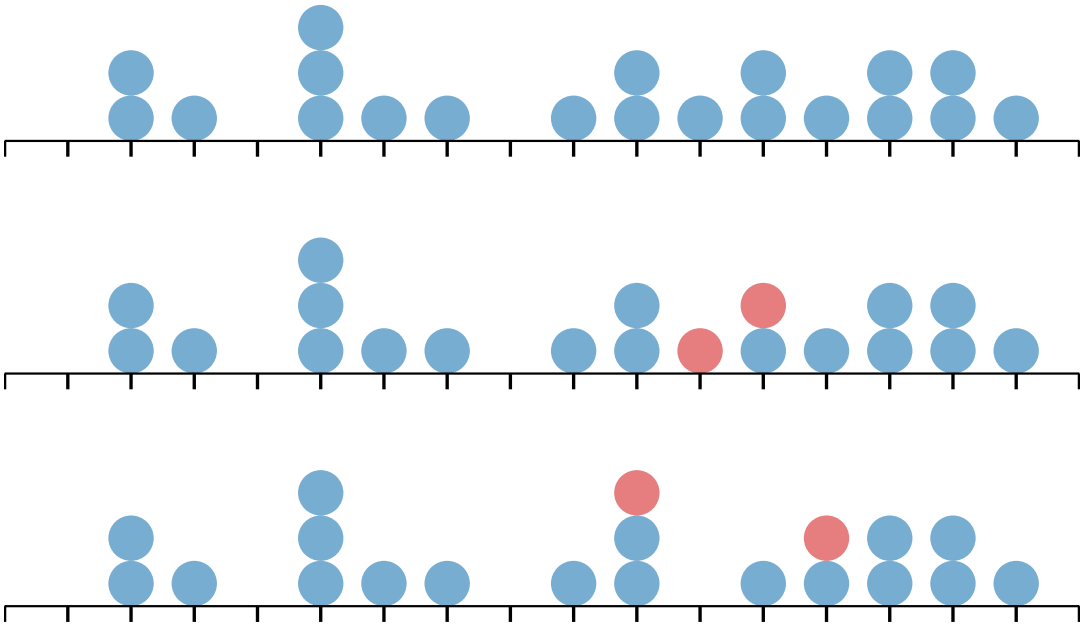


Simulating nondiffusive dynamics in reaction-diffusion systems

Utilizing a simple model of correlated motion in a single-species annihilation system

Master's thesis in Complex Adaptive Systems

ALFRED WEDDIG KARLSSON

DEPARTMENT OF PHYSICS

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2023
www.chalmers.se

Master's thesis 2023

Simulating nondiffusive dynamics in reaction-diffusion systems

Utilizing a simple model of correlated motion in a single-species annihilation system

ALFRED WEDDIG KARLSSON

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Physics
Chalmers University of Technology
Gothenburg, Sweden 2023

Simulating nondiffusive dynamics in reaction-diffusion systems
Utilizing a simple model of correlated motion in a single-species annihilation system
ALFRED WEDDIG KARLSSON

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

Master's Thesis 2023
Department of Physics
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: System of particles on a lattice moving according to a model of correlated motion used in the thesis. A pair of particles, marked in red, mutually hop away from each other such that the center of mass of the system is conserved.

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2023

Simulation of nondiffusive dynamics in reaction-diffusion systems
Utilizing a simple model of correlated motion in a single-species annihilation system
ALFERD WEDDIG KARLSSON
Department of Physics
Chalmers University of Technology

Abstract

Reaction-diffusion systems are a class of mathematical models with broad applications across physics, chemistry and biology. While very versatile, a limitation of these systems is that they assume diffusion as the only means of motion. To allow for broader application, especially in the realm of condensed matter physics, there is a need to take into account other transport processes. This thesis aims to implement an existing Monte Carlo algorithm for efficient simulation of single-species reaction-diffusion systems on a lattice, and subsequently extend it to also handle nondiffusive dynamics. The considered nondiffusive model is a simple toy model of correlated motion that preserves center of mass, and the nondiffusive nature of this kinetic model is both justified theoretically and demonstrated by simulation. It is then used in conjunction with single-species annihilation with the goal of finding how the particle density scales over time with such dynamics.

The implementation of the base algorithm shows behavior consistent with theory and previous computational results. In order to show the nondiffusivity of the correlated motion model, the time evolution of the mean squared displacement is examined and it is found to be subdiffusive and slightly slower than expected. Whether this slower evolution is due to limitations in the theory or the computational model is unknown. When simulating two-particle single species annihilation, the particle density is observed to quickly reach the same steady state of 2/11 inverse length units independent of parameter settings, and no apparent power laws are found in the approach to this steady state. What these results suggest is that there exist certain invariant configurations of the lattice in which no more reactions can occur. This leaves open the opportunity for more reasearch into the equilibrium states of various system geometries and types of reaction.

Keywords: diffusion, reaction, annihilation, correlated motion, Monte Carlo, mean-squared displacement, steady state.

Acknowledgements

I would like to express my sincerest gratitude to the Department of Physics at the University of Gothenburg for taking me in as part of their team during this project. I especially want to thank my supervisor Johannes Hofmann for suggesting the project and for all his guidance and encouragement. Without him, I would have nothing new to report. I also want to thank my examiner Bernhard Mehlig for helping me get started, as well as his many lectures that in large part inspired my interest in the field of statistical physics in the first place. My gratitude also goes out to my opponent Ilias-Marios Sarris for all the helpful discussions and for being a good friend and colleague during this past spring.

Finally, I want to thank my friends and family for always being by my side.

Alfred Weddig Karlsson, Gothenburg, June 2023

Contents

1
Introduction

Diffusion-limited reactions are chemical reactions in which spatial separation between reactants are significant enough to keep local reactions from occurring. The spontaneous spreading out of the reactants in space through diffusion therefore plays a crucial role in the rate of reaction. These processes are often represented by a class of mathematical models known as *reaction-diffusion systems*. A reaction-diffusion system can be formulated as

$$\frac{\partial}{\partial t}\vec{v} = \mathbb{D}\nabla^2\vec{v} + R(\vec{v}) \tag{1.1}$$

where $\vec{v}(\vec{x}, t)$ is some vector quantity (e.g. a concentration or probability distribution), $\mathbb{D}$ is a diagonal matrix of the diffusion coefficients in each spatial dimension, and $R(\vec{v})$ is some function representing local reactions. Put simply, it is a diffusion equation with an additional term that accounts for local reactions. What constitutes a "reaction" is broad and can mean any typical reactions between different chemicals, but also conversions from one species of particle to another, excitations, branching processes, reproduction, coalescence, or annihilation, for instance.

Owing to their generality, these types of models have proven to be very versatile beyond the context of chemical reactions and into any diffusive systems that feature interactions between particles. As such, they have seen widespread use across various disciplines of physics, chemistry and biology, and other disciplines where the formalism of statistical physics find use. Examples of applications include predator-prey dynamics in ecology, morphogenesis in biochemistry, exciton interactions in condensed matter physics, and pattern formation in general.

In reaction-diffusion systems, there generally exists a critical spatial dimension above which the diffusion process mixes the system enough that a universal mean-field theory can accurately describe the dynamics — any particle is almost equally likely to react with any other particle. This is known as the reaction-limited regime. Below the critical dimension, however, diffusion fails to mix the system enough and reactants are exceedingly likely to only react with their initial nearest neighbors. This slows down reaction because reactants become less likely to come into contact with each other, and it is instead diffusion that limits the rate at which reactions can occur. This is known as the diffusion-limited regime. The critical dimension is of theoretical importance because it sets a boundary above which the system evolves in an already well-understood universal fashion, but below which its behavior is novel.

However, diffusion is hardly the only mechanism by which molecules can move. If we take other mechanisms into account, or drive the dynamics of the system somehow, could this affect the critical dimension? Could, for instance, some form of correlated kinetics increase the critical dimension such that it forces the system to exhibit behavior outside the mean-field paradigm?

For instance, consider anomalous diffusion, a family of processes in which, for a number of possible reasons, the mean squared displacement is not linear in time. Such processes will result in a very different diffusion term in equation (1.1) and as such cannot be tackled with the same methods. There could be attractive or repulsive forces between particles that inhibit the Gaussian-distributed displacement of typical diffusive processes. Or, in extreme cases, there could be no motion possible for any one particle on its own. An example of this can be found in condensed matter physics, in a quasiparticle known as a fracton.

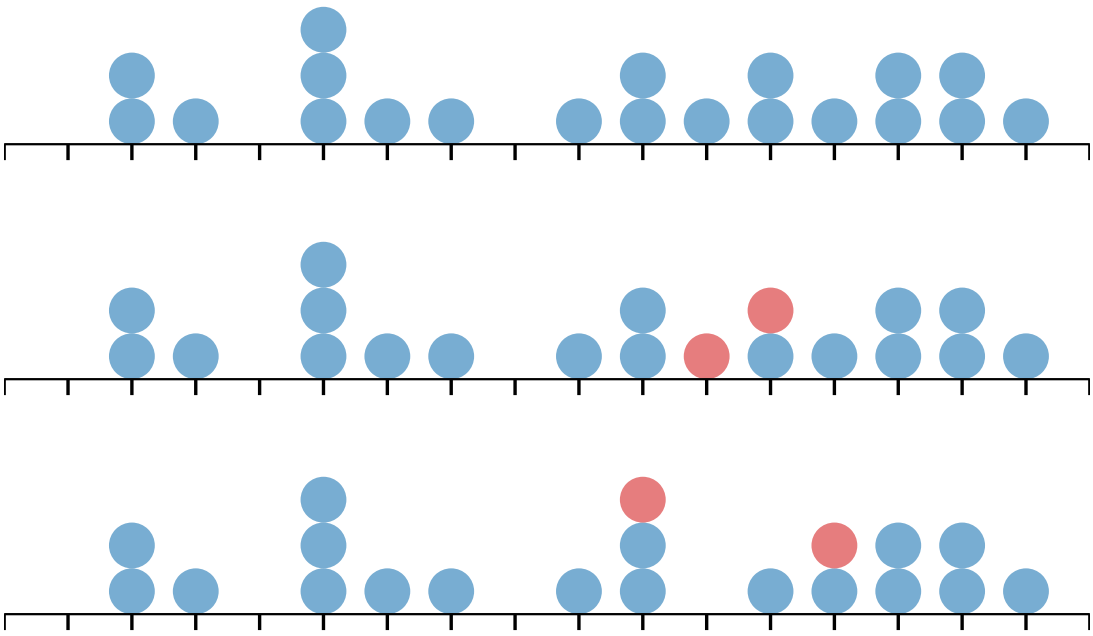


Figure 1.1: Simple model of correlated pair-hopping dynamics on a one-dimensional lattice. The pair of particles marked in red "see each other" on line 2 and jointly hop away from each other in line 3, conserving the center of mass.

Fractons are rather unique in that they are excitations that cannot move by themselves, even with forces applied on them, but rather only as a part of a bound state with another fracton [1]. Another unusual property is the conservation laws fractons follow. They move by "pushing" off of each other in such a way that center of mass and dipole moments are conserved. A simplified picture of this kind of motion is shown in figure 1.1, in which two particles mutually hop away from each other.

1.1 Aim of the project

This project aims to explore the impact of non-diffusive dynamics on the critical dimension by means of Monte Carlo simulations. The toy model will be a single-species annihilation system on a one-dimensional lattice. That is, the project will focus on reactions of the form

$$A + A \longrightarrow \emptyset \tag{1.2}$$

for some particle species A, although the implementation allows for a larger number of reactants as well. The implementation is based on the Monte Carlo method used by S.C. Park [2].

Why choose the annihilation reaction in particular? The main reason is that there are theoretical methods for predicting the outcome exactly. It is also simple to represent in simulation and has an easy to interpret quantity to follow over time, namely the particle population, or alternatively, the particle density. Additionally, it functions as a sort of "base case" for many reactions. Single-particle reactions of species A that yield new particle species B, C, etc. as products can be modeled as annihilation or coalescence reactions if the products do not have any meaningful interactions with A. In this way, analyzing annihilation reactions can yield insight into a broader set of reactions than merely literal annihilations.

One non-diffusive kinetic model is considered: a simple model of correlated motion consisting of pairs of particles hopping mutually away from or towards each other akin to figure 1.1. A theoretical justification for why this kinetic model is not diffusive is presented and also substantiated by simulation. Using this motion together with annihilation reactions, the goal is to find if the particle density decays with some power law, as well as what the system's steady states are.

2

Theory

This chapter provides the theoretical context of diffusion as well as diffusion-limited reactions. First, an overview with some heuristic arguments is presented. Next, the necessary details of how to mathematically represent diffusion, diffusion-limited reaction as well as the correlated hopping model are presented one by one.

Consider a reaction of the form in equation (1.2), i.e. single-species annihilation. Let $n(t)$ represent the number of particles in the system. Under annihilation, n should clearly decay over time. With the assumption that the reactants are homogeneously distributed across the space at all times, i.e. that diffusion keeps the system well-mixed, we can formulate a simple rate equation using the law of mass action:

$$\frac{\mathrm{d}n}{\mathrm{d}t} = -\lambda n(t)^2 \tag{2.1}$$

where λ is the reaction rate. This equation for n can be solved analytically, and one finds that $n \sim t^{-1}$ in the long term as the system approaches homogeneity through diffusion.

However, this result in not always accurate. It only works in sufficiently high spatial dimensions, and there exists a critical dimension d_c below which the decay takes on different, slower forms. Dimensional analysis will similarly fail to capture the observed dynamics below d_c. Some heuristic arguments, used by Krapivsky et al [3], can take us closer to understanding this critical dimension.

First, consider the fact that a random walk is recurrent in one and two dimensions, meaning a random walker will almost certainly return to any of its previous positions eventually. In contrast, it is transient for three dimensions and above, meaning a random walker might not ever return. The recurrence in low dimensions means that walkers generally stay in a neighborhood around their initial position for long periods of time. This leads to the formation of depletion zones, i.e. areas of very low density due to high numbers of reactions that the recurrent random walk is too "slow" to fill up again. Transient walks, on the other hand, will have a high probability of exploring a much larger neighborhood in the same time frame, meaning that it is more likely to even out the gaps left by annihilations. The existence of these depletion zones in one and two dimensions leads to the assumption of a well-mixed system being invalid in these cases, and thus the rate equation fails to describe the kinetics accurately.

Second, we know from the variance of random walks that a walker will visit a spherical region of radius $\sqrt{Dt}$, where D is the diffusion constant, during time t. This radius is also the characteristic size of the previously discussed depletion zones, as the number of particles in this region must be of order 1 since any higher number of particles than that would have already annihilated. Thus an estimate of the typical separation between particles is proportional to $(\sqrt{Dt})^d$, where d is the dimension, meaning that the particle count should roughly decay as $n \sim (Dt)^{-d/2}$. At $d = 1$ this time-dependency matches observations of diffusion-limited coalescence [4], while for $d = 2$ an additional logarithmic correction factor is required. The case $d > 2$ is the well-mixed paradigm in which $n \sim t^{-1}$ independent of d. As we shall see, a more thorough theoretical approach will in fact show results consistent with this.

2.1 Diffusion

As a starting point, we will use the typical diffusive dynamics to verify that the discussed reaction-diffusion dynamics behaves as expected. There is a mathematical difference between a discrete random walk and a continuous Brownian motion process, but we shall see that the discrete lattice will show behavior that matches the continuous space case in the end. Until this is proven, positions are denoted by m in discrete space and x in continuous space, and afterwards x is used for both cases. The diffusive dynamics will then be replaced with fracton-inspired pair-hopping dynamics, and the non-diffusive nature of these dynamics will be shown analytically.

2.1.1 Discrete space

We start by constructing the master equation of a symmetric discrete-space random walk in one dimension. Figure 2.1 shows an example of a random walk step. Let $P_m(t)$ denote the probability of finding a particle at lattice site m at time t, and γ a rate of hopping from one site to one of its neighbors. A master equation describes how this probability changes over time based on the current state of the system. A particle can enter site m by hopping either from $m-1$ or $m+1$. Likewise, a particle can leave site m by hopping to either one of its two neighboring sites. In figure 2.1, we can see an example of a rightward hop, leading to a decrease in the number of particles at site m and an increase in $m + 1$. The probability $P_m(t)$ has therefore decreased, while $P_{m+1}(t)$ has increased. Formalizing this, we see that the master equation for $P_m(t)$ must be:

$$\frac{\partial P_m}{\partial t} = \gamma(P_{m-1} + P_{m+1} - 2P_m). \tag{2.2}$$

This master equation can be solved exactly by utilizing the discrete Fourier transform and its properties,

$$\mathcal{F}\{P_m(t)\} \equiv P(k,t) = \sum_{m=-\infty}^{\infty} e^{-ikm} P_m(t), \tag{2.3}$$

$$\mathcal{F}\{f(m) - f(m \pm 1)\} = (1 - e^{\pm ik}) f(k). \tag{2.4}$$

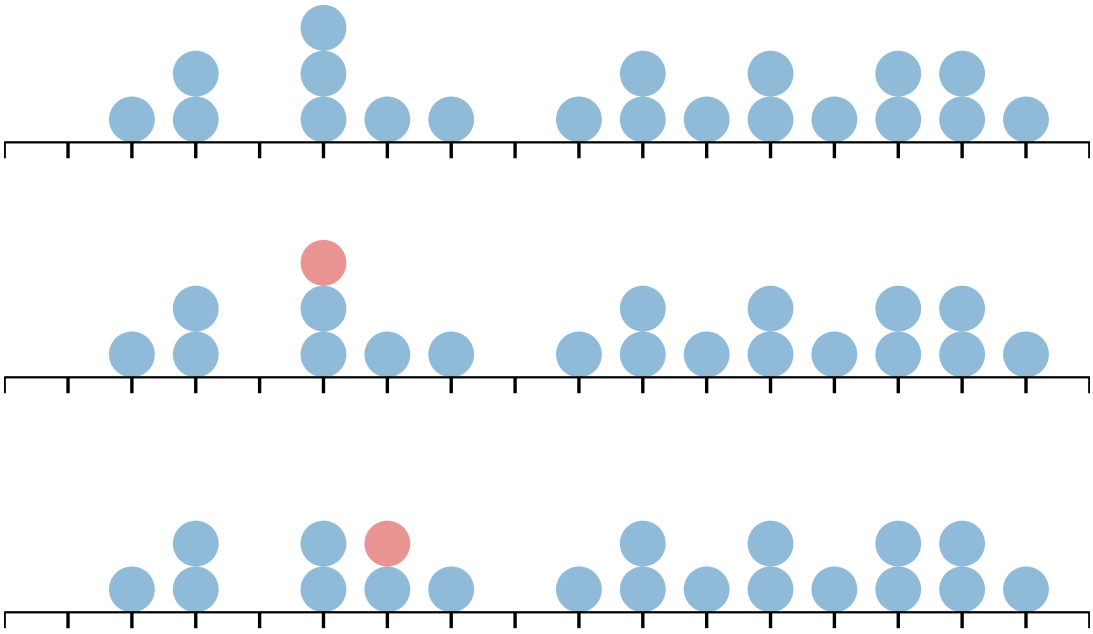


Figure 2.1: Example of a discrete random walk step on a lattice populated by particles. On line 2, the particle in red is chosen for a diffusion step, and on line 3 it has moved one step to the right. A move to the left is equally likely.

Applying these on equation (2.2) yields

$$\frac{\partial P(k,t)}{\partial t} = \gamma(e^{ik} + e^{-ik} - 2)P(k,t) = \gamma(2\cos(k) - 2)P(k,t), \tag{2.5}$$

which has the simple solution

$$P(k,t) = e^{2\gamma(\cos(k)-1)t} = e^{2\gamma t\cos(k)}e^{-2\gamma t}. \tag{2.6}$$

We can identify the first factor on the right-hand side with a property of the modified Bessel functions of the first kind I_m:

$$e^{z\cos(k)} = \sum_{m=-\infty}^{\infty} e^{ikm} I_m(z). \tag{2.7}$$

Inverse transforming equation (2.6) therefore results in

$$P_m(t) = I_m(2\gamma t)e^{-2\gamma t}. \tag{2.8}$$

Thus we have arrived at a probability distribution describing the discrete-space random walk. Note, however, that this derivation required the assumption of an infinite lattice. This is fine as long as the random walk has not yet hit the boundaries, but as soon as the boundaries have been reached, a different approach is needed. When verifying that this infinite-lattice result is consistent with the continuous-space case, we will also find an elegant way to show what late-stage distribution the periodic boundary condition leads to.

2.1.2 Continuous space

To contrast, let us consider the continuous-space case, which is governed by the very similar *diffusion equation*

$$\frac{\partial P}{\partial t} = D\frac{\partial^2 P}{\partial x^2}, \tag{2.9}$$

where D is the diffusion coefficient, here taken to be constant. Note that this continuous diffusion equation can be acquired from the discrete master equation (2.2) by dividing it by the squared lattice spacing $(\Delta x)^2$, which turns it into a discretized second derivative, and letting $\Delta x \to 0$. Then D and the previous hopping rate γ must be related according to

$$D = \gamma(\Delta x)^2. \tag{2.10}$$

Of course, this continuous case does differ from the discrete model; however, as the system grows in size the continuous model is an increasingly accurate approximation. We will see that the final results are very similar and the discrepancy is negligible on the relevant time scales.

To check the behavior of the diffusion dynamics, consider the diffusion propagator $P(x, x'; t, t')$ defined by the equation

$$\left[\frac{\partial}{\partial t} - D\frac{\partial^2}{\partial x^2}\right] P(x, x'; t, t') = \delta(x - x')\delta(t - t') \tag{2.11}$$

where $\delta(x)$ is the Dirac delta distribution. When D is constant in time, we can set $t' = 0$ without loss of generality. Equation (2.11), coupled with periodic boundary conditions, has a set of eigenfunctions

$$\varphi_n(x) = \frac{1}{\sqrt{L}} e^{ik_n x}, \quad \text{where } k_n = \frac{2\pi n}{L}. \tag{2.12}$$

A solution to the propagator is therefore the superposition

$$P(x, x'; t) = \theta(t)\frac{1}{L}\sum_{n=-\infty}^{\infty} e^{ik_n(x-x') - Dk_n^2 t} \tag{2.13}$$

Already at this point we can find a limiting behavior, namely the long time limit. Note that the term $-Dk_n^2 t$ is strictly negative for $n \neq 0$, so as $t \to \infty$, the exponent approaches $-\infty$ and thus the $n \neq 0$ terms all go to zero. The term $n = 0$, however, has a zero exponent and thus the exponential function is 1. Thus equation (2.13) reduces down to just one term,

$$P(x, x'; t \to \infty) = \theta(t)\frac{1}{L}. \tag{2.14}$$

Thus we see that the long time limit of the propagator in the periodic boundary conditions is a uniform distribution.

To deal with the short-time limit, let us consider the Fourier transform of the summand with respect to n and utilize the Poisson summation:

$$\sum_{n=-\infty}^{\infty} f(n) = \sum_{\mu=-\infty}^{\infty} F(\mu) \quad \text{where } F(\mu) = \mathcal{F}\{f(n)\}(\mu). \tag{2.15}$$

With the summand

$$f(n) = \frac{1}{L} e^{2\pi i n(x-x')/L - D(2\pi n/L)^2 t} \tag{2.16}$$

the Fourier transform with respect to n is

$$F(\mu) = \frac{1}{\sqrt{4\pi D t}} e^{-(x-x'-\mu L)^2/4Dt} \tag{2.17}$$

and thus an alternate representation of the propagator is

$$P(x, x'; t) = \theta(t) \frac{1}{\sqrt{4\pi D t}} \sum_{\mu=-\infty}^{\infty} e^{-(x-x'-\mu L)^2/4Dt}. \tag{2.18}$$

Whereas the previous representation (2.13) gave us information about the long-time limit, this representation (2.18) will instead tell us about the short-time limit. As $t \to \infty$, the summand's exponent goes to 0, i.e. the summand goes to 1, and the normalization coefficient goes to 0. We can see that it does not diverge, but it also gives no useful information. But for shorter t, the summand forms a Gaussian distribution together with the normalization coefficient. The propagator is thus a sum of Gaussian distributions of differing mean values $x' + \mu L$ but the same variance $2Dt$. Since we are limiting our view to a region of length L, the contributions from the terms with $\mu \neq 0$ are negligible when $\sqrt{2Dt} \ll L$, meaning the $\mu = 0$ term is the only one of practical importance. When the dust settles, what is left of the propagator is

$$P(x, x'; t) = \theta(t) \frac{1}{\sqrt{4\pi D t}} e^{-(x-x')^2/4Dt}, \quad t \ll \frac{L^2}{D}. \tag{2.19}$$

The center point x' can be set to 0 without loss of generality. Notice that the time $t = L^2/D$ is a characteristic time scale of the system's evening out — at times of this order, the standard deviation of the particle distribution is of the same order as the size of the lattice.

2.1.3 Compatibility of the discrete and continuous cases

We have showed that the continuous diffusion equation (2.9) gives a Gaussian distribution (2.19) when boundaries are irrelevant. Is this result consistent with the distribution (2.8) found in the discrete-space case? It can be shown that the asymptotic behavior of equation (2.8) is precisely equation (2.19), as

$$I_m(2\gamma t) e^{-2\gamma t} \xrightarrow[t\to\infty]{} \frac{1}{\sqrt{4\pi\gamma t}} e^{-m^2/4\gamma t}. \tag{2.20}$$

To prove this fact, we can go back to the discrete master equation (2.2) and utilize probability-generating functions. A probability-generating function $F(z,t)$ for a probability distribution $P_m(t)$ is defined as

$$F(z,t) = \sum_{m=-\infty}^{\infty} z^m P_m(t) \tag{2.21}$$

where $z \in \mathbb{C}$. Cauchy's residue theorem allows us to reconstruct the probability distribution $P_m(t)$ with a contour integral, which will be the key to the proof:

$$2i\pi P_m(t) = \oint \frac{F(z,t)}{z^{m+1}} \mathrm{d}z. \tag{2.22}$$

First, we need to find the probability-generating function of P_m. Starting from equation (2.2), we have

$$\frac{\partial F}{\partial t} = \frac{\partial}{\partial t} \sum_{m=-\infty}^{\infty} z^m P_m(t) = \sum_{m=-\infty}^{\infty} z^m \frac{\partial P_m(t)}{\partial t} = \sum_{m=-\infty}^{\infty} z^m \gamma (P_{m-1} + P_{m+1} - 2P_m). \tag{2.23}$$

Since $m \in (-\infty, \infty)$, it is possible to simply shift the summation index over by one step, $m \to m \pm 1$ without changing the sum. This can be utilized if we split up the sum into its individual terms:

$$\begin{aligned}
\sum_{m=-\infty}^{\infty} z^m (P_{m-1} + P_{m+1} - 2P_m) &= \sum_{m=-\infty}^{\infty} z^m P_{m-1} + \sum_{m=-\infty}^{\infty} z^m P_{m+1} - 2 \sum_{m=-\infty}^{\infty} z^m P_m \\
&= \sum_{m=-\infty}^{\infty} z^{m+1} P_m + \sum_{m=-\infty}^{\infty} z^{m-1} P_m - 2 \sum_{m=-\infty}^{\infty} z^m P_m \\
&= z \sum_{m=-\infty}^{\infty} z^m P_m + \frac{1}{z} \sum_{m=-\infty}^{\infty} z^m P_m - 2 \sum_{m=-\infty}^{\infty} z^m P_m \\
&= zF(z,t) + \frac{1}{z} F(z,t) - 2F(z,t),
\end{aligned} \tag{2.24}$$

meaning we have the differential equation

$$\frac{\partial F(z,t)}{\partial t} = \gamma \Big(z + \frac{1}{z} - 2 \Big) F(z,t) \quad \Longrightarrow \quad F(z,t) = c(z) e^{\gamma t (z + 1/z - 2)}, \tag{2.25}$$

where $c(z)$ remains to be determined with an initial condition. If we let the initial condition be $P_m(t=0) = \delta_{m,0}$, i.e. all particle start at the center of the lattice, then

$$F(z, t=0) = c(z) e^0 = c(z) = \sum_m P_m(0) = \sum_m \delta_{m,0} = 1, \tag{2.26}$$

i.e. for the initial condition $P_m(0) = \delta_{m,0}$ we get that $c(z) \equiv 1$.

We are now ready to use the residue theorem to invert the probability-generating function. Since the only singularity is in $z = 0$, we can choose the unit circle as our curve, for instance. We now have

$$P_m(t) = \frac{1}{2i\pi} \oint_{|z|=1} \frac{e^{\gamma t (z + 1/z - 2)}}{z^{m+1}} \mathrm{d}z = \frac{1}{2i\pi} \oint_{|z|=1} e^{\gamma t (z + 1/z - 2) - (m+1)\ln(z)} \mathrm{d}z. \tag{2.27}$$

This will be solved using the method of steepest descent, which can be used for integrals of the form:

$$\int_C f(z)e^{\lambda g(z)}\mathrm{d}z, \tag{2.28}$$

where C is some path, $f(z)$ and $g(z)$ are analytic functions along C, and λ is some large number. In our case, f is the identity function, C is the unit circle and λ corresponds to the dimensionless time γt. We need to find saddle points of g:

$$g(z) = \gamma t(z + 1/z - 2) - (m+1)\ln(z) \implies g'(z) = \gamma t\Big(1 - \frac{1}{z^2}\Big) - \frac{m+1}{z}. \tag{2.29}$$

Setting the derivative to zero and assuming that m is large enough to make $m+1 \sim m$ yields the saddle points:

$$g'(z) = \gamma t\Big(1 - \frac{1}{z^2}\Big) - \frac{m}{z} = 0 \implies z_0 = \frac{m \pm \sqrt{m^2 + 4\gamma^2 t^2}}{2\gamma t}, \tag{2.30}$$

and we can see that as $t \to \infty$, $4\gamma^2 t^2$ dominates in the radicand and we get

$$z_0 \to \frac{m}{2\gamma t} \pm \frac{\sqrt{4\gamma^2 t^2}}{2\gamma t} = \frac{m}{2\gamma t} \pm \frac{2\gamma t}{2\gamma t} = \frac{m}{2\gamma t} \pm 1. \tag{2.31}$$

Now introduce $\alpha = m^2/2\gamma t$, which allows us to write the saddle points as

$$z_0 = \frac{\alpha}{m} \pm 1. \tag{2.32}$$

The exponent can then be expanded to

$$g(z) = \frac{m^2}{2\alpha}(z - z_0)^2 - \frac{\alpha}{2} + \mathcal{O}\Big(\frac{1}{m}\Big). \tag{2.33}$$

Since the only singularity is in the origin, we can deform the integration curve to any shape as long as the deformation does not cross the origin. By choosing the integration curve to be a new circle C' with radius $\to \infty$ that crosses the saddle point z_0 such that it is completely vertical and ascending at z_0. All other contributions are assumed to oscillate rapidly such that they average to 0, leaving only the vertical line through z_0 behind:

$$\begin{aligned} P_n(t) &= \frac{1}{2\pi i}\oint_{C'} e^{-\alpha/2}e^{-\gamma t(z-z_0)^2}\mathrm{d}z \quad [\text{Set } r = i(z - z_0)] \\ &= e^{-\alpha/2}\frac{i}{2\pi i}\int_{-\infty}^{\infty} e^{-\gamma t r^2}\mathrm{d}r = e^{-\alpha/2}\frac{1}{2\pi}\int_{-\infty}^{\infty} e^{-\gamma t r^2}\mathrm{d}r \\ &= e^{-\alpha/2}\frac{1}{2\pi}\sqrt{\frac{\pi}{\gamma t}} = \frac{1}{2\sqrt{\pi\gamma t}}e^{-m^2/4\gamma t} = \frac{1}{\sqrt{4\pi\gamma t}}e^{-m^2/4\gamma t}. \end{aligned} \tag{2.34}$$

This proves the limit (2.20), and so we get exactly the same Gaussian distribution as in equation (2.19) with the mean 0, exchanging x for m. Thus for long time scales the discrete- and continuous-space models are equivalent.

Another neat feature of the probability-generating function is that we can derive

the mean and variance of the distribution quickly. Starting from equation (2.21), we can see that

$$\frac{\partial}{\partial z}F(z,t) = \sum_{m=-\infty}^{\infty} m z^{m-1} P_m(t) \implies \frac{\partial}{\partial z}F(z,t)\Big|_{z=1} = \sum_{m=-\infty}^{\infty} m P_m(t) = \langle m \rangle. \tag{2.35}$$

In the same fashion, we can prove that

$$\frac{\partial^2}{\partial z^2}F(z,t)\Big|_{z=1} = \langle m^2 \rangle - \langle m \rangle = \mathrm{Var}(m) + \langle m \rangle^2 - \langle m \rangle. \tag{2.36}$$

In our case, we have

$$F(z,t) = e^{\gamma t(z+1/z-2)} \implies \langle n \rangle = \gamma t\Big(1 - \frac{1}{z^2}\Big)e^{\gamma t(z+1/z-2)}\Big|_{z=1} = 0 \tag{2.37}$$

and with a zero mean, the variance simply becomes the second derivative evaluated at $z = 1$,

$$\begin{aligned}\langle m^2 \rangle &= \frac{\partial^2}{\partial z^2}F(z,t)\Big|_{z=1} = \gamma t\Big(\frac{2}{z^3}\Big)e^{\gamma t(z+1/z-2)} + \gamma^2 t^2\Big(1 - \frac{1}{z^2}\Big)^2 e^{\gamma t(z+1/z-2)}\Big|_{z=1} \\ &= 2\gamma t e^0 = 2\gamma t\end{aligned} \tag{2.38}$$

which proves one of the characteristic properties of diffusive processes: the mean squared displacement, i.e. the spatial variance, is linear in time. If the mean squared displacement instead follows a non-linear power law $t^a, a \in \mathbb{R}^+$, the dynamics instead belong to the more general family of processes known as *anomalous diffusion*. A process of this kind is called *superdiffusive* if $a > 1$ and *subdiffusive* when $a < 1$.

Repeating the derivation for a general lattice spacing Δx, which was previously assumed to be 1, shows that the continuity limit yields:

$$\langle x^2 \rangle = 2(\Delta x)^2 \gamma t \xrightarrow[\Delta x \to 0]{} 2Dt, \tag{2.39}$$

which is consistent with the continuous diffusion process in equation (2.19).

To illustrate that the Bessel distribution (2.8) approaches the Gaussian distribution (2.19) sufficiently fast for this application, we can use a graphical approach. Figure 2.2 shows how the discrepancy between these functions shrink on a small time scale. Note that the characteristic time scale of the Gaussian distribution is L^2/D, which is generally huge compared to the shown time points for any sensible lattice size L. Since the lattice spacing is taken to be 1, D and γ have the same value. The quick coalescence indicates that the discrete-space distribution behaves just like the continuous-space distribution in the long term. With the insight that the discrete and continuous cases are equivalent in practice, there is no longer any need to distinguish between discrete and continuous space. From now on, x will be used to denote a lattice site rather than m.

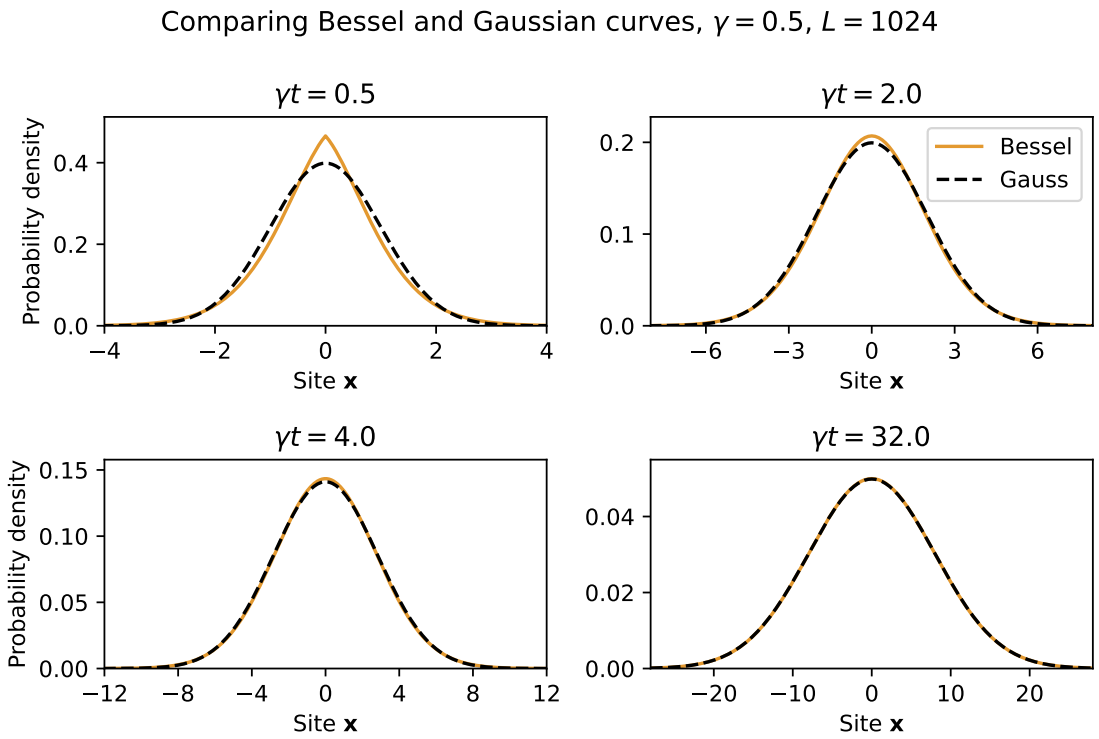


Figure 2.2: The theoretically predicted discrete-space distribution (left-hand side of equation (2.20)) compared to the continuous-space distribution (right-hand side of equation (2.20)) with hopping rate $\gamma = 0.5$. Even within a relatively short timespan, the distributions match up very closely.

2.2 Reaction and diffusion

With the theory of diffusion laid out, it is time to introduce reactions to the picture. The annihilation reaction can be handled rather simply in a discrete-space model. If several particles occupy the same lattice point, then a pair of them will react with some rate λ. If a pair reacts, then the particle count is reduced by 2 at the site. In continuous space, there would be some range of interaction between the particles. By choosing a sufficiently large lattice spacing and appropriate reaction rate, the discrete-space model can sidestep the need for a range of interaction and yield equivalent statistics for less computational power. The generalization to three-particle reactions or beyond is straightforward in silico, though the time-scaling of the population count/density takes on a different nature.

Moving on with the discrete-space model, consider reactions on the form

$$kA \xrightarrow{\lambda} \emptyset, \tag{2.40}$$

where $k \in \mathbb{N}$ is the number of reactants in the annihilation reaction. When $k = 2$ we get back the previously considered case. However, this section will handle the most general case of k-particle annihilations. The master equation for the reaction

is

$$\begin{aligned}\frac{\partial P(n,t)}{\partial t} &= \lambda(n+k)(n+k-1)\dots(n+1)P(n+k,t) \\ &\quad - \lambda n(n-1)\dots(n-k+1)P(n,t)\end{aligned} \tag{2.41}$$

where $P(n,t)$ is the probability to have n particles at time t and λ is the reaction rate. This equation holds for each site individually. Similarly to equation (2.1), the law of mass action under the well-mixed assumption gives the differential equation

$$\frac{\mathrm{d}n}{\mathrm{d}t} = -\lambda n(t)^k \tag{2.42}$$

where λ is once again the reaction rate. Solving this equation gives the asymptotic scaling

$$n \sim (\lambda t)^{-1/(k-1)}, \quad t \to \infty. \tag{2.43}$$

We can note that for the previously discussed case $k = 2$, we get back the same mean-field scaling $n \sim t^{-1}$. However, the critical dimension does not necessarily equal 2 anymore. This is because multi-particle annihilation necessarily leads to slower reaction rates, as the probability of enough particles occupying the same site decreases with increasing number of reactants, and so the limiting effect of diffusion becomes less significant. It is known that the critical dimension for k-particle annihilation reactions follows a simple rule [5, 6]:

$$d_\mathrm{c} = \frac{2}{k-1}, \tag{2.44}$$

meaning that for $k = 2, 3, 4$ the critical dimension is 2, 1 and < 1 respectively. Thus, it is possible to numerically verify the mean-field theory by increasing the number of reactants instead of the number of spatial dimensions. For 3-particle annihilation, the critical dimension is 1 and so it follows the mean-field paradigm up to a logarithmic correction factor, whereas 4-particle annihilation follows the mean-field paradigm exactly.

This critical dimension, as well as the proper late-time density scaling, can be derived analytically by considering the problem as a quantum many-body system. Worth reiterating is that this system is completely classical and no quantum mechanics are needed to describe it; however, the formalism of second quantization for quantum many-body problems nevertheless happens to be viable here, as first shown by M. Doi [7]. The full derivation is laborious and beyond the scope of this project, but a few key insights will be summarized below following the procedure as laid out by J. Cardy [8].

A good place to start is to translate the discrete-space reaction-diffusion model into a quantum many-body system. We define a vacuum state $|0\rangle$, meaning the state of all sites holding zero particles. Then, let us define the annihilation and creation operators a_j and $a_j^\dagger$ on site j, as Hermitian conjugates of each other. Placing a single

particle at site j is expressed as $a_j^\dagger |0\rangle$. Any general state can then be expressed on the form

$$|\{n_j\}_j\rangle = \prod_j (a_j^\dagger)^{n_j} |0\rangle \,, \tag{2.45}$$

where n_j is the occupation count at site j. The annihilation and creation operators commute as follows:

$$[a_j, a_k^\dagger] = \delta_{j,k}, \tag{2.46}$$

$$[a_j, a_k] = [a_j^\dagger, a_k^\dagger] = 0. \tag{2.47}$$

With these definitions, we can construct a state $|\psi(t)\rangle$ in Fock space such that

$$|\psi(t)\rangle = \sum_{\{n_j\}} p(\{n_j\}, t) \prod_j (a_j^\dagger)^{n_j} |0\rangle \tag{2.48}$$

where $p(\{n_j\}, t)$ is the probability of a certain configuration $\{n_j\}$ at time t and the sum is over all possible configurations. One can then show that the time evolution of $|\psi(t)\rangle$ can be written as a kind of Schrödinger equation:

$$\frac{\mathrm{d}\,|\psi(t)\rangle}{\mathrm{d}t} = -H\,|\psi(t)\rangle \,, \tag{2.49}$$

which has the typical solution

$$|\psi(t)\rangle = e^{-Ht}\,|\psi(0)\rangle \,. \tag{2.50}$$

This Hamiltonian H consists of a diffusive term H_d and a reactive term H_r. These terms are derived using the master equations for the diffusion and the reaction respectively, leading to the diffusive term

$$H_d = D \sum_{\langle ij \rangle} (a_i^\dagger - a_j^\dagger)(a_i - a_j), \tag{2.51}$$

and the reactive term

$$H_r = -\lambda \sum_i (1 - (a_i^\dagger)^k) a_i^k. \tag{2.52}$$

With the quantum system set up, the challenge lies in making sense of $e^{-(H_d + H_r)t}$. This problem has been tackled before [5, 8–10] and the details are an irrelevant detour. Instead, we hone in on the the result of interest to us, namely the case $d = 1$, $k = 2$ which has the exact density decay [10]:

$$\rho(t) \sim \frac{1}{\sqrt{8\pi D t}}, \quad t \to \infty, \tag{2.53}$$

where $\rho(t) = n(t)/L$ is the particle density and L is the lattice size. We will come back to this result when analyzing the simulation results in chapter 4.2.

2.3 Beyond diffusion: simple model of correlated hopping

An approach to modeling fracton kinetics in one dimension is to have the motion of a pair of particles consist of simply mutually hopping away from each other, as exemplified by figure 1.1. To achieve detailed balance, the exact opposite motion of hopping inwards towards each other is also possible and equally probable. Initially the most general form will be considered, where the left and right particles in the pairing can hop different distances b_1 and b_2 respectively, and it will be shown that this leads to diffusive behavior when $b_1 \neq b_2$, and as such only the case $b_1 = b_2$ will be of interest.

Let a be the distance at which two particles can form a hopping pair, the "pairing range". Furthermore, let b_1 and b_2 be the hopping distances of the left and right particle respectively. Then the inward hopping process has the pairing range $a+b_1+b_2$. Note that this means that in the pairing, the left particle always hops a distance b_1 and the right particle always hops b_2, regardless of if the hopping is inward or outward.

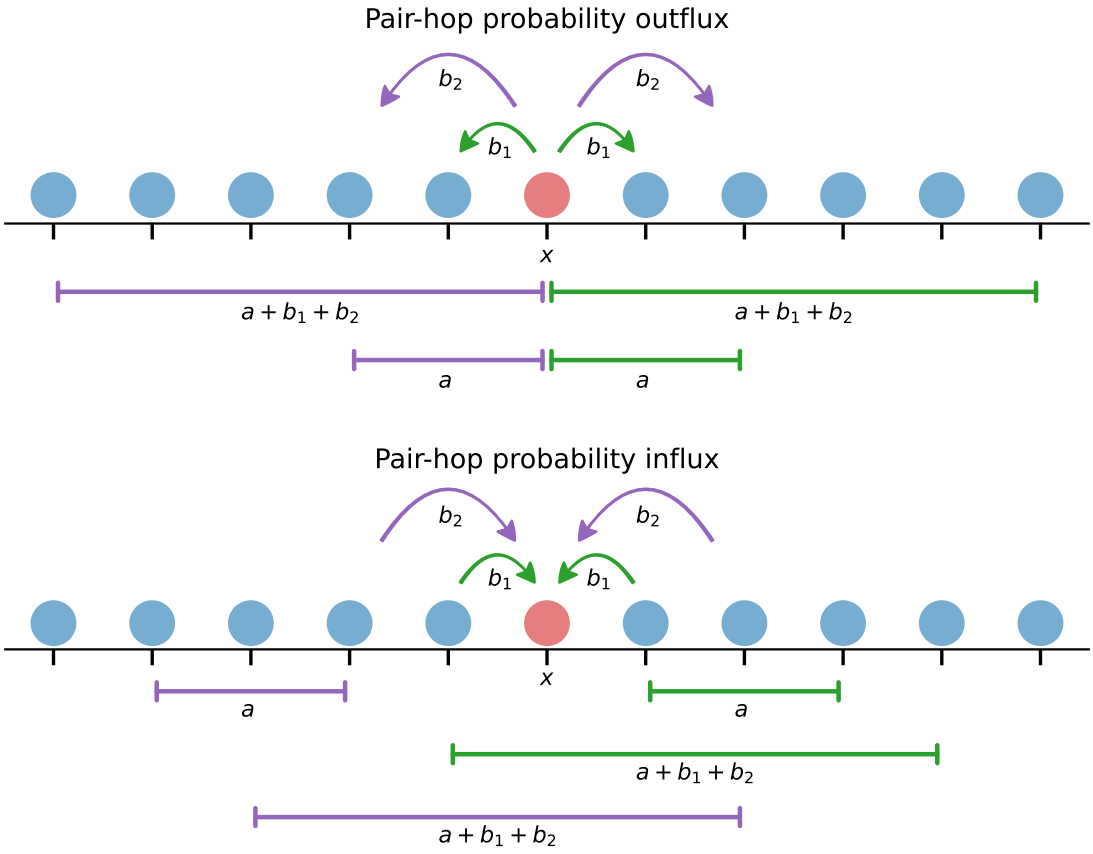


Figure 2.3: Visualization of all possible hops that contribute to the probability flux in the master equation (2.54) for the sample site x. This figure uses the sample parameters $a = 2$, $b_1 = 1$, $b_2 = 2$. Influx gives positive terms, outflux gives negative.

Figure 2.3 shows the possible paths a particle can take out of or in to a given site, in this case the central site. The bars below the number line mark which particles pair with each other to achieve each hop. Since particles need to be present at both ends of the bar for the hop to occur, the probabilities at these sites need to be multiplied. Using this as a guide, the master equation for this process is then found to be the rather intimidating

$$\begin{aligned}\frac{\partial P(x)}{\partial t} = \alpha\Big[&P(x-b_2)P(x-b_2-a) + P(x+b_1)P(x+b_1+a)\\ &+P(x-b_1)P(x+b_2+a) + P(x+b_2)P(x-b_1-a)\\ &-P(x)P(x-a-b_1-b_2) - P(x)P(x+a+b_1+b_2)\\ &-P(x)P(x-a) - P(x)P(x+a)\Big],\end{aligned} \tag{2.54}$$

where $P(x)$ refers to the probability of finding a particle at site x and α is the rate of hopping . Since this is a nonlinear master equation, solving it is no trivial feat. It might be helpful to instead replace the probability with the population $n(x)$. Doing this, we get the equation for the time evolution of the density at site x according to

$$\begin{aligned}\frac{\partial n(x)}{\partial t} = \alpha\Big[&n(x-b_2)n(x-b_2-a) + n(x+b_1)n(x+b_1+a)\\ &+n(x-b_1)n(x+b_2+a) + n(x+b_2)n(x-b_1-a)\\ &-n(x)n(x-a-b_1-b_2) - n(x)n(x+a+b_1+b_2)\\ &-n(x)n(x-a) - n(x)n(x+a)\Big],\end{aligned} \tag{2.55}$$

i.e. we get the same exact equation again but with a different interpretation. Assume now that there exists some equilibrium population n_0. Further assuming that the system is close to this equilibrium, we can linearize the equation around the equilibrium according to $n(x) = n_0 + \phi(x)$, and subsequently tossing away any terms not linear in the perturbation ϕ. Putting all remaining terms in a convenient order, this lands us in

$$\begin{aligned}\frac{1}{n_0\alpha}\frac{\partial\phi(x)}{\partial t} = &-\phi(x-b_1-b_2-a) - \phi(x+b_1+b_2+a)\\ &+\phi(x-b_1-a) + \phi(x+b_1+a)\\ &+\phi(x-b_2-a) + \phi(x+b_2+a)\\ &+\phi(x-b_2) + \phi(x+b_2)\\ &+\phi(x-b_1) + \phi(x+b_1)\\ &-\phi(x-a) - \phi(x+a) - 4\phi(x).\end{aligned} \tag{2.56}$$

Now we will make further approximations by way of the fourth-order Taylor expansion

$$\begin{aligned}\phi(x+\delta) = \phi(x) + \delta\frac{\partial}{\partial x}\phi(x) + \frac{\delta^2}{2!}\frac{\partial^2}{\partial x^2}\phi(x)\\ + \frac{\delta^3}{3!}\frac{\partial^3}{\partial x^3}\phi(x) + \frac{\delta^4}{4!}\frac{\partial^4}{\partial x^4}\phi(x)\end{aligned} \tag{2.57}$$

This monstrous expansion will be handled piece by piece. First, note that the zeroth order terms $\phi(x)$ vanish simply because there are an equal number of positive and negative terms. Consider now the first order terms, which all have the common first-derivative factor and in which only the δ for each term varies:

$$\begin{aligned}&\Big[-(-b_1-b_2-a)-(b_1+b_2+a)+(-b_1-a)+(b_1+a)+(-b_2-a)+(b_2+a)\\&+(-b_2)+b_2+(-b_1)+b_1-(-a)-a\Big]\frac{\partial\phi(x)}{\partial x}=0\end{aligned} \tag{2.58}$$

since all terms cancel pair-wise due to the alternating signs. For the same reason the third-order terms also cancel. The second order terms, however, add up pair-wise rather than cancel:

$$\begin{aligned}&\frac{1}{2}\Big[-(-b_1-b_2-a)^2-(b_1+b_2+a)^2+(-b_1-a)^2+(b_1+a)^2+\dots\Big]\frac{\partial^2\phi(x)}{\partial x^2}\\&=\frac{1}{2}\Big[-2(b_1+b_2+a)^2+2(b_1+a)^2+2(b_2+a)^2+2b_2^2+2b_1^2-2a^2\Big]\frac{\partial^2\phi(x)}{\partial x^2}\\&=\Big[-(b_1+b_2+a)^2+(b_1+a)^2+(b_2+a)^2+b_2^2+b_1^2-a^2\Big]\frac{\partial^2\phi(x)}{\partial x^2}\\&=\Big[-(b_1^2+b_2^2+a^2+2b_1b_2+2b_1a+2b_2a)+(b_1^2+2b_1a+a^2)\\&\quad+(b_2^2+2b_2a+a^2)+b_2^2+b_1^2-a^2\Big]\frac{\partial^2\phi(x)}{\partial x^2}\\&=\Big[-2b_1b_2+b_1^2+b_2^2\Big]\frac{\partial^2\phi(x)}{\partial x^2}=(b_1-b_2)^2\frac{\partial^2\phi(x)}{\partial x^2},\end{aligned} \tag{2.59}$$

and we see that the first nonzero contribution is of second order when $b_1 \neq b_2$, meaning it is diffusive for asymmetric hopping. We also see that the second-order term cancels for the case of symmetric hopping $b_1 = b_2$, motivating the need for a fourth order Taylor expansion. Looking further into the symmetric hopping dynamics, i.e. the case that conserves center of mass, we let $b_1 = b_2 \equiv b$. Our first nonzero contribution in this case is of the fourth order, and it takes on the form

$$\begin{aligned}&\frac{1}{24}\Big[-2(2b+a)^4+4(b+a)^4+4b^4-2a^4\Big]\frac{\partial^4\phi(x)}{\partial x^4}\\&=\frac{1}{12}\Big[-(16b^4+32b^3a+24b^2a^2+8ba^3+a^4)+2(b^4+4b^3a+6b^2a^2+4ba^3+a^4)\\&\qquad+2b^4-a^4\Big]\frac{\partial^4\phi(x)}{\partial x^4}\\&=\frac{1}{12}\Big[-(12b^4+24b^3a+12b^2a^2)\Big]\frac{\partial^4\phi(x)}{\partial x^4}=-(b^4+2b^3a+b^2a^2)\frac{\partial^4\phi(x)}{\partial x^4}\\&=-b^2(b+a)^2\frac{\partial^4\phi(x)}{\partial x^4}.\end{aligned} \tag{2.60}$$

With this, we have landed in a much less intimidating approximation of the symmetric pair-hopping dynamics

$$\frac{\partial\phi(x)}{\partial t}=-\tilde{D}\frac{\partial^4\phi(x)}{\partial x^4}, \tag{2.61}$$

where $\tilde{D} = n_0 \alpha b^2 (b+a)^2$ is an analogue of the diffusion constant D. To recap what these parameters mean, n_0 is an assumed equilibrium state of the particle count, α is the hopping rate in discrete space, a is the pairing range and b is the (symmetric) hopping distance. The conclusion we can draw from this is that the symmetric pair-hopping appears to be non-diffusive as it does not follow the diffusion equation. Instead, it is an example of the broader category *anomalous diffusion*. A suitable name for the constant $\tilde{D}$ could be the constant of anomalous diffusion. In the implementation and results, $\tilde{D}$ will never be used since we do not fully understand the discrepancies between this rough approximative theory and the numerical simulation, and D is instead used to denote hopping rate as the diffusion constant and hopping rate are equivalent when the lattice spacing is unity. Due to the lack of exact theory, treating D as a hopping rate in the pair-hopping is the easiest option.

Finally, we can note that the condition that particles can only move in pairs is a restriction that is certain to lead to steady states. One class of steady state is a configuration where none of the particles are at the pairing distance of each other, meaning no particles can form pairs and therefore they cannot move at all. This class will be called "immobile". Another class is one where particles form clusters separated by large gaps, such that the particles can only move within their cluster but never reach any other cluster. We will call this class "clustered". Both classes have extreme cases with a maximum particle density at certain highly ordered configurations when annihilations are introduced to the system. Let us take a look at one example of each class for $k = 2$ reactants, the pairing range $a = 1$ and hopping distance $b = 1$.

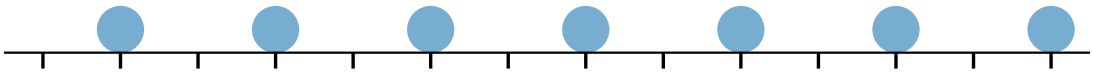


Figure 2.4: Configuration where no motion is possible for odd pairing ranges.

The first class has a maximum steady state density $\rho = 1/2$. This occurs at odd pairing ranges when the particles are found at every other site, such that the possible pairing distances fall in between populated sites. See figure 2.4 for an illustration of this configuration. Every other site is occupied with exactly one particle, as any higher amounts would annihilate, meaning the density is $1/2$.

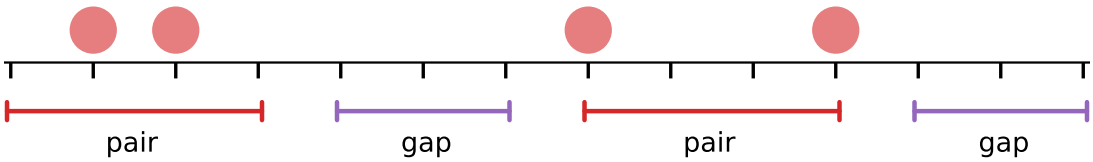


Figure 2.5: Configuration where motion is periodic and confined to clusters.

The second class has a maximum steady state density $\rho = 2/7$. Figure 2.5 shows how this configuration looks with $a = b = 1$. It consists of two-particle clusters that can move back and forth with each other. This cluster takes up 4 sites. In between the clusters there are gaps of length at least 3. This means that particles in different clusters are always at least 4 sites away from each other, ensuring that pairings can never occur between the clusters. Packing this configuration as tightly as possible, there are 2 occupied sites for every seven sites, giving the maximum density 2/7.

Similar configurations can be found for other parameters. Of course, these two examples in figures 2.4 and 2.5 forming from annihilations driven by random motion would be exceedingly rare. Instead, they exemplify invariant structures that make up steady states. A third trivial yet important structure is the vacuum state, i.e. a completely empty lattice. The steady states that annihilation system with pair-hopping dynamics settles into will feature a mix of these three invariant structures, though the probabilities of each of them occurring are unknown. The question of which steady state densities the system settles into is therefore not trivial.

3
Methods

The easiest way to simulate the diffusion-limited annihilation process is to compute the probabilities of diffusion and reaction and simply updating the state of the system with these probabilities in each iteration, with a small enough time step to accurately model continuous time. However, this approach is extremely computationally inefficient, especially for modeling long-term behavior with low particle densities. For this reason, S.C Park [2] devised another algorithm that captures the same statistical behavior but requiring way fewer operations. As described prior, the goal of this project is to implement and modify this algorithm to capture kinetics other than diffusion, in particular the pair-hopping kinetics described previously in chapter 2.3. This chapter will first summarize the algorithm before describing an implementation in Python. Then modifications will be made to simulate the pair-hopping kinetic model instead of diffusion.

3.1 Overview of the algorithm

The basic idea in Park's method is to create a list of all possible tuples of particles that can move or react. Which actions are performed during a short time frame can then be simulated by choosing an element of the list randomly, and incrementing time by this time frame divided by the total number of actions that were possible. In this way, the problem is reduced to integer combinatorics — the calculations of which are more accurate and faster to perform than the large number of floating point operations and random number generations that would've been needed in a direct simulation. The irregular time increments also make sure that time flows quicker in the simulation when fewer actions are possible, i.e. when actions are more rare. Furthermore, it is rather general, as the particular operations that are performed on a tuple can be simply swapped out to simulate any type of single-species reactions, like spontaneous generation, multi-particle annihilation, branching or coalescence, for instance. In principle, all reaction on the form

$$kA \xrightarrow{\lambda_{km}} (k+m)A \tag{3.1}$$

can be modeled, where $k \in \mathbb{N} \cup \{0\}$ is the number of reactants and $m \geq -k$ is the number of particles added or subtracted during the reaction. Multiple kinds of reactions can occur in the same system, though their rates λ_{km} are not necessarily the same. We will limit ourselves to the annihilation reaction $m = -k$ in the one-dimensional case for this implementation, though the extension to higher dimensions

is straightforward and the following summary will make no assumptions of reaction type or dimensionality.

Let $\rho_{\mathbf{x}}$ be the number of particles occupying lattice point $\mathbf{x} \in \mathbb{N}^d$, with ρ denoting the array consisting of all values of $\rho_{\mathbf{x}}$ in the space. To deal with the n-tuples of particles, we introduce the function $h(\rho_{\mathbf{x}}, n)$, defined as

$$h(\rho_{\mathbf{x}}, n) = \binom{\rho_{\mathbf{x}}}{n} \epsilon_n, \tag{3.2}$$

where

$$\epsilon_n = \begin{cases} 1, & D\delta_{n,1} + \sum_m \lambda_{nm} \neq 0 \\ 0 & \text{otherwise.} \end{cases}$$

That is, h counts the number of available n-tuples at site $\mathbf{x}$, with ϵ_n filtering out all the values of n that do not have any associated reaction. The number $N_{\mathbf{x}}$ of available tuples relevant for reactions at site $\mathbf{x}$,

$$N_{\mathbf{x}} = \sum_n h(\rho_{\mathbf{x}}, n),$$

are called "available states" by Park, and the same terminology will be used here. Furthermore, the total number of available states in the system is denoted by M, that is

$$M = \sum_{\mathbf{x}} N_x.$$

To keep track of all the available states, we define two arrays that Park call `list[]` and `act[][]`. Each element s of `list[`s`]` are on the form $(\mathbf{x}, \ell)$ where $\ell \in [1, N_{\mathbf{x}}]$, i.e. it enumerates each available state at site $\mathbf{x}$. Thus the length of `list[]` is M. The second array `act[][]` is the inverse, i.e. `act[x][`ℓ`]` $= s$.

In each iteration of the simulation, an element of `list[]` is chosen by rolling a random positive integer up to M. Which reaction to perform is then chosen by determining n as the smallest n that satisfies

$$\ell \leq \sum_{k=0}^{n} h(\rho_{\mathbf{x}}, k) \tag{3.3}$$

for the chosen $\mathbf{x}$ and ℓ. All possible reactions with this n according to equation (3.1) are then performed with a probability proportional to λ_{nm}. In the specific case when $n = 1$, in addition to any reactions that may or may not take place, a diffusion step is performed with probability $2dD\Delta t$. That is, the acceptance probabilities for each action is

$$P_{\text{acc}} = \begin{cases} n!\lambda_{nm}\Delta t & \text{if } n\text{-particle reaction} \\ 2dD\Delta t & \text{if diffusion step} \end{cases}. \tag{3.4}$$

Because these acceptance probabilities actually need to be interpreted as probabilities, we have the following constraint on the parameters:

$$(2dD\delta_{n,1} + n!\lambda_{nm})\Delta t \leq 1, \tag{3.5}$$

meaning the time step Δt needs to be chosen according to the parameters being used.

Now that the system has been updated, `list[]` needs to be updated to correspond to the new configuration. One could in principle compute `list[]` again from scratch, however this is extremely wasteful for large systems where the vast majority of states are still available after an update. This is where `act[][]` comes in. As previously mentioned, its elements are each value of s, i.e. the index with which you can find the element `list[`s`]`$= (\mathbf{x}, \ell)$. Thus `act[][]` can be used to restructure `list[]` and remove all states that are no longer available, and subsequently add all new available states for the new configuration. Note that since the state is chosen uniformly, the ordering of the arrays do not matter, which means that the updating stage is simplified to just making sure all available states are contained once and only once somewhere in the arrays. Of course, `act[][]` also needs to be updated, which can be done with the help of `list[]`. In other words, the two arrays are used to update each other back and forth, circumventing the need to compute the available states from scratch and saving us many, many redundant computations. The details of this updating scheme will be provided in the implementation section below. Once ρ, `list[]` and `act[][]` have been updated, time is incremented by $\Delta t/M$ and the next iteration starts.

3.2 Implementing reaction-diffusion dynamics

A prototype of the program was written in Python. This choice of language trades computational speed for ease of writing. Park's algorithm relies heavily on manipulating arrays, which is a task Python is designed to do intuitively at the cost of speed. This prototype therefore allows the focus to fall on implementing and extending the algorithm. For faster performance, an implementation in a lower level language like C can be written based on the Python prototype once it has shown itself to behave correctly.

The lattice is implemented as an array of integers, with each element representing the number of particles currently occupying the site. That is, the most basic structure in the program is the array ρ. For the purpose of diffusion-limited annihilation, ρ is initialized with a Poisson distribution, that is

$$\rho_{\mathbf{x}}(t = 0) \sim \mathrm{Poi}(\rho_0) \tag{3.6}$$

for some mean initial particle count per site ρ_0 in each site $\mathbf{x}$. This means that the total initial particle population is

$$\sum_{\mathbf{x}} \rho_{\mathbf{x}}(t = 0) \sim \mathrm{Poi}(\rho_0 L)$$

where L is the number of lattice points. To move a particle from site $\mathbf{x}$, subtract 1 from $\rho_{\mathbf{x}}$ and add 1 to $\rho_{\mathbf{x}+\Delta x}$, where $\Delta x \in \{1, -1\}$, i.e. the lattice spacing is fixed at 1. Reactions are just as simple in this model, as it is assumed reactions can only occur between particles occupying the same site. And so, a reaction on the form in

equation (3.1) occurring at site $\mathbf{x}$ is performed by adding m to $\rho_{\mathbf{x}}$. In the case of kA-annihilation, $m = -k$ and so it entails subtracting k from $\rho_{\mathbf{x}}$.

The array `list[]` is a 2D array of integers of size $M \times 2$. Worth noting is that the sites $\mathbf{x}$ are indexed in `list[]` starting from 1, whereas arrays in Python start from 0. This means that any time a site is chosen from `list[]`, the index has to be decremented by one before use. The benefit of this is for debugging purposes — since ℓ is never zero, any time a zero shows up in `list[]` something must have gone wrong.

To implement `act[][]`, one could use an array. This array would need to be of shape $L \times M$ to represent all possible available states. However, in most cases it will be sparse and only a very small fraction of the elements will hold any meaning. For large lattice sizes and particle counts this requires a massive amount of memory to store. Even if one limits the array size to $L \times \max_{\mathbf{x}}(N_{\mathbf{x}})$, i.e. the smallest it can be at any iteration, the need to keep track of $\max_{\mathbf{x}}(N_{\mathbf{x}})$ to check if the array needs to be padded will also reduce performance, and will still require large amounts of memory in certain cases. To sidestep these issues, it is advantageous to implement `act[][]` not as an array but instead as a hash table. Python comes with a built-in hash table structure known as a dictionary (keyword `dict`). Utilizing this, $\mathbf{x}$ and ℓ will be a pair of *keys* to a nested hash table rather than indices of a matrix, and thus `act[][]` does not have to contain any redundant elements.

In each iteration, one row of `list[]`, containing one $\mathbf{x}$ and one ℓ, is randomly chosen uniformly. The corresponding n is chosen from the chosen ℓ according to equation (3.3), by having a variable `states_sum` initialized to equal $\rho_{\mathbf{x}}$. If `states_sum` is then greater than or equal to ℓ, $n = 1$ is returned. Otherwise, `states_sum` is incremented by the right-hand side of the inequality (3.3), one value of k at a time, until the inequality holds. At this point $n = k$ is returned.

A particularly tricky part of the algorithm is the updating of `list[]` and `act[][]`. We can note that ℓ only enumerates available states and bears no information about any specific states. The number of available states only increases with increasing particle counts and vice versa. Thus, an increase in particle count at a site can only add new states and a decrease can only remove states. We can therefore get all the information we need about the list updating by computing the number of available states $N_{\mathbf{x}}$ at site $\mathbf{x}$ before and after an action is performed on that site. The following Python function performs the updating using the new and old values of $N_{\mathbf{x}}$:

```
def update_states_at_site(states_list: np.ndarray, act: dict,
                          site: int, num_states_before: int,
                          num_states_after: int, M: int):

    # First loop: remove states one at a time
    if num_states_before > num_states_after:
        for k in range( num_states_after, num_states_before):
            s = act[site][k][0]
            x = states_list[M-1][0]
            ell = states_list[M-1][1]
```

```
            # Copy the currently final element of list[] to
            # take the place of the newly inaccessible state
            states_list[s-1][0] = states_list[M-1][0]
            states_list[s-1][1] = states_list[M-1][1]

            # Put the new index of that element into act[][]
            # and delete the inaccessible state's element
            act[x][ell] = [s]
            del act[site][k]

            # Decrement the number of available states
            M -= 1
        # Cut out the duplicate elements at the end of list[]
        states_list = states_list[:M, :]

    # Second loop: add new states one at a time
    if num_states_before < num_states_after:
        # Extend list[] to the new number of available states
        num_new_states = num_states_after - num_states_before
        states_list = np.concatenate((states_list,
                                      np.zeros([num_new_states, 2],
                                      dtype=int)), axis=0)

        for k in range( num_states_before, num_states_after):
            # Increment the number of available states
            M += 1

            # Store the new available state at the end of list[]
            states_list[M-1, :] = [site, k]

            # Add the new state to act[][]
            if site not in act:
                act[site] = {}
            if k not in act[site]:
                act[site][k] = [M]
    return (states_list, act, M)
```

This function is an adaptation of the pseudocode in Park's paper [2, fig. 2]. Note that `list` is a reserved keyword in Python, which is why `states_list` is used instead. The second loop is rather straightforward, as it simply adds new values of ℓ to `list[]` and the corresponding elements to `act[][]`. The first loop, which runs when $N_{\mathbf{x}}$ is reduced is more interesting and less intuitive. In short, it copies the final element in `list[]` to take the place of the first no longer available state, and keeps track of the final element's site and state in the variables `x` and `ell` to save the new index of this element in `act[x][ell]`. The no longer available state is then deleted from `act[][]`. It then repeats this process for the second last element and second no longer available state, the third last element and third no longer available state, and so on, until every state that is no longer available has been overwritten. After the loop, `list[]` is sliced to remove the duplicate elements. A diffusion step changes the particle count at two sites, and a reaction at one site. The function `update_states_at_site()` is therefore called twice for a diffusion step, and once for a reaction.

3.3 Implementing correlated particle motion

To implement a simple model of the fracton dynamics described in section 2.3, the same structure of `list[]` and `act[][]` can be reused, but with a new interpretation of available states. Unfortunately, due to how the new particle kinetics are handled, one-particle reactions, i.e. single-particle branching or spontaneous vanishing, can no longer be modeled with this approach without some explicit workarounds, as we shall see. This project will only concern the one-dimensional case, though a generalization to higher dimensions is possible.

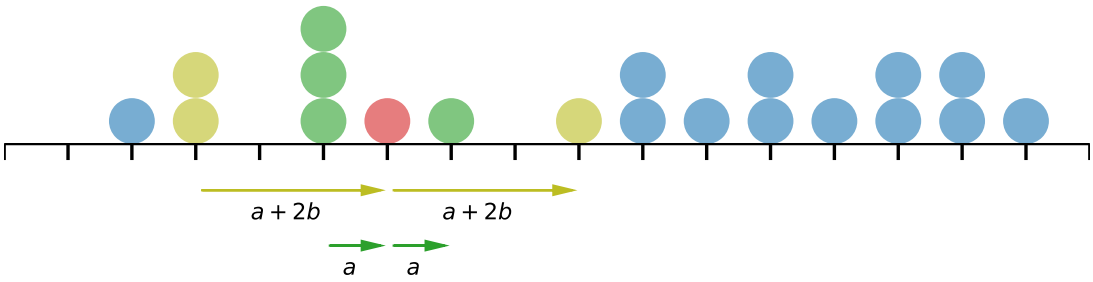


Figure 3.1: Example of the pairings a selected particle, marked with red, can participate in, using a pairing range $a = 1$ and hopping distance $b = 1$. Pairings for outward hopping are marked in green and for reverse hopping are marked in olive. The directions of the arrows signify how the particles always "look to the right" for partners to pair up with, and how the pairings are counted as available states on the site at the base of the arrow.

Previously, a diffusion step was performed if $\ell \leq \rho_{\mathbf{x}}$, where $\rho_{\mathbf{x}}$ refers to the number of particles at site $\mathbf{x}$. Now, we are not interested in each particle individually, but rather which *pairs* they can form with other particles in a site some distance $a > 0$ away. We will assume that for a given site $\mathbf{x}$, the algorithm only "looks aheads", i.e. to the right or in the positive direction, for possible pairings. See figure 3.1 for an illustration. The new number of available states at site $\mathbf{x}$, which we will denote $\tilde{N}_{\mathbf{x}}$, is therefore

$$\tilde{N}_{\mathbf{x}} = (\rho_{\mathbf{x}}\rho_{\mathbf{x}+a} + \rho_{\mathbf{x}}\rho_{\mathbf{x}+a+2})\delta_{n,1} + \sum_n h(\rho_{\mathbf{x}}, n), \tag{3.7}$$

i.e. the states corresponding to $n = 1$ are now dependent not on the number of particles at site $\mathbf{x}$, but instead on the product of the number of particles at $\mathbf{x}$ and the number of particles at $\mathbf{x} + a$, as well as $\mathbf{x}$ and $\mathbf{x} + a + 2$ for the detailed balance condition. The states corresponding to reactions remain unchanged for all $k \neq 1$. Reactions with $k = 1$ can no longer be represented in this procedure since the $n = 1$ case does not correspond to choosing one particle out of a total of $\rho_{\mathbf{x}}$.

As the case $n = 1$ now corresponds to two different actions, we need a way to tell them apart. For this purpose, we introduce a character variable `action`. This variable can take on the three values `'n'` for a near-pairing, `'f'` for a far-pairing,

and `'r'` for a reaction. In this model, `states_sum` is first incremented by $\rho_{\mathbf{x}}\rho_{\mathbf{x}+a}$. If the inequality holds at this point, $n = 1$ and `action` $=$ `'n'` is returned. If not, it is incremented by $\rho_{\mathbf{x}}\rho_{\mathbf{x}+a+2}$, returning $n = 1$ and `action` $=$ `'f'` is returned if the inequality holds. If the inequality still does not hold, a reaction is chosen in the same way as in the diffusive model, and $n = k$ is returned along with `action` $=$ `'r'`.

Note that this is just one method of many possible to differentiate between the hopping events and the reactions. One could in principle use different values of n to tell them apart, which would prevent us from having to pass around one extra variable, as n holds no intrinsic meaning beyond keeping track of what the program should do next. It would be valid to instead use $n = 1$ to mean outward hopping, $n = 2$ for outward hopping and $n > 2$ to encode reactions involving $k = n - 2$ reactants, for instance. This would reintroduce the possibility of 1-particle reactions, as $n = 3$ would correspond to $k = 1$. The benefit of using the previously outlined method over this alternative is that the previous method is consistent with how Park's algorithm always allows for motion when $n = 1$, and n always refers to the number of reactants even in a simulation allowing several different reaction types. Ultimately, it boils down to an arbitrary choice based on individual taste and what is most intuitive for any given application.

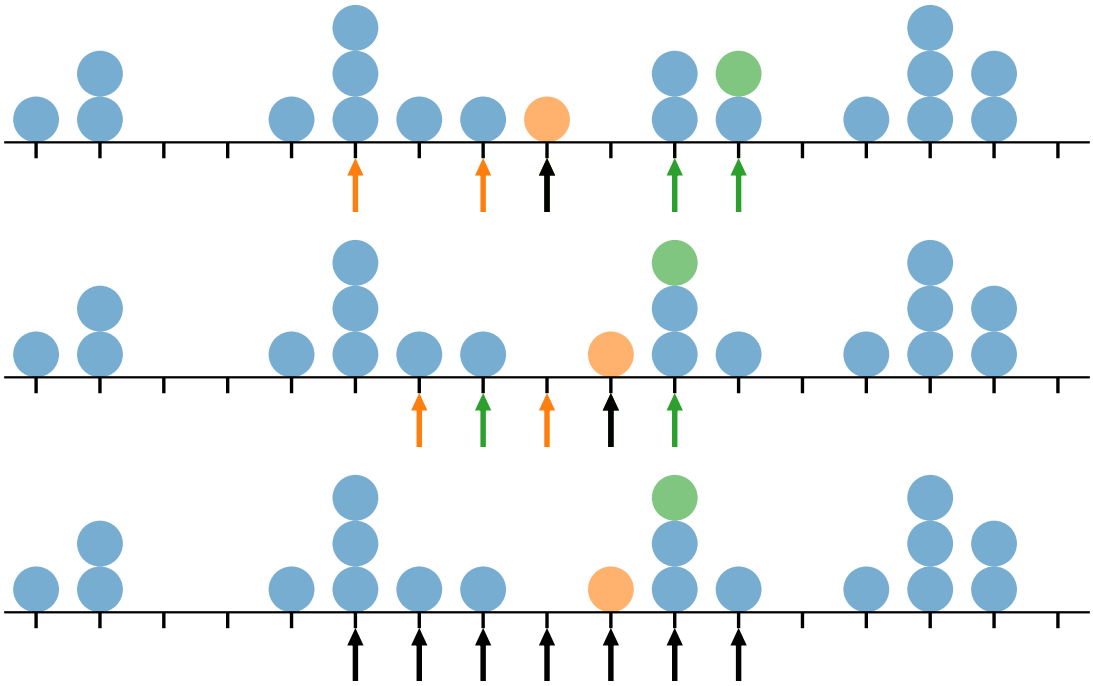


Figure 3.2: Illustration of the sites that are affected by a pair-hopping step. The orange particle pairs up with the green particle, shown on the first row, to perform an inward hop, shown in the second row. On the first and second rows, the sites which are dependent upon the orange particle's position are marked with orange arrows, and vice versa for the green particle, with overlaps marked with black. The third row tallies all the sites that are affected.

The pair-hopping dynamics have one significant difference from the diffusive dynamics: the number of sites that have their available states changed after an action is taken. Thankfully, with the way the function `update_states_at_site()` was written, it does not need to be changed at all. Instead, we will need to keep track of how many sites need to be updated, compute their $N_{\mathbf{x}}$ values before and after an action, and call `update_states_at_site()` on each of them just like before. For each site whose particle count changes, three sites have their available states changed: the site itself, and the sites a and $a + 2$ steps to its left. The number of sites affected by a reaction is 3. After a pair hopping action, four sites have their particle counts changed. However, there is much overlap between these four sites and their respective pair sites to their left. Figure 3.2 shows an example of this overlap, and we see that we get a total of seven distinct sites being affected. It is these seven sites that need to have their states compared before and after the pair-hop.

4

Results

Before the results of the non-diffusive kinetics are presented, let us verify that the implementation in this project yields behavior consistent with the diffusion theory as well as the second quantization theory in chapter 2.2. Next, the power law scaling of the mean squared displacement is examined for both kinetic models and contrasted with each other. Finally, the density scaling under the pair-hopping model and annihilation reaction is presented and steady state configurations are discussed. Due to slow performance in the simulation for high reactant counts, a satisfactory amount of data for the cases of $k \geq 3$ reactants was not achieved and only the $k = 2$ reactant case is presented.

4.1 Diffusive results: pure diffusion

To check that the algorithm correctly captures the statistics of diffusion, we can check that the distribution of particles follows equation (2.6). Initially placing all particles at the center of the lattice as shown in figure 4.1, we expect the distribution of particles to follow equation (2.6) at all time steps until the boundary is reached. For clarity's sake, the lattice is shifted such that the center is the origin $n = 0$.

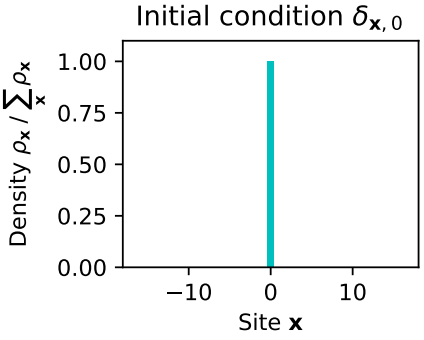


Figure 4.1: Initial condition for the diffusion verification runs, exemplified here by a system size $L = 32$. The lattice has been shifted such that the center is the origin.

Figure 4.2 contains three snapshots of the time-evolution of such a system for three different lattice sizes. For dimensionless time, the discrete hopping rate $\gamma = 0.5$ is used, which is equivalent to the diffusion constant D since the lattice spacing is $\Delta x = 1$. We can notice that the ensemble-average particle distribution follows equation (2.8) remarkably well regardless of lattice size.

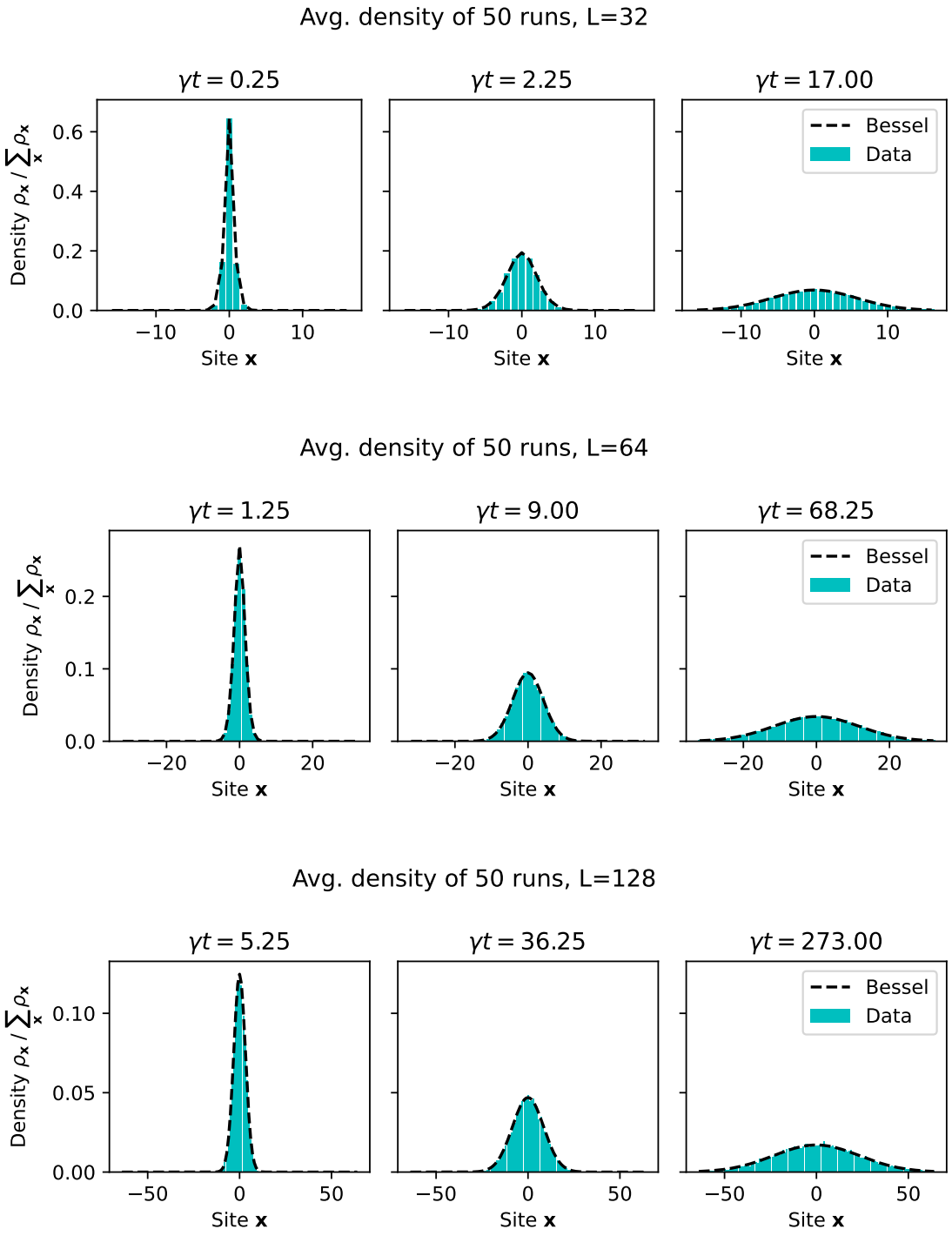


Figure 4.2: Ensemble-average time evolution of the distribution of particles. Lattice sizes are $L = 32, 64, 128$ respectively. Parameters in common are $\rho_0 = 3$, $D = 0.5$, $\lambda = 0.5$ and the ensemble consists of 50 independent runs. Note that the y axes have different scales in each row of plots, and the time scales differ as well. The dashed curves show the theoretically predicted distribution at each time according to equation (2.8).

Once the particles have reached the boundary, the periodicity causes the distribution to slowly even out. The final state of the distribution is uniform, as predicted by equation (2.14). Figure 4.3 shows a few snapshots of the transition between the Gaussian and uniform paradigms for the lattice size $L = 64$, starting from where figure 4.2 ends.

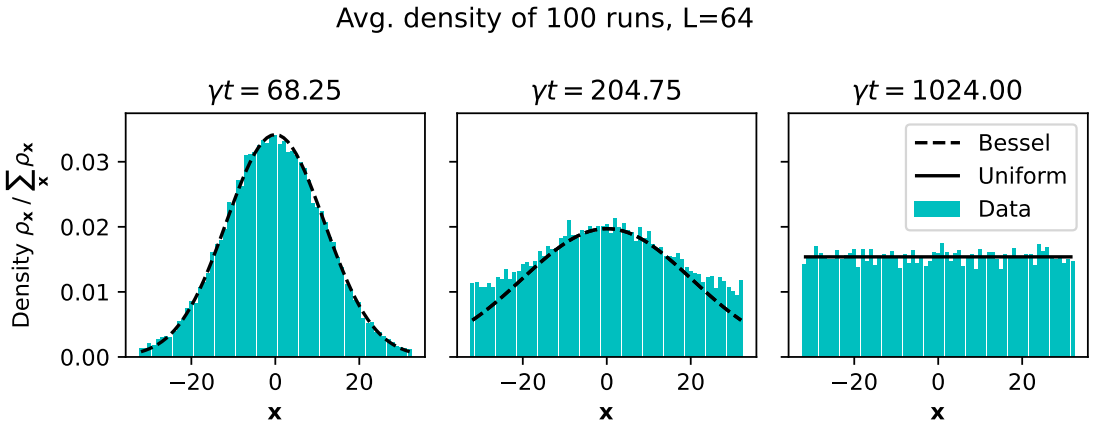


Figure 4.3: Continuation of the diffusion in the $L = 64$ case in figure 4.2, with 50 more runs in the ensemble. Note that in the middle case, the distribution has started to deviate considerably from the Bessel curve from equation (2.8), but is still far from uniform. In the third case, the diffusion process has gone on for long enough for the distribution to reach the uniform distribution $1/L$ as predicted from equation (2.14)

We see that the distribution does indeed become uniform. Worth noting is that a characteristic time scale was found to be L^2/D, which with the dimensionless time in this case would be $L^2/\Delta x^2 = 64^2 = 4096$. The final snapshot shown is at $\gamma t = Dt = 1024$, which is only one fourth of the previously noted characteristic time. It turns out that this characteristic time is a slight overestimate of the time scale at which the distribution becomes homogeneous in this case. That is, the evening out happens faster than the Gaussian's standard deviation reaches the size of the entire lattice. Even so, it remains a useful estimate for when the distribution definitely has reached uniformity.

4.2 Diffusive results: 2-particle annihilation

The second quantization calculations for the 2-particle annihilation reaction as laid out in chapter 2.2 predict an asymptotic density scaling

$$\rho(t \to \infty) \sim \frac{1}{\sqrt{8\pi Dt}} \tag{4.1}$$

independent of initial density and reaction rate. To check that the simulation's behavior is consistent with the theoretical prediction, we can run the simulation

with annihilation enabled and check how the particle density scales over time for various parameter settings. Unless otherwise specified, the default parameters of these runs are

$$\begin{cases} \text{Lattice size } L & = 2^{16} = 65536 \\ \text{Lattice spacing } \Delta x & = 1 \\ \text{Initial density } \rho_0 & = 1 \\ \text{Reaction rate } \lambda & = 0.5 \\ \text{Diffusion constant } D & = 0.5 \\ \text{Time resolution } \Delta t & = 0.5 \end{cases} \tag{4.2}$$

and one parameter out of ρ_0, λ and D will be varied at a time. An ensemble average of 100 independent sample runs is used for each parameter setting.

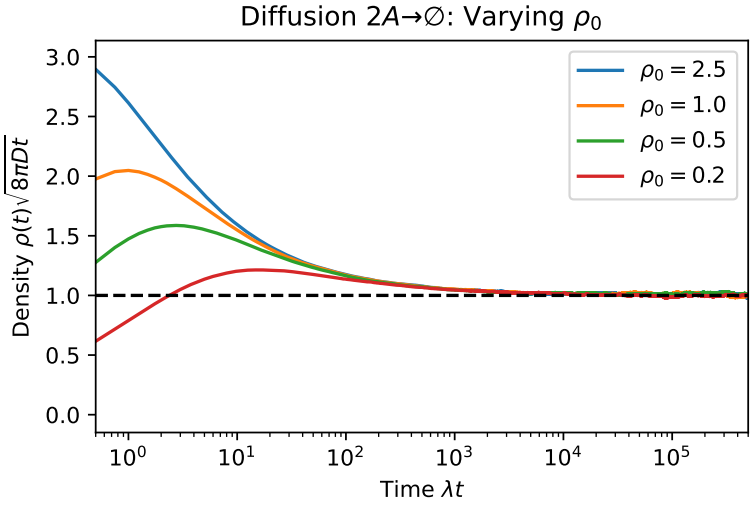


Figure 4.4: Time evolution of the total particle density ρ with various initial densities ρ_0. The density is normalized with the theoretical asymptotic scaling $(8\pi Dt)^{-1/2}$. Note that regardless of initial density, all plots approach a constant 1, indicating that the density indeed scales like $(8\pi Dt)^{-1/2}$ in silico in the long term, regardless of initial density.

To begin, we vary the initial density ρ_0. As a reminder, the initial configuration is given by equation (3.6), i.e. Poisson distributed with the mean initial occupation count in each site being ρ_0, meaning the initial density is ρ_0 on average but not exactly. Figure 4.4 shows a log-linear plot of how the density evolves over time with four sample settings of ρ_0. To make the axes dimensionless, the time is scaled by λ and the density by the inverse of the predicted power law in equation (4.1), i.e. $\sqrt{8\pi Dt}$. With this setup, the density curve will approach a constant 1 if it behaves the predicted asymptotic power law. We see that this is indeed the case; the normalized density approaches 1 regardless of initial density.

Next, consider the reaction rate λ. The second quantization theory predicts that the density scaling is independent of λ in the diffusion-limited regime. To verify

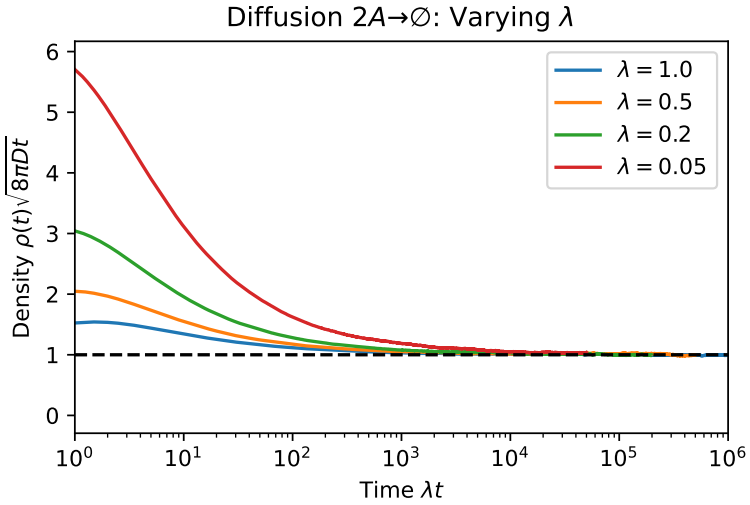


Figure 4.5: Time evolution of the total particle density ρ with various reaction rates λ. The density is normalized with the theoretical asymptotic scaling $(8\pi Dt)^{-1/2}$. Here all plots approach a constant 1 just like in figure 4.4, indicating that the long-term density scaling is independent of reaction rate.

this, we employ the same time and density scales and the resulting time evolution is shown in figure 4.5. Once again the curves all approach unity regardless of λ, showing that the diffusion-limited annihilation process is independent of reaction rate. Worth noting is that in figure 4.5, the time scale actually varies between runs as it is based off of the reaction rate.

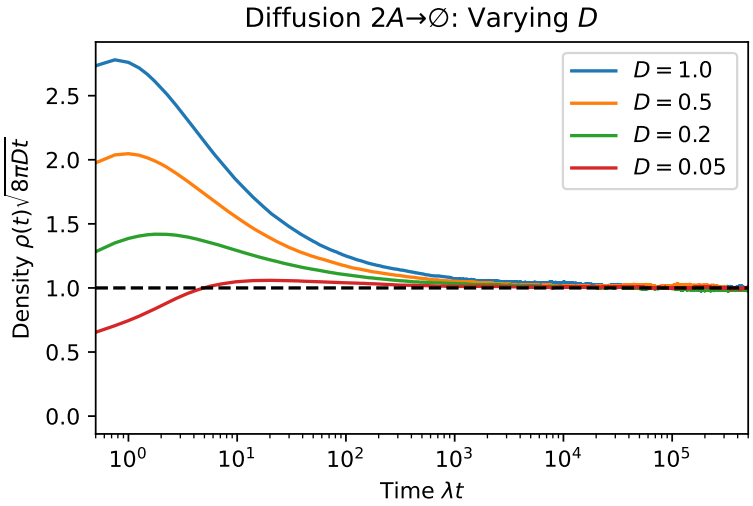


Figure 4.6: Time evolution of the total particle density ρ with various diffusion constants D. The density is normalized with the theoretical asymptotic scaling $(8\pi Dt)^{-1/2}$. Just like in figures 4.4 and 4.5, the plots approach a constant 1, indicating that the density indeed scales like $(8\pi Dt)^{-1/2}$ in the long term.

Finally, consider the diffusion constant D. The density drop is not independent of D unlike ρ_0 and λ. The theory predicts a dependence $\sim \sqrt{D}$ though, so by including it in the reference density scale, we expect the curves to approach unity once more. Figure 4.6 shows that this is the case, confirming the theoretically predicted asymptotic power law $\rho(t) \sim (8\pi Dt)^{-1}$.

In conclusion, the simulated diffusion-limited annihilation with $k = 2$ reactants follows the theory remarkably well, even with a somewhat small ensemble size of 100 independent runs. This is consistent with the results presented in the original paper proposing the algorithm [2]. Having verified that the base algorithm performs as expected, the pair-hopping modification can now be examined.

4.3 Pair-hopping results: kinetics

Since the pair-hopping kinetics require particles to occupy several distinct sites, the Dirac delta initial conditions cannot be used to analyze the time-evolution anymore. A way to get around this is to instead use a narrow Gaussian distribution. By tracking how the variance of this distribution changes over time, it is possible to determine the power law by which the mean squared displacement changes over time. As described in chapter 2.1.2, a linear power law means that the kinetics remain diffusive while a power less than 1 or greater than 1 indicates sub- or superdiffusive kinetics, respectively.

To analyze the dynamics, we use the mean density per site of 100 independent runs with system size $L = 64$, pairing range a and "diffusion constant" (which is essentially just a hopping rate) $D = 0.5$, and find how the variance evolves over time. The most straightforward way to compute the variance is by definition. As a refresher, the definition of the variance is

$$\mathrm{Var}(x) = \langle x^2 \rangle - \langle x \rangle^2, \tag{4.3}$$

where $\langle \cdot \rangle$ averages over each particle. In our case, we have the population count $n(x)$ in each site x. Thus, we can compute the mean particle position by summing over each site x and weighing each term by $n(x)$:

$$\langle x \rangle = \frac{1}{L} \sum_{x=-L/2}^{L/2} n(x)x \tag{4.4}$$

with which we can then compute the variance according to

$$\mathrm{Var}(x) = \frac{1}{L} \sum_{x} n(x)(x^2 - \langle x \rangle^2). \tag{4.5}$$

This variance is also the mean squared displacement. Figure 4.7 shows the time evolution of the distribution at a few select snapshots of the ensemble average distribution over 100 runs, using lattice size $L = 64$ and hopping rate $D = 0.5$. The

variance computed using the scheme above is visualized as Gaussian curves with the corresponding variance in the particular time point and plotted above the distributions.

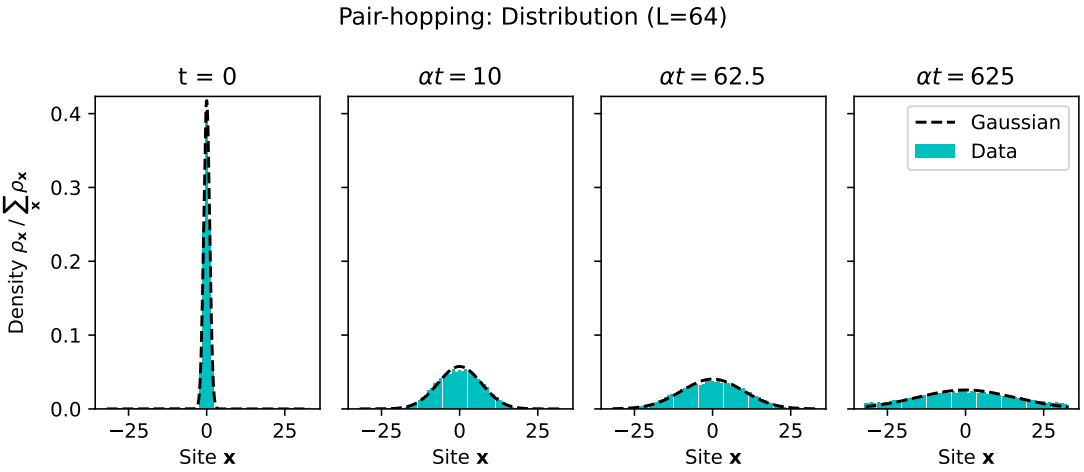


Figure 4.7: Time evolution of the particle distribution under the pair-hopping dynamics. The initial configuration is a Gaussian with variance 1.

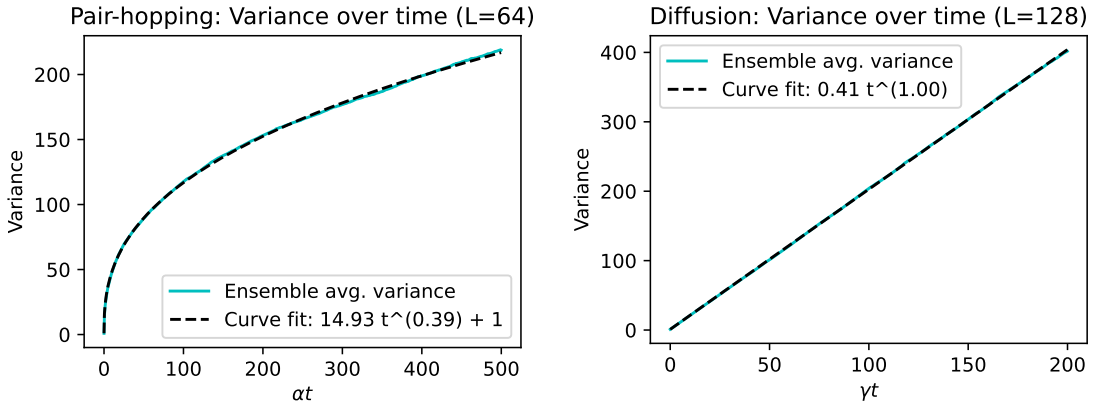
To find the power law by which the mean squared displacement grows, we can perform a nonlinear least-squares fit to the variance data of the form

$$f_{\text{fit}}(t) = \mu t^{\nu} + c \tag{4.6}$$

using a tool like `scipy.optimize.curve_fit()`, for instance. However, it is important to note that this variance is only valid as long as the distribution has not yet hit the boundaries. The algorithm in equations (4.4) and (4.5) cannot take into account total distance traveled across the boundary limit. It is therefore important to check when the boundaries have been reached and not sample any variance data beyond that time.

Figure 4.8 shows how the variance of the data in figure 4.7 grows as time passes, up to around the time when the particles start to reach the boundary. As a comparison, it also shows the variance in a diffusive simulation using size $L = 128$ and diffusion constant $D = 0.2$. Additionally, it shows a curve fit on the form in equation (4.6) for both dynamics. Since the initial variance was set to 1, we already know that $c = 1$ and so only the parameters μ and ν need to be fitted. We see that the diffusion has a linear growth with impressive accuracy, as predicted by equation (2.38). We also find by the curve fit that the linear coefficient is $\mu = 0.41$, i.e. roughly double the hopping rate, which matches the prediction from equation (2.38).

On the other hand, the pair-hopping has the power law $\sim t^{0.39}$. This means that the dynamics are a form of anomalous diffusion, which is consistent with the theoretical result in equation (2.61). More specifically, the dynamics are subdiffusive since the exponent is less than 1. It is however unclear what the prefactor $\mu = 14.93$ represents as there is currently no known exact theory for this kinetic model.



(a) Pair-hopping. **(b)** Diffusion.

Figure 4.8: Time evolution of the distribution variance for the pair-hopping dynamics (left) and for duffisive dynamics (right), both using an ensemble of 100 runs. A curve fit on the form in equation (4.6) has been performed with $c = 1$, which resulted in an exponent of about 0.39 and 1 respectively. For the pair-hopping, the hopping rate $\alpha = 0.5$ was used. The diffusion used hopping rate $\gamma = 0.2$.

4.4 Pair-hopping results: 2-particle annihilation

It is now time to check the behavior under the annihilation reaction with the pair-hopping dynamics. Unless otherwise specified, the default parameters of these runs are

$$\begin{cases} \text{Lattice size } L & = 2^{16} = 65536 \\ \text{Lattice spacing } \Delta x & = 1 \\ \text{Initial density } \rho_0 & = 1 \\ \text{Reaction rate } \lambda & = 0.5 \\ \text{“Diffusion constant” } D & = 0.5 \\ \text{Time resolution } \Delta t & = 0.5 \\ \text{Pairing range } a & = 1 \end{cases} \tag{4.7}$$

and just like in the diffusive case in chapter 4.2, one parameter at a time is varied. All plots use the ensemble average of 100 independent runs.

To start with, let us check how the parameters affect the transient scaling profile. First, we vary the initial density ρ_0. Figure 4.9 shows four sample settings of ρ_0. The time axis is scaled by the transition rate λ, but the density axis is left as is. To look for any potential power laws, a log-log plot is shown to the right. We see a rapid drop in density down to a steady state at about $\rho = 2/11$, in which no more annihilations occur. This steady state appears independent of initial density.

Next, we vary the hopping rate α, which is denoted D in simulation to reflect its role as an acceptance rate like the diffusion constant in the original diffusive algorithm. Figure 4.10 shows four sample settings. Once again the time is scaled by

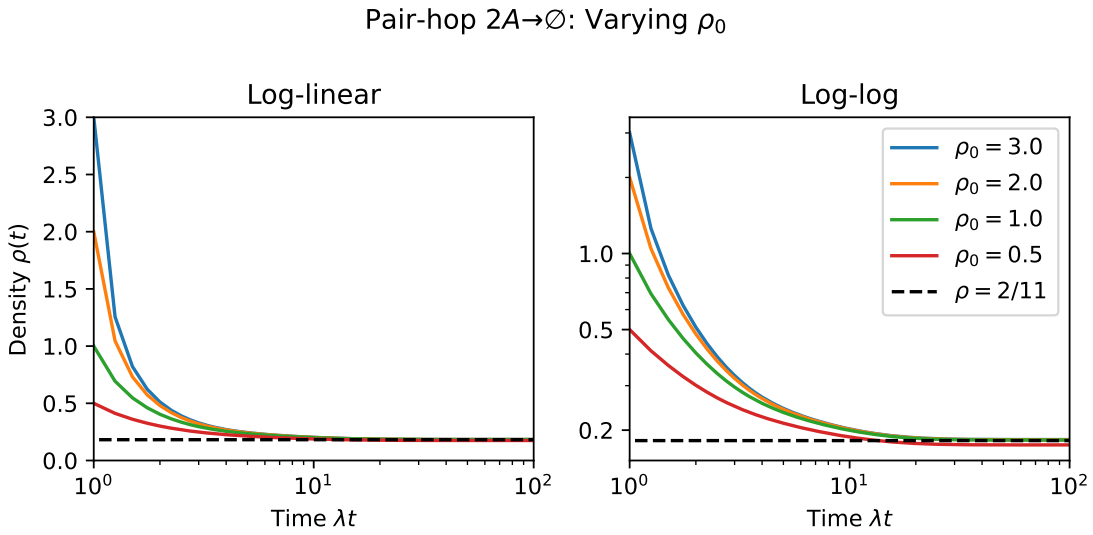


Figure 4.9: Time evolution of the total particle density ρ with various initial densities ρ_0 for pair-hopping dynamics.

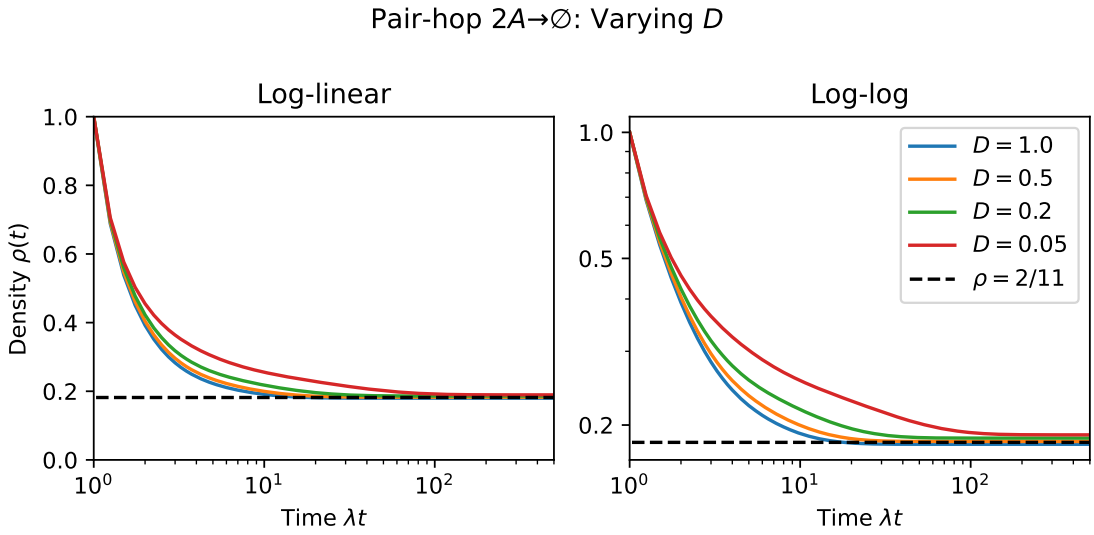


Figure 4.10: Time evolution of the total particle density ρ with various hopping rates α (here denoted D for its analogous role in the diffusion model) for pair-hopping dynamics.

λ and the density is not rescaled. We see that the initial density drop is the same, but the decay slows down more over time the lower the D. The mean steady state is once again roughly $\rho = 2/11$ independent of D. The log-log plot shows no clear sign of any power law.

Finally, we vary the reaction rate λ. Figure 4.11 shows four sample settings. Both a dimensionless and absolute time axis is shown to highlight the different curve shapes of each setting of λ. Here the initial decay is slower for lower reaction rates, which is

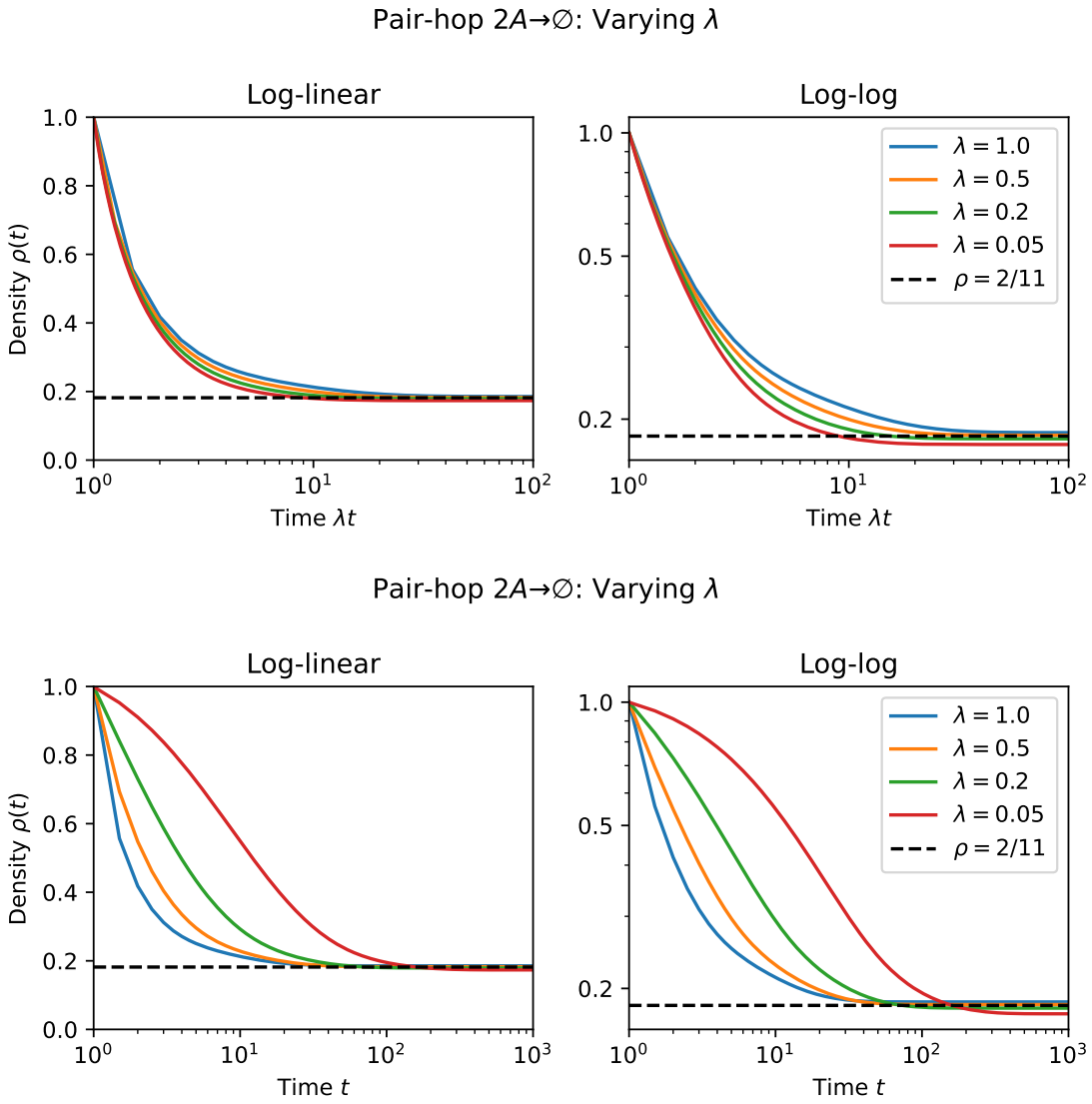


Figure 4.11: Time evolution of the total particle density ρ with various reaction rates λ for pair-hopping dynamics. This time both a dimensionless time axis (above) and absolute time axis (below) are shown, because a consistent time scale better visualizes the dramatically different shapes of the curves. Below $\lambda = 0.2$, the density drop begins taking on a sigmoid shape.

expected behavior in the reaction-limited regime. The same mean steady state near $\rho = 2/11$ is reached again. As most clearly seen in the top right plot, the lower the λ, the lower the actual steady state density, although no case deviates meaningfully from 2/11. No apparent power law can be found in this log-log plot either, though it seems that the density curve takes on a sigmoid shape when $\lambda < 0.2$. This is interesting as it suggests that reactions occur more often at slightly lower densities than at high densities when the reaction rate is low.

Having established that a steady state exists for all the tested cases, another question is how consistently this particular density value is reached for a given setting.

Figure 4.12 shows the mean density scaling for an ensemble of 100 independent runs, as well as the ensemble standard deviation as an error estimate. To make the error estimate visible, it has been exaggerated by a factor of 25. Only one sample ensemble is shown, though all previously shown runs with all the different parameter settings exhibited the same kind of behavior. We can see that the standard deviation remains approximately constant as time passes, and that it is very, very small. This indicates that the behavior is very consistent with small deviations from the mean in each individual run.

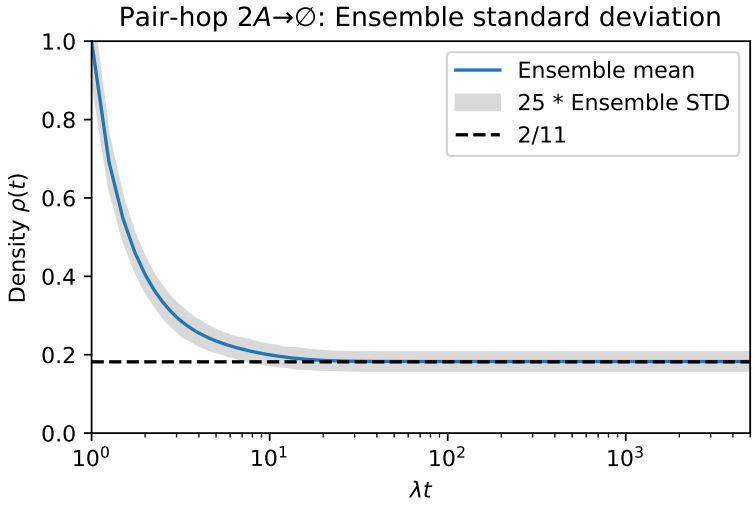


Figure 4.12: Time evolution of the total particle density ρ for some sample parameter setting, averaged over an ensemble of 100 runs, as well as an error estimate based on the ensemble standard deviation. Even with the deviation being exaggerated 25-fold, it is clear that the deviation is very small.

Finally, let us take a look at what the steady state might look like. Figure 4.13 shows a sample lattice configuration in the steady state for three consecutive time steps. The red particles can form pairs and move, while blue particles are frozen in place. We see the typical structures discussed in chapter 2.3: the red particles can never reach any blue particles, nor any of the other red particles except one, meaning there is no further mixing of the system. All that can happen is that red particles hop inward and outward in place forever.

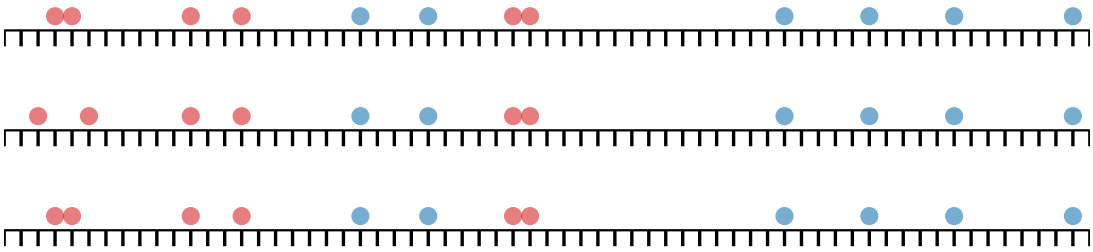


Figure 4.13: Example of a pair-hopping annihilation steady state. Marked in red are particles that can move, whereas all other particles are blue. Note that due to the gaps between the clusters of particles, the red and blue particles can never reach each other. The system is no longer getting mixed by kinetics and thus no more reactions can occur.

5
Conclusion

To summarize, this project consisted of implementing a Monte Carlo algorithm for simulating single-species reaction-diffusion systems. The implementation has shown itself to behave in accordance with the most precise known theory in all tested aspects. It was then extended to feature a new model of motion. The stated goal of the project was to explore how this new kinetic model affects the critical dimension of the reaction-diffusion system, and while this was not achieved, the results instead indicates that there are interesting structures in the steady states of this model that warrant further investigation. Below follows a discussion of all results as well as possible future work to be done.

5.1 Discussion

The results presented in chapter 4.1 are consistent with the diffusion theory described in chapter 2.1. They confirm that the implementation performs as desired with the simplest kinetic model, with the correct time scaling of the mean squared displacement and the transition from an (accurate yet approximate) Gaussian distribution to uniformity. It is in this light that the results in chapter 4.2 should be understood, as they truly do show the asymptotic long-term scaling of the density as predicted by the most accurate known theory of diffusion-limited annihilation, i.e. the second-quantization representation from chapter 2.2. In particular, the complete independence from the reaction rate in the long term in figure 4.5 exemplifies the limiting effect of diffusion. Note however that while all four samples coalesce at seemingly the same time scale, this is because the time scale is set by the reaction rate, and smaller reaction rates do in fact lead to a longer transient before reaching the diffusion-limited regime. The accurate performance of the base implementation gives some credence to the new results with the pair-hopping kinetic model since both versions are built from the same base. If one behaves reasonably, the other probably does as well.

Regarding the pair-hopping kinetics, the results in chapter 4.3 are significant in that they confirm that the model is in fact not diffusive. The simulated power law exponent of 0.39 in the time-evolution of the mean squared displacement is small enough that we can be certain that its nonlinearity is no coincidence. However, because we do not yet have any accurate theory for this time evolution, we do not know how sensitive the results are to the simulation's system size, how to interpret the anomalous diffusion constant $\tilde{D}$ in relation to the other parameters, nor how

much the discrete lattice and continuous space cases agree with each other. The exact value calculated from the simulation should therefore be taken with a grain of salt. A quick dimensional analysis of equation (2.61) suggests that there are four spatial units for each time unit, suggesting that a characteristic length scale for the dynamics grows as the fourth-root of time, giving a mean squared displacement that grows as the square root of time, i.e. an exponent of 1/2. The simulation result of 0.39 is not far off enough to entirely reject this estimate, but is also nowhere near confirming it. This exponent is a problem worth looking into further. Implementing asymmetric hopping to check if it does indeed lead to diffusion is another extension to consider.

Moving on to annihilation driven by pair-hopping, no obvious power law tendencies could be identified from the results in 4.4. The closest match is what appears to be a slightly different decay profile in the lowest D tested in figure 4.10, in which the red curve looks almost linear briefly in the log-log plot. This brings up an important note: perhaps the default setting of $D = 0.5$ is too large to be entirely reasonable. To see why, consider that for diffusion we had a constraint on D to make the acceptance probability an actual probability, i.e. the first term in the inequality (3.5). Due to the lack on an exact theory on the pair-hopping kinetics, the equivalent acceptance probability for a pair-hopping step and subsequently the constraint on D are both unknown. This should have no bearing on these results though, as the hopping rates used here are purely hypothetical and serve as a proof-of-concept. However, a proper acceptance probability needs to be determined before using this algorithm for practical purposes.

On the other hand, a result we can be sure of is that all the tested parameter settings had the same late-time behavior, just like in diffusion-limited annihilation. Not only that, they all also lead to the same steady state density of around 2/11. Worth noting is that while this is a close approximation to the steady state density observed in these simulations, it is only an approximation and closer inspection could yield something more precise.

As discussed in chapter 2.3, there are three invariant structures under the pair-hopping kinetics: complete vacuum, configurations where the distance between the particles falls between the two possible pairing distances, and configurations of isolated pairs of particles that can only move with each other. The low deviation seen in figure 4.12 can be understood as follows: the lattice can be split up into sublattices, so a large system can be thought of as a set of smaller systems that exhibit one of the three invariant structures. The Monte Carlo simulation essentially samples a set of these invariant structures based on their probability of occurring. For this reason, each run of a large system essentially already computes an average steady state density by itself. The larger the system, the lower the deviation. What the results show then, is that the average of these three structures consistently leads to a density of about 2/11. This is specific for 2-particle annihilation and higher numbers of reactants would yield completely different steady states.

There is one other case that the current implementation does not yet support: the pairing range $a = 0$, meaning particles on the same site mutually reject each other or both hop into the same site. Under 2-particle annihilation, this case is interesting because its reverse motion directly leads to a pairing that can subsequently annihilate. The steady states of this system could have a very different structure from that of $a \neq 0$ and is worth considering.

5.2 Outlook

Moving forward, there are many new aspects to explore. The first and possibly most important step is to optimize the implementation so that it runs quick enough to allow more complex simulations with reasonable run time. This implementation was written in Python, a language that is readable and intuitive but slow as a tradeoff. For the best possible performance and the most opportunity for further optimization, a new implementation in a language like C or C++ or even Rust should be considered.

Once a more efficient implementation has been made, one could use it to test more reactants or higher dimensions, which could verify the mean-field theory and critical dimension of the diffusion-limited annihilations, and also any critical behavior and new steady states in the pair-hopping model. One could also construct a new kinetic model that emulates some proper physical system better than the pair-hopping kinetics, which is more akin to a proof-of-concept or a toy model for center-of-mass-conserving systems. Furthermore, by introducing more occupation arrays, one could simulate multiple particle species which could be useful for applications like simulating dipole-conserving motion akin to the briefly discussed fractons, or a broader set of reactions for chemical processes.

Finally, there is room for developing more sophisticated theory for the behavior of the pair-hopping dynamics. Rather than the same approach as the diffusive case that lead to a nonlinear master equation (making it not actually a master equation, strictly speaking) one could instead try to construct a theory based on a two dimensional probability of possible pairings. It is unclear whether such an approach is easily compatible with reactions, but to understand the nature of the kinetics it could be helpful. One could also study the steady states through the lens of the previously mentioned invariant structures in the considered case as well as other higher-dimensional cases, or with other reactions. This might help shed light on the enigmatic density of 2/11.

Bibliography

[1] M. Pretko, X. Chen, and Y. You, "Fracton phases of matter," *International Journal of Modern Physics A*, vol. 35, no. 06, p. 2 030 003, Feb. 2020. DOI: 10.1142/s0217751x20300033. [Online]. Available: https://doi.org/10.1142%2Fs0217751x20300033.

[2] S.-C. Park, "Monte Carlo simulations of bosonic reaction-diffusion systems," *Phys. Rev. E*, vol. 72, p. 036 111, 3 Sep. 2005. DOI: 10.1103/PhysRevE.72.036111. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevE.72.036111.

[3] P. L. Krapivsky, S. Redner, and E. Ben-Naim, *A Kinetic View of Statistical Physics*. Cambridge University Press, 2010, ISBN: 978-0-521-85103-9.

[4] J. Allam *et al.*, "Measurement of a reaction-diffusion crossover in exciton-exciton recombination inside carbon nanotubes using femtosecond optical absorption," *Phys. Rev. Lett.*, vol. 111, p. 197 401, 19 Nov. 2013. DOI: 10.1103/PhysRevLett.111.197401. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.111.197401.

[5] B. P. Lee, "Renormalization group calculation for the reaction ka to oe," *Journal of Physics A: Mathematical and General*, vol. 27, no. 8, p. 2633, Apr. 1994. DOI: 10.1088/0305-4470/27/8/004. [Online]. Available: https://dx.doi.org/10.1088/0305-4470/27/8/004.

[6] K. Kang and S. Redner, "Fluctuation-dominated kinetics in diffusion-controlled reactions," *Phys. Rev. A*, vol. 32, pp. 435–447, 1 Jul. 1985. DOI: 10.1103/PhysRevA.32.435. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevA.32.435.

[7] M. Doi, "Second quantization representation for classical many-particle system," *Journal of Physics A: Mathematical and General*, vol. 9, no. 9, p. 1465, Sep. 1976. DOI: 10.1088/0305-4470/9/9/008. [Online]. Available: https://dx.doi.org/10.1088/0305-4470/9/9/008.

[8] J. Cardy, *Reaction-diffusion processes*, https://www-thphys.physics.ox.ac.uk/people/JohnCardy/warwick.pdf, Lecture notes, accessed: 2023–05-22, Jul. 2016.

[9] J. Hofmann, "Corrections to reaction-diffusion dynamics above the upper critical dimension," *Physical Review E*, vol. 105, no. 2, Feb. 2022. DOI: 10.1103/physreve.105.024127. [Online]. Available: https://doi.org/10.1103%2Fphysreve.105.024127.

[10] A. Lushnikov, "Binary reaction 1+1→0 in one dimension," *Physics Letters A*, vol. 120, no. 3, pp. 135–137, 1987, ISSN: 0375-9601. DOI: https://doi.org/10.1016/0375-9601(87)90714-6. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0375960187907146.

DEPARTMENT OF SOME SUBJECT OR TECHNOLOGY
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY