter in (3.12). That is because, as we have anticipated, the assumption made in Section 3.4 are generally not true: the estimators $\hat{P}_j$ are not independent. The probability of reaching the next level L_{j+1} does not only depend on the current level L_j but also on the specific starting point in A_j which, in turn, depends on the entrance points found in the previous iteration, starting at L_{j-1}, and so on. In other words, P_j is not constant over $\{x : L_j > \phi(x) > L_{j-1}\}$ but, rather, it is an average:

$$P_j = E[P(\phi(X(t_j) > L_j \mid \phi(X(t_{j-1}) > L_{j-1})]$$

and the variance of $\hat{P}_j$ depends on ϕ [24].

It has been proved [12] that by taking into consideration the effects of the importance function

$$\frac{-P^2 \log(P)}{C} \leq Var(\hat{P}_{AMS}) \leq \frac{2P(1-P)}{C}, \tag{3.14}$$

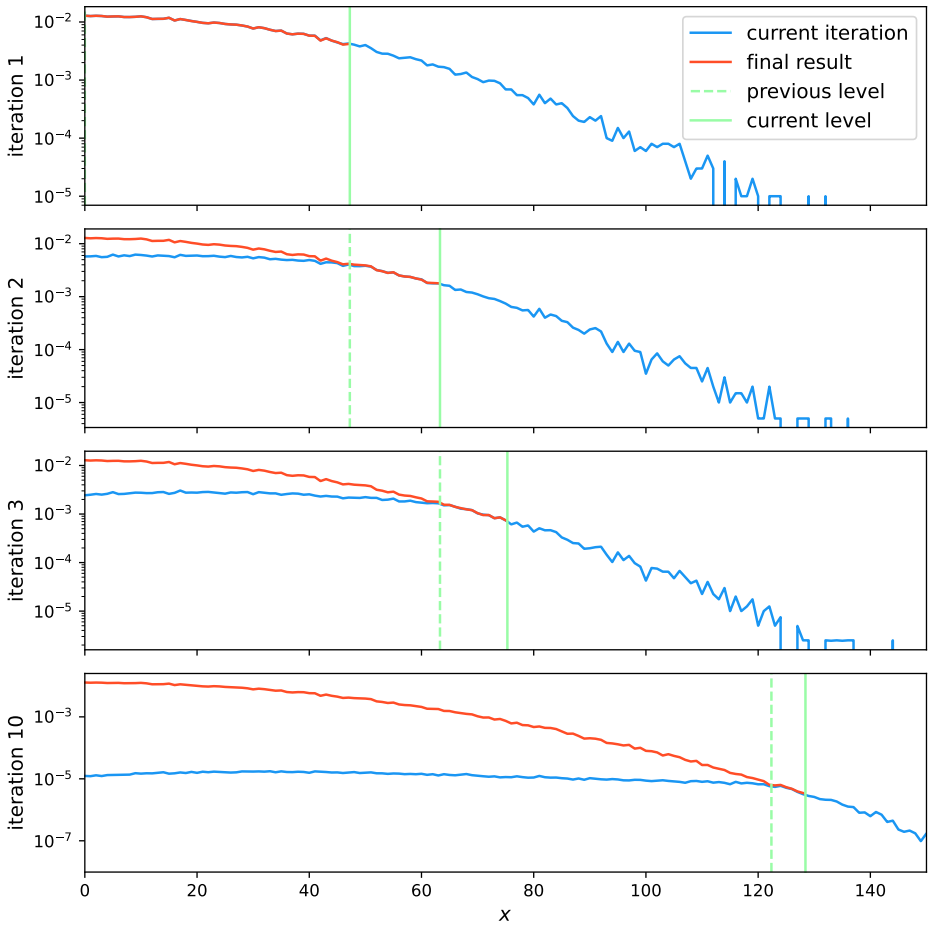

Figure 3.3: The evolution of a probability distribution estimate computed using AMS. The stochastic process taken as an example in the picture is a symmetric random walk $X(t)$ on the line starting at the origin, and the corresponding rare event is $X(T) = x \gg 0$ (for more details see Section 4.1). At every iteration, the algorithm estimates the probabilities P_x (blue line) that have not met condition (3.15) yet, i.e., those to the right of the previous level (vertical dashed green line). To visualise the levels, they have been mapped to the x-axis through the inverse of the importance function: $x_j = \phi^{-1}(L_j)$. Then, the current level (vertical solid green line) is computed according to (3.11). The part of the current estimate between the previous and current levels is added to the final result (red line), in the sense that it will not be recomputed further, since it will meet condition (3.15) at the following iteration. Notice that the first iteration corresponds to a standard Monte Carlo run and that, as the iterations increase and the current level advances, the tails become more defined. In the final result, the relative magnitude of fluctuations remains stable across scales thanks to the variance reduction performed by AMS.

which means that, even knowing the optimal importance function, the error is always strictly positive, but, on the other hand, it is always bounded and, at worse, double as bad as the standard Monte Carlo method (3.2). The upper bound is intuitive to understand: just like the worst weight function for importance sampling, the worst importance function for importance splitting is zero at the rare event and strictly positive elsewhere. This means that, after cutting off the "worst" half of the trajectories, on average, the rare event will be observed in the remaining half only if $N/2 \geq 1/P$. Indeed, if the algorithm gets stuck in local maxima, the only possible solutions are to either increase N, as previously

mentioned, or to find a better importance function.

Comparing this to (3.7) one can conclude that in the best case, importance sampling is finitely better than AMS, but in the worst case, importance sampling is infinitely worse than AMS. Nevertheless, there is no definitive way of comparing the two algorithms since in practice the function choice will always fall in between the two extremes. Instead, it is up to the user to evaluate which one better adapts to the case in question.

In [23] some theoretical results regarding importance functions are summerised. In general, the optimal importance function cannot be found a priori, except for simple one-dimensional cases. Instead of relying on intuition and trial-and-error, more sophisticated algorithms can be used to progrmmatically construct an importance function but, ultimately, like for the weight function in importance sampling, knowing the optimal importance function is at least as hard as knowing the target probability itself. Indeed, the ideal importance function is the probability of reaching A from the state in question. So developing complex algorithms or dedicating too many resources to trying to find it is counterproductive. Moreover, this search for a better importance function can be formalised as a standard optimisation problem and standard computational methods may be employed to solve it.

3.8 Computing Probability Distributions

Computing a probability distribution $P_x = P(X = x)$ with $x \in [a, b]$, and not just one probability P, can be done in a single run of the algorithm by continuing to split after reaching the first event $X = a$, which may not be rare, until $X = b$ is reached. This allows to use the same generated data, dramatically reducing the computation time.

Assuming that $P_x \geq P_{x-1}$, one must make sure not to recompute P_x after crossing its corresponding level, or, in other words, if

$$\phi(x) < L_{j^*}, \tag{3.15}$$

where j^* is the current iteration. In fact, the algorithm can only estimate probabilities P_x such that

$$\prod_{j}^{j^*} P_j \leq P_x \leq \prod_{j}^{j^*} \frac{P_j}{N}, \tag{3.16}$$

where the product is strictly decreasing after every iteration, eventually making some probabilities too large to be computed. If we were to estimate P_x when condition 3.15 is met, it would be necessarily biased towards lower values. Furthermore, notice that there does not have be to a one-to-one relationship between levels L_j and events $X = x$; multiple events can be reached in the same iteration, and one iteration may not reach any new events.

When computing probability distributions, the algorithm will behave as shown in Figure 3.3.

3.9 Finite Differences vs Gillespie

A naive way to simulate stochastic processes with rate α is to discretise the continuous process by slicing a time interval $[0, T]$ in many sub-intervals of length Δt and computing the probability $\alpha \Delta t \ll 1$ that an event will occur in each sub-interval. This is known as the *finite differences method*. Since the resulting trajectory tends to the theoretical expectation as Δt tends to 0, its accuracy and efficiency are inversely proportional to one another and a trade-off must be found.

Sometimes the reaction rate of a stochastic process can depend on time, directly or indirectly. For example, recall that $\alpha = \lambda n(n-1)/2$ for binary annihilation systems, where n decreases with time.

This makes the choice of Δt only apparently trickier, as it can be dynamically updated using the rule

$$\Delta t = \frac{f}{\alpha},$$

where f is a small constant factor, such as 10^{-3}.

A *Gillespie algorithm* is the common alternative to the finite differences method: instead of checking every Δt whether an event will occur or not, one can sample from the distribution of the time interval between two events and jump to the next immediately. In detail, if an event occurs with rate α, the distribution of the time in-between events is exponential with parameter α.

If the system is described by multiple reactions, each with rate α_i, then the mean time to the next reaction is $\sum_i \alpha_i$ and the probability of each reaction is proportional to its rate. The most efficient way of selecting the next event is illustrated in Algorithm 4.

Algorithm 4 Gillespie algorithm

Input: K reaction rates $\alpha_1, ..., \alpha_K$.
Output: Next reaction and when it will occur.
Compute $\alpha_0 = \sum_{i=1}^{K} \alpha_i$.
Sample a random number r_1 uniformly distributed in $[0, 1]$, and compute the time to the next reaction $\tau = -\alpha_0^{-1} \ln(r_1)$.
Sample another random number r_2 uniformly distributed in $[0, 1]$. The index j of the next reaction is such that $r_2 \geq \alpha_0^{-1} \sum_{i=1}^{j-1} \alpha_i$ and $r_2 < \alpha_0^{-1} \sum_{i=1}^{j} \alpha_i$.

Since the Gillespie algorithm does not depend on the choice of a parameter and only samples a strictly necessary number of random numbers, it is generally preferred. However, when using an importance splitting method, the Gillespie algorithm should be avoided in one particular case. Since it skips directly from one event to the next, if the importance function grows in-between events, that is, it grows with time as the other variables remain unchanged, then its maximum along a trajectory may not be recorded, and the splitting point could also be chosen incorrectly, resulting in entirely wrong estimates. To compensate for this, one would need to employ some form of interpolation or, more simply, the finite differences method.

3.10 Simulating Diffusion

Diffusion is also a stochastic process, which could be simulated through the finite differences method by discretising Langevin's equation (1.2) as following:

$$x(t + \Delta t) = x(t) + \sqrt{2D\Delta t}\xi,$$

where ξ is a normally distributed random number. But, in order to apply Gillespie's algorithm, we discretise not time, but space, dividing the domain into K compartments A_i, which can be interpreted as points of a lattice. Single particles transition from a compartment to a neighbouring one with rate $D = \tilde{D}/a_0^2$, where a_0 is the compartment length (here we are just recovering the discrete hopping rate from Section 2.5), which means that hops from compartment A_i happen with rate $\alpha_i = dDn_i$, where d is the dimension. In addition, we assume that all and only particles in the same compartment are close enough to react with each other, so, in each compartment, reactions occur with independent rates.

For instance, a one-dimensional binary annihilation system can be simulated applying Gillespie's algorithm to the following system of reactions:

$$A_1 \underset{}{\overset{\text{D}}{\rightleftharpoons}} A_2 \overset{\text{D}}{\rightleftharpoons} A_3 \overset{\text{D}}{\rightleftharpoons} ... \overset{\text{D}}{\rightleftharpoons} A_K$$

$$2A_i \xrightarrow{\lambda} \varnothing \qquad i = 1, ..., K$$

This example assumes no-flux boundary conditions; for periodic boundaries just add the reactions $A_1 \rightleftharpoons A_K$.

The compartment-based approach is easily generalised to the case of multiple chemical species A, B, ... by keeping track of the number of particles for each species for each compartment through the lists A_i, B_i, ...

Since the Gillespie algorithm generates correct stochastic trajectories, the only computational error introduced is due to the spatial discretization.

3.11 Memory Optimisation

Because AMS, in order to compute the next level and the splitting points, requires to store all N trajectories from the current level to the last, it demands significant memory resources. This can become an issue when adding diffusion to the simulation, as the number of events per unit of time increases exponentially.

The most trivial memory optimisation strategy, which presents no drawbacks, is to only store the state of the system when the importance function has strictly increased compared to the previous state. Other states would not be considered anyway to determine levels nor splitting points. This has also an obvious positive side effect on computation time, as fewer potential points have to be considered.

A more advanced strategy, mentioned in [25], which trades space complexity for time complexity, consists in only saving the starting point, the random-number-generator seed and the maximum importance value for each trajectory. Then, the first level is computed as normal using the maximum importance values, but the crossing points of the successful trajectories are found by re-computing the trajectories from the start using the same seed. The unsuccessful trajectories will re-start from these checkpoints, but with a different seed assigned, and this process will be repeater for every iteration.

As one can imagine, this second strategy is quite extreme, as recomputing entire trajectories is definitely expensive, but it might be indispensable if the elevated memory demand causes the program to crash.

4

Results

In this chapter, we test AMS on a set of different problems, discussing its accuracy and efficiency, and our choices regarding the importance functions. In Sections 4.1, 4.2 and 4.3, we benchmark the computational results against well-understood analytical approximations, exact solutions and previous computational findings, respectively. In Section 4.4, we compare them to the semiclassical approximation for a binary annihilation system, thereby reaching the conclusion of this thesis.

4.1 Symmetric Random Walk

First we start with a simple example. Let $X(t)$ be a continuous-time, discrete-space, one-dimensional and symmetric random walk. The rare event considered is a large deviation x from the mean position (zero) at time $T \gg 1$. This is a stochastic process with rate λ. It is described by the following Master equation:

$$\frac{d}{dt}P_x(t) = \frac{\lambda}{2}(P_{x-1} + P_{x+1} - 2P_x).$$

Its solution, obtained through the generating function method, is

$$P_x(t) = e^{-t}I_x(t), \tag{4.1}$$

where I is the modified Bessel function of the first kind.

In the simulations, for simplicity, we set $\lambda = 1$ and $T = 1000$ (measured in units of $1/\lambda$). Figure 4.1 compares the computational results obtained using the standard and AMS variations of the Monte Carlo method. For the latter $K = 0.8N$ and $\phi(x,t) = x/\sqrt{t+1}$ were used. The intuitive reason behind the importance function is that quantity $x/\sqrt{t}$ represents the normalised deviation from the average, while $+1$ is needed to avoid a singularity in $t = 0$.

To reasonably compare the two, we make use of the notion of time complexity already introduced in Section 3.5. We have executed the Splitting algorithm with $N = 100$ trajectories and averaged the results over 1000 independent runs. Since each run required on about $M = 40$ adaptively chosen levels, SMC has been executed with $N = 2000$ trajectories, also averaged over 100 independent runs. We should reiterate that the AMS complexity is an overestimation, because trajectories are not all restarted at each level, and never from the initial condition; in reality, the algorithm is much faster than expected. In fact, our C implementations of SMC and AMS take about 75 and 10 seconds to run, respectively, on the same single-core general-purpose computer. However, since we will argue that the latter is incomparably more efficient than the former, we will allow SMC an advantage, and will not dwell further of this detail.

As one can observe in Figure 4.1, the SMC method fails completely to compute small probabilities, below about 10^{-6}, simply because the rare event has not occurred in any of the $2 \cdot 10^5$ simulated

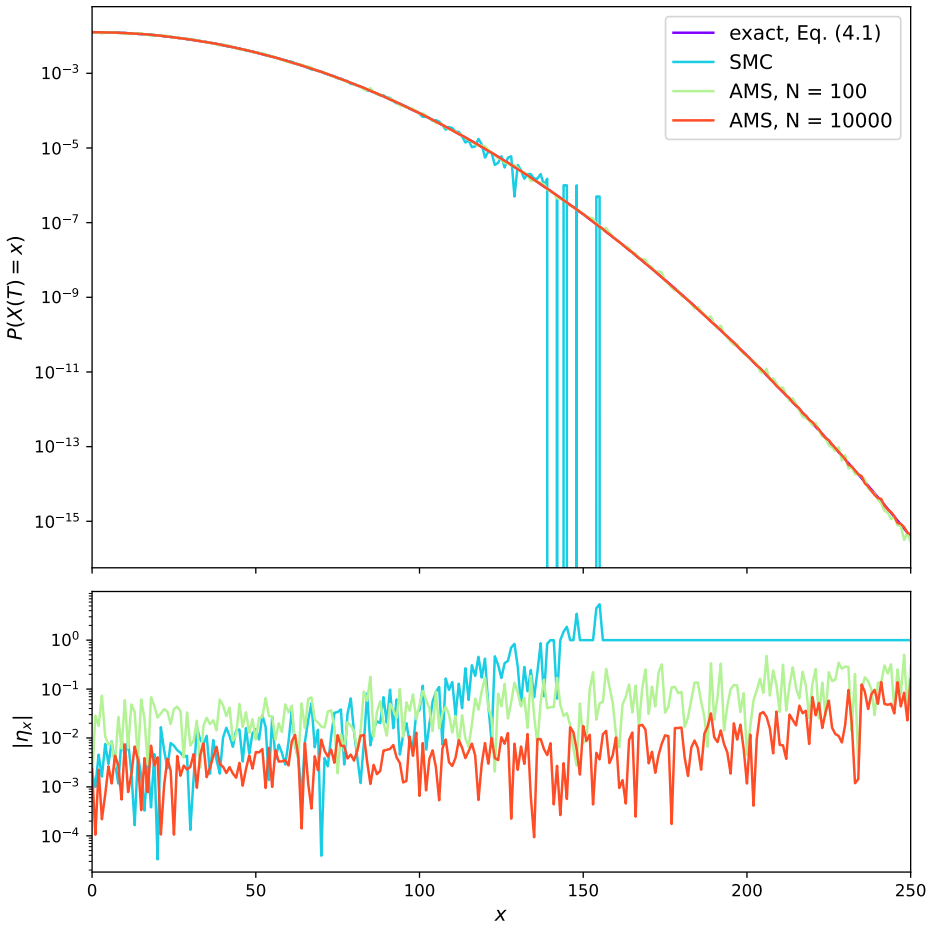

Figure 4.1: Probability of a large displacement $x \gg 0$ from the mean for a symmetrical random walker $X(t)$ on the line. Top: probability of a one-dimensional symmetric random walker with step rate $\lambda = 1$ to be at a distance x from the origin at $T = 1000$. The exact solution (4.1) is compared to the estimates obtained using standard Monte Carlo (SMC) and Adaptive Multilevel Splitting (AMS) methods. Bottom: the absolute value of the relative error between the estimates and the exact solution.

trajectories, exactly as one would expect. On the other hand, the importance-splitting relative error $\eta_x = (P_x - \hat{P}_x)/P_x$ fluctuates around zero up to probabilities of about 10^{-15}, albeit with increasing oscillations. In particular, the absolute value of the relative error η_x increases from about 10^{-2} to 1. We should also note that, eventually, as the magnitude of the probabilities in question approach zero, the relative error of either algorithm may briefly become larger than 1 before defaulting to it, which is the error of simply guessing $P = 0$. Nonetheless, the higher efficiency of AMS is undeniable, as it is able to maintain $\eta_x < 1$ far longer than SMC. The same SMC code would need to be run 10^9 times to estimate probabilities up to the order of 10^{-15} with a relative accuracy of 1, meaning that it would take roughly 10^{11} seconds, or about 3000 years, to run.

Figure 4.1 also shows that the results of AMS correctly tend to the exact values as N increases, since the same algorithm, run with $N = 10000$, produces significantly smaller fluctuations. In particular, $|\eta_x|$ ranges from 10^{-3} to 10^{-1} in the same x interval.

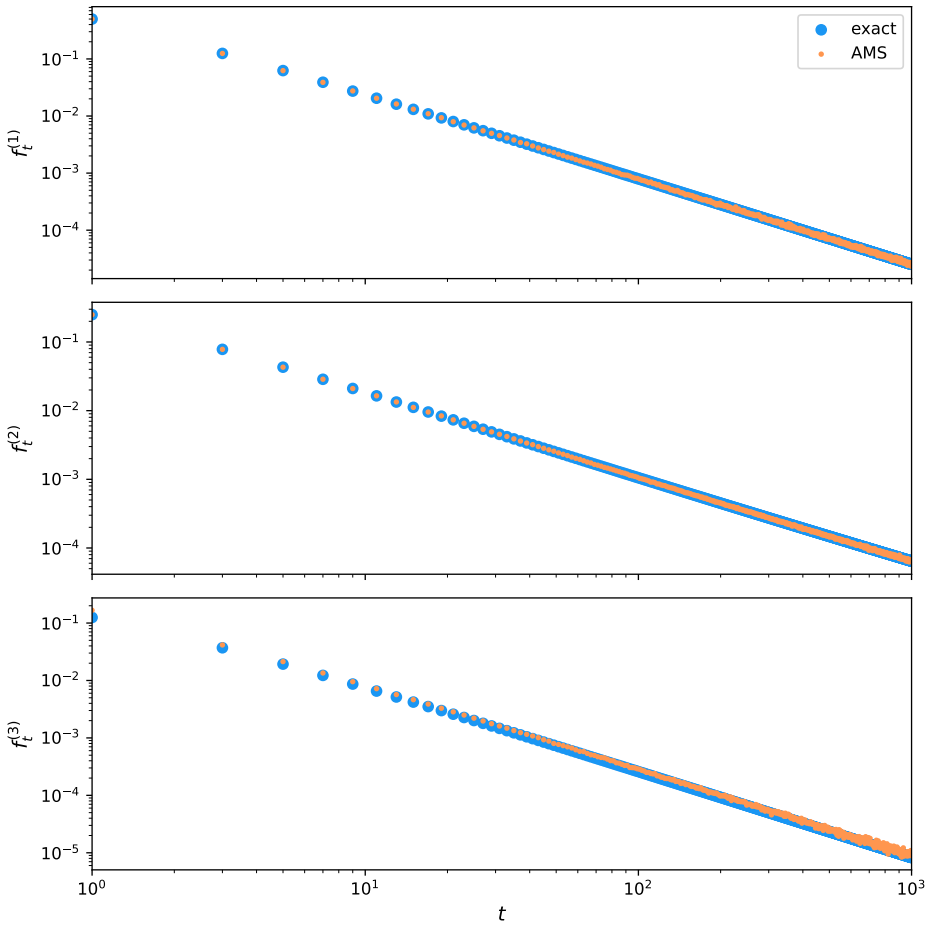

Figure 4.2: First return to the origin probability $f_t^{(d)}$ for a simple random walk for $d = 1, 2, 3$. The numerical solution obtained by Taylor-expanding (4.5) is compared to the AMS estimate.

4.2 Return to the Origin

Now let X_t be a discrete, symmetric random walk on $\mathbb{Z}^d$ starting in the origin. We want to compute the first-passage probability f_t of returning to the origin after exactly t steps:

$$f_t^{(d)} = P\left(X_t = 0, X_{t'} \neq 0 \; \forall \; 0 < t' < t\right). \tag{4.2}$$

To find the analytical solution first define the return probability $r_t = P(X_t = 0)$ and notice that $r_t = 0$ if t is odd, so we can limit our analysis to r_{2n} with $n \in \mathbb{N}$. In one dimension, the number of paths that are in the origin after $2n$ steps is the number of paths that have and equal amount of steps to the right and to the left, that is $\binom{2n}{n}$. Multiplying by the probability of each path we obtain

$$r_{2n}^{(1)} = 2^{-2n}\binom{2n}{n}.$$

Moreover, it can be proved [26] that

$$F(x) = \frac{R(x) - 1}{R(x)}, \tag{4.3}$$

where F and R are the generating functions of f and r, respectively. Together with the property that f_{2n} can be recovered as the $(2n)$-th coefficient of the Taylor expansion of $F(x)$ in $x = 0$, it leads to

$$f_{2n}^{(1)} = \frac{1}{(2n-1)2^{2n}} \binom{2n}{n}. \tag{4.4}$$

In higher dimensions, $X_t^{(d)}$ is in the origin if and only if all its projections on each of the axes, which are simply rescaled one-dimensional random walks, are in the origin. It means that

$$r_{2n}^{(d)} = (r_{2n}^{(1)})^d. \tag{4.5}$$

For $d > 1$ there is no closed form for $f_{2n}^{(d)}$, $d > 1$, but it can be computed up to arbitrarily large n via Taylor expansion.

To estimate the first return probability, we ran AMS with $N = 1000$, averaged the results over 1000 runs and chose

$$\phi(x,t) = \begin{cases} t & x(t') \neq 0 \ \forall \ t' \leq t \\ 0 & \text{otherwise} \end{cases} \tag{4.6}$$

as importance function. This is a simple but effective choice: importance linearly increases as long as the walker has not returned to the origin, and is zero afterwards. It has the rather insignificant drawback of having to keep a flag for each trajectory to record whether or not it has returned yet.

The results are compared in Figure 4.2. The results are displayed for $t > 10^3$, and so for probabilities $f_{2n}^{(d)} < 10^{-5}$, which are much higher than what we consider as rare. This is due to the fact that numerically Taylor-expanding (4.3) up to very large n and, therefore, t soon becomes unfeasible, and complicates further for higher values of d. So we were not able to compute the exact solution for larger values of t. Even though the variance of AMS, too, increases with d, it becomes the only way to tackle the problem in a reasonable amount of time. This reiterates the necessity for computational methods when closed form solutions do not exist.

Interestingly, we also had to intentionally set K to a low, non-optimal value, such as $0.1N$, therefore increasing the computation times. Compared to the exponential decay we have encountered in the other examples considered in this thesis, the power-law decay of this distribution tends to spread out the return times; thus, the levels selected by high values of K depend on few, far-ahead trajectories and the stopping condition (3.15) is met before the bulk of the trajectories reach the rarer events. By picking low values of K we artificially slow down the algorithm, allowing small probabilities to be computed using data from many trajectories, and not just a few high-variance ones.

The results perfectly overlap for $d = 1, 2$, but a constant bias can be observed in the AMS estimates for $d = 3$. Although we cannot explain it, since the code used to generate the date was the same for $d = 2$ and $d = 3$, we guess that it could be related to the fact that a symmetric random walk becomes transitory in the latter case.

Lastly, note that neither the finite-differences method nor the Gillespie algorithm were employed, as this problem is time-discrete, and so not defined by a stochastic differential equation.

4.3 Survival in Lotka-Volterra Model

The Lotka-Volterra model [4, 27] describes the dynamics of a system of two populations, one of predators of size N_1 and one of preys of size N_2 (notice that this is the first and only multi-species model analysed in this thesis). The model assumes that, in the absence of the other species, the populations sizes decay and grow exponentially, respectively. In particular, preys have a per-capita

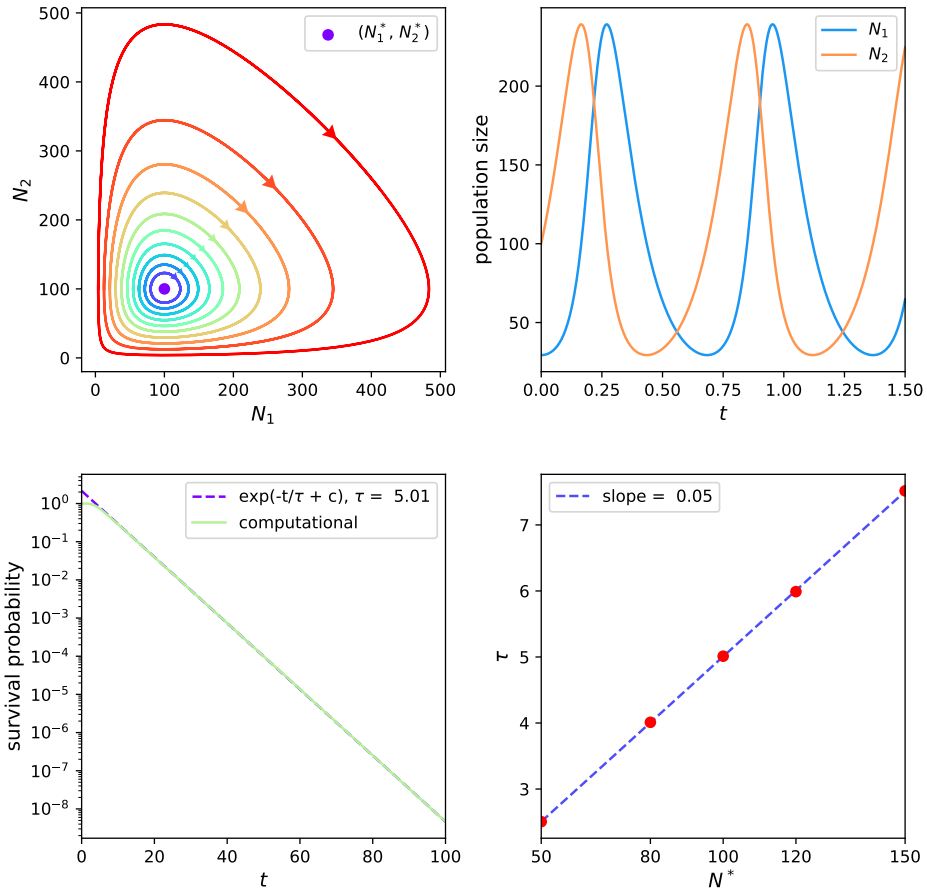

Figure 4.3: A Lotka-Volterra system with $\mu = 10$, $\sigma = 10$, $\lambda = 0.1$. Top left: the phase space with the coexistence fixed point and some sample mean-field trajectories. Top right: the time-evolution of N_1 and N_2 along a sample mean field trajectory showing oscillatory behaviour. Bottom right: the survival probability over time for a realisation starting at $(N_1^*, N_2^*) = (100, 100)$ and its exponential fit. Bottom left: the linear relation between the system size $N = N_1^* = N_2^*$ and the extinction time scale τ.

birth rate μ, while predators have a per-capita death rate σ. In addition, the species interact when a predator captures a prey, generating an offspring in the process, which happens with frequency proportional to the number of predators-prey pairs and rate λ.

This system can be modelled by a set of three chemical reactions:

$$N_1 \xrightarrow{\sigma} \varnothing \qquad N_2 \xrightarrow{\mu} 2N_2 \qquad N_1 + N_2 \xrightarrow{\lambda} 2N_1 \,. \tag{4.7}$$

The mean-field evolution of the average population sizes $\langle N_1 \rangle$ and $\langle N_2 \rangle$ is described by the following system of differential equations:

$$\begin{cases} \dfrac{\partial \langle N_1 \rangle}{\partial t} = -\sigma \langle N_1 \rangle + \lambda \langle N_1 \rangle \langle N_2 \rangle \\ \dfrac{\partial \langle N_2 \rangle}{\partial t} = \mu \langle N_2 \rangle - \lambda \langle N_1 \rangle \langle N_2 \rangle \end{cases} . \tag{4.8}$$

Despite being too simplistic to describe any real-world system, the model manages to give rise to a fundamental behaviour generally observed in nature, that is population oscillations, rather than the existence of a stable equilibrium state [4]. All the exact solutions of (4.8) are, in fact, closed orbits around the fixed point $(N_1^*, N_2^*) = (\mu/\lambda, \sigma/\lambda)$, as shown in Figure 4.3. Hence, this is the case of a conservative system. All the trajectories are marginally stable, which means that realisations with noise will neither be attracted nor repelled. Those that start in (N_1^*, N_2^*) will fluctuate around it, continuously shifting between orbits, and, eventually, end up in one of the two trapping regions of the system when they hit one of the two axes, i.e., when one of the two species goes extinct: if that happens to the preys, then the predators have no way of generating new offspring and they will die exponentially fast; if the predators go extinct instead, then the preys cannot die anymore and their number blows up to infinity. These two situations correspond to the fixed points $(0, 0)$ and $(0, \infty)$, respectively.

What we want to compute is the time evolution of the survival probability of the system as a whole, i.e., the probability that both species still coexis, in the stochastic case described by (4.7).

As predicted in previous numerical experiments [27], the survival probability tends to zero with an exponential decay and time scale τ. Figure 4.3 shows that the algorithm was able to estimate probabilities up to the order of 10^{-9} with remarkable precision using only $N = 100$ trajectories and averaging over 1000 independent runs. As importance function we have picked

$$\phi(t, N_1, N_2) = \frac{t}{|N_1 - N_1^*||N_2 - N_2^*| + 1},$$

which, intuitively, rewards trajectories that stay close to the coexistence fixed point for longer periods of time. In fact, simply trying to get far from the axes is not a good strategy to survive: points (N_1, N_2) such that $N_1 \gg N_1^*$ and $N_2 \gg N_2^*$ are part of orbits which, in contrast, pass very close to the axes as they loop around the fixed point. In these bottlenecks, very small fluctuations are enough to cause the extinction of one population. The best strategy is, instead, to loop very closely around (N_1^*, N_2^*), so that fluctuations towards the axes can be later compensated by ones in opposite direction.

Figure 4.3 also shows the linear relation between the system size N^*, when $N^* = N_1^* = N_2^*$, and the extinction time scale τ, which was also empirically recorded in [27].

4.4 Binary Annihilation

In this section we finally combine the theoretical approximation derived in Chapter 2 with the computational methods of Chapter 3 to study zero-dimensional binary annihilation systems:

$$2A \xrightarrow{\lambda} \varnothing \,. \tag{4.9}$$

We are interested in the probability of the system having n particles at time t (measured in units of $1/\lambda$) and, particularly, for $n \gg \langle n \rangle(t)$ (right tail) and $n \ll \langle n \rangle(t)$ (left tail). These two cases must be computed separately, because they represent two opposite rare event. For the former, as importance function of a state (n, t'), we have chosen the expected final value $\langle n \rangle(t)$ itself:

$$\phi(n, t') = \frac{1}{\frac{1}{n} + \lambda(t - t')},$$

taking advantage of the Markovian, i.e., memory-less, property of the system. This choice has the advantage that the mean-field behaviour is well-known even for higher-dimensional systems [9], and therefore can be trivially extended.

It should be pointed out that this is one such case as those discussed in Section 3.9, where the importance function strictly increases in-between two subsequent events, and thus the Gillespie algorithm is

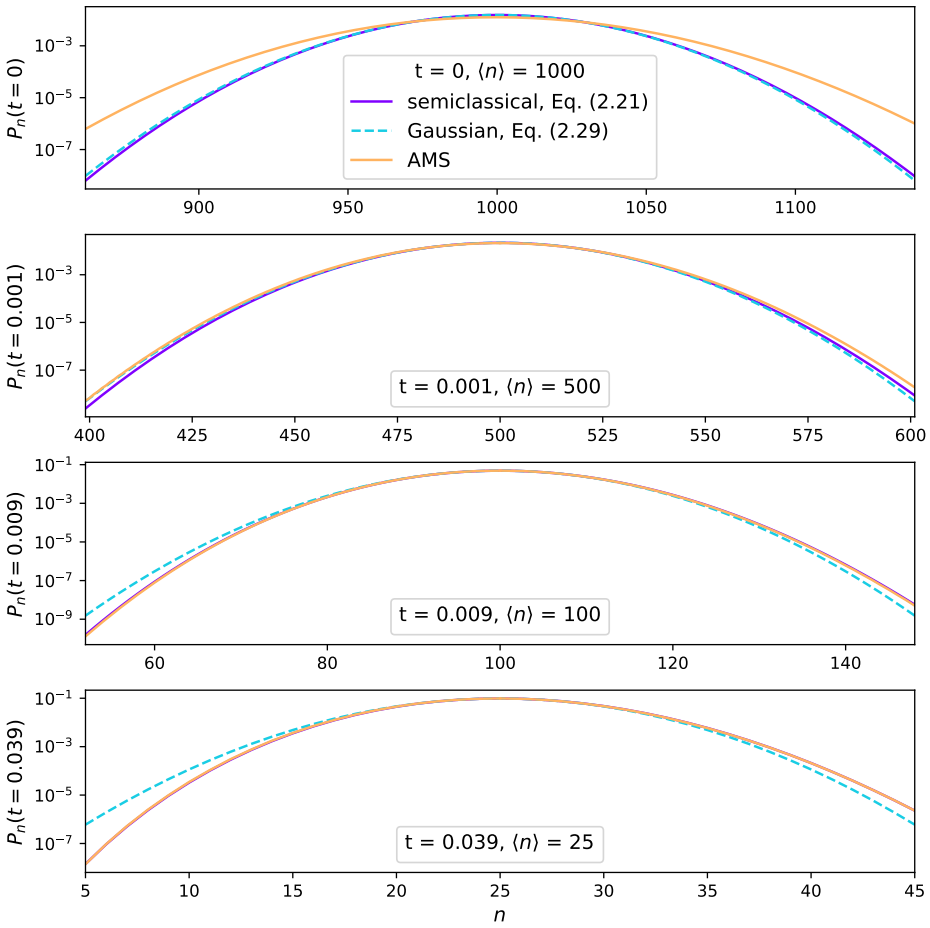

Figure 4.4: The semiclassical approximation (2.21) compared to the computational results for a zero-dimensional binary annihilation system with $\lambda = 1$ at four subsequent times $t = 0$, 0.001, 0.009, 0.039, when $\langle n \rangle = 1000$, 500, 100, 25, respectively.

not suited to simulate this process for our purposes. Indeed, notice that

$$\frac{1}{\frac{1}{n} + \lambda(t - t')} < \frac{1}{\frac{1}{n} + \lambda(t - t' - \Delta t)} \tag{4.10}$$

for any $\Delta t > 0$. Intuitively, we can say that the longer the system goes without an annihilation event happening, the greater the likelihood that the final number of particles will be higher. Notice that equation (4.10) does not provide any information on the difference $\phi(n, t') - \phi(n - 2, t' + \Delta t)$, which can be both positive or negative depending on the (positive) value of Δt.

For the left-hand tail $n \ll \langle n \rangle$, we have maintained the same importance function, but mirrored other parts of the algorithm such as storing the importance minima μ_i along trajectories, and sorting the vector in descending order.

The AMS results, compared to their respective semiclassical approximations (2.21), are presented in Figure 4.4. We notice an initial transition between the imposed Poisson distribution and the

semiclassical approximation, i.e., when

$$\langle n \rangle = \frac{1}{\frac{1}{n_0} + \lambda t} \approx \frac{1}{\lambda t}$$

does not hold true yet. The approximation (2.29), in fact, does not depend explicitly on the time passed from the start of the process t, but only on the instantaneous average $\langle n \rangle$. However, after this initial transition we note an almost perfect overlap between the two lines. This evidence proves that both of them are highly accurate approximations of the exact distribution, as they were obtained through widely different and unrelated approaches.

Moreover, when using appropriate values for N and the number of independent runs to average over, in this case 10^4 and 10^3, respectively, the computational results show no noticeable fluctuations. This computation is also arguably inexpensive in terms of time. When choosing an approximately-optimal value for K, such as $0.9N$, as exhaustively discussed in Section 3.6, a single run, executed on a single core of a general-purpose computer, takes less than 15 seconds.

Lastly, let us discuss the leading-order approximations (2.29). The Gaussian approximation for $n \approx \langle n \rangle$ deviates significantly from the exact solution at late times. The tail expressions from $n \gg \langle n \rangle$ and $n \ll \langle n \rangle$ have not been included in the plot, as they were entirely inaccurate at this probability range, as already shown in Figure 2.2. Regardless, what we can notice is that the tails are indeed asymmetric, as analytically predicted. And the asymmetry is accentuated at late times, as the Gaussian approximation worsens.

5

Conclusions

In this final chapter, we draw the conclusions of this thesis, arguing that we successfully reached our goal of proposing a highly efficient computational method from small-probability estimation in reaction-diffusion systems. In Section 5.1 we summarise and highlight the key findings of the first three chapters, focusing particularly on Chapter 4. In Section 5.2 we outline future projects.

5.1 Discussion

Chapter 2 provided the theoretical basis for the study of reaction-diffusion systems, and an in-depth explanation of the only other work specifically dealing with rare events [8], deriving an analytical approximation for the probability distribution function under some assumptions. The chapter has a pedagogical focus: no previous knowledge of the Doi-Peliti formalism, nor quantum mechanics in general, is required. Moreover, the derivation is logically ordered, uses coherent notation despite drawing from multiple sources, and includes every relevant and non-trivial step.

Chapter 3 provided a background on computational (small-) probability estimation, focusing on Monte Carlo importance splitting and its variants. It summarises the most significant theoretical results in the literature regarding its efficiency in comparison to that of other methods. The latter sections focus specifically on efficient reaction-diffusion simulations.

In Chapter 4 we begin by finally putting the theoretical results of the previous chapter to the test. In Section 4.1 we compared the efficiency of adaptive multilevel splitting to that of standard Monte Carlo when computing probabilities as low as 10^{-15} with a relative precision of 1, surprisingly finding the former to be 10^9 times more efficient than the latter, even higher than our theoretical prediction derived in Section 3.5. This evidence unmistakeably proves the necessity of using appropriate variance-reduction techniques when estimating rare-event probabilities. In Section 4.2 we also showed that AMS can be a valid alternative to numerical-calculus methods, which can become unfeasibly slow as the complexity of the equations involved increases.

The advantages of adaptive multilevel splitting also include its ease of use, as it requires only to know the system update rules, and to devise a reasonable importance function. In many cases, this can be straightforward, such as using the normalized deviation from the mean when aiming for random walks that travel far, or the maximum time before returning to the origin when aiming for random walks that return at late times. The importance function proposed in Section 4.3 is not as intuitive, but still understandable when reflecting on the characteristics of the system. The requirement to "come up" with a more-or-less intuitive importance function is the weak point of the algorithm, as discussed in Section 3.7. However, this limitation is inherent to the problem, and therefore unavoidable: speeding up the computation of the classical definition of probability (3.1) inevitably involves a trade-off. Although we successfully, and easily, found an importance function for every problem we examined, there may be more complex cases where this can prove to be impossible.

Our most significant result was combining, in Section 4.4, the semiclassical approximation obtained in Chapter 2 with the adaptive multilevel splitting prediction. After an initial transition period, the two estimates exhibit a surprisingly high degree of agreement, down to probabilities as low as 10^{-10}. This is, to our knowledge, the first time the accuracy of the semiclassical approximation has been demonstrated. At the same time, this also reiterates the merits of AMS, for which another intuitive importance function was found, i.e., the mean-field expectation.

However, note that, unlike the semiclassical approximation, AMS can be applied to more than just the particle number. As recalled above, AMS can be employed to study events that are not merely defined by a specific state of the system but also, for example, by the occurrence of reaching said state for the first time. Another shortcoming of the former is its complete lack of results for multispecies systems, which, in contrast, we were able to investigate using the latter, in Section 4.3. Overall, versatility is another important strength of AMS, which theoretical methods lack.

5.2 Outlook

The next step in continuing this project is to take proper diffusion into account. We have analysed a two-species system in Section 4.3, but diffusion would require a much greater number of species, one for each lattice site. The number of events (reactions or hops on the lattice) per unit time would increase significantly, giving rise to computational challenges in terms of both time and space. In Section 3.11 we have already illustrated a possible, but extreme, solution to the latter, but more strategies should be designed and evaluated.

Once these practical implementation challenges are overcome, the introduction of a spatial dependency will allow us to study a much broader class of problems where diffusion plays a critical role, known as *diffusion-controlled reactions* [28]. By setting aside the well-mixing hypothesis, we are able to describe non-homogeneous distributions of reactants, arising, for example, from the presence of spatially-extended catalysts, or of far-apart reactant sources and sinks. Indeed, the spatial component is of paramount importance in the vast majority of the applications discussed in Chapter 1, since, without it, the possibility of modelling real-world system is extremely limited. Rare events in this broad class of models have not been addressed yet in the literature, leaving ample opportunities for potential discoveries.

Even in simple toy models, and even when starting from a uniform distribution, the introduction of a spatial component brings about non-trivial effects. The reaction itself may create so-called *depletion regions* when the reaction time-scale is much faster than the time diffusion takes to replenish the concentration of reactants. In annihilation systems, this determines a critical dimension d_c, below which the decay of the number of particles does not follow the mean-field prediction [29].

As a final remark, recall that, although here we focused our analysis on reaction-diffusion systems, adaptive multilevel splitting can be employed to estimate rare-event probabilities of potentially any Markov process, offering endless possibilities for applications.

6

Bibliography

[1] Mark A. Donelan and Anne-Karin Magnusson. The Making of the Andrea Wave and other Rogues. *Scientific Reports*, 7(1):44124, March 2017. Publisher: Nature Publishing Group.

[2] J. D. Murray, editor. *Mathematical Biology 2: Spatial Models and Biomedical Applications*, volume 18 of *Interdisciplinary Applied Mathematics*. Springer, New York, NY, 2003.

[3] Philip Ball. Forging patterns and making waves from biology to geology: A commentary on Turing (1952) 'The chemical basis of morphogenesis'. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, 370, April 2015.

[4] J.D. Murray. *Mathematical Biology 1: An Introduction*. Springer, third edition, 2002.

[5] Oleksandr Mikhnenko, Paul Blom, and Thuc-Quyen Nguyen. Exciton Diffusion in Organic Semiconductors. *Energy Environ. Sci.*, 8, May 2015.

[6] Michael K. L. Man, Julien Madéo, Chakradhar Sahoo, Kaichen Xie, Marshall Campbell, Vivek Pareek, Arka Karmakar, E Laine Wong, Abdullah Al-Mahboob, Nicholas S. Chan, David R. Bacon, Xing Zhu, Mohamed M. M. Abdelrasoul, Xiaoqin Li, Tony F. Heinz, Felipe H. da Jornada, Ting Cao, and Keshav M. Dani. Experimental measurement of the intrinsic excitonic wave function. *Science Advances*, 7(17):eabg0192, April 2021. Publisher: American Association for the Advancement of Science.

[7] Alexander Altman and Benjamin D. Simons. *Condensed Matter Field Theory*. Cambridge University Press, 2nd edition, 2010.

[8] Vlad Elgart and Alex Kamenev. Rare event statistics in reaction-diffusion systems. *Physical Review E*, 70(4):041106, October 2004.

[9] Johannes Hofmann. Corrections to reaction-diffusion dynamics above the upper critical dimension. *Physical Review E*, 105(2):024127, February 2022.

[10] Angran Li, Ruijia Chen, Amir Barati Farimani, and Yongjie Jessica Zhang. Reaction diffusion system prediction based on convolutional neural network. *Scientific Reports*, 10(1):3894, March 2020. Publisher: Nature Publishing Group.

[11] Zhixing Cao, Rui Chen, Libin Xu, Xinyi Zhou, Xiaoming Fu, Weimin Zhong, and Ramon Grima. Efficient and scalable prediction of stochastic reaction–diffusion processes using graph neural networks. *Mathematical Biosciences*, 375:109248, September 2024.

[12] Frédéric Cérou, Arnaud Guyader, and Mathias Rousset. Adaptive Multilevel Splitting: Historical Perspective and Recent Results. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, pages 1–32, 2019.

[13] Radek Erban, S Jonathan Chapman, and Philip K Maini. A Practical Guide to Simulations of Reaction-Diffusion Systems.

[14] M. Doi. Second quantization representation for classical many-particle system. *Journal of Physics A: Mathematical and General*, 9(9):1465, September 1976.

[15] L. Peliti. Path integral approach to birth-death processes on a lattice. *Journal de Physique*, 46(9):1469–1483, September 1985. Publisher: Société Française de Physique.

[16] Uwe C. Täuber. Reaction-diffusion systems. In *Critical Dynamics*. Cambridge University Press, 2014.

[17] Johannes Knebel. Application of Statistical Field Theory to Reaction-Diffusion Problems. Technical report, April 2010.

[18] Nicholas C. Metropolis. The Beginning of the Monte Carlo Method. *Los Alamos Science Special Issue*, 15:125–130, 1987.

[19] Herman Kahn and Theodore E. Harris. Estimation of particle transmission by random sampling. *National Bureau of Standards applied mathematics series*, (12):27–30, 1951.

[20] Herman Kahn. Use of Different Monte Carlo Sampling Techniques. Technical report, RAND Corporation, January 1955.

[21] Gerardo Rubino and Bruno Tuffin, editors. *Rare Event Simulation using Monte Carlo Methods*. John Wiley & Sons, Ltd., 2009.

[22] Frédéric Cérou, Pierre del Moral, Teddy Furon, and Arnaud Guyader. Rare event simulation for a static distribution. Technical report, 2009.

[23] Pierre L'Ecuyer, Valerie Demers, and Bruno Tuffin. Splitting for Rare-Event Simulation. In *Proceedings of the 2006 Winter Simulation Conference*, pages 137–148, Monterey, CA, USA, December 2006. IEEE.

[24] M.J.J. Garvels and D.P. Kroese. A comparison of RESTART implementations. In *1998 Winter Simulation Conference. Proceedings (Cat. No.98CH36274)*, volume 1, pages 601–608, Washington, DC, USA, 1998. IEEE.

[25] S. Baars, D. Castellana, F. W. Wubs, and H. A. Dijkstra. Application of Adaptive Multilevel Splitting to High-Dimensional Dynamical Systems. *Journal of Computational Physics*, 424:109876, January 2021. arXiv:2011.05745 [physics].

[26] Charles M. Grinstead and J. Laurie Snell. *Introduction to Probability*. American Mathematical Society, 2003.

[27] Matthew Parker and Alex Kamenev. Extinction in Lotka-Volterra model. *Physical Review E*, 80(2):021129, August 2009. arXiv:0905.3728 [cond-mat, q-bio].

[28] Denis S Grebenkov. Diffusion-Controlled Reactions: An Overview. *Molecules*, 28(22):7570, November 2023. Publisher: MDPI.

[29] K. Kang, P. Meakin, J. H. Oh, and S. Redner. Universal behaviour of N-body decay processes. *Journal of Physics A: Mathematical and General*, 17(12):L665, August 1984.

DEPARTMENT PHYSICS
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY