Crowding: Using Statics to Understand the Dynamics of Densely Packed Hard Particles

Patrick Nima Raanes

New College

Oxford University

Thesis submitted for the degree of

M.Sc. in Mathematical Modelling and Scientific Computing

September 3, 2010

Dedication

To my parents.

Acknowledgements

I would like to thank my supervisors, Dr. Mason Porter, Dr. Nick Jones and Dr. Radek Erban. The topic of researc was their idea, and they have contributed greatly with guidance for the whole duration of the project. Dr. Radek Erban has assisted with expertise in stochastic systems and RSA, Dr. Nick Jones has contributed with clearsighted physical, and biochemical knowledge, and Dr. Mason Porter has contributed with mathematical acumen and experienced supervision. I must also thank Prof. Tomaso Aste of Kent University for his contribution of general knowledge in the field of packing, and with whom I have kept a running email conversation for the duration of the project.

I am also grateful to all of my coursemates, as they have provided a highly motivating study environment, and the academic discussions we have had have been instrumental to my progress over the whole year. Thanks must likewise be directed to Dr. Kathryn Gillow, course organiser of this M.Sc. programme, for her expert assistance in all matters related to this course, and indeed to the rest of the organisers of the course, which has been ample with good lecturers, all of which has prepared me for writing this thesis.

I must also extended my thanks to the "Johan Helmich Janson og Marcia Janson" endowment fund for their kind contribution in support of my studies at Oxford this year.

Abstract

We do a comparative study of the spacial configurations of particles that arise from the stochastic processes of Random Sequential Adsorption (RSA) and Brownian Dynamics (BD). We relate this to biochemical application areas and, in particular, crowded environments. We describe some of the necessary theory of BD, RSA and kernel density estimation, and we develop a variety of BD simulation algorithms to tackle issues relating to collision handling. We undertake studies in one and two dimensions, and show that classical RSA does not generate the same type of particle configurations as BD, although the difference between the two is largely dependent on factors such as crowdedness and polydisperisty. In one dimension, we achieve better approximations to BD particle configurations through simple modifications of the RSA procedure.

Contents

Chapter 1

Introduction

1.1 Motivation

The notion of *crowded media* is of high importance in cellular biochemistry. It differs from that of concentrated media in that no single macromolecular[1] species occurs at high concentration, but, taken together, the macromolecules occupy a substantial fraction (typically around 20% to 30%) of the total volume [5].

The macromolecular particles in such crowded microbiological media (typically cell cytoplasms) undergo certain forms of dynamics that determine how the particles are distributed within the medium. If a "snapshot" is taken at a given instance, the spacial configuration of the particles should be possible to quantify statistically, yielding interesting quantities such as the contact probability between particles. As described in Section 1.3, the statistical properties of the spacial particle configurations have wide-ranging effects on the inner workings of cells, making it important to be able to estimate these properties.

One way to model the dynamics inside a cell is through the stochastic process called *Brownian motion*, or *Brownian dynamics*[2] (BD). Simulations of BD are generally less computationally heavy than most forms molecular dynamics (simulations of approximations of Newton's classical physical laws of motion), but even BD typically requires several iterations, and further complications arise as a consequence of the high frequency of particle collisions in crowded environments.

An alternative way to establish the spacial configurations of the particles is offered by another stochastic process, called *Random Sequential Adsorption* (RSA). This is

[1] Molecules such as nucleic acids, proteins, carbohydrates and lipids, which are significantly larger than the solvent molecules.

[2] Brownian dynamics is often the term applied to describe the dynamics of an ensemble of particles undergoing Brownian motion.

a packing method which is often used to model the adsorption of macromolecules onto surfaces [14], but we want to investigate if it can be used to replicate the spacial particle configurations of BD.

Although both BD and RSA are stochastic methods, it is far from obvious that they should engender the same spacial particle configurations, and indeed our results show that they do not. That being said, our results also show that simple modifications of the classical RSA procedure allow for much better approximations of the BD configurations.

1.2 Summary of sections

The thesis is organised as follows. In Section 1.3 we introduce the necessary biochemical background to motivate our studies. The rationale for employing BD is explained in Section 1.3.1, and the theory behind crowdedness in Section 1.3.2. In Chapter 2 we consider the one-dimensional case of RSA and BD, introducing along the way necessary tools and other considerations regarding the interpretation of the simulation data. We begin by introducing analytical RSA in Section 2.1, before describing the basics of density estimation in Section 2.2. This enables us to start with numerical RSA, and describe the composition of the simulations experiments in Section 2.3. We then delve into analytical and numerical BD in Sections 2.5 and 2.6, and in Section 2.7 we compare the spacial particle configurations of RSA and BD in one dimension. We discuss one-dimensional polydispersity in Section 2.8 and the effect of varying the density in Section 2.9. In Section 2.10 we discuss some of the relevant modelling and implementation issues of this project, and in Section 2.11 we attempt to explain the results obtained in one dimension. In Chapter 3 we treat two-dimensional BD and RSA, re-investigating many of the same issues as we did in one dimension. In Chapter 4 we briefly discuss modifications to the classical RSA algorithm, and in Chapter 5 we summarise our methods and findings, and discuss potential future areas of research.

1.3 Biochemical background

In acknowledgement of the fact that this is a project with a direct application in (physical) biochemistry, we here present an introduction to the relevant biochemistry. For more details see [1, 4, 10].

Paraphrasing the review article [11], the effects of crowding on intra-cellular biochemistry help to explain (at least) two very general properties of living cells:

- Modest changes in the volume of a cell are associated with a broad spectrum of diverse intracellular processes that are too large to be accounted for by mass action laws (see Section 1.3.1).
- Every type of cell so far examined is equipped with mechanisms for the maintenance of cellular volume.

Crowdedness also leads to so-called “anomalous diffusion”, which is the sub-linear scaling of the mean-squared displacement of the molecules over time [13]. This has a bearing on many biochemical processes for which the diffusion rate is decisive, including transport systems and the signal speed through signalling pathways [13]. Furthermore, crowding affects reaction rates and chemical equilibria. This happens both indirectly through changes in diffusion rates, and through other phenomena, as described in Section 1.3.2.

1.3.1 Intra-cellular biochemistry

Although deterministic rate laws have proven very successful in biochemistry, they have certain limitations. First, these are empirical laws that give simple relations between the concentrations of the reactant species. They are typically based on the law of mass action, which considers the reactions to be macroscopic under convective or diffusive stirring, and do not take into account the fact that what is “really” going on is the “random” collision of particles with the correct geometry. More to the point, they do not take into account the obvious discrete character of the quantities involved, the inherent randomness of the system[3], and the resulting fluctuations for very small systems, such as cells [15]. The knowledge of such fluctuations dates back about 150 years, when Scottish botanist Robert Brown took notice of their existence while studying microscopic living phenomena. This famously lead Albert Einstein to study the phenomenon, and we have since grown accustomed to the ubiquitous presence of Brownian motion on the microscopic scale.

The cell lies between two extreme scales: that of individual atoms and molecules and that of whole multi-cellular organisms. Whereas those two “extremes” have been the subject of biochemical studies for a long time, the cell constitutes a relatively recent area of study [12] which has only really opened up in the last two decades or so [8]. On this level, Brownian motion plays an important role, and knowledge of the biochemical processes is often obtained through snapshots of BD simulations, which yield the spacial configuration of the macromolecules at specific points in time.

[3] Originating in the intractable movement of small particles.

1.3.2 Excluded volume theory

The main problem for intra-cellular biochemistry is the sometimes huge discrepancies between *in vitro* and *in vivo*[4] chemical properties and processes. The solution medium inside a cell is crowded with macromolecules that occupy between 7% and 40% of the volume. Although the influence of high fractional volume occupancy by the macromolecules has been recognized since the 1960's [8], it is often, and unwarrantably, regarded as unwanted, "unchemical" effects, and some researchers try to eliminate it by strategies such as extrapolation of results to zero macromolecular concentration [11]. In fact, the discrepancy between *in vitro* reactions, where the total concentration of macromolecules is around 1 mg/ml, and *in vivo* reactions, where the concentration is around 40 mg/ml, can be of several orders of magnitude [11]. The discrepancy is now understood to be caused by a neglect of the effect of *nonspecific interactions*[5] between these background macromolecules. As is described in [11] and [5], this is explained through *excluded volume theory*, which takes into account the spacial distribution of particles.

First, the diffusion rate decreases as the medium gets more densely packed. This sometimes implies lower reaction rates, as reactants will encounter "new" reactants less frequently. For some reactions, though, it can also lead to higher reaction rates, as newly dissociated particles are "caged in" by the crowdedness, and so re-associate, thus re-composing the previously dissociated molecule. Moreover, it also affects the ratios between different reaction rates, as the smaller molecules will move around more easily than the larger ones.

Secondly, chemical equilibria are affected by crowdedness, as can be analysed through the *free energy*[6] of the particle configurations. Somewhat counter-intuitively, when a medium is substantially crowded, adding more molecules to the system can actually decrease its entropy. This affects chemical equilibria, because the reaction state with fewer particles may induce a higher entropy and hence a lower free energy. As an example, consider the following reversible dimerisation reaction taking place in

[4] *In vitro* and *in vivo* mean "in glass" and "in life" respectively, and signify the contrast between chemistry taking place in a lab (e.g. a test tube) and chemistry taking place in live environments (e.g. a cell).

[5] Non-specific interactions include steric and/or electrostatic repulsion and electrostatic and/or hydrophobic attraction, although the only one considered in this project is steric interaction, which is the mutual repulsion of molecules as a consequence of their finite size.

[6] Free energy is a composite measure of a certain potential which nature always "seeks" to minimise. In this context, the free energy can be expressed by $H = U - \Theta S$, where U is the internal energy of the system, Θ is the temperature, and S is the entropy.

a crowded environment:

$$A + A \rightleftharpoons A_2. \tag{1.1}$$

Assuming that the size of the dimer is much reduced (i.e. comparable to the size of the monomer), the equilibrium will be "pushed to the right" by the fact that the dimer represents one molecule less than the two monomers, and will thus contribute *more* to the entropy of a *crowded* system.

Thirdly, both reaction rates and equilibria are affected by changes in *thermodynamic activity* (defined below). To see how equilibria are affected, consider the same dimerisation reaction as above. Chemists are used to the equilibrium association constant depending on the relative concentration of the species involved, i.e.

$$K_{12} = \frac{c_2}{c_1^2},$$

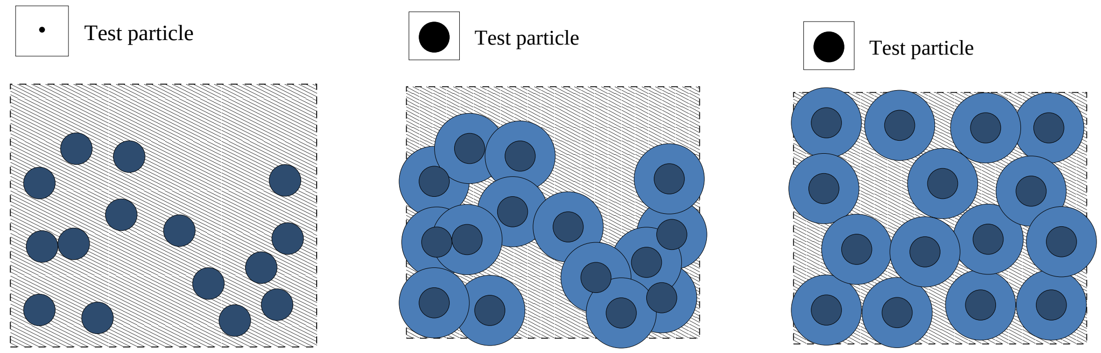
where c_1 is the concentration of the monomer and c_2 is the concentration of the dimer. For crowded media, however, the equilibrium constant cannot be expressed only through the concentrations. Instead it must depend on the thermodynamic activities, a_i, of the species involved. In the case of (1.1), the equilibrium constant can be expressed as

$$K_{12}^0 = \frac{a_2}{a_1^2},$$

and if the macromolecules in the solution only interact through steric repulsion, then the thermodynamic activities are defined through

$$\frac{a_i}{c_i} = \frac{v_{\text{tot}}}{v_i},$$

where v_{tot} and v_i denote the total volume and the volume available to species i, respectively. The key connection to spacial particle configurations is that the volume available to a given species is highly dependent on the spacial particle configuration, as illustrated through Figures 1.1 and 1.2. Consider for example a given test particle, for which we want to determine the available volume. If the test particle is small compared to the background macromolecules, then nearly all of the volume not occupied by the background molecules is available to it (Figure 1.1a). If, on the other hand, the test particle is of the same size as the background molecules, then the volume available to it is much smaller (Figure 1.1b). Furthermore, given a certain number of particles, their exact spacial configuration also heavily influences the available volume (contrast Figures 1.1b and 1.1c). Finally, even intracellular "walls" impact the available volume (contrast Figures 1.2a and 1.2b).

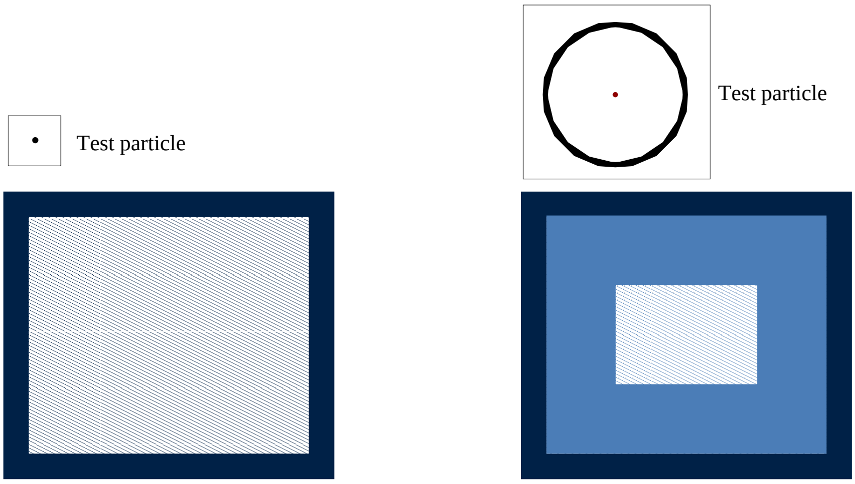


(a) A small test particle in a disorganised, crowded medium

(b) A large test particle in a disorganised, crowded medium

(c) A large test particle in a relatively organised, crowded medium

Figure 1.1: Crowded media with different particle configurations result in different fractions of available space for a given test species. The area available to the test particle is designated by the light shading. The rings surrounding each background macromolecule in Figures 1.1b and 1.1c represent the unoccupied area that nevertheless is unavailable because the particles cannot overlap. Fraction of area occupied by background macromolecules: 15.5% in all three figures.



(a) A small test particle in a confined space.

(b) A large test particle in a confined space.

Figure 1.2: Macromolecular confinement: volume exclusion by pore boundaries. The dark blue frame represents the confinement container, such as a cell pore. The light blue, inner frame represents the excluded volume with respect to the test particle.

Chapter 2

One-dimensional RSA and BD

We will in this chapter consider RSA and BD in one dimension, and also introduce some of the tools necessary to interpret the results of the numerical simulations and computations. We will discuss modelling issues, algorithm issues, implementation issues, and compare the results from RSA and BD simulations.

RSA, as opposed to BD, does not start out with a given distribution of particles; it is not a method which describes the dynamics of a system, but a packing method that gradually fills up a d-dimensional domain. As defined in [14], the classical form of RSA is a stochastic process in which finite-sized particles are sequentially, permanently, and randomly positioned in the d-dimensional domain with the condition that no particles can overlap[1].

Both BD and RSA can be applied in any number of dimensions. Rigorous analytical results for RSA are mostly restricted to one dimension, though, and are discussed in Section 2.1. In one dimension, RSA is also known as the *car parking problem*, because visualising one-dimensional particles inserted onto a line bears an obvious resemblance to cars parking on a strip of road. In view of this analogy, the terms "car" and "particle" will be used interchangeably, as some concepts are common for both one-, and higher-dimensional systems.

2.1 Analytical results on RSA

The car parking problem was first introduced by Hungarian mathematician Alfred Rényi in a paper published in 1958 where he solves a continuous version of the problem. There are also lattice-based models, such as the one studied and solved in [7],

[1] As examined in [6], there are also variations on this classical RSA procedure where partial overlapping is allowed.

and which in the infinitesimal limit yield the continuous version, but here we shall focus on the continuous version, as set out in [14]. The results on one-dimensional RSA are listed in [14], but without any derivation, so I have worked through the relevant details, as presented here.

Consider a test line, or a strip of tarmac, of length L, assumed empty at $t = 0$, and suppose that cars of a fixed length λ attempt to park at random positions on that line at a certain rate per unit length k_a. Assume that the test line is a circle, i.e. that it employs periodic boundary conditions (BCs). The RSA process is illustrated in Figure 2.1. The times given in the figure are approximate, given that it is a stochastic process, in time as well as space. Now assume that we run the RSA process over and over on an infinite number of copies of the initial, empty system. By the law of large numbers, the mean of the stochastic quantities involved are then deterministic. Consequently, if the times in Figure 2.1 represented the mean of these copies, they would then be deterministic, and would read $t = 0$, $t = 1/(Lk_a)$, $t \geq 2/(Lk_a)$ and $t \geq 6/(Lk_a)$, with the inequalities being due to the fact that successful car parking only occurs if the "new" car does not overlap with the "old" ones.

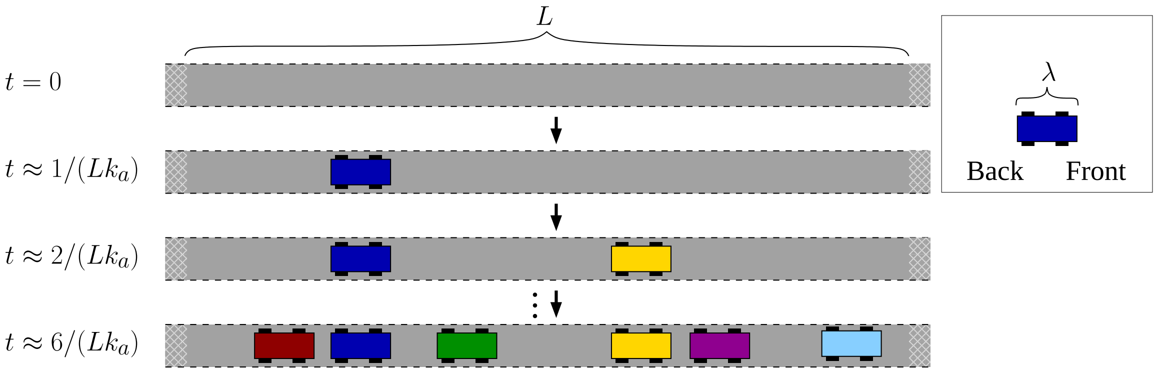


Figure 2.1: Illustration of the RSA process in one dimension (car parking problem) at progressive times.

Define $P(t)$ to be the arithmetic mean (over all of the copies) number of cars per unit interval (i.e. a number density) at time t, and $\rho(t)$ to be the mean volumetric density, i.e. $\rho = P\lambda$. Let $\Phi(t)$ be the mean fraction of space available for parking at a t, and note that this is not the same as the fraction of unoccupied space, $(1 - \rho(t))$. This is because a parked car effectively shades a space of length λ behind (or alternatively, in front) of itself from the insertion of a new car. This phenomenon was further discussed as "excluded volume theory" in Section 1.3.2.

Let $G(h, t)\,\mathrm{d}h$ represent the mean number of gaps, per unit length, of gap length between h and $h + \mathrm{d}h$. This function is referred to as the gap density function, and it is central to this thesis. Note that $G(h,t)$, along with $\rho(t)$, $P(t)$ and $\Psi(t)$, is not only defined for the RSA procedure, but are also valid definitions for BD procedures. The difference lies in how they develop with time, and where there is an ambiguity we will explicitly specify whether these quantities refer to particle configurations established by BD or RSA. We will also repeatedly refer to estimates of $G(h,t)$, which will be denoted $\hat{G}(h,t)$ (see Section 2.2 for density estimation).

The rate of change of the number density is given by

$$\frac{\mathrm{d}P(t)}{\mathrm{d}t} = k_a \Phi(t), \tag{2.1}$$

and the relation tying together Φ and G also follows from the definitions:

$$\Phi(t) = \int_{\lambda}^{\infty} (h-\lambda) G(h,t)\,\mathrm{d}h. \tag{2.2}$$

In this expression, the lower limit of integration and the $(h-\lambda)$ factor is due to volume exclusion, as discussed in Section 1.3.2. In view of the fact that there is one car for each gap, the number density of cars is

$$P(t) = \int_{0}^{\infty} G(h,t)\,\mathrm{d}h, \tag{2.3}$$

whereas the uncovered line is related to $G(h,t)$ through

$$1 - P(t)\lambda = \int_{0}^{\infty} hG(h,t)\,\mathrm{d}h. \tag{2.4}$$

The governing differential equation of the system is

$$\frac{\partial G(h,t)}{\partial (k_a t)} = \begin{cases} 2\displaystyle\int_{h+\lambda}^{\infty} G(\hat{h},t)\,\mathrm{d}\hat{h} & h \leq \lambda, \\ 2\displaystyle\int_{h+\lambda}^{\infty} G(\hat{h},t)\,\mathrm{d}\hat{h} - (h-\lambda)G(h,t) & h > \lambda. \end{cases} \tag{2.5}$$

The differentiation by $k_a t$ instead of t is necessary to normalise the rate expressions, and simplifies notation. The second term in the expression for $G(h,t)$ on the interval $h > \lambda$ corresponds to the destruction of a gap of length h by the insertion of a car into that type of gap; it is only present when the gap length h is larger than the length of a car λ. The first term for both intervals corresponds to the creation of a gap of length h, and can be derived as follows: a "new" gap of length h can arise from any "old" gap of length $\hat{h} \geq h + \lambda$, whence the lower limit of the integral; the rate (normalised by k_a) at which a car is inserted into such a gap is equal to the total length of such gaps that is available for insertion, $(\hat{h} - \lambda)G(\hat{h},t)$; this then has to be multiplied with the probability that the old gap $\hat{h}$ is split into exactly the right proportions (h and $\hat{h} - h$), $2\,\mathrm{d}\hat{h}/(\hat{h} - \lambda)$; the factor 2 comes from the fact that there are two symmetric (around the midpoint) spots where a car can be parked in order to create the new gaps; thus the integrand is

$$2\frac{\mathrm{d}\hat{h}}{(\hat{h} - \lambda)}(\hat{h} - \lambda)G(\hat{h}, t) = 2G(\hat{h}, t)\,\mathrm{d}\hat{h}.$$

Equation (2.5) can be solved for $h > \lambda$ by introducing the ansatz

$$G(h, t) = F(k_a\lambda t)e^{-k_a(h-\lambda)t}. \tag{2.6}$$

Substituting this into (2.5) for $h > \lambda$, we obtain

$$\frac{\partial}{\partial(k_a t)}\left[F(k_a\lambda t)e^{k_a\lambda t - k_a h t}\right] = 2\int_{h+\lambda}^{\infty} F(k_a\lambda t)e^{k_a\lambda t - k_a\hat{h}t}\,\mathrm{d}\hat{h} - (h - \lambda)F(k_a\lambda t)e^{k_a\lambda t - k_a h t}.$$

Using the substitution $u = k_a ht$, carrying out the differentiation on the left hand side and the integration on the right hand side, we get

$$\begin{aligned}\lambda\left[F'(u) + F(u)\right]e^u e^{-k_a ht} - hF(u)e^u e^{-k_a ht} &= 2F(u)e^u\frac{1}{k_a t}e^{-u}e^{-k_a ht}\\ &\quad - (h - \lambda)F(u)e^u e^{-k_a ht}.\end{aligned}$$

Multiplying both sides by $k_a te^{k_a ht}$ yields

$$\begin{aligned}u\left[F'(u) + F(u)\right]e^u - (k_a th)F(u)e^u &= 2F(u) + uF(u)e^u - (k_a ht)F(u)e^u\\ F'(u) - 2\frac{e^{-u}F(u)}{u} &= 0.\end{aligned}$$

This is a first-order ordinary differential equation, and can be solved by an integrating factor, yielding

$$F(u) = A\exp\left(2\int_{\alpha}^{u}\frac{e^{-x}}{x}\,\mathrm{d}x\right),$$

for some $0 < \alpha < u$ and some $A \in \mathbb{R}$. This can be further rewritten as

$$\begin{aligned}F(u) &= A\exp\left(-2\int_{\alpha}^{u}\frac{-e^{-x}}{x}\,\mathrm{d}x\right)\\ &= A\exp\left(-2\left[-\ln\frac{u}{\alpha} + \int_{\alpha}^{u}\frac{1 - e^{-x}}{x}\,\mathrm{d}x\right]\right)\\ &= B\exp\left(-2\left[-\ln u + \int_{0}^{u}\frac{1 - e^{-x}}{x}\,\mathrm{d}x\right]\right)\end{aligned}$$

for some $B \in \mathbb{R}$. Hence,

$$\begin{aligned}F(u) &= Bu^2\exp\left(-2\int_{0}^{u}\frac{1 - e^{-x}}{x}\,\mathrm{d}x\right) \qquad (2.7)\\ &= Bu^2 Y(u), \qquad (2.8)\end{aligned}$$

where we have introduced $Y(u) = \exp\left(-2\int_0^u \frac{1-e^{-x}}{x}\,\mathrm{d}x\right)$ to simplify notation.

Having solved for $h > \lambda$, (2.5) can now be solved for $h \leq \lambda$. Using the initial condition $G(h,0) = 0$ for all $h > 0$, and integrating the first line of (2.5), we obtain

$$\begin{aligned}
G(h,t) &= 2k_a \int_0^t \int_{h+\lambda}^{\infty} G(\hat{h},\hat{t})\,\mathrm{d}\hat{h}\,\mathrm{d}\hat{t} \\
&= 2k_a \int_0^t \int_{h+\lambda}^{\infty} F(k_a\lambda\hat{t})e^{-k_a\hat{h}\hat{t}+k_a\lambda\hat{t}}\,\mathrm{d}\hat{h}\,\mathrm{d}\hat{t} \\
&= 2k_a \int_0^t \frac{F(k_a\lambda\hat{t})}{k_a\hat{t}} e^{-k_a(h+\lambda)\hat{t}+k_a\lambda\hat{t}}\,\mathrm{d}\hat{t}.
\end{aligned}$$

Defining $\hat{u} = k_a\lambda\hat{t}$, we get

$$\begin{aligned}
G(h,t) &= 2\int_0^t F(\hat{u})\frac{1}{\hat{t}}e^{-k_a h\hat{t}}\,\mathrm{d}\hat{t} \\
&= 2\int_0^{k_a\lambda t} \frac{F(\hat{u})}{\hat{u}} \exp\left(-\frac{h}{\lambda}\hat{u}\right)\,\mathrm{d}\hat{u}.
\end{aligned}$$

We can determine the value of B by using the initial condition $P(0) = 0$. Using equation (2.4), carrying out the integration, and letting $t \to 0$, one finds that $B = 1/\lambda^2$. Summarising, the final expression for $G(h,t)$ is then

$$G(h,t) = \begin{cases} F(k_a\lambda t)e^{-k_a(h-\lambda)t} & \text{for } h < \lambda, \\ 2\displaystyle\int_0^{k_a\lambda t} \frac{F(\hat{u})}{\hat{u}} \exp\left(-\frac{\hat{u}h}{\lambda}\right)\,\mathrm{d}\hat{u} & \text{for } h \geq \lambda. \end{cases} \tag{2.9}$$

Having solved for $G(h,t)$, we can now directly solve for $P(t)$. This can be done starting from either equation (2.1) or (2.4), although equation (2.3) arguably offers the most straightforward route, as shown here. Given that $G(h,t)$ is defined distinctly on the two intervals $0 < h < \lambda$ and $\lambda \leq h < \infty$, we need to do two separate integrations. Consider first the interval $0 < h < \lambda$:

$$\begin{aligned}
\int_0^{\lambda} G(\hat{h},t)\,\mathrm{d}\hat{h} &= \int_0^{\lambda} \frac{2}{\lambda^2} \int_0^{k_a\lambda t} e^{-\frac{\hat{h}}{\lambda}\hat{u}}\hat{u}Y(\hat{u})\,\mathrm{d}\hat{u}\,\mathrm{d}\hat{h} \\
&= \frac{2}{\lambda^2}\int_0^{k_a\lambda t} -\frac{\lambda}{\hat{u}}\left[e^{-\hat{u}} - 1\right]Y(\hat{u})\,\mathrm{d}\hat{u} \\
&= \frac{2}{\lambda}\int_0^{u} \left[1 - e^{-\hat{u}}\right]Y(\hat{u})\,\mathrm{d}\hat{u},
\end{aligned} \tag{2.10}$$

where we use the substitution $u = k_a\lambda t$.

Next consider the interval $\lambda \leq h < \infty$:

$$\begin{aligned}\int_{\lambda}^{\infty} G(\hat{h},t)\,\mathrm{d}\hat{h} &= \int_{\lambda}^{\infty} \frac{1}{\lambda^2} e^{u} e^{-k_a \hat{h} t} u^2 Y(u)\,\mathrm{d}\hat{h} \\ &= \frac{1}{\lambda^2} e^{u} u^2 Y(u) \frac{1}{k_a t} e^{-k_a \lambda t} \\ &= \frac{1}{\lambda} u Y(u). \end{aligned} \tag{2.11}$$

As (2.10) is an integral expression, we also want to express (2.11) as such; rewriting (2.11) through integration by parts,

$$\begin{aligned} uY(u) &= \int_0^u Y(\hat{u})\,\mathrm{d}\hat{u} + \int_0^u \hat{u} Y'(\hat{u})\,\mathrm{d}\hat{u} \\ &= \int_0^u Y(\hat{u}) - 2\hat{u}\frac{1-e^{-\hat{u}}}{\hat{u}} Y(\hat{u})\,\mathrm{d}\hat{u} \\ &= \int_0^u -Y(\hat{u}) + 2e^{-\hat{u}} Y(\hat{u})\,\mathrm{d}\hat{u}. \end{aligned}$$

This yields

$$\int_{\lambda}^{\infty} G(\hat{h},t)\,\mathrm{d}\hat{h} = \frac{1}{\lambda}\int_0^u -Y(\hat{u}) + 2e^{-\hat{u}} Y(\hat{u})\,\mathrm{d}\hat{u}. \tag{2.12}$$

Summing up (2.10) and (2.12), we get

$$\begin{aligned} P(t) &= \frac{1}{\lambda}\int_0^u 2\left(1-e^{-\hat{u}}\right) Y(\hat{u}) - Y(\hat{u}) + 2e^{-\hat{u}} Y(\hat{u})\,\mathrm{d}\hat{u} \\ &= \frac{1}{\lambda}\int_0^u Y(\hat{u})\,\mathrm{d}\hat{u} \\ &= \frac{1}{\lambda}\int_0^{k_a \lambda t} \exp\left(-2\int_0^{\hat{u}} \frac{1-e^{-x}}{x}\,\mathrm{d}x\right)\,\mathrm{d}\hat{u}. \end{aligned} \tag{2.13}$$

Equations (2.9) and (2.13) give us analytical expressions for $G(h,t)$ and $P(t)$ for the car parking problem. These are very useful, as they can be used to evaluate the accuracy of the numerical results.

2.1.1 Evaluating the analytical RSA expressions

Analytical expressions for $G(h,t)$ and $P(t)$ were developed in Section 2.1. The evaluation of these is not entirely straightforward, and merits some discussion.

The complications stem from the fact that the expression for $G(h,t)$, equation (2.9) involves some problematic integrals. The integral in the expression for $F(u)$, equation (2.7), does not need to be explicitly evaluated through integration, as it can be expressed as exponential integrals for which MATLAB has standard functions.

On the other hand, (2.9) does need numerical integration in order to solve for $h > \lambda$. This was done using standard MATLAB quadrature functions, (the `quad` family, see below for discussions) but since these cannot handle vector input values, a loop is needed in order to evaluate $G(h,t)$ at multiple gap lengths h.

The numerical RSA procedures (see Section 2.3) are run up until a certain target volumetric density, ρ_0, is reached. Define T_{RSA} to be the point in time such that $\rho_0 - \rho(T_{\text{RSA}}) = 0$. Given that equation (2.9) for $G(h,t)$ requires the time of the RSA procedure as input, whereas the numerical experiments run the RSA procedure with ρ_0 as input, a method is needed to estimate T_{RSA} from ρ_0. The RSA simulations do give estimates of T_{RSA},

$$\hat{T}_{\text{RSA}} = N\Delta t_{\text{RSA}},$$

where N is the total number of insertion attempts necessary to reach ρ_0 and $\Delta t_{\text{RSA}} = 1/k_a$. However, these are not very accurate when the number of particles in the system, n, is small.

However, since the function $\rho(t)$ is strictly increasing for increasing t, $\rho_0 - \rho(t)$ is strictly decreasing for increasing t. Therefore, and especially when using $\hat{T}_{\text{RSA}}$, it is not too hard to find one point at which $\rho_0 - \rho(t)$ is negative, and one at which it is positive. These two points can then be given as starting values to MATLAB's `fzero`, which uses a combination of bisection, secant, and inverse quadratic interpolation methods, to find T_{RSA} [9].

In order to evaluate $\rho(t)$ one could use equation (2.13), or alternatively (2.3). In the latter case, which is the one employed in this project, the upper limit of integration is infinite and this causes some problems for the numerical integration. If `quad` is used, then an finite, yet sufficiently large limit has to be supplied. The problem is that after a certain point, as one extends the upper limit, the integral was observed to decrease towards zero, even though $G(h,t)$ is strictly positive! The exact reason for this is probably a combination of stiffness[2] and the discontinuity in the first derivative of $G(h,t)$ at $h = \lambda$, as neither of these factors were individually enough to cause this problem[3]. The solutions that we found to work was either to explicitly carry out the integration in two parts, or to use `quadgk`, which can accommodate discontinuities through so-called *way points*.

[2] The stiffness in question is the fact that $G(h,t)$ has the shape of a decreasing exponential function which, when viewed from "afar", looks like it's discontinuous around zero.

[3] Using `quad` on equally big intervals for functions that are equally stiff as $G(h,t)$ on large intervals (but analytically $\mathcal{C}^\infty$ on the whole interval, as opposed to $G(h,t)$) works fine, while `quad` over equally large intervals with $h \cdot G(h,t)$ as the integrand also works fine, even though there's still the discontinuity.

2.2 Density estimation

Density estimation is the attempt to infer a probability distribution function (PDF) from a large enough set of observed data that one assumes is a random sample from a population distributed according to an underlying PDF [3]. It is an important tool in experimental science, also when the experiments are run on a computer, as is the case with this project, where we use it predominantly to estimate the shape of the gap density function $G(h,t)$, thus obtaining simulation estimates, $\hat{G}(h,t)$.

2.2.1 Kernel smoothing techniques

The genesis of density estimation is the familiar histogram. Histograms are difficult to superimpose for comparisons, but more importantly, they have the disadvantage of being defined on a partition. Therefore, what is often applied is a *kernel smoothing technique* [3]. Kernel smoothing is a form of non-parametric[4] curve fitting that is employed to reveal traits of the data set. Whereas a histogram amasses the data into certain bins which partition the space containing the data, kernel smoothing can be regarded as its continuous counterpart, giving continuous curves instead of bars. In kernel smoothing, the generalisation of the bin width for histograms is the *bandwidth* and it regulates the smoothness of the estimated curves[5]. The density estimate, $\hat{f}$, at x, of the underlying PDF, f, based on the set of data, $\{x_i\}_{1\leq i\leq n}$, is:

$$\hat{f}_{w_b}(x) = \frac{1}{nw_b}\sum_{i=1}^{n} K\left(\frac{x_i - x}{w_b}\right), \tag{2.14}$$

where K is the kernel function. Two of the most common choices for K is that of a normal PDF (i.e. $K \sim \mathcal{N}$) and that of a uniform PDF (i.e. $K \sim \mathcal{U}$), both of a given variance, denoted σ, and centred on zero for symmetry. It is easy to show that as long as the kernel is a PDF, then the density estimate, $\hat{f}(x)$, will also be a PDF.

All, save one, of the density estimation in this thesis employ normal PDFs for the kernel. The one which doesn't is Figure 2.4, which uses a uniform, symmetric PDF for the kernel. As can be seen by comparison with the other figures, the result is generally less smooth for a given bandwidth. This is because, for a given point, the uniform kernel only takes into account data within a distance w_b from that point, whereas the normal kernel takes into account all of the data, albeit with less weight on distant points.

[4] Non-parametric in the sense that one does not try to fit the data to a known type of an underlying model.

[5] Conversely, a histogram can be regarded as a kernel smoothing technique using a box kernel and only doing the estimation at points that are w_b apart.

2.2.2 Selecting the bandwidth

A measure that is often used to establish optimality of bandwidths is the *Mean Integrated Square Error* (MISE), expressed by

$$\mathrm{MISE}\left(\hat{f}_{w_b}\right) = \mathbb{E}\left(\int \left[\hat{f}_{w_b}(x) - f(x)\right]^2 \mathrm{d}x\right). \tag{2.15}$$

As noted in [9], although the real density f is of course generally unknown, assuming that it is a normal PDF of variance σ^2, and minimising (2.15), leads to

$$w_b = \left(\frac{4}{3n}\right)^{1/5} \sigma. \tag{2.16}$$

Noteworthy observations from this formula are the two scalings $n^{-1/5}$ and σ, which imply that the bandwidth grows for increasing variation and shrinks for increasing data sample size.

Although the data samples of gap lengths used in estimating $G(h,t)$ is not normally distributed, using the same bandwidth selection formula (2.16) as for normally distributed data still seems reasonable; there is only one discontinuity in the first derivative of $G(h,t)$ (2.9), as discussed in Section 2.4; furthermore, plots of $G(h,t)$ (for example in Figure 2.2) reveal that $G(h,t)$ does not otherwise exhibit large fluctuations in of the first derivative (i.e. large second derivatives), but rather decreases smoothly towards zero.

Moreover, often a "robust"[6] estimate of the variance is preferred to the regular empirical standard deviation. This works better for density distributions with long tails[7], something that also seems justified in the case of $G(h,t)$, and the default choice of MATLAB's `ksdensity` is derived from the *Median Absolute Deviation*, or MAD, expressed as

$$\mathrm{MAD}\left(\{x_i\}\right) = \mathrm{median}_i\left(|\mathrm{median}_j(x_j) - x_i|\right). \tag{2.17}$$

From (2.17) one can derive a consistent estimate of the standard deviation σ by multiplying by a certain factor, which in the case of normal distributions is approximately 1/0.68, where 0.68 arises as approximately the 3rd quartile of $\mathcal{N}(0,1)$. We did some numerical experimentation, and the results show that the MAD of the data of the gap densities $\{h_i\}_{1\leq i\leq n}$ (as generated by simulations) is between 78% and 84% of the empirical standard deviation, independently of the system size.

[6] An estimate which is not easily thrown off by extreme sample values.

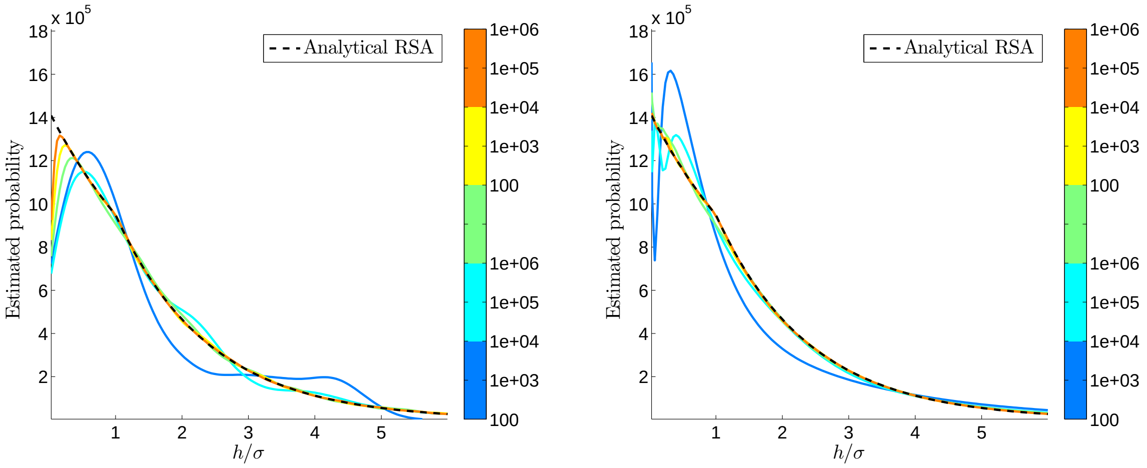
[7] Distributions with long tails can be said to have relatively large standard deviations compared to, e.g. a uniform distribution, which has no tail.

In summary, we use the following formula to establish the kernel density estimation bandwidth:

$$w_b = \left(\frac{4}{3n}\right)^{1/5} \mathrm{MAD}\left(\{\mathrm{x_i}\}\right). \tag{2.18}$$

2.2.3 Data on bounded supports

Given that the length of a gap is a positive quantity, the data from the numerical experiments, $\{h_i\}_{1\leq i\leq n}$, will lie inside $[0,\infty)$. However, $G(h,t)$ does not tend to zero for $h \to 0$, but instead peaks at $h = 0$. This is a problem, because the density estimate for low values of h will be dragged down by the absence of values for $h < 0$. Figure 2.2a shows the result of this, as all of the curves, except the analytical one, bend downwards as they approach zero. This is obviously an artefact of the kernel smoothing technique, and cannot be remedied indefinitely by increasing the system size, n, because computing power is limited.



(a) Without log-transformation of data.

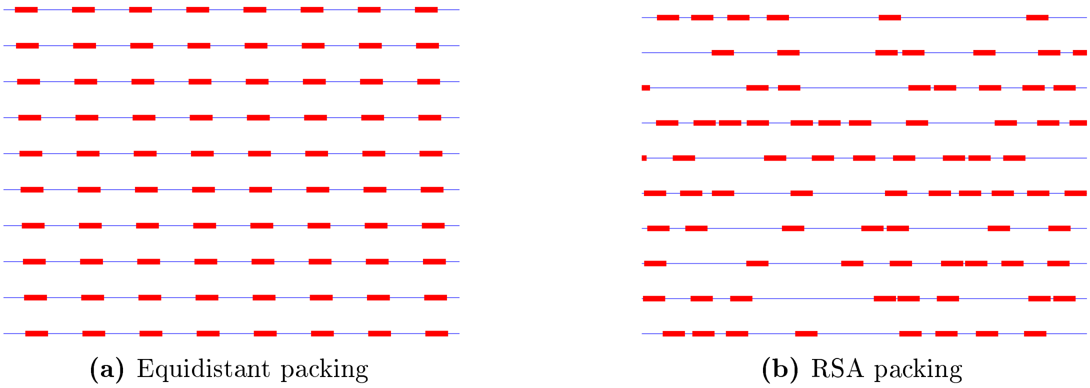
(b) With log-transformation of data.

Figure 2.2: Comparison of kernel density estimation with and without log transformation. The graphs are gap density functions, for varying system size, n, indicated by the colour bars.

A possible solution is to apply a transformation to the bounded region which transforms it into an unbounded one. Taking logarithms is suitable for a domain bounded below by zero, transforming the positive real line into the entire real line. As with any change of variables, the chain rule then requires the multiplication by the derivative of the transformation, in this case $\frac{\mathrm{d}}{\mathrm{d}x}\log(x) = 1/x$. The bandwidth is calculated from the transformed data. The result can be observed in Figure 2.2b;

although for small systems this method produces large oscillations (through $1/x$), for large enough systems (the orange lines) the numerical curves are visually indistinguishable from the analytical one, all the way up to $h = 0$. However, as discussed further in Section 2.4, the log-transformation of the domain is not employed in this thesis, mostly due to the oscillations it causes for small values of h when n is below 1×10^4.

2.3 Simulating RSA


(a) Equidistant packing

(b) RSA packing

Figure 2.3: Demonstration of the visualisation of the system of cars on a line that was used for debugging. The test line is broken up into several lines to make the most out of the display area.

It is convenient to begin with implementing RSA before BD, as RSA naturally deals with some of the issues common to both processes. The test line is typically set to unit length, and cars are sequentially inserted onto the line until a target volumetric density, ρ_0, is reached. The system size, i.e. the number of cars, n, is be provided alongside ρ_0 as a parameter to the experiment, and the (fixed) length of the cars is calculated by

$$\lambda = \frac{L}{n}(1 - \rho_0) \tag{2.19}$$

in order that $n\lambda = L(1 - \rho_0)$. Note that in the case of varying car lengths, considered in Section 2.8, (2.19) only gives the mean car length.

The default form of the RSA procedure is set out in Algorithm 1. Other variations include polydispersity, as discussed in Section 2.8, and variations on the implementation is given in Section 2.3.2.

```
Algorithm 1: Random Sequential Adsorption (RSA)
Input: $L$, $n$, $\rho_0$, $k_a$
Output: A list of cars packed by RSA, $T_{\text{RSA}}$
Let $\rho = 0$
while $\rho \leq \rho_0$ do
    Draw $\xi$ from $\mathcal{U}[0,1]$
    Increment the RSA time by $\Delta t_{\text{RSA}} = 1/k_a$
    if The position indicated by $L\xi$ is available for insertion then
        Insert the car position and length in the arrays
        Increment $\rho$ by $\Delta\rho = \lambda/L$
    else
        Reject the insertion attempt
```

2.3.1 Why $G(h,t)$ characterises the distribution

Once RSA simulations have been run, the spacial particle configuration can be studied by plotting density estimates of the gap density function $G(h,t)$. Given that different system sizes, n, and domain lengths, L, may be used, the gaps are all given relative to the car lengths, λ, or to the mean gap length of the system, $\bar{h} = \lambda(1-\rho)/\rho$.

The central question that is asked in this thesis is whether the spacial particle configurations obtained from BD matches those obtained from RSA. The reason why the gap density function can be used in answering this question in one dimension is that, apart from the order of the gaps, the gap density function $G(h,t)$ fully characterises the particle configuration.

In the case of RSA, the ordering is doubtlessly independent of the gap lengths. In other words, the length of gap i gives no information on gap the lengths of gaps $i+1$ and $i-1$, or those further away. This follows from the definition of RSA, which employs a uniform probability of insertion, and it implies that knowing $G(h,t)$ is sufficient to characterise the spacial particle configurations.

Unlike RSA, it is not immediately evident that for BD that there is this independence between the gap lengths; although an attempt at showing this from theoretical arguments is made in Section 2.11, I am not aware of any rigorous mathematical proof of this (keeping in mind the finite size of the particles). Therefore a numerical study was undertaken, and Figure 2.4 shows the probability that a gap of length h_{ref} follows a gap of length h, relative to the abundance of gap length h. Each curve corresponds to a different h_{ref}, and since the abundances of h_{ref} also differ, the curves have been scaled according to their abundance. Figure 2.4 shows that both for RSA

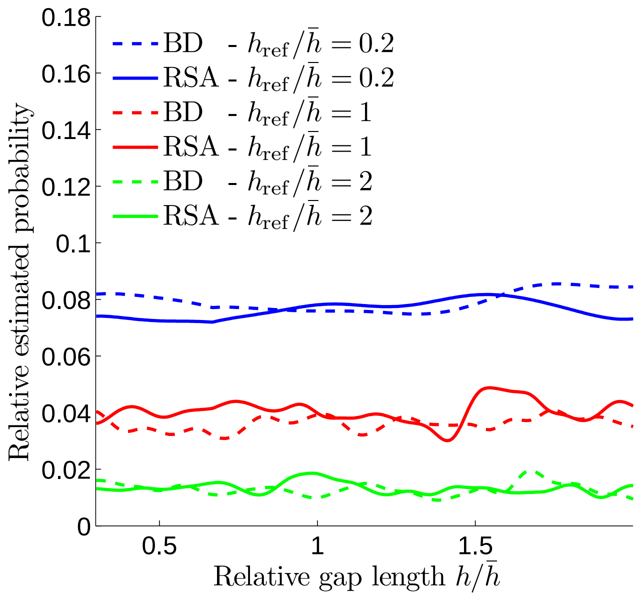


Figure 2.4: Plot illustrating that the order of gaps is independent of their length. $n = 1 \times 10^6$, $\rho = 0.3$. Density estimation uses a uniform Kernel function, unlike all of the other density estimates in the thesis.

and BD, and regardless of h_{ref}, the general tendency is that of a uniform probability distribution. This uniformity means that for any given type of reference gap h_{ref}, the conditional probability of the following gap being of length h is equal to the unconditional probability of that gap being of length h. By induction, this observed, pairwise independence can then be propagated throughout the sequence of cars, leading to the conclusion that the length of a given gap is independent of any of the gaps around it.

2.3.2 Implementation issues relating to RSA

At first the car/particle configurations were stored in simple two-column arrays; the first column would hold the "Front" of the cars, while the second would hold the length of each car. Unlike higher-dimensional systems, one-dimensional systems have a natural ordering based on the position of the cars. Furthermore, it is an inherent characteristic of working with non-overlapping cars that the order is fixed. Due to considerations such as search speed and structure, the arrays would be kept sorted (i.e. in the same order as the cars on the line) over the course of the RSA packing procedure. With hindsight this now appears to have been a misjudgement, as the extra coding work required for this does not outweigh the dubious advantage in running time of the code[8].

Another implementation option is to keep a running list of the space available for the insertion/parking of a car, at the cost of considerable bookkeeping. Because the random variable, ξ, specifying the insertion position is mapped to this space, each insertion "attempt" will then always succeed. Due to the bookkeeping, we found this method to be several times slower than Algorithm 1, depending on the system size (for $n = 1 \times 10^4$ it was about four times slower). It also requires extra calculations to give an estimate of the packing time T_{RSA}, with $\Delta t_{\text{RSA}}(t) = \Phi(t)L/k_a$. For these reasons, and because it is difficult to generalise this method to higher dimensions, the bookkeeping RSA version was also only given preliminary use in the beginning of the project.

2.4 Convergence of RSA simulations

The one-dimensional simulations are computationally light-weight compared to two-dimensional systems, and so system sizes of up to $n = 1 \times 10^6$ can be simulated on a time scale of a day on a modern laptop (see Appendix B for computer specifications). As can be observed from the subfigures of Figure 2.5, the numerical curves get closer and closer to the analytical curve for increasing n, clearly converging for $n \to \infty$.

The magnified plot (Figure 2.5b) serves to highlight this convergence for a specific interval, where the difference between the curves is easily observable due to the peaking of the curves. Observe that the curves corresponding to $n \geq 1 \times 10^4$ are practically collinear with the analytical curve until a very noticeable change in curvature occurs. Let h_I denote the point at which this change occurs, typically estimated visually, although $\arg\max_h \hat{G}(h,t)$ provides a good automatic estimator. Clearly, the dominating "form" of convergence for $n \geq 1 \times 10^4$ is that h_I shrinks as n increases. This differs from the typical form of convergence that one expects from estimations with $n \to \infty$, where the vertical difference between curves gradually decreases over a given interval. The shrinking of h_I is therefore not so much a consequence of the higher accuracy provided by larger samples; it is rather due to the fact that the bandwidth, w_b, as calculated from (2.18), gradually decreases for increasing n.

The bandwidth used for $n = 1 \times 10^6$ is about 0.07λ, while the corresponding curve does not start swinging downwards until $h = 0.2\lambda$. This means that for $h > 3w_b$, the

[8] The running time is of course dependent on the size of the system, n, but keeping the list sorted during the RSA procedure increases the running time compared to not doing so. One-dimensional BD is much faster to simulate with an ordered list, but because the order is permanently set, the ordering need only take place once before the simulation.

downturn effect of the kernel smoothing is no longer visually discernible. The reason is of course that the weight attributed to points at that distance is negligible; the cumulative density function of $\mathcal{N}(0, 1)$ at -3 is is 0.00135.

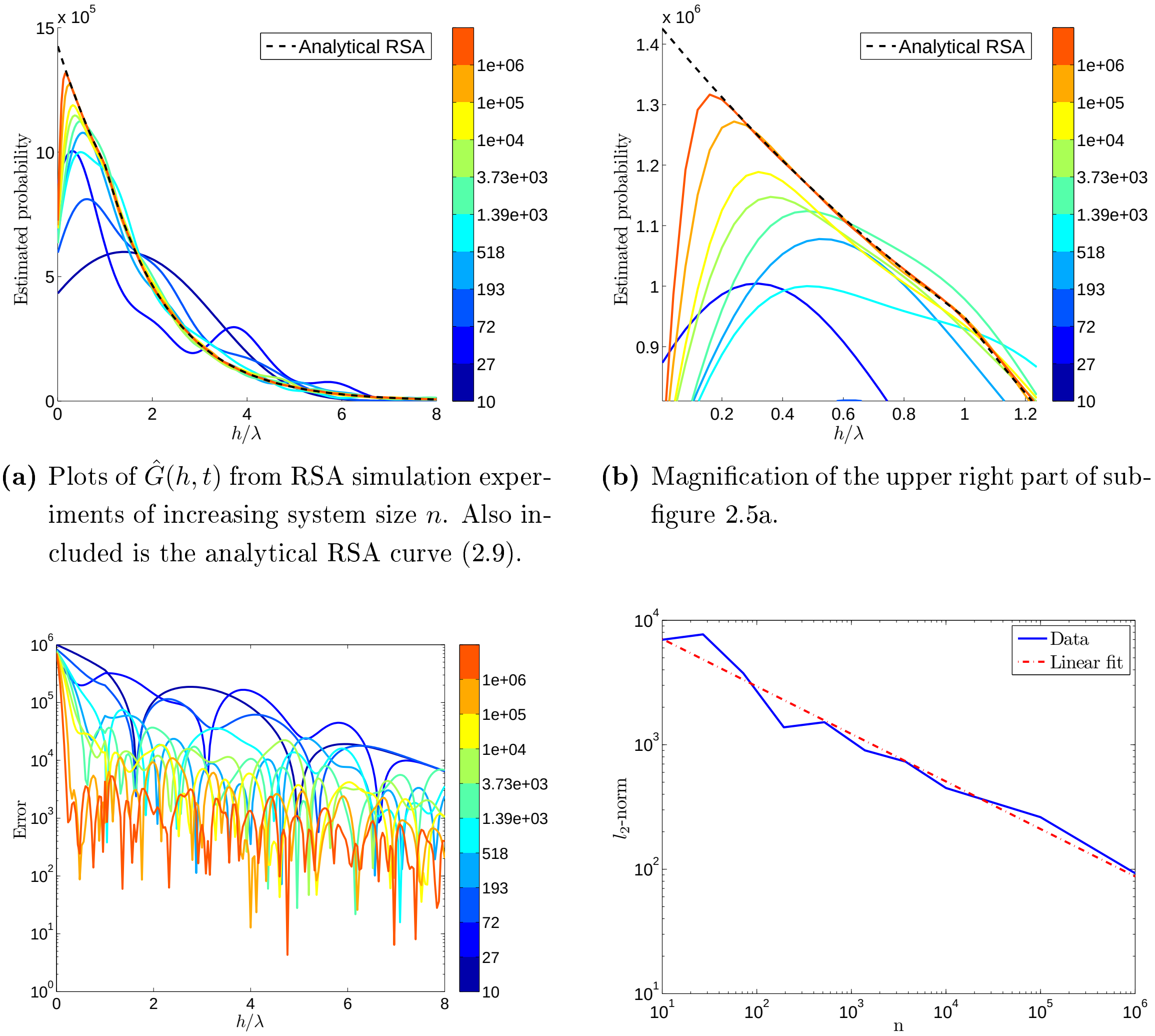
What is also observable from Figure 2.5b is that only for $n \geq 1 \times 10^5$ can the curves be said to follow the kink at $h = \lambda$, and so for any type of simulations, whether RSA or BD, the density estimates of $G(h, t)$ should not be expected to possess this feature. This kink arises as consequence of the piecewise definition of $G(h, t)$ (2.9) which guarantees $\mathcal{C}^0$ continuity, but not higher, and it is very useful as a pronounced feature of the RSA curves which can be used in comparisons.

It is frequently of interest to have a measure of the difference between functions. The l_2 norm is probably the most common choice for numerical data points, and as such it is the one we will use. The question arises, however, whether it would be wise to include the interval $[0, h_I]$ in the range of application of the l_2 measurements. Although the downwards bend for $h < h_I$ is an artefact of the kernel smoothing, the answer is not obviously to exclude $[0, h_I]$, as, for instance, two equal data sets would engender two identical curves, over the interval $[0, h_I]$. Moreover, one could argue that the downwards bend is a blessing in disguise, as it does help to visually distinguish the curves. Nevertheless, due to the added complication of varying system sizes, n, and bandwidths, w_b, both of which affect the width of the artificial downturn (i.e. h_I), we restrict the range on which the l_2 norm is applied to $[h_I, \infty)$. It should be noted, however, that this has a bearing on results such as the convergence rate, which in the case illustrated in Figure 2.5d is approximately halved when the $[0, h_I]$ is taken into account. An alternative solution to restricting the interval of application could be to use a log transformation before applying the kernel smoothing technique, as discussed in section 2.2.3. However, this is prone to extreme oscillations for small h, and therefore we do not apply this method. In summary, if density estimation is carried out for M points, and I represents the index of h_I, the expression applied to calculate the restricted l_2 norm of the difference between two curves is

$$\|G - \hat{G}\|_{2,\text{res}} = \frac{1}{M - I + 1} \sum_{i=I}^{M} (G(h_i) - \hat{G}(h_i))^2, \tag{2.20}$$

where, unlike for the rest of this thesis, the indices i do not represent the car/particle/gap index, but the index of an abscissa point h_i.

Figure 2.5d shows a log-log plot of the restricted l_2 error between the analytical and numerical curves. In view of the preceding discussion on the restriction of the l_2 norm this should not be taken to convey a lot of information in absolute terms, but is more useful in establishing convergence rates. As can be seen from the slope of the regression line in Figure 2.5d, the restricted l_2 error for numerical RSA appears to converge at the rate of $n^{-2/5}$. The data curve is clearly linear, although it should also be noted that with a higher resolution on the abscissa (as a result of a larger number of simulations with different system sizes), the curve would have been more jagged. Presumably the convergence of the error would continue to decline as shown in Figure 2.5d, if system sizes larger than $n = 1 \times 10^6$ are simulated, up until some point depending on machine precision.



(a) Plots of $\hat{G}(h,t)$ from RSA simulation experiments of increasing system size n. Also included is the analytical RSA curve (2.9).

(b) Magnification of the upper right part of subfigure 2.5a.

(c) Semilog-y plots of the absolute error between the analytical RSA curve and the numerical curves, for increasing system size n.

(d) Plot of $\|G(h, T_{\mathrm{RSA}}) - \hat{G}(h, T_{\mathrm{RSA}})\|_{2,\mathrm{res}}$, with $h_I = \arg\max_h \hat{G}(h, T_{\mathrm{RSA}})$ for the lowest value of n. Logarithmic axes. Slope of the linear fit is about -0.4.

Figure 2.5: Convergence studies for RSA simulations. Colour bars indicate n.

2.5 Analytical results on Brownian motion

The advantage of modelling the dynamics of particle systems by Brownian Dynamics (BD) is that it reduces the dimensionality (and hence complexity) of the system. Instead of explicitly considering all of the particles within a system, one focuses on the particles of interest, i.e. the macromolecules, and represents the other particles in the system as having the effect of random background noise on the macromolecules. The following results on Brownian motion can be found in most basic stochastic textbooks, and their derivation is given here for completeness.

Let $\mathbf{X}(t) \in \mathbb{R}^d$ be the position of particle i (in this section we drop the particle index, i, for clarity) at time t, the stochastic differential equation representation of this particle undergoing Brownian motion is

$$\mathrm{d}\mathbf{X}(t) = D\,\mathrm{d}\mathbf{W}(t) + \mathbf{F}(\{\mathbf{X}(t)\}\,;t)\,\mathrm{d}t, \tag{2.21}$$

where $\mathbf{F}$ represents deterministic drift, $\{\mathbf{X}(t)\}$ is the set of positions of all of the particle in the system, D, is the diffusion constant, and $\mathbf{W}(t)$ is a Wiener process, i.e. a stochastic process with independent increments such that any component, W_j, of $\mathbf{W}$ follows a normal distrjbution of variance equal to the duration of the BD process, T_{BD}. In other words,

$$W_j(t + T_{\mathrm{BD}}) - W_j(t) \sim \mathcal{N}\left(0, T_{\mathrm{BD}}\right),$$

for all t. The drift term is used to model inter-particle interactions and external forces. As no such forces are considered in this thesis, however, we shall immediately set it to zero. When numerically simulating Brownian motion, one of course has to discretise time. Hence Equation (2.21) becomes

$$\mathbf{X}^{k+1} = \mathbf{X}^k + D\sqrt{\Delta t_{\mathrm{BD}}}\,\xi^{k+1}, \tag{2.22}$$

where $\mathbf{X}^k$ is the position of the particle at the k^{th} step of the simulation, ξ^k is a normally distributed, d-dimensional, random variable of mean zero and variance $\sigma = I_d$ (identity matrix), and $\Delta t_{\mathrm{BD}} = T_{\mathrm{BD}}/N$. The square root of Δt_{BD} provides the correct scaling, as seen by the following argument:

$$X_j^N - X_j^0 = \sum_{k=1}^{N}\left(X_j^k - X_j^{k-1}\right) = \sum_{k=1}^{N} D\sqrt{\Delta t_{\mathrm{BD}}}\,\xi^k = D\sqrt{\Delta t_{\mathrm{BD}}}\sum_{k=1}^{N}\xi^k.$$

But the variance of two independent, normally distributed variables is the sum of their variance. Thus, since the ξ^k are independent,

$$\begin{aligned} X_j^N - X_j^0 &= D\sqrt{\Delta t_{\mathrm{BD}}}\,\sqrt{N}\xi \\ &= D\sqrt{T_{\mathrm{BD}}}\xi, \end{aligned}$$

where ξ denotes a random variable that follows $\mathcal{N}(0, I_d)$. Hence, we observe that the total potential displacement of the N steps is equal to that of the continuous definition of Brownian motion (2.21), scaling with $\sqrt{T_{\text{BD}}}$. It is then easy to calculate that the expected value for the absolute displacement also scales with $\sqrt{T_{\text{BD}}}$:

$$\mathbb{E}\left(|X_j^N - X_j^0|\right) = D\sqrt{\frac{2}{\pi}T_{\text{BD}}}. \tag{2.23}$$

2.6 Simulating BD

The BD simulations, although seemingly straightforward, offer quite few choices and challenges, even in one dimension. These are mostly due to collisions between the particles, and are discussed in Sections 2.6.1.

Equation (2.22) is the starting point for the simulations, and Algorithm 2 describes a generic, one-dimensional BD algorithm.

```
Algorithm 2: General Brownian dynamics (BD)
Input: A list of $n$ cars with given positions, $\{\mathbf{X}^0\}$, and lengths, $D$, $\Delta t_{\text{BD}}$, $T_{\text{BD}}$
Output: The same list of cars, but with new positions
Calculate $N = T_{\text{BD}}/\Delta t_{\text{BD}}$
Set $k = 1$
while $k \leq N$ do
    Draw $\xi^k \in \mathbb{R}^n$ from $\mathcal{N}[0,1]^n$
    Move cars according to $\mathbf{X}^k = \mathbf{X}^{k-1} + D\sqrt{\Delta t_{\text{BD}}}\,\xi^k$
    Check for overlapping cars or cars which have switched order
    Treat these overlappings (collision handling)
```

2.6.1 Collision handling

In reflection of "reality", all of the particles should be moved simultaneously, for each iteration, i.e. not one-by-one, or sequentially. However, the finite time step might cause collisions between the cars (cars that either overlap, or have switched order). This should be resolved by some form of collision handling, where ideally some form of molecular dynamics resolves the collisions. For example, if two cars are found to overlap, then one can bounce them apart a distance equal to that with which they overlap, as a reflection.

The problem arises when multiple overlappings happen on successive cars, due to clustering[9]. Or perhaps even car $i-1$ and car $i+1$ (i.e. cars which should have one car and two gaps separating each other) are now overlapping. If one were to make all of the correct bounce-backs for a sequence of successive collisions, it would be necessary to draw out the trajectory of the cars from one iteration to the next. Another compounding difficulty is the fact that resolving one collision zone might create another on its fringes, whose resolution might create another, and so on. This could in principle all be resolved as long as the collision handling has a sufficient level of sophistication, but it would be at great computational cost, and would require keeping track not only of the current position of the particles, but also their previous position. The question then is whether that would be worth it, or if it actually just defeats the purpose of BD simulations.

Note that this is really a consequence of simulating finite-sized particles in a crowded environment. Unless Δt_{BD} is infinitesimal, or one does actually trace out the trajectories of the particles from k to $k+1$, the collision handling is probably bound to affect the statistical properties of the dynamics, making the BD simulation depart somewhat from the "pure" continuous form of Brownian motion (2.21), to which all BD simulations should converge as $\Delta t_{\mathrm{BD}} \to 0$. Although I did not analytically attempt to analyse these properties, the recognised objective was to minimise this departure from the continuous form. "Common sense", mixed with numerical experimentation, was an important source of guidance. The result was the implementation of several algorithms to resolve the collision problem, and is discussed in Sections 2.6.2 and 2.6.3.

As discussed in [13], many forms of BD have been developed by researchers for different purposes. For example, some employ adaptive time-stepping and some use discretised domains (lattices). Some of these would surely have been useful to this project. However, they would have to be effective in crowded environments, and not too intricate, as it is important that they do actually implement Equation (2.22), and not some slight variation of it. Other variations might be hard to analyse, and in this thesis our principal aim is to approximate BD by RSA, not the other way around. With the time limit involved and the difficulty of finding and analysing other BD methods, I therefore developed and used my own BD algorithms.

[9] Although there are no particular attractive forces in the system, some clusters are bound to arise simply because there are so many cars in the system.

2.6.2 A short study of a failed BD algorithm

Before we discuss the algorithms that were used with success in this project in Section 2.6.3, consider Algorithm 3, which substitutes Line 7 of Algorithm 2, and provides an interesting example of a collision handling method that is not very successful. The problems with Algorithm 3 are two-fold: (1) in limiting certain movements it employs a relatively drastic collision resolution method, whose influence on the statistical properties of the BD might be correspondingly drastic, and (2) it is computationally heavy for large n.

Algorithm 3: Rejective BD (collision handling method)

```
while There are any overlapping cars do
    if There are two successive overlappings then
        Reject/cancel the movements carried out by $\xi^k$
        Break while-loop
    else
        Bounce apart those cars that overlap by a distance equal to that with
        which they overlap
        Check again how many overlapping cars there are
if Movements were rejected then
    Do not increment $k$
else
    Increment $k$
```

What is interesting about Algorithm 3 is that it allows for some interesting heuristics in explaining the observed, dreadful simulation times. Assume, for the sake of explanation, that there are n_c distinct clusters[10] of cars in the system, where n_c is proportional to n: $n_c = \gamma n$, $\gamma \ll 1$. Also assume that the probability of any given cluster producing at least one double overlapping, for iteration $k \to k+1$, is $1-\beta$, $\beta \in [0,1]$. In other words, $1-\beta$ is the probability that any given cluster makes Algorithm 3 reject the proposed set of movements, $D\sqrt{\Delta t_{\mathrm{BD}}}\xi^k$, for $k \to k+1$. In order for the next iteration to be accepted, none of the n_c clusters must produce double overlappings. Assuming independence between clusters, the probability of not having any double overlappings is $\beta^{n_c} = \beta^{\gamma n}$. This implies that the probability of accepting

[10] It is surely possible to put a strict definition on clustering, but only the intuitive concept of "regions of extra high density" is necessary for this argument.

the proposed set of movements decreases exponentially with n, and this makes large systems practically impossible to simulate via Algorithm 3.

The obvious solution to the problem is to decrease the iteration time Δt_{BD}, as this should reduce the number of overlappings overall, as discussed in Section 2.6.5. This, however, was not observed to be very successful for Algorithm 3, as the time saved by not having as many rejected iterations was fully compensated for by the extra iterations of the decreased Δt_{BD}. This could again be explained by some rather interesting reverse-engineered heuristics, but we conclude this section simply with the empirical observation, as it stands.

2.6.3 Working BD algorithms

With the lessons taken from the trial and error of Algorithm 3 it seems that one must compromise a little on the strictness of the collision handling. What if each overlapping is treated separately, regardless of how the "bounce-back" may affects other neighbouring cars, and only once. This is the method of Algorithm 4, which again replaces line 7 of Algorithm 2.

Algorithm 4: Simultaneous BD (collision handling method)

```
1 for Each collision between two cars do
2     Bounce apart those cars that overlap by a distance equal to that with
      which they overlap
3     Check again how many overlapping cars there are, and store this
      information
4 Increment $k$
```

The collision handling of Algorithm 4 might create some new remaining, unresolved overlappings each iteration, but not as many as it resolved in the first place. Moreover, it is relatively inexpensive, as the number of operations in the collision handling procedure, per iteration, only scales with n, and not e^n, as with Algorithm 3. Furthermore, the total number of unresolved collisions, ζ, is monitored (line 3 of Algorithm 4). Let $\Psi = \zeta/(nN)$ be the average number of unresolved collisions of a BD procedure per iteration per car. Then Ψ gives a measure of how valid a BD simulation is, and can be used as a "quality guarantee" of a given BD simulation.

There are of course many intermediate versions of Algorithms 4 and 3. For example, one could do two repetitions of Algorithm 4, or one could keep looping until there were no unresolved collisions left. Neither would ensure that the simulation is close to continuous BD, though, and they would come at greater computational cost.

Another, quite different, approach consists in compromising on the simultaneity of the movements of the particles (as is in effect already done in the collision handling of Algorithm 4). This is the method of Algorithm 5, which replaces lines 5 to 7 of Algorithm 2. In order to avoid "prioritising" one car ahead of others, Algorithm 5 also randomises the order of movement of the cars from one iteration to the next. This is represented by the random permutation R on the numbers $\{1, \cdots, n\}$. Note that no collision handling is necessary in Algorithm 5; as the cars are moved sequentially, the bounds on the movement of car i for a given iteration, k, are set by the neighbouring cars, $i-1$ and $i+1$, which are stationary while car i is moved. If car i overtakes those bounds, it is simply reflected back onto the delimited segment.

Algorithm 5: Sequential Brownian dynamics (BD)

```
for $i = 1, \cdots, n$ do
    Get a random car index: $j(i) = R(\{1, \cdots, n\}; i)$
    Move car $j$ according to $X_j^k = X_j^{k-1} + D\sqrt{\Delta t_{\text{BD}}}\,\xi_j^k$ within segment
    delimited by neighbouring cars
Increment $k$.
```

Although Algorithm 4 is the algorithm that has been used to produce all of the plots for the one-dimensional BD procedures, both Algorithm 4 and 5 have their advantages and disadvantages. These are related to Δt_{BD} and T_{BD}, and are discussed in Sections 2.6.4 and 2.6.5.

2.6.4 Selecting T_{BD}

Two key parameters to all BD algorithms are the time step, Δt_{BD}, and the total time of the process, T_{BD}. The question is how they should be set.

Recall that the total, expected, unrestricted movement of a particle undergoing Brownian motion, either continuous (2.21) or discrete (2.22), scales with $\sqrt{t}$:

$$X_i(t) - X_i(0) = D\sqrt{t}\xi \sim \mathcal{N}(0, t).$$

The movement of the non-overlapping cars is not unrestricted, though, and the result is that the resulting movement after N iterations (corresponding to $t = T_{\text{BD}}$) is approximately a uniform PDF on the limited interval, as long as T_{BD} is sufficiently large, as shown by the following argument.

Define T_0 as the point in time when the standard deviation of $X_i(t) - X_i(0)$ is equal to $\bar{h}$: this will serve as the reference time, against which all other times are measured. As illustrated in Figure 2.6, let $f(x) \sim \mathcal{N}(0, \sigma^2)$ be the PDF of $X_i^N - X_i^0$ (i.e. $\sigma = D\sqrt{\Delta t_{\mathrm{BD}}}$), and suppose that cars $i-1$ and $i+1$ are both a distance l_0 away from the front and rear of car i respectively (i.e. the movement of car i is limited to an interval of length $2l_0 = \bar{h}$). Let $a \in [0,1]$ be such that $x = al_0$, and $b > 0$ be such that $\sigma = bl_0$.

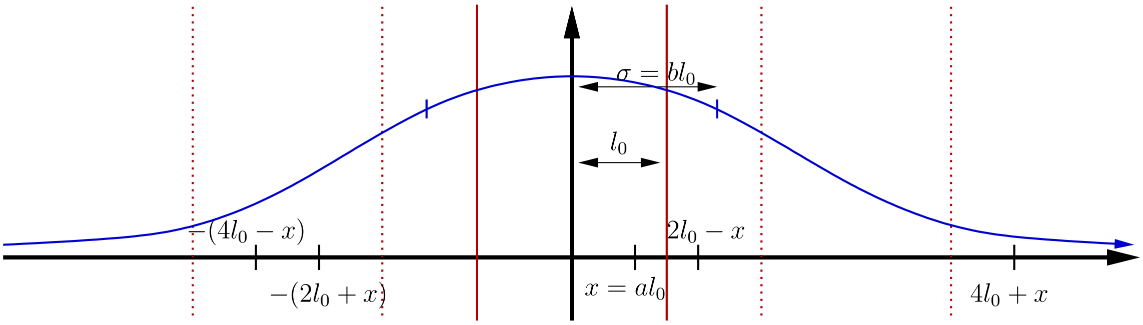


Figure 2.6: Illustration of the PDF of the expected, total movement of an unrestricted car undergoing Brownian motion. The solid, red lines represent the restriction on the movement on the car, and are a distance $2l_0$ apart. The dotted, red lines lie at multiples of $2l_0$ away from the centre. The ticks on the abscissa mark the points on the curve which will lie on top of x after the Gaussian curve is folded back on itself, as shown in Figure 2.7.

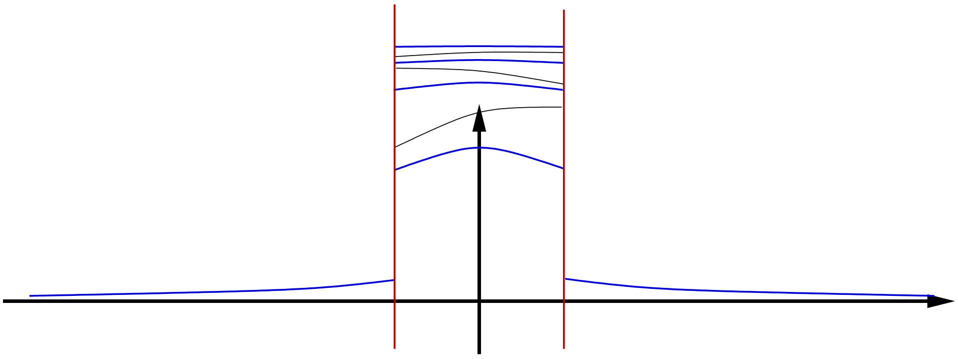


Figure 2.7: Illustration of how a Gaussian PDF that is "reflected" on the segment walls (neighbouring cars) effectively turns into a uniform PDF on that segment. Think of the tails of the distribution as being folded inwards and put in layers on top of each other.

The restriction on the movement of car i means that the PDF of $X_i^N - X_i^0$ becomes the result of "reflecting" the tails of the Gaussian tail back upon itself, as illustrated in Figure 2.7. Thus the PDF of $X_i^N - X_i^0$, $f(x)$, after restriction, is

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}}\left[\exp\left(-\frac{x^2}{2\sigma^2}\right) + \sum_{k=1}^{\infty}\left\{\exp\left(-\frac{(2kl_0+x)^2}{2\sigma^2}\right) + \exp\left(-\frac{(2kl_0-x)^2}{2\sigma^2}\right)\right\}\right]$$

for $|x| < l_0$, and $f(x) = 0$ otherwise. Substituting in $x = al_0$ and $\sigma = bl_0$, we get

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}}\left[\exp\left(-\frac{a^2}{2b^2}\right) + \sum_{k=1}^{\infty}\left\{\exp\left(-\frac{(2k+a)^2}{2b^2}\right) + \exp\left(-\frac{(2k-a)^2}{2b^2}\right)\right\}\right]$$

for $|x| < l_0$. This series does not, to my knowledge, have a closed-form expression, but is easily evaluated numerically, which we do in order to confirm that the "folded up" Gaussian PDF does indeed result in a uniform PDF on the restricted interval. Figure 2.8 shows plots of $f(x)/f(0) = f(al_0)/f(0)$ for a few different values of b. As can be seen, even for $b = 0.82$, i.e. when $\sigma \leq l_0$ and $T_{\text{BD}} \leq T_0$, $f(a)$ is practically uniform, with $f(l_0)/f(0) = 0.86$ being the minimal ratio for $0 \leq a \leq 1$. For higher values of b, the PDF is 1 for all $0 \leq a \leq 1$.

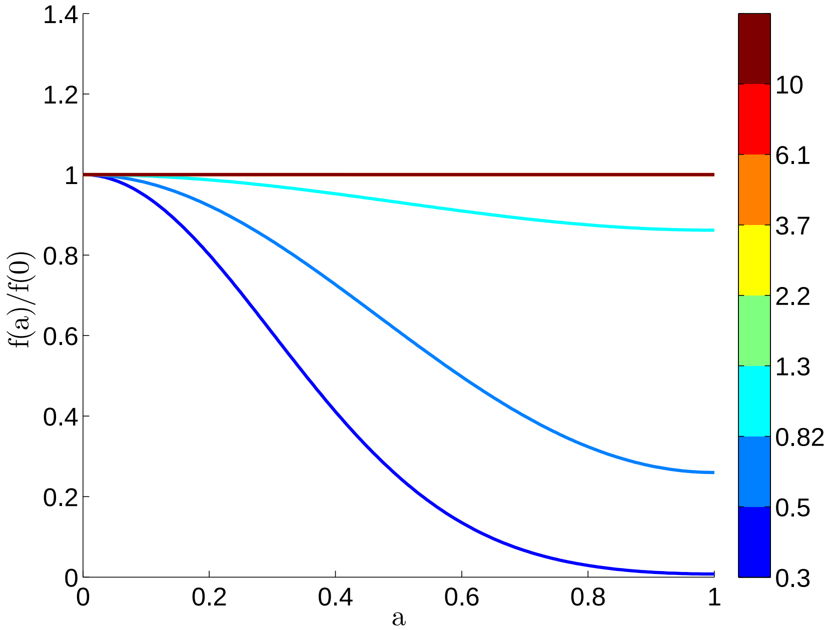


Figure 2.8: Plots of $f(x)/f(0) = f(al_0)/f(0)$ for different values of b (indicated by colour bar). Most of the curves are identically one, but are obscured by the curve for $b = 10$.

Additionally, the average position of the cars hardly changes over the course of one simulation (i.e. the drift is zero). This can be deduced by the central limit theorem, and is confirmed by numerical simulations: running BD with $T_{\text{BD}}/T_0 = 4$, i.e. $b = 2$, caused the average position to change by $\approx 0.01\bar{h}$. Therefore, although the borders fluctuate a little, the space in which a car can move is not only restricted over one iteration, but it is effectively restricted for the entire duration of the BD simulation.

This implies that convergence to a steady state of the car configuration should happen for T_{BD} not much larger than T_0, because by then $f(a)$ looks like a uniform distribution between the two neighbouring cars. Exactly how large T_{BD} has to be is difficult to deduce from this analysis, though, as there is still some uncertainty related to the effect of the fluctuations of the neighbouring cars. To get a better answer, one should study the convergence towards a steady state of the particle configuration, as is done is Section 2.7, but from Figure 2.8 one would expect that T_{BD} only needs to be on the order of T_0.

2.6.5 Selecting Δt_{BD}

Decreasing Δt_{BD} decreases the probability that two cars collide when moved, as the expected movement of a car scales with $\sqrt{\Delta t_{\mathrm{BD}}}$ (2.23). On the other hand, decreasing Δt_{BD} increases N, and the simulation will take more time (at least unless the collision handling procedure is very sensitive to the number of overlappings). Therefore, choosing the optimal Δt_{BD} involves compromising between these two effects.

Let $\bar{d} = D\sqrt{\Delta t_{\mathrm{BD}}}$ be the standard deviation of the PDF of the unrestricted movement over one iteration, $X_i^1 - X_i^0$, of car i. As Figure 2.9 shows, Ψ, the number of overlappings per iteration per car, increases at a rate of approximately $\bar{d}^{1.7} \propto (\sqrt{\Delta t_{\mathrm{BD}}})^{1.7} = (\Delta t_{\mathrm{BD}})^{0.85}$, which is more than a linear increase with $\bar{d}$.

When is Ψ sufficiently low to ensure that the BD simulations are sufficiently close to the continuous version of BD? The question can again be answered by studying the convergence of the particle configurations as $\Delta t_{\mathrm{BD}} \to 0$, as is done in Section 2.7; when $\Psi \leq 0.01$ it seems that decreasing Δt_{BD} has no further effect, and so $\Psi \leq 0.01$ was generally employed as the maximum allowed value of Ψ (the value of Ψ is always given). As can be deduced from Figure 2.9, this happens at around $\bar{d}/\bar{h} \approx 0.1$.

The big advantage of the sequential BD algorithm (Algorithm 5) is that the time step can be very large, since there are never any collisions. Setting $\Delta t_{\mathrm{BD}} = T_{\mathrm{BD}}$ will not work, though, probably because the non-simultaneity of the movements. It was found through simulations that one needed at least $N = 2$ in order for the car configuration to converge, but a few more were generally employed, as a safety margin.

Given that fewer iterations are needed for Algorithm 5 than for Algorithm 4, one would hope that Algorithm 5 could be run much faster. This is not the case though, as by its sequential nature, it is not vectorisable as a MATLAB function, and this makes it much slower, and therefore we persist in employing Algorithm 4. Presumably

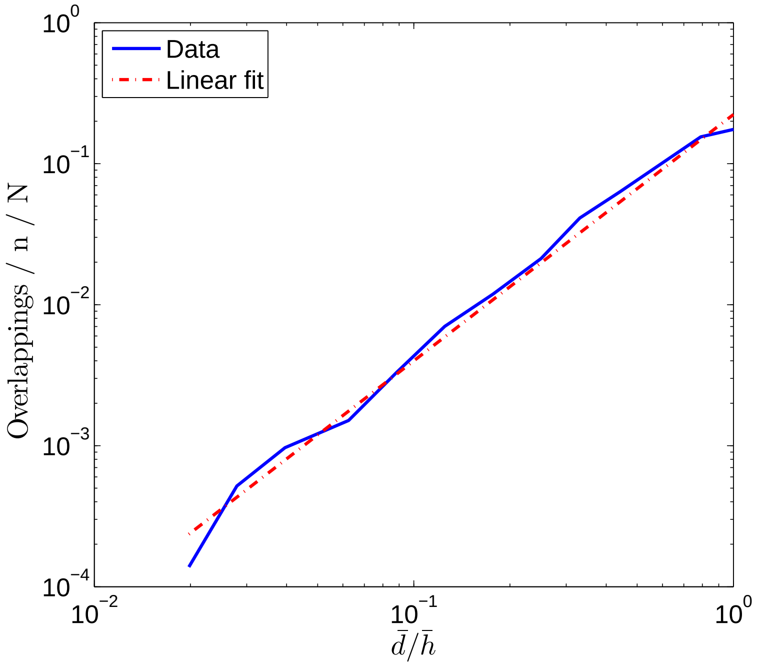


Figure 2.9: Log-log plot illustrating the increase in overlappings with increased time step Δt_{BD} for Algorithm 4. $\rho = 0.4$, slope of linear regression curve: 1.7.

it would have an advantage in other programming languages though.

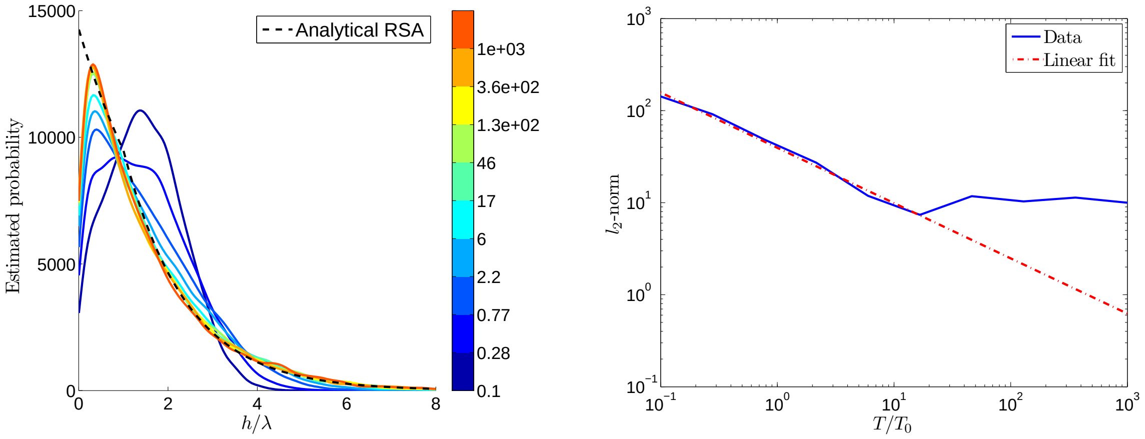
2.7 Comparison of BD and RSA

In order to rigorously assess the probability that two data samples originate from the same underlying probability law, one would have to perform a statistical test on the data samples from the simulations. This has not been applied in this thesis, though, because of time limitations and because visual comparison of the curves and their qualitative convergence tendencies were considered sufficient as a beginning.

In anticipation of the discussions below, it should be stated from the outset that despite the similarities of the curves, the conclusion drawn in one dimension is that RSA and BD to not lead to the same particle distributions.

Figure 2.10 gives plots of the gap density function for increasing T_{BD}. The simulations have been produced with each snapshot as the starting particle configuration for the next, with initial configuration being a uniform (equidistant) distribution, making the curves for small T_{BD} resemble a central spike. For higher values of T_{RSA}, this feature disappears, and the curves approach the analytical RSA curve.

Figure 2.10b gives a log-log plot of the restricted l_2 norm, of the difference between the RSA and BD curves. The slope is approximately -0.6, and so it seems that the



(a) Plots of $\hat{G}(h,t)$. The colour bar indicates the ratio T_{BD}/T_0.

(b) Plot of $\|G(h,T_{\mathrm{BD}}) - \hat{G}(h,T_{\mathrm{BD}})\|_{2,\mathrm{res}}$, with $h_I = \arg\max_h \hat{G}(h,T_{\mathrm{BD}})$ for the lowest value of T_{BD}. Logarithmic axes. Slope of the linear fit is about -0.6.

Figure 2.10: Convergence studies of BD simulations of increasing diffusion time T_{BD}. The system starts off from an equidistant distribution. $n = 1 \times 10^4$, $\rho = 0.4$. The value of Δt_{RSA} is such that $\Psi < 0.01$.

difference decreases at the rate of $T_{\mathrm{BD}}^{-0.6} \approx 1/\sqrt{T_{\mathrm{BD}}}$, which is proportional to the increase in the expected total distance of an unrestricted car undergoing Brownian motion (2.23).

The decrease in the restricted l_2 norm of the difference stops at a certain point, when $T_{\mathrm{BD}}/T_0 \approx 10$. At first we thought to this was a consequence of the limited accuracy given the size of the system. That is, if n got larger, the error would get smaller. However, we now consider it a consequence of the actual difference between RSA and BD distributions, as will be more apparent in Sections 2.8 and 2.9.

Figure 2.11 takes a slightly different approach from the simulations that produced the plots of Figure 2.10. Instead of starting off from an equidistant configuration, the particles here start off from a RSA configuration. This "wastes" less time, as Figure 2.10a clearly shows that the stable regime of the BD system possesses particle configurations that are (at least) close to the RSA distribution.

Skipping the transitory regime allows the simulations of Figure 2.11 to employ a smaller T_{BD} and hence also a smaller time step, and also more particles, without increasing the simulation time. This should reduce the inaccuracies, and increase our confidence that the curves are close to what would be obtained if $T_{\mathrm{BD}} \to 0$ and $\Delta t_{\mathrm{BD}} \to 0$. Another BD curve, labelled "Final BD" has also been produced, using extra small time steps. The simulations for Figure 2.11 take about 10 hours on a

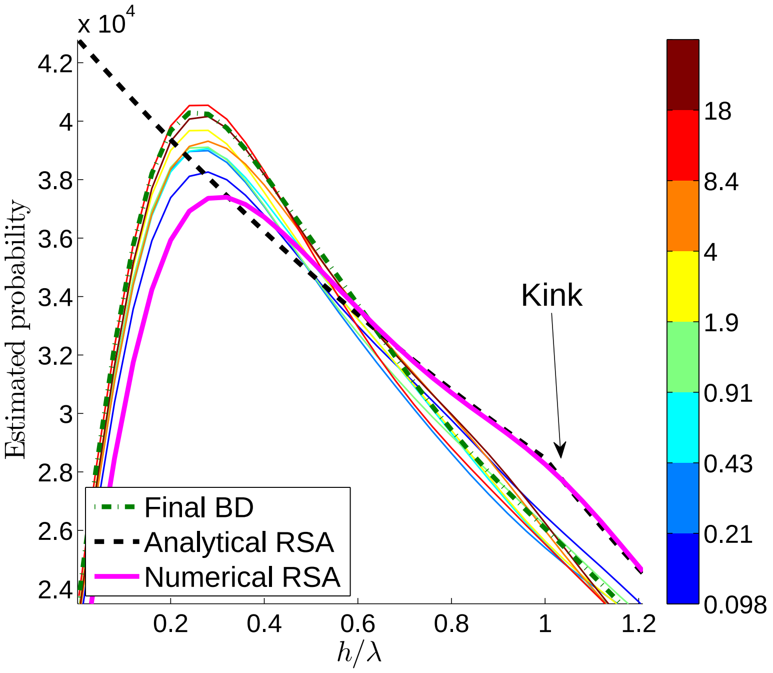


Figure 2.11: Convergence studies for BD simulations of increasing diffusion time T_{BD}. The system starts off from an RSA distribution. The value of Δt_{BD} is such that $\Psi < 0.01$ for all of the BD curves, except the "Final BD" curve where $\Psi \approx 0.0009$ and $T_{\mathrm{BD}}/T_0 = 22$. $n = 3 \times 10^4$, $\rho = 0.4$.

modern laptop.

What Figure 2.11 shows under scrutiny is that RSA and BD do in fact not give equal particle configurations in one dimension. In order to draw this conclusion one must recall the discussion from Section 2.4 about accuracy relating to system size n, and consider the following. The system size ($n = 3 \times 10^4$) is large enough that the numerical RSA curve on Figure 2.11 is very accurate; it follows closely the analytical RSA curve across the kink and up to the peak, at which point the effect of density estimation bends the curve downwards. Hence one would expect the BD curves to stay very close to these curves if they originate from the same underlying PDF. They does not do this, as in fact it seems that it converges[11] to a particle configuration whose gap density function lies slightly above the RSA gap density for $h/\lambda \leq 0.6$ and slightly below for $h/\lambda \geq 0.6$. From this we can carefully begin to draw the conclusion that the particle configurations of BD and RSA are not, in general, equivalent, although the next two sections will make this more evident.

[11] Converges both in terms of increasing T_{BD} and also for $\Delta t_{BD} \to 0$, given the "Final BD" curve.

2.8 Polydispersity

Polydispersity refers to having a mixture of different kinds of particles. In one dimension, though, this variation is limited to variations in the lengths of the cars. For the BD procedures that were implemented in this project, in one dimension, this has no effect on the distribution, as is discussed in Section 2.11. For RSA the impact of polydispersity is not so obvious, and numerical simulations are needed in order to investigate it. The way we implemented polydispersity in one dimension is by sampling the car lengths from a given PDF that is composed by centring a Gaussian curve at some value, and reflecting the bits of its tail that are negative back onto $[0, \infty)$.

Figure 2.12 plots curves of the gap density function for BD and RSA, magnifying the region of small h. For polydisperse RSA there exist two general versions of RSA, namely *competitive* and *non-competitive*. The difference between them is that non-competitive RSA does not change the particle type if an insertion fails, but keeps to insert the same particle until it succeeds, whereas competitive RSA draws a new particle for each insertion attempt. It is not obvious how to achieve the target density, ρ_0, in competitive RSA, since the car lengths do not necessarily add up to $\rho_0 L$. In non-competitive RSA, on the other hand, the car lengths can all be chosen ahead of the insertion phase, and scaled correctly. As is seen in Figure 2.12, the BD gap density curve lies somewhat above and below the other curves for $h < 0.6\lambda$ and $h > 0.6\lambda$.

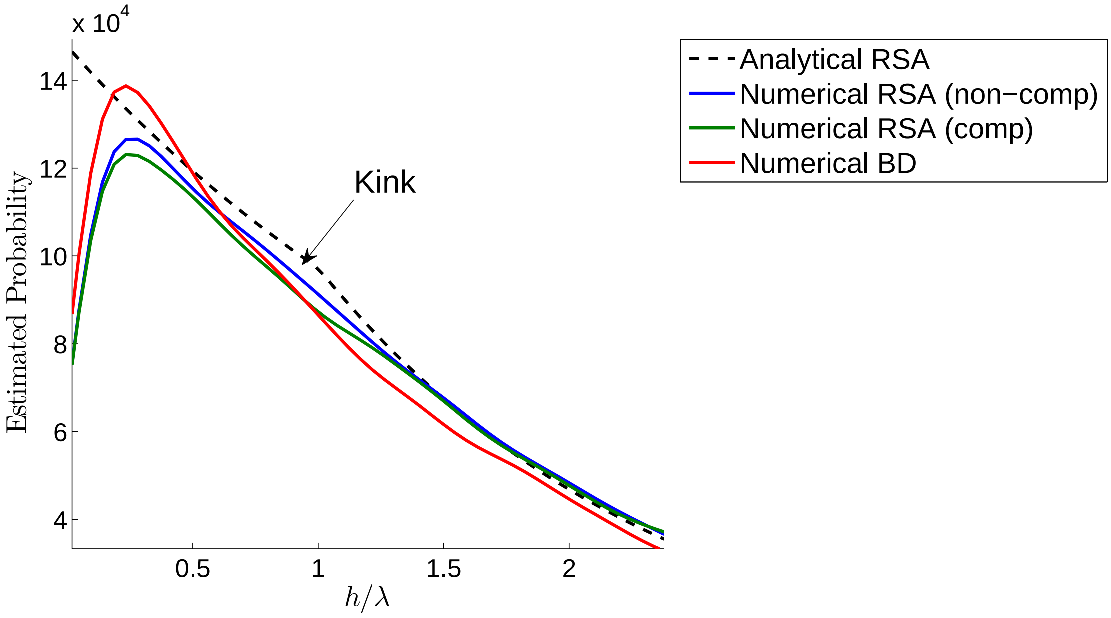


Figure 2.12: Plot of gap densities for polydisperse systems. The analytical RSA curve is for a fixed car length. $n = 1 \times 10^5$, $\rho = 0.4$.

Unlike fixed-size numerical RSA curves, as observed in previous Figures (e.g. Figure 2.5b), neither of the two numerical, polydisperse RSA curves in Figure 2.12 follows the kink at $h = \lambda$ of the analytical RSA curve[12]. This ought to be expected, as $h = \lambda$ now only represents the *mean* of the car lengths, and not their actual size.

Again however, the curves all lie sufficiently close to each other that one could easily be led to think that they follow the same underlying probability law. This, however, would be contrary to what one may expect from polydisperse RSA, and suggests that one should be careful to conclude that the spacial car configurations, although leading to *similar* gap densities, are not the same.

2.9 The influence of crowdedness

As mentioned, this project aims to experiment with a volumetric density of about 0.3 – 0.4. Therefore, although it was well-known to me (through numerical experimentation) that changing ρ (while keeping n constant) had a significant impact on the shape of $G(h,t)$, experimentation with other values of ρ was limited, as extreme values of ρ (e.g. 0.01 or 0.73) were regarded mostly as being outside of the scope of the project. With hindsight, this now seems to have been poor reasoning, or a rather a lack thereof, because extreme values of ρ allow for much clearer comparisons, and finally allows us to confidently draw the conclusion that BD and RSA do not give rise to equivalent particle configurations.

Figure 2.13 shows plots of the gap density function for a very high value of ρ, near the jamming limit of about 0.7475 [14]. In this case, the kink at $h = \lambda$ is much more pronounced than what it is for $\rho \approx 0.3$. The numerical RSA curves fit tightly too the analytical one, kink included. In this case, however, there is no doubt that BD result in different car configurations than RSA, as the BD curve has no tendency towards a kink, but is very smooth, and it also peaks well above the numerical RSA curve.

[12] Note that the analytical RSA curve, from (2.9), is for fixed-size cars. One-dimensional, polydisperse RSA is studied analytically in [6], but due to time limitations, this was not implemented here.

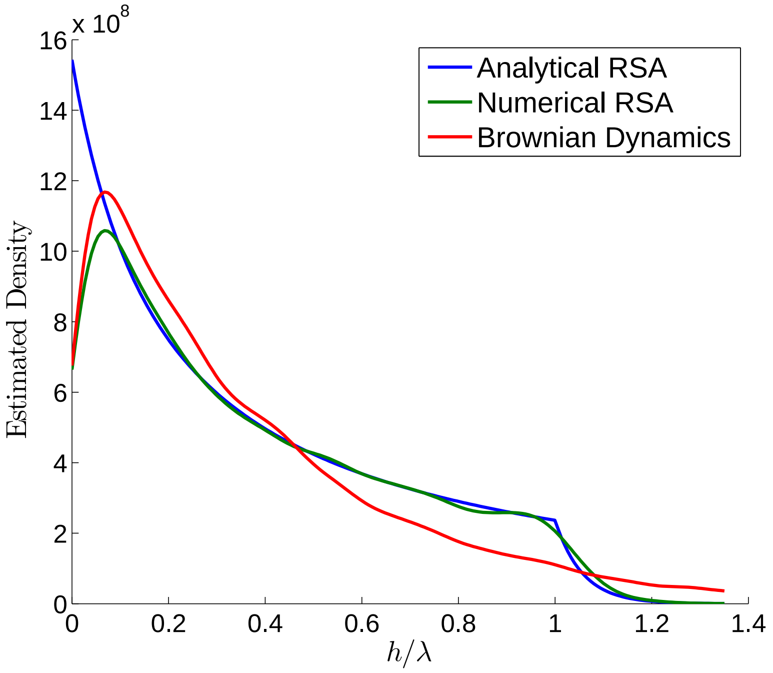
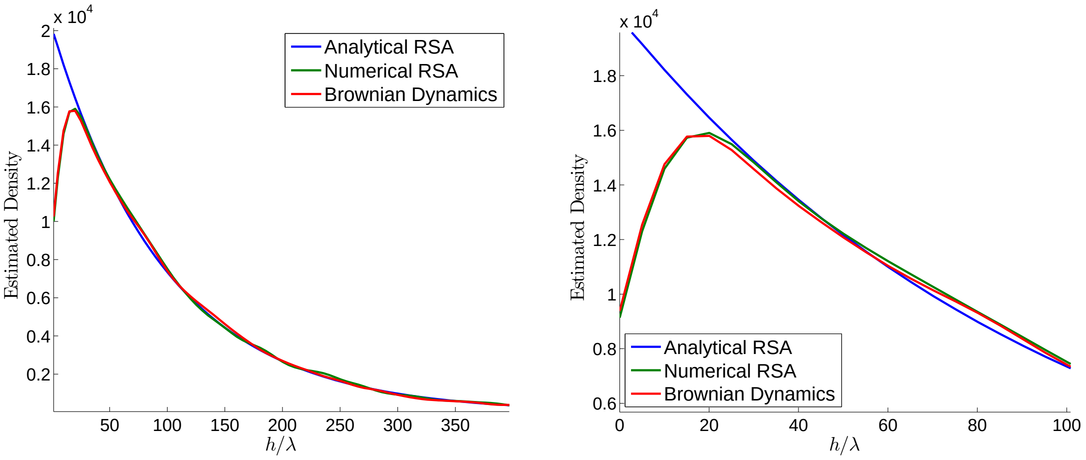


Figure 2.13: Plots of $\hat{G}(h,t)$ for $\rho = 0.73$ and $n = 2 \times 10^4$. $T_{\mathrm{BD}}/T_0 = 9$, Δt_{BD} such that $\Psi = 0.017$.

The other extreme case, $\rho \ll 1$, plotted in Figure 2.14, yields the opposite result – the curves are now virtually indistinguishable. Note that as $\lambda \to 0$, $\lambda/\bar{h} \to 0$, and consequently the range $h < \lambda$, practically disappears from the plots of $G(h,t)$[13]

[13] Magnifying the region just around $h = \lambda$, the kink is still discernible, but only just.



(a)

(b) Magnification of upper right part of Figure 2.14a.

Figure 2.14: Plots of $\hat{G}(h,t)$ for $\rho = 0.01$ and $n = 2 \times 10^4$. $T_{\text{RSA}}/T_0 = 9$, Δt_{BD} such that $\Psi = 0.018$.

These results leads us to conclude that in the limit of $\rho \to 0$ the BD and RSA car configurations are theoretically the same (for $T_{\text{BD}} \to \infty$), while this is clearly not the case for $\rho \approx 0.74$.

2.10 Further modelling and implementation issues

Over the course of this project, many choices were made concerning the modelling and implementation of the experiments. Some of these have a relatively small impact on the results, while others, like the density ρ, have considerable effect.

For example, for generality, and in order to be able to concatenate test lines (although this turned out to be of little relevance), functionality to use different test line lengths L was implemented. However, λ is proportional to L (2.19), and $G(h,t)$ depends on λ. Therefore, if one wants to be able to compare simulations with different L, one has to rescale h and $G(h,t)$ in order that the curves from the various simulations correspond (this is of course only possible since these quantities scale linearly). Also, if the simulation estimate of T_{RSA} supplied to the function for calculating the analytical expression for $G(h,t)$ (see Section 2.1.1) is to be accurate, it is necessary to rescale k_a with L so that Δt_{RSA} is correct.

Issues such as the one above make it apparent that conducting the simulations would be a lot easier if as many parameters as possible are encapsulated, and this is why the simulations were implemented object-orientedly. Otherwise the complexity of comparing many different simulation experiments is much more daunting.

2.10.1 Boundary conditions

Periodic boundary conditions (BCs) were used predominantly for most of the experiments, and is employed throughout this report. This means that particles going across one edge of the domain enters on the opposite edge. But we also experimented with "closed" BCs, meaning that particles are reflected back from the domain walls.

Although seemingly trivial, in one dimension this was quite problematic to implement. First, writing algorithms for both methods must take into account the fact that there are n gaps when periodic BCs are used, whereas there are $n+1$ when closed BCs are used. Secondly, if periodic BCs are employed, car number 1 should always be to the right of (i.e. non-overlapping, and in the right order with) car number n, and vice-versa. If closed BCs are used, though, car number 1 should always be to the right of the left edge of the domain, and car number n should always to to the left of the right edge of the domain. This ambiguity causes problems for functions that move the cars, for example. A solution that proved effective is to implement low-level functions into the object-oriented classes that take care of the ambiguity in accordance with the BCs used by that simulation experiment. Hence, for example, the BD algorithms could be implemented without worrying about which BCs are used.

Although boundary conditions can have an impact on small systems, or systems where a drift term is present, they otherwise become negligible as $n \to \infty$.

2.10.2 Many small systems versus one large system

One important implementation issue is whether to use one large system of particles, or many initial copies of a small one (such that the number of cars used in the small system, times the number of copies of that system, equals the number of cars for the big system n). In terms of accuracy of the estimated gap densities, we found that this makes no difference, but that it does matter for speed.

As for the RSA procedure, we found that the small-systems approach beats the big-system approach by a speed factor varying somewhat irregularly between 3 to 30, for n between 1×10^2 and 1×10^4. For BD, however, we found the opposite to be true; when the number of copies and the number of cars in the small systems were both 1×10^2 (i.e. $n = 1 \times 10^4$), the big-system approach beat the small-systems approach by a factor of about 2.

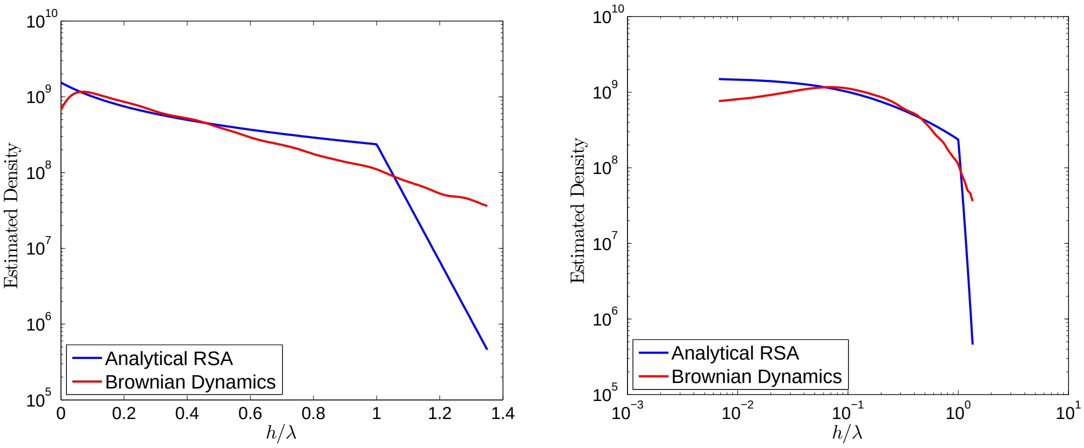
The reason why the big-system approach was faster for BD seems to be due to the additional overhead involved in calling all of the subroutines many more times for the small-system approach.

However, the reason why RSA is faster for small-systems approach lies in the nature of RSA. Assume for the sake of explanation that there are 10 small systems corresponding to one big system, and that the process in both cases has reached a significant density ρ. Let $\Delta\rho_{\text{small}} = \lambda_{\text{small}}/L$ be the resulting change in density from the insertion of one car to the small systems. Then $\Delta\rho_{\text{big}} = \Delta\rho_{\text{small}}/10$. For the 10 small systems, one insertion is necessary to increase ρ to $\rho + \Delta\rho_{\text{small}}$. For the big system, 10 insertions are necessary. The difference is that the insertions into the big systems will take place at progressive densities: ρ, $\rho{+}\Delta\rho_{\text{big}}$, $\cdots$, $\rho{+}9\Delta\rho_{\text{big}}$, whereas the insertion into the small systems take place only at a density of ρ. This means that the rates of insertion will progressively decrease for the big system, whereas the rate of insertion for the small systems is fixed. Therefore, for crowded environments, the RSA procedure takes a lot longer for the big system than for the 10 small ones.

Overall, what tended to take the longest time were the BD simulations. Therefore, and also because it is comparatively slightly more complex to run many small experiments instead of one big one, the big-system approach was the prevailing approach in one dimension.

2.11 Explaining the numerical results

The general trend of the gap density plots is that of a smooth decrease from small values of h to bigger values of h. And as long as ρ is not to close to the jamming limit, the BD and RSA curves lie very close. Consider however some alternatives; a spiked gap density distribution would be exceedingly unlikely, corresponding to a uniform (equidistant) configuration; also, large gaps are unlikely, as both RSA and BD have a propensity to fill in such gaps, and so anything resembling a uniform PDF for the gap density is also highly unlikely.



(a) Semilog-y plot.

(b) Log-log plot.

Figure 2.15: Semilog-y and log-log plots of the analytical RSA gap density function and estimates of the BD gap density function for $\rho = 0.73$.

As can be deduced straight from the expression for $G(h, t)$ (2.9), for $h > \lambda$, $G(h, t)$ is exponential in h. Moreover, Figure 2.15 attempts to shed some light on the behaviour of $G(h, t)$ for all values of h. In order to accentuate the characteristics of the gap density function for RSA, ρ has been near the jamming limit. As can be seen, the (first part of the) analytic RSA curve bends upwards in the semilog-y plot, and downwards in the log-log plot, so for $h < \lambda$, $G(h, t)$ could be said to be something in between an exponential function and a power of h. On the other hand, as shown in the semilog-y plot, the BD curve is clearly exponential in nature.

Explaining this with some rigour gets a lot easier once the following dual "point of view" is taken into account[14]. When it comes to BD in one dimension, the car lengths are irrelevant and can all be amassed at one end of the test line, whilst the gap lengths are maintained by point-cars (i.e. cars with no length) on the line. This dual point of view is illustrated in Figure 2.16, and was already implicitly used in Section 2.6.4. The justification is simple; the movement of the particles as defined through Algorithm 2 is not influenced whatsoever by the length of the cars, except in establishing the lengths of the gaps between the cars[15].

[14] The idea of this dual point of view came from private correspondence with Prof. T. Aste of Kent University [2], although the following reasoning is my own.

[15] The dual point of view can also be applied to the RSA process, but only for fixed-size RSA, and it also requires some more sophistication.

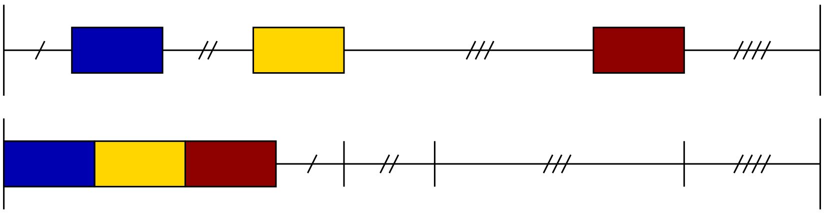


Figure 2.16: An equivalent way of looking at lengths for one-dimensional BD.

Therefore, one can conclude that the gap length distribution established by BD should arise from minimising the entropy of n identical points on a line. Since there is no reason this should give rise to discontinuities of any sort the BD curves should not be expected to mould to the kink that is observed in RSA curves.

In order to explicitly derive the exponential nature of the BD curves, one could argue as follows[16]. The way the reflective collision handling of Algorithm 4 works, cars are bounced apart a distance equal to that with which they previously overlapped, while their mean position is maintained. This, from the dual point of view, looks just as if the cars had actually passed by each other. In other words, the BD simulations from the dual point of view is just like the free, *unrestricted* BD of n point-cars on a reduced line of length $(1-\rho)L$. This implies that the position of each point-car at T_{BD} follows a uniform PDF on the reduced line (as long as T_{BD} is sufficiently large, as was discussed in Section 2.6.4), and this is equivalent to the way RSA gaps of length $h \geq \lambda$ are formed[17]. Hence the same reasoning that follows from (2.5) can be applied to BD, and this gives an exponential curve.

Note also that the revelation of the dual point of view of with unrestricted point-cars, seems to imply for BD that the ordering of the gaps is independent of their lengths, as was investigated numerically in Section 2.3.1.

[16] This is an alternative method to that of minimising the entropy, which was not attempted in this thesis.

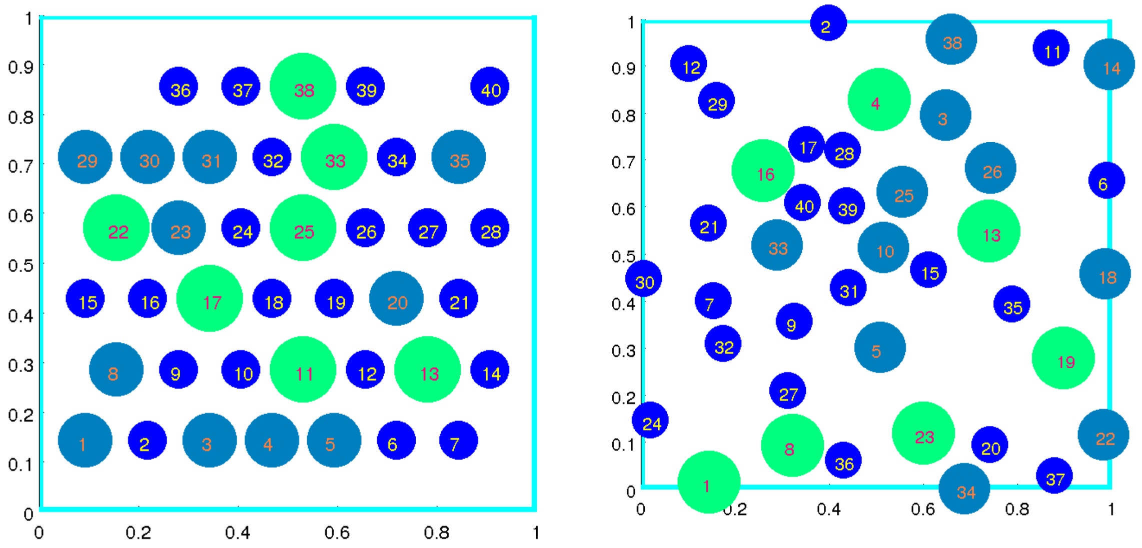
[17] All RSA gaps of length $h > \lambda$ have been created from insertion at a point drawn from a uniform PDF. Unlike gaps of length $h < \lambda$, however, they have also “faced” destruction by insertion of a car at a point also drawn from a uniform PDF.

Chapter 3

Two-dimensional RSA and BD

3.1 Conducting numerical experiments in two dimensions

With the conclusions of one dimension in mind, we now make the jump into two dimensions. Less discussions are given in two dimensions, as many of the algorithms employed are similar in one and two dimensions, and so many of the lessons learnt from one dimension are applicable in two dimensions too. As Figure 3.1 shows, the



(a) Uniform packing

(b) RSA packing

Figure 3.1: Demonstration of the visualisation of the system of particles on a domain that was used for debugging. The particles are labelled, to help in tracking them, as there is no natural ordering in two dimensions.

two-dimensional particles that we use are all discs, and polydispersity is modelled employing different discrete size *types*. Furthermore, only non-competitive RSA is studied in two dimensions.

The particles involved in each simulation experiment, are specified through an array $[s_1,\ s_2,\ \cdots s_q]$ specifying the relative sizes (areas) of the particles, and another $[p_1,\ p_2,\ \cdots p_q]$ with the relative proportion in area that they occupied. "Relative" here means that these arrays need not be normalised, and q is the number of different size types to be used. A certain volumetric density ρ and a total number of particles n has to be specified, as well as the width W and height H of the domain. With these numbers one can deduce the different absolute sizes $[S_1,\ S_2,\ \cdots S_q]$ and numbers $[n_1,\ n_2,\ \cdots n_q]$ of particles of given sizes to be used in the experiment, as they should satisfy the following system of equations. Their sum should add up to n, i.e.

$$\sum_{i=1}^{q} n_i = n. \tag{3.1}$$

(We again break with the convention of this thesis here by using i as an index for size type, instead of as a particle index.) The sum of their areas should add up to the total occupied area,

$$\sum_{i=1}^{q} n_i S_i = \rho W H. \tag{3.2}$$

The absolute areas of the particles are specified through $[s_1,\ s_2,\ \cdots s_q]$, i.e.

$$\frac{S_1}{S_i} = \frac{s_1}{s_i} \qquad \forall i \in 2, \cdots, k, \tag{3.3}$$

and the proportion of the total area that each particle type occupies should satisfy

$$\frac{n_1 S_1}{n_i S_i} = \frac{p_1}{p_i} \qquad \forall i \in 2, \cdots, k. \tag{3.4}$$

This gives $2k$ equations for the same number of unknowns, and solving them results in

$$n_i = n\rho \frac{\frac{\prod_{j=1}^{q} s_j}{s_i}}{\sum_{j=1}^{q} p_j \frac{\prod_{j=1}^{q} s_j}{s_i}}, \tag{3.5}$$

where the formula has been expressed in such that it is easy to vectorise in MATLAB. The expression for S_i is similar:

$$S_i = \frac{\rho W H s_i}{\sum_{j=1}^{q} n_j \frac{s_j}{s_i}}. \tag{3.6}$$

3.1.1 Measuring distances

Given that there is no natural order of particles in two-dimensions, their spacial distribution is measured by nearest-neighbour distances, $d_{\min}(i)$ for all $1 \leq i \leq n$. This gives rise to a nearest-neighbour density function analogous to the gap density function used in one dimension, and we will therefore denote and refer to the nearest-neighbour density function in the same way, i.e. as the gap density function $G(h,t)$. Formally, $G(h,t)\,\mathrm{d}h$ denotes the mean number of particles, per unit area, whose nearest neighbour lies within a distance between h and $h + \mathrm{d}h$ to those particles. Note that the distances are measured not between centres of discs, but as the shortest distance between any point in one disc to any point in another. Additionally, one could divide $G(h,t)$ by $h+r$, or $2\pi(h+r)$, where r is the particle radius, in order to compensate for the effect of the growth of the area of a ring of inner radius $h+r$, but this is not employed here. Given that no analytical solution is known in two dimensions for the RSA procedure [14], only numerical curves of the gap density function are plotted.

In order to determine the nearest-neighbour distances, we first determine the global pair-wise distances, $d(i,j)$, for all $1 \leq i,j \leq n$. This information can be assembled in a symmetric matrix that we will refer to as the *global distance* matrix, D_{glob}, such that element (i,j) of D_{glob} is $d(i,j)$. The nearest-neighbour distance is then found by finding the minimal element, outside of the diagonal, of each row (or column) of D_{glob}, i.e.

$$d_{\min}(i) = \min_{i \neq j} \; d(i,j).$$

The aforementioned operations are all vectorisable in MATLAB, and so this method is relatively fast, but D_{glob} does require a lot of memory; for n particles, $D_{\text{glob}} \in \mathbb{R}^{n \times n}$, and consequently require $8n^2$ bytes of contiguous memory[1]. The alternative would be to loop over all particles, and establish $d_{\min}(i)$ one-by-one, which would be slower, but requires less memory. Therefore, with system sizes up to $n = 1 \times 10^4$, which is the biggest system size that was used for the two-dimensional experiments, it was necessary to use a workstation with more memory than the aforementioned modern laptop (see Appendix B for computer specifications). As an example of the time of the simulations, Figure 3.6 took 105 hours to produce on such a workstation.

[1] This assumes that the numbers are `floats` in MATLAB, and that one does not take advantage of the symmetry of D_{glob}.

3.1.2 Further algorithm and modelling issues

Section 2.10.2 discusses the advantages and disadvantages of carrying out one big simulation experiment versus using many copies of a small experiment. In one dimension, the time taken for BD simulations is a little less for the big-system approach than for the small-systems approach. In two dimensions, this would probably not hold true however, as the number of operations involved in checking for overlappings scales with n^2 (n comparisons for n particles each iteration). This is also true for RSA. Therefore I expect that carrying out many smaller experiments would have been a better approach in two dimensions, as this would also not require such large amounts of memory, although this approach was not implemented, due to the limited time available.

As for the BD algorithms, and collision handling, the lessons learnt from one-dimensional experiments are very helpful. Again, a certain number of unresolved overlappings per iteration is considered acceptable, (see Section 3.3 for convergence studies) as the computational and programming costs of fully resolving them are too big. The collision handling in two dimensions is similar to that in one dimension in that each particle is bounced back by a distance equal to that with which they overlapped. In two dimensions, though, this operation also includes the calculation of the angles for this reflection.

The collision checking procedure in two dimensions take much longer time than in one dimension, as it has to check for overlappings between all of the particles. There is however a noteworthy trick which makes the process faster. To begin with, one can measure the L_1 distance between the particles. Only the ones that are found to be within a distance of the sum of their radii then need to be checked in the L_2 norm , which requires more computer flops to calculate, for collisions[2].

Recall that the big advantage of the sequential algorithm in one dimension is that it allows for huge time steps. In one dimension the bounds on the movement of a given car is easily found by looking at the two neighbouring cars. This does not directly translate into two dimensions, however, and therefore even the two-dimensional inheritor of Algorithm 5 needs some form of collision handling, and the size of the time step is therefore limited. Still, since only one particle is moved at a time, the sequential method leads to fewer unresolved collisions, and a lower value of Ψ, as no two particles are be bounced “simultaneously”.

On the other hand, checking for overlaps takes place n times per iteration for the sequential algorithm, while the simultaneous algorithm, inheritor of Algorithm 4, only

[2] This idea I owe to Maria Bruna, D.Phil. student at the Oxford Centre for Collaborative Applied Mathematics (OCCAM), whose work is also within BD.

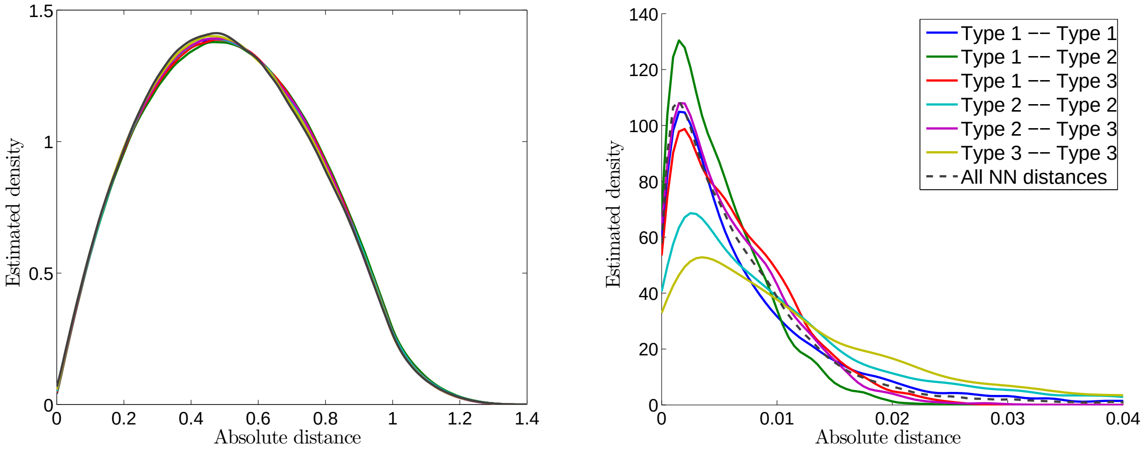
does it once per iteration. Moreover, the reduction in Ψ for the sequential algorithm is very slight, as is revealed by the following asymptotic analysis: for small time steps, the probability of that any given overlapping takes place is very small. Therefore, the probability that two neighbouring overlappings, which after the bounce-backs give rise to an unresolved overlapping, take place, is approximately the square of the probability of the first probability. Therefore, given the increase in computational cost of the sequential algorithm, the simultaneous algorithm was again generally preferred, and it is this that is used to produce the results in this thesis for two dimensions.

In one dimension, the distances are all considered relative to the car length (average if it varied) or the mean gap length. There is also a fixed relation between these two, namely $\rho\bar{h} = (1-\rho)\lambda$, which justifies the car length as a reference distance. In two dimensions there is no such obvious choice of reference distance. Although the average radius is analogous of the car length, due to the involvement of square powers in two dimensions, the average radius does not have such an obvious relationship with the average nearest-neighbour gap length $\bar{d}$, nor with the square root of the total unoccupied volume $\sqrt{(1-\rho)WH}$, both of which could also be used as a reference distance. Nevertheless, we use the average radius, $\bar{r}$, as the reference distance because it is simple, and important in establishing the time step; in determining Δt_{BD}, one should consider the average nearest-neighbour distance and the square root of the total unoccupied volume (as was discussed in one dimension in Section 2.6.5); but one also has to make sure that Δt_{BD} is small enough that the expected displacement each iteration is smaller than the radius of a particle. Otherwise one risks having particles that, over the course of one iteration, fully pass by each other without a collision even being detected. Therefore, both Δt_{BD} and T_{BD} are set with respect to T_0, the time at which the expected displacement is equal to the reference length, $\bar{r}$.

3.2 Polydispersity

One of the aims of this project is to investigate the effects of polydispersity, as in the case of having, say, three different types of macromolecules making up the crowded environment. The question is whether particles of different types have different contact probabilities.

Figures 3.2a and 3.2b show some plots to this purpose. As mentioned, in order to establish the nearest-neighbour distances, D_{glob} was first established, and it is density estimates of $d(i,j)$, for all $1 \leq i,j \leq n$, that are plotted in Figure 3.2a. Furthermore, Figure 3.2a assimilates the data into different categories, according to the types of the particles. For instance, the “Type 1 – Type 2” curve can be thought of as counting



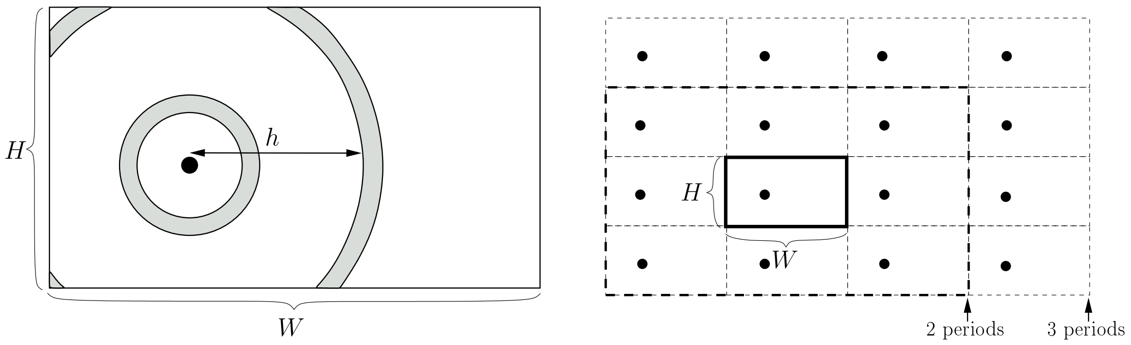
(a) Global distances

(b) Nearest-neighbour distances

Figure 3.2: Distribution established by RSA. Plot of density estimates of $G(h,t)$ for individual types. $n = 1 \times 10^4$, $\rho = 0.3$. $[s_1,\ s_2,\ s_3] = [1,2,3]$, $[p_1,\ p_2,\ p_3] = [1,1,1]$. Size of domain: 1×1.

the average number of particles of type 2 that are inside a ring of a given radius and width, centred at a particle of type 1. This is illustrated in Figure 3.3a. Although the particle configuration studied in Figure 3.2a was established by RSA, the fact is that even an equidistant distribution would induce curves that are overwhelmingly similar to the ones in Figure 3.2a (as was numerically verified), since both RSA and equidistant distributions have the same, uniformly distributed, volumetric particle density. There would be slight differences for small values of h, but these would tend to be smoothed out by the kernel density estimation. Hence we see the importance of employing the nearest-neighbour distances as measures of the spacial configuration of the systems.

It should be noted that, as illustrated in Figure 3.3a, the distances are measured as if the system uses closed BCs, despite the fact that the simulations uses periodic BCs. If true periodic boundary condition measurements are performed (i.e. where the periodicity is repeated infinitely, as illustrated in Figure 3.3b), h would not be bounded above (as it is now by $\sqrt{H^2 + W^2}$), and the curves on Figure 3.2a would be increasing linearly (given the linear relationship between circumference and radius). In order to measure nearest-neighbour distances assuming periodic BCs, one would have to measure nine times the amount of distances that what is currently done by D_{glob}, as illustrated in Figure 3.3b. Further periodicity is not necessary for nearest-neighbour measurements, because these would never be nearer than any of those nine-fold measurements. This increase is not insurmountable, but its effect on the



(a) Measurement of $d(i, j)$, and how non-periodic BCs affect it.

(b) Visualisation of repeating periodicity.

Figure 3.3: Illustration of the effect of boundary conditions upon distance measurements.

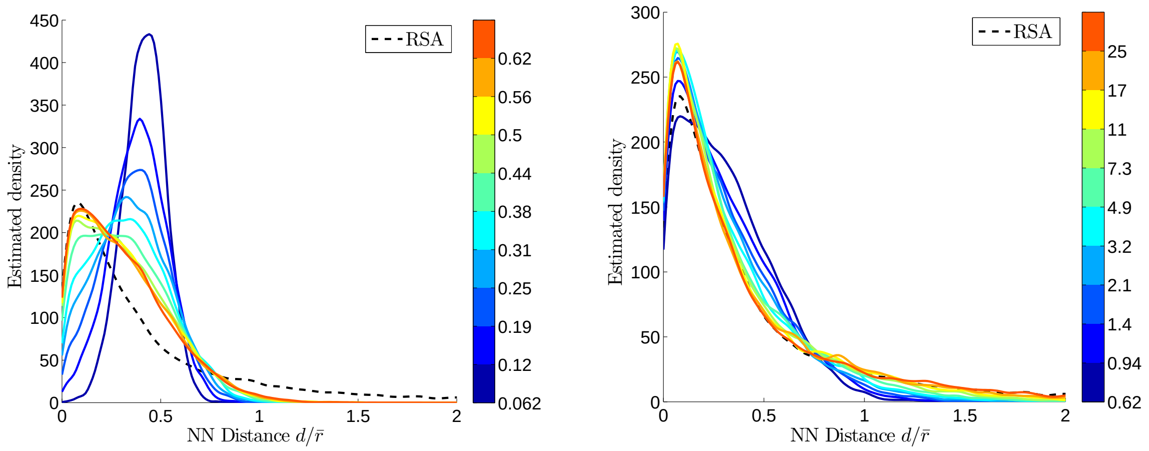
nearest-neighbour gap densities is negligible, since the number of particles on the rims is very small compared to those further inside the domain. Consequently, the distance measurements are done assuming closed BCs.

Figure 3.2b plots the nearest-neighbour distances, and although the curves in this case are not overlapping, that is not because different size types result in different gap densities. It is due to the fact that the individual densities are different, and this means that the density curves are more stretched out for the lower density types. Further simulations were carried out where only particles of type 1 were present, and in the same amount as in Figure 3.2b, and the resulting gap density curve was found to be equal (within expected stochastic variations) to the "Type 1 – Type 1" curve of Figure 3.2b. In other words, polydispersity, implemented as size difference, does not seem to have an effect on the particle configurations, although one could argue that we have not gone far enough in our investigations, and should try out more extreme differences in the sizes of the particles.

3.3 Comparison of BD and RSA

The way we compare and draw conclusions on the similarity of the particle configurations is the same as it was in one dimension. See Sections 2.3.1 and 2.7 for more details.

Figure 3.4 shows how the distribution converges from a equidistant distribution for increasing T_{BD}. In Figure 3.4b, the RSA curve is mostly obscured by the curves for the intermediate and higher values of T_{BD}, because the curves again lie relatively close to one another. But with the lessons drawn from one dimension, we know that we need to be careful on drawing the conclusions, and therefore Figure 3.4 does not provide enough details to conclude whether RSA and BD in two dimensions result in the same particle configurations.



(a) Small values of T_{BD}.

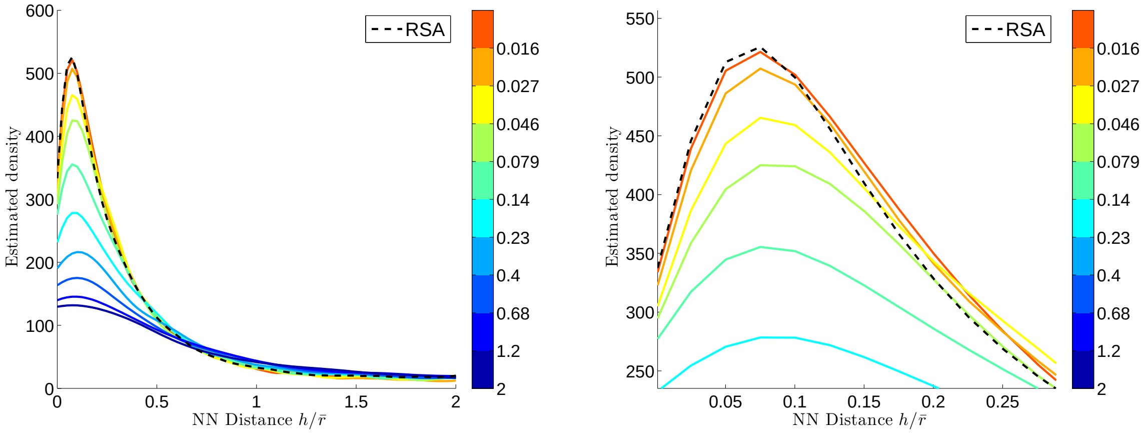
(b) Large values of T_{BD}.

Figure 3.4: Plots of $\hat{G}(h,t)$ for increasing T_{BD}. The BD simulations start out from a equidistant particle configuration. T_{BD}/T_0 indicated by the colour bar. Δt_{BD} such that Ψ is below 0.04 for all curves. $n = 1 \times 10^4, \rho = 0.3$. Only one type of particle size present in system.

As can be seen by the apparent convergence of the curves in Figure 3.4, $T_{\mathrm{BD}} \approx 5$, should be a sufficient time span in order for a BD system to reach a stable regime, and therefore Figures 3.5 and 3.6 use this values of T_{BD}. In order to do more in-depth studies on the matter of convergence, one has to study how the BD curves converge for $\Delta t_{\mathrm{BD}} \to 0$. This is provided by Figures 3.5 and 3.6.

Figure 3.5 seems to suggest the startling conclusion that the two-dimensional particle configurations of RSA and BD are equal. Startling, because this was shown not to be the case in one dimension. Note that the RSA curve in two dimensions, as can be seen on all of the figures of this section do not exhibit any form of kink at any point, as was the case for $h = \lambda$ in one dimension. This is a clear indication that the nature of the RSA process is qualitatively different in two dimensions, and this extra smoothness could be used to argue why one should expect the RSA and BD curves to lie closer in two dimensions than what they do in one dimension.

The curves in Figure 3.5 do seemingly converge, because the distance between them decreases for decreasing values of Δt_{BD}. However, all of the curves exhibit the trend of peaking with strictly increasing heights for decreasing Δt_{BD}. With the stochastic inaccuracies of the system ($n = 1 \times 10^4$), though, one would expect some oscillation around the stable distribution, if the curves have converged (as in Figure



(a)

(b) Magnification of upper-left part of Figure 3.5a

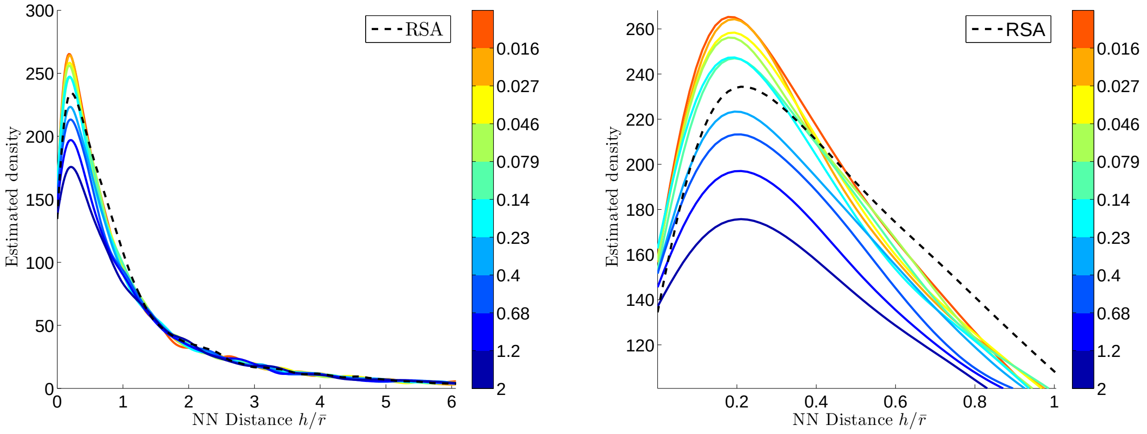
Figure 3.5: Plots of $\hat{G}(h,t)$ for decreasing Δt_{BD}. The BD simulations start out from a particle configuration established by RSA. $T_{\text{BD}}/T_0 = 5$, $\Delta t_{\text{BD}}/T_0$ indicated by the colour bar, although Δt_{BD} is such that Ψ is around 0.51 for $\Delta t_{\text{BD}}/T_0 = 2$ and decreases to $\Psi \approx 0.019$ for $\Delta t_{\text{BD}}/T_0 = 0.016$. $n = 1 \times 10^4, \rho = 0.5$. Only one type of particle size present in system.

2.11). Figure 3.5 does not exhibit such oscillation, and so it is not certain that the curves have actually converged.

Therefore more simulations experiments are needed, with even smaller values of Δt_{BD}. This is provided in Figure 3.6, where also a lower density has also been employed. This lower density (for a fixed n) means that T_0, the time at which the estimated displacement of an unrestricted particle is equal to $\bar{r}$, is also lower than it was for Figure 3.5. Hence, although the ratios $\Delta t_{\text{BD}}/T_0$ are the same for the two figures, the values of Δt_{BD} relative to the gap lengths is actually lower in Figure 3.6 than it is in Figure 3.5, and this induces fewer overlappings per iteration, i.e. lower values of Ψ, and more confidence in the simulation results.

The conclusion that can be drawn from Figure 3.6 contradicts that which was suggested by Figure 3.5. In view of the lower values of Ψ for Figure 3.6, and the lack of certainty that the curves of Figure 3.5 have converged, the conclusion seems to be that in two dimensions, again, RSA and BD do not generate the same particle configurations.

However, Figure 3.5 was produced using $\rho = 0.5$, whereas Figure 3.6 used $\rho = 0.3$. So one can not be entirely confident that this holds for all densities. The one-dimensional experiments suggest that the difference between RSA and BD is largest when ρ is near the jamming limit, though. And for $\rho \to 0$ (i.e. using point particles)



(a)

(b) Magnification of upper-left part of Figure 3.6a

Figure 3.6: Plot of $\hat{G}(h,t)$ for different values of Δt_{BD} for BD simulations that start out from a particle configuration established by RSA. $T_{\mathrm{BD}}/T_0 = 10$. $\Delta t_{\mathrm{BD}}/T_0$ indicated by the colour bar, although Δt_{BD} is such that $\Psi \approx 0.18$ for $\Delta t_{\mathrm{BD}}/T_0 = 2$ and decreases to $\Psi \approx 0.0024$ for $\Delta t_{\mathrm{BD}}/T_0 = 0.016$. $n = 1 \times 10^4, \rho = 0.3$. Only one type of particle size present in system.

the distributions are certainly equal[3]. Therefore it would be surprising if $\rho = 0.5$ actually leads to more similarity than $\rho = 0.3$ Unfortunately the simulations for Figure 3.5 took 24 hours on the workstation, and those of Figure 3.6 took 105 hours, and due to time limitations no further experiments could be pursued.

[3] This was verified numerically, but it is also relatively clear that RSA and BD are equivalent in theory for $\rho \to 0$.

Chapter 4

Modifying the RSA algorithm

The modification of the RSA algorithms can take several forms. One option is to do something similar to the collision handling methods of BD (see Section 2.6.1); instead of rejecting potentially overlapping insertions, one bounces the particle being inserted a distance of u ($u > 0$) times the overlap away from the already inserted particle. A second option is to manipulate the insertion probabilities on the available intervals. This is studied in [6], but with the objective of modelling some form of polymer adsorption, and not to approximate BD distributions, as we do here[1].

Both of these alterations were tried out in one dimension. Recall that the objective is to make the RSA gap density curve approximate the BD gap density curve. The first, employing the bounce-backs, is not very successful, as shown in Figure 4.1a. More values of u were tried out than those plotted in Figure 4.1a, all without notable success, and the main effect of varying u seems to be stretching the curve along the x-axis.

In order to describe the second method, again let $2l_0 = h - \lambda$ be the length the space available for insertion of a car in a given gap, and $x = al_0$, with $-1 \leq a \leq 1$. Denote $a_\lambda = (l_0 - \lambda)/l_0$ the value of a which corresponds to $x = l_0 - \lambda$. This is illustrated in Figure 4.2. Also, define $\chi(a)$ to be a scaled PDF denoting the probability of insertion of a car at position $x = al_0$ inside a given gap ($\chi(a)$ is scaled such that $\max\limits_{-1 \leq a \leq 1} \chi(a) = 1$). The objective then is to make the RSA gap density curve approximate the BD curve by tuning $\chi(a)$ in certain ways. What would be interesting is if this tuning could be grounded in some theoretical argument on the RSA algorithm. However, the rationale we use is simply to make the following observation; the largest discrepancy, at least for ρ near the jamming limit, is the kink

[1] Note also that this second option is a generalisation of another method, which was not implemented, where one varies the probability of insertion as a function of the size of the gap, but not within the gap itself.

at $h = \lambda$. Therefore, tuning $\chi(a)$ in such a way that fewer gaps of length λ are created should attenuate the kink. The desired property is therefore to make $\chi(a)$ smaller at $x = -l_0 + \lambda$ and $x = l_0 - \lambda$, the two symmetric points in a gap where insertion would create a gap of length λ. One way to have this property is to define $\chi(a)$ to be a quartic,

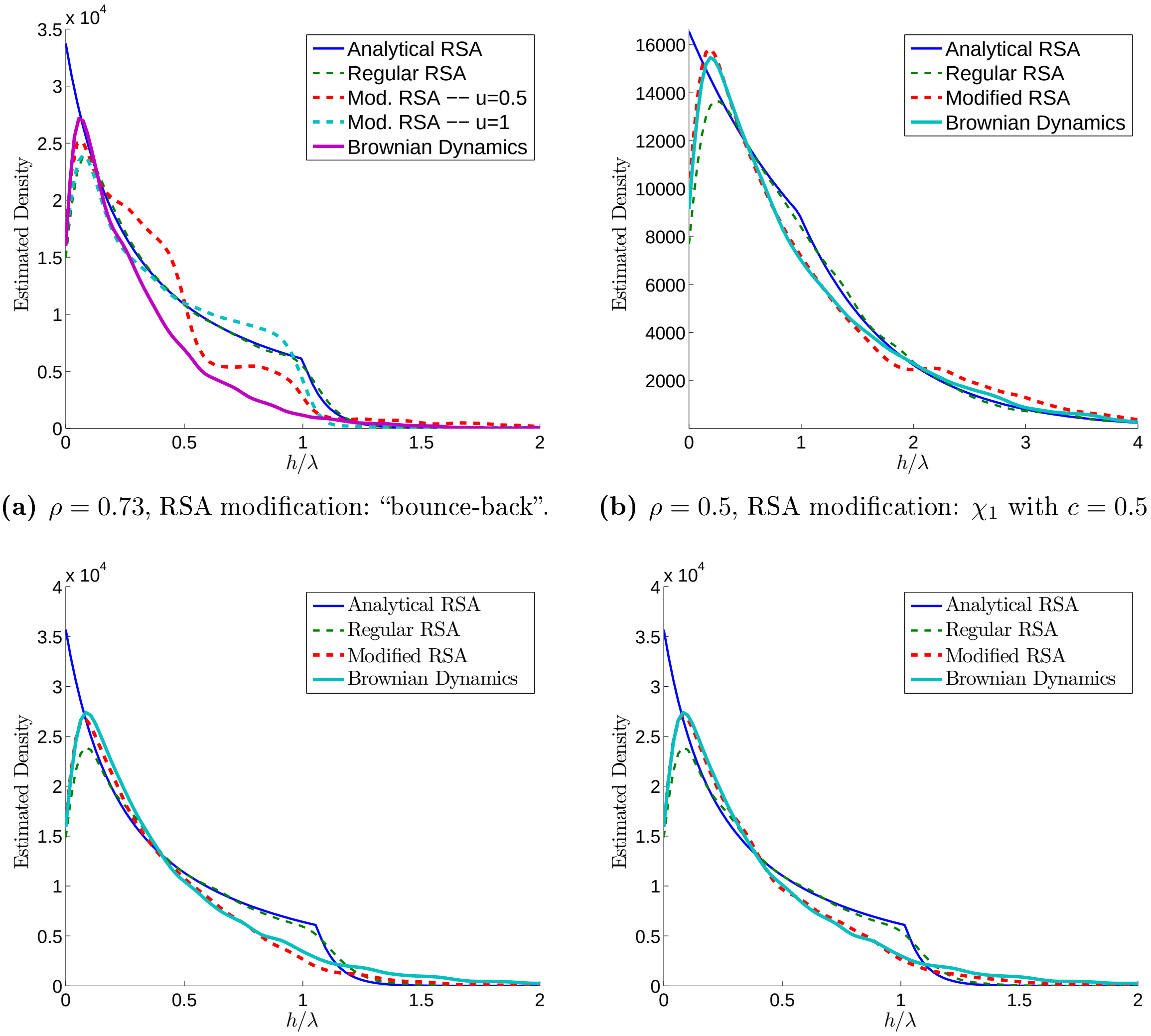
$$\chi_1(a) = c + \frac{1-c}{a_\lambda^2(1-a_\lambda)^2}\left[(a-a_\lambda)^2\left(a-(1-a_\lambda)\right)^2\right], \tag{4.1}$$

where the parameter c determines $\chi_1(a_\lambda)$. A plot of $\chi_1(a)$ is provided in Figure 4.3.

The result of using (4.1) can be seen by observing how the modified RSA curve overlaps with the BD curve in Figures 4.1c and 4.1b. The method is highly successful; the kink is fully eliminated, and the curve now overlaps with the BD curve across the peak and to some extent along the tail. The latter two improvements are side effects of the elimination of the kink; as fewer cars are inserted where they would create gaps of length λ, the frequency of occurrence of other gaps must increase.

A further fine-tuning was attempted. This is referred to as $\chi_2(a)$ in Figures 4.1d and 4.3. The fine-tuning was done because we observed over several trials that the modified RSA curve, although lying very close to the BD curve, would typically lie slightly below it for $h < 0.45\lambda$ and slightly above it for $h > 0.45\lambda$ (for $\rho = 0.73$). In an attempt to remedy this, a Gaussian "bump" is added locally around $h = 0.45\lambda$ which switches sign at $h = 0.45\lambda$. Trials show that this typically made the modified RSA curve approximate the BD curve a little better. The question is, however, whether the improvement is worth the cost of setting three additional parameters (the height, mean, and spread of the Guassian bump). This question is also relevant for $\chi_1(a)$, although there is only c that needs to be tuned for $\chi_1(a)$.

Of course, one could employ methods that measure the l_2 difference between curves and run optimisation routines to minimise this difference. Or one could employ empirical formulae based on trial and error to determine the parameters. But as long as the modifications are based on trial and error, and not grounded in theoretical comparisons of BD and RSA, the tuning of these parameters, and the addition of more parameters, remains slightly artificial. Nevertheless, having only one parameter, c, to tune, can be said to be a fairly successful modification.



(a) $\rho = 0.73$, RSA modification: "bounce-back".

(b) $\rho = 0.5$, RSA modification: χ_1 with $c = 0.5$

(c) $\rho = 0.73$, RSA modification: χ_1 with $c = 0.1$.

(d) $\rho = 0.73$, RSA modification: χ_2, as illustrated in Figure 4.2.

Figure 4.1: Resulting approximations to the BD gap density curve by the modified RSA algorithms.

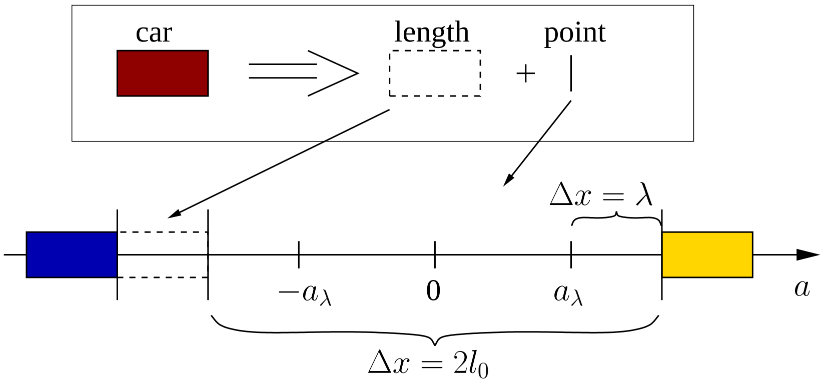


Figure 4.2: Illustration of dual point of view of insertion of a car.

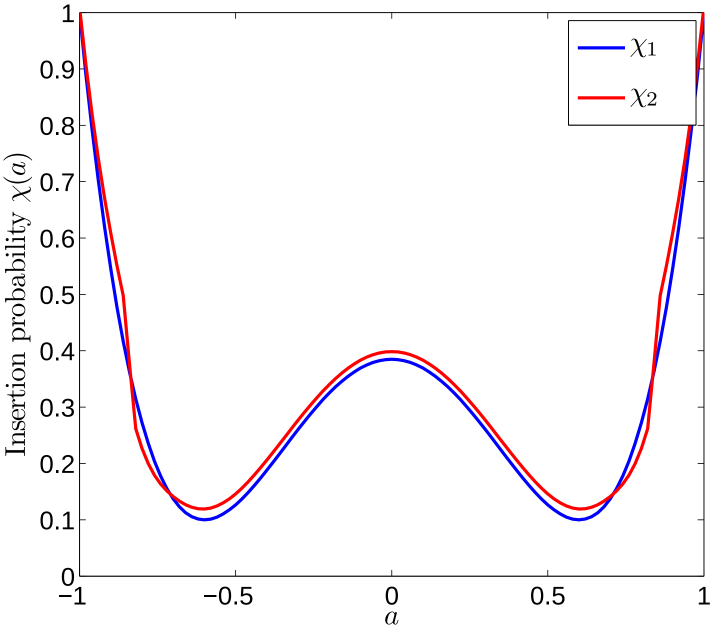


Figure 4.3: Insertion probability functions $\chi_1(a)$ and $\chi_2(a)$.

Chapter 5

Conclusions

5.1 Summary

This project has undertaken a comparative study of the spacial particle configurations that arise from the stochastic processes of BD and RSA. Factors that have been taken into account includes crowdedness and polydispersity. The principal motivation has been to replicate the particle configurations of BD using classical, or modified RSA.

The analytical expression for the gap density function of one-dimensional RSA was useful in measuring the errors of estimates obtained by simulation, and helped in asserting criteria for the comparisons. Otherwise the study has been mostly numerical, relying on some analytical results and numerical convergence studies to guide the tuning of the BD simulation parameters, and some theory of density estimation. The simulations were been implemented in object-oriented MATLAB code, which can be downloaded from `https://sites.google.com/site/patrickatoxford/home`. Although there is room for improvement in efficiency, we can simulate two-dimensional systems of 1×10^4 particles on workstations (see Appendix B) on a time scale from hours to days, and hence the accuracy of the simulation results is high enough to draw conclusions.

Multiple issues were encountered relating to collision handling in the BD simulations, and we developed and compared several algorithms to deal with this. Other issues encountered includes numerical integration of non-smooth, stiff functions; root-finding techniques; kernel smoothing, bandwidth selection and the effect of bounded supports; optimisation of the data structure storing location of the particles; error measurements; distance measurements and the implementation of polydispersity.

Results have been obtained in both one and two dimensions that suggest that even though the similarity of the particle configurations of BD and RSA varies depending on certain system parameters and algorithmic variations, the spacial configuration of particles are generally not equivalent. However, with a simple modification of the RSA insertion probability for a given gap, the approximation to BD particle configurations is much improved. As far as we are aware, this type of comparison has not been undertaken before[1], and therefore the results and conclusions reached in this thesis constitute an interesting, and potentially, new perspective on both BD and RSA.

5.2 Future work

Although we do answer the basic question of the thesis, and go some way in modifying RSA to approximate BD, there are still plenty of possible directions for future research on this topic.

5.2.1 Improving the simulation algorithms

Many of the improvements that could be made to this project relate to the BD algorithms and collision handling. As we already explored some basic variants of the BD algorithms (see Section 2.6), this effort should start with looking at algorithms developed in other biochemical simulation projects, and a good starting point would be [13].

One can of course always attempt larger simulations, employing more particles n, longer T_{BD} and shorter Dt_{BD}. However, this is probably not the most worthwhile improvement, as our convergence studies show that the accuracy of our simulations is sufficient for most of the purposes in this thesis.

It would also be interesting to implement the sequential form of the BD algorithm (discussed in Section 2.6.3 in a programming language which is efficient with loops, while also supporting object-oriented code, such as `C++`. This could offer dramatic gains in simulation time, although this only applies in one dimension.

As discussed in Section 3.1.2, it would probably be wise to implement the multiple-copies approach in two dimensions, instead of running one big simulation, as was done in this thesis project.

5.2.2 Further analysis

Theoretical arguments in two dimensions relating to why RSA and BD give rise to different particle configurations, like those of Section 2.11, should be formulated.

[1] This is backed up by private communication with Prof. T. Aste, who is an expert in the field of packing of granular materials [2].

Another potential improvement to the project would be to try to set the BD algorithms of Section 2.6 on a solid mathematical foundation, in turn relying less on heuristics and empirical convergence studies to validate the results of the simulations.

Furthermore, as discussed in Section 2.8, it would be useful to study and implement an analytical form of for polydisperse RSA in one dimension.

One might also want to look into the theory of statistical hypothesis testing, and apply it to the simulation data, in order to rigorously assess the probability that the particle configurations of RSA and BD originate from the same underlying probability distribution.

5.2.3 Further modifications

There is a multitude of possible variations of the parameters and the composition of the simulation experiments. In addition to crowdedness, BCs and size differences, one could for instance experiment with the effect of drift and intermolecular forces.

Under the banner of polydispersity, it would be interesting try out more extreme size differences, or implement other shapes than discs and spheres, and investigate whether this affects nearest-neighbour distances and contact probabilities. Furthermore, one could implement conservation of momentum in the BD algorithms, both in collision handling and by employing different diffusion constants D for different types of particles.

Further modification of the RSA algorithm, to approximate BD particle configurations should also be studied, and trying out some of these modifications in two dimensions should also be of high priority. Theoretical arguments to support these modifications would also be of great interest.

5.2.4 Higher dimensions

One further area of future research is of course to go from two to three dimensions, or even higher. This should not be very difficult to implement, as it presumably only involves adding an extra dimension to all of the methods and data structures that are already implemented. This is in contrast to the switch from one to two dimensions, where the nature of the simulations were drastically changed[2]. For example, Figure 5.1 shows a possible visualisation of three dimensional experiments; it was made with a few simple modification of the function used for making Figure 3.1, illustrating the

[2] E.g. insertion of a car into a gap partitions that gap into two separate intervals. In higher dimensions, no such partition takes place. Also, any particle can encounter any other particle over the course of a BD simulation, whereas in one dimension the order of cars is fixed.

simplicity of transitioning from two to three dimensions. The ease with which the

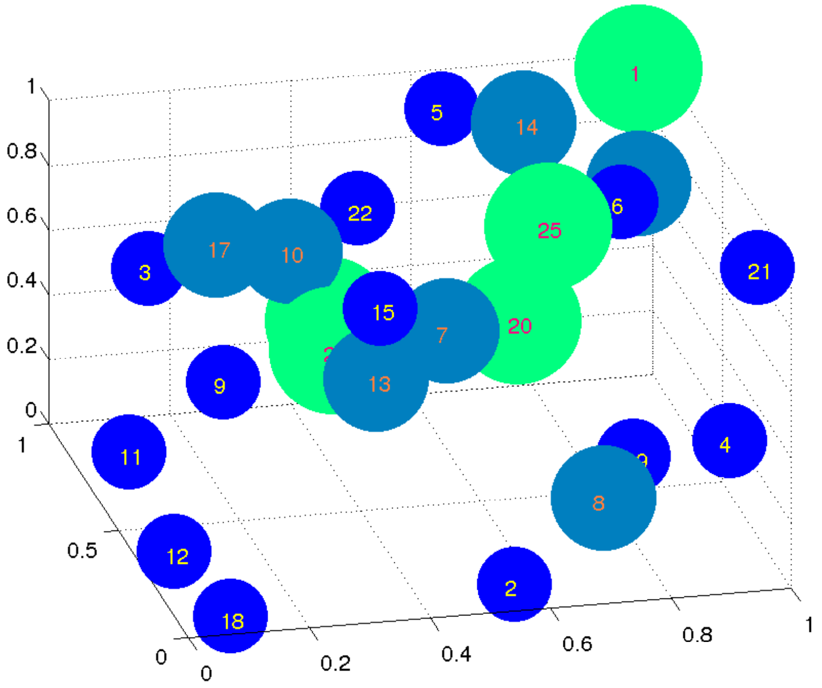


Figure 5.1: Visualisation of three dimensional experiments made from a simple update of the function used to make Figure 3.1.

generalisation to three dimensions can be implemented stems from the fact that the nature of two and three dimensional systems is not far apart, and this also makes me conjecture that the results of a comparison between RSA and BD in three dimensions would be similar to the results in two dimensions; although the spacial particle configurations are similar, and more so than in one dimension, they are not statistically the same.

Appendix A

Symbols and abbreviations

Most used symbols:

- a and b: factors such that $x = al_0$ and $\sigma = bl_0$
- d: number of dimensions
- $f(x)$: generic density estimate, and PDf of $X_i(t) - X_i(0)$
- $G(h,t)$: gap/nearest-neighbour density function
- h gap length
- $\bar{h}$: average gap length
- h_i: gap length between car i and car $i + 1$
- h_I: $\arg\max_h \hat{G}(h,t)$ of a modification to RSA
- H: height of two-dimensional domain
- i: car/particle index (also used temporarily for other indices)
- k: iteration variable of BD and RSA
- k_a: rate constant
- l_0: $\frac{1}{2}(h - \lambda)$ for a given h
- L: length of test line
- n: system size: total number of cars/particles
- N: total number of iterations of BD and RSA

- $P(t)$: number density
- q: total number of different size types
- r: radius of two-dimensional discs
- t: time, employed both for RSA and BD
- T_{RSA}: total time of packing
- T_{BD}: total time of diffusion/Brownian motion.
- $u = k_a \sigma t$ and also multiplicative factor for the “bounce-back”
- W: width of two-dimensional domain
- Δt_{RSA}: insertion time increment for RSA simulations
- Δt_{BD}: time step of BD simulations
- $\rho(t)$: volumetric density
- ρ_0: target density for RSA procedure
- σ: standard deviaton (generic)
- $\zeta(t)$: total number of unresovled collisions for a BD simulation
- λ: car lenght (fixed)
- ξ random variable, distributed either normally, or uniformly
- $\Phi(t)$: proportion of space available for insertion
- $\chi(a)$: insertion probability function for modified RSA algorithms
- $\Psi(t)$: number of unresolved collisions per iteration per car

Abbreviations:

- RSA: Random Sequential Adsorption
- BD: Brownian Dynamics
- PDF: Probability Distribution Function
- MISE: Mean Integrated Square Error
- MAD: Median Absolute Deviation

Appendix B

Computer specifications

Modern laptop:

- Processor: Intel Core 2 Duo P8600 (CPU Freq: 2.4GHz)
- Memory: 4GB

Workstation:

- Processor: Dual Core AMD Opteron(tm) Processor 875 (CPU Freq: 2.2GHz)
- Memory: 32GB

Bibliography

[1] M. Al-Habori. Macromolecular crowding and its role as intracellular signalling of cell volume regulation. *The International Journal Of Biochemistry & Cell Biology*, 33(9):844–864, 2001.

[2] Tomaso Aste. Personal communication. 2010.

[3] A.W. Bowman and A. Azzalini. *Applied smoothing techniques for data analysis: the kernel approach with S-Plus illustrations.* Oxford University Press USA, 1997.

[4] R.J. Ellis. Macromolecular crowding: an important but neglected aspect of the intracellular environment. *Current opinion in structural biology*, 11(1):114–119, 2001.

[5] R.J. Ellis. Macromolecular crowding: obvious but underappreciated. *Trends in Biochemical Sciences*, 26(10):597–604, 2001.

[6] R. Erban, J. Chapman, K.D. Fisher, I.G. Kevrekidis, and L.W. Seymour. Dynamics of polydisperse irreversible adsorption: a pharmacological example. *Arxiv Preprint Physics/0602001*, 2006.

[7] J.J. González, P.C. Hemmer, and J.S. Høye. Cooperative effects in random sequential polymer reactions. *Chemical Physics*, 3(2):228–238, 1974.

[8] D. Hall and A.P. Minton. Macromolecular crowding: qualitative and semiquantitative successes, quantitative challenges. *Biochimica et Biophysica Acta (BBA)-Proteins & Proteomics*, 1649(2):127–139, 2003.

[9] The MathWorks. Matlab Documentation, R2007b, 2006.

[10] A.P. Minton. Implications of macromolecular crowding for protein assembly. *Current Opinion in Structural Biology*, 10(1):34–39, 2000.

[11] A.P. Minton. The influence of macromolecular crowding and macromolecular confinement on biochemical reactions in physiological media. *Journal of Biological Chemistry*, 276(14):10577, 2001.

[12] D. Ridgway, G. Broderick, and M.J. Ellison. Accommodating space, time and randomness in network simulation. *Current opinion in biotechnology*, 17(5):493–498, 2006.

[13] K. Takahashi, S.N.V. Arjunan, and M. Tomita. Space in systems biology of signaling pathways-towards intracellular molecular crowding in silico. *Federation of European Biochemical Societies (FEBS) Letters*, 579(8):1783–1788, 2005.

[14] J. Talbot, G. Tarjus, P.R. Van Tassel, and P. Viot. From car parking to protein adsorption: an overview of sequential adsorption processes. *Colloids and Surfaces A: Physicochemical and Engineering Aspects*, 165(1-3):287–324, 2000.

[15] T.E. Turner, S. Schnell, and K. Burrage. Stochastic approaches for modelling in vivo reactions. *Computational Biology and Chemistry*, 28(3):165–178, 2004.