Simple numerical techniques for mesoscale polymer models

Koopman, E.A.

Publication date
2014
Document Version
Final published version

Simple numerical techniques for mesoscale polymers

Evert Koopman

Simple numerical techniques for mesoscale polymers

Evert Koopman

Simple numerical techniques for mesoscale polymer models

Simple numerical techniques for mesoscale polymer models

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Universiteit van Amsterdam,
op gezag van de Rector Magnificus
prof. dr. D. C. van den Boom
ten overstaan van een door het college voor promoties
ingestelde commissie,
in het openbaar te verdedigen in de Agnietenkapel

op woensdag 4 juni 2014, te 12.00 uur

door

Evert Adriaan Koopman
geboren te Delft

Promotiecommissie

Promotor:

- Prof. dr. P.G. Bolhuis

Co-promotor:

- Dr. C.P. Lowe

Overige leden:

- Prof. dr. ir. P.J. Schoenmakers
- Prof. dr. G. Rothenberg
- Dr. D. Dubbeldam
- Prof. dr. P.D. Iedema
- Prof. dr. ir. T.J.H. Vlugt
- Dr. M. Dreischor
- Prof. dr. B. Mulder

Faculteit der Natuurwetenschappen, Wiskunde en Informatica

The research reported in this thesis was carried out at the Van 't Hoff Institute for Molecular Sciences, Faculty of Science, University of Amsterdam (Nieuwe Achtergracht 166, 1018 WV, Amsterdam, The Netherlands).
The research described in this thesis forms part of the research programme of the Dutch Polymer Institute (DPI), P.O.Box 902, 5600 AX Eindhoven, the Netherlands, DPI project #364.

Contents

Chapter 1

Introduction

The diversity in length and time-scales present in polymer systems has made this a rich and complicated field of science. Not only that, but there are also many industrial applications for polymer systems such as rubbers, plastics and gels. Since the field is so large and the number of pages finite, we will only attempt to treat a small part of it in this thesis.

At the lowest level of polymer science is the basic building block of all polymers, the monomer. A monomer is typically a small molecule that in certain conditions reacts with other monomers of the same or a different type to form larger chain-like molecules. In the special case where there is just one type of monomer, they typically react to form long linear molecules. Depending on the conditions, the polymers can become millions of repeat units long.

Of course there are many other possible polymer types. One can think of chains that have branches or chains that interconnect with each other at certain points. The macroscopic properties of such systems can be very important from an industrial point of view, so being able to predict such properties from the known structure of the monomers and external conditions is one of the goals of polymer science in general. Relevant macroscopic properties could for example be the viscosity of a solution, the elasticity of a largely interconnected system (rubbers) or the heat capacity.

Starting the analysis of such complex systems may seem like a daunting task due to the broadness of the field. However, many workers have contributed both from a theoretical and experimental view point. We will focus mainly on the theoretical approach. With theory we not only mean classical "pen and paper" theories, but also very large sets of coupled equations that can only be solved by computers. Fortunately for us, some eminent scientists, such as Flory and de Gennes who were both awarded a Nobel prize for their work [1, 2], made significant advances describing some clearly defined limits

of the field. One of the first well described and now well understood polymer problems is that of the ideal chain. Simply put the ideal chain is a set of volumeless particles only connected to one another by a harmonic spring [3]. This simple system has been thoroughly investigated and can be solved analytically. While not exactly the most realistic model, particularly because it neglects excluded volume effects, it successfully explains the counter-intuitive observation that polymeric materials generally stiffen when heated. Further, it serves as a starting point on which to build more realistic models.

Some extensions to the ideal chain model that we will treat are the inclusion of a solvent [4], interconnections between chains and giving monomers a volume [1]. All those extension have been studied in detail before by many other people, but in contrast with the ideal chain model, none of them have been completely solved analytically. The lack of analytical solutions is sometimes not a conceptual problem, but more of a technical one. In theory we could model a complete macroscopic system containing 10^{23} atoms at all length scales larger than the Planck length on any given time-scale using numerical solutions to the Schrödinger equations. As one might expect, this is quite a lot of work to do, probably requiring more computational resources than are available in the foreseeable future.

A solution to the complexity of analytical solutions is to carefully select the approximations to make. Typically this selection means discarding or simplifying equations relating to small length scales and short time-scales. For many systems this selection approach works very well, especially if the longest and shortest relevant time and length scales are well separated. For polymer systems the longest and shortest relevant time-scales are not always close, but often we can reduce the amount of equations and degrees of freedom to an amount that we can handle. It might still be too difficult to do by hand on a blackboard, but with the help of computers we can make quite some progress. There are many techniques to help the process of decreasing the amount of equations one needs to solve to describe systems correctly on longer length scales than the atomic scale. In the next few sections we will describe a few that we will use later on in this work. We have chosen the techniques because they work on the length scales we are interested in, and because they have the potential to make long polymer computational problems tractable. Because of the nature of polymers they are all particle methods. The methods listed below are just a few select ones, there are of course many more methods to adress different length and time-scales and alternative continuum descriptions.

1.1 Polymers

Both the application range and the diversity of the fields of science interested in polymer behaviour are too large to even begin to describe in this section. The most commonly encountered polymer in a human made material is plastic while most organisms themselves contain many types and shapes of polymers.

We will however try to describe polymers mostly in the more mathematical sense of the word. Taking a single building block and connecting many identical copies of this block in some way we get a polymer. In this work the most "complicated" building block is a spherical blob that interacts with copies of itself through a simple distance dependent potential. The building blocks (monomers) are connected through a second type of potential. Except for the chapter on polymer networks, the building blocks are only allowed up to two connections each, resulting in only linear chains. Surprisingly enough this simplification of reality will allow us to accurately describe some types of behaviour of such large molecules.

Now some techniques prove to be particularly suited for computer simulations on systems with polymers, amongst them Molecular Dynamics, Dissipative Particle Dynamics and the Lowe-Andersen thermostat.

1.2 Molecular Dynamics

Molecular Dynamics is a technique for simulating molecules at atomic length scales. Because the method has been around for quite some time there are many variations. The simplest one consists only of identical particles interacting through a semi-empirical pair potential and an integration scheme. Usually the Lennard-Jones potential is chosen such that the pair potential u_{ij} is

$$u_{ij} = 4\epsilon \left[(\frac{\sigma}{r_{ij}})^{12} - (\frac{\sigma}{r_{ij}})^{6} \right]$$

with ϵ and σ being some characteristic unit of energy and length respectively [5]. Despite its simplicity it does a remarkable job of describing the phase behaviour of argon [6].

People have since its conception modified this method to have it describe much more complex systems in detail. We will not use any of these modifications except a barostat and a thermostat. Since by itself the method only describes a system with a constant volume V, number of particles N and energy E (which is a very uncommon experimental situation), some fundamental changes need to be made to have the system sample different ensembles.

The two other ensembles that we will consider in this thesis are the NVT and NPT ensembles. They are very similar in that although they are "constant"

temperature and pressure, these quantities actually fluctuate and only the time averaged values should be the desired constant value. In most methods keeping the temperature or pressure constant is achieved by some sort of coupling to an infinite system with the desired properties. Our adaptation of these methods is conceptually identical, but we will show that especially in the constant temperature systems we can make the sampling more consistent with real systems with just some minor variations.

The choice of using Molecular Dynamics with the Lennard-Jones potential is historical. This potential describes the interactions between simple atoms with a reasonable degree of realism. From all the simulation methods working at the level of relatively small molecules, this method has been the most extensively researched and described. Conceptually it is another step up from an ideal gas case, the first step was including repulsive interactions between particles, using a Lennard-Jones potential adds attractive forces between particles. With attractive and repulsive forces, time integration methods and some "control knobs" (the thermostat and barostat) we have the key ingredients to study at least qualitative polymer behaviour.

1.3 Periodicity

One of the biggest problems with numerical simulations of real life materials is the extreme discrepancy between the size of a real macroscopic system and the maximum size of a simulated system. A very typical Molecular Dynamics system with 1000 molecules of H_2O would represent a cubic block of water with each side about 3nm. As the reader probably knows most interesting experiments on water require a bit more water. Luckily many macroscopic phenomena find their origin on the simulated length scale or lower. The biggest issue is trying to most accurately represent the system on such a length scale, in theory a perfect representation of a system on a small scale would allow one to accurately describe phenomena originating from processes in real systems on these length scales. This means that simulations of a real system should try not to introduce things into the system that are not there in a real system. The easiest difference to spot between any real system and a simulated system is the edge. A real system also does have an edge, but the ratio of the atoms at the edge of a system to the atoms not at the edge is many orders of magnitudes lower than in a simulated system.

There are three common approaches for handling the edges in a simulated system. The easiest is not to include explicit edges at all. Another approach is to place artificial walls around the simulated sample and define interactions between the wall and the system. The last approach is to introduce walls, but make each wall a window to an identical copy of the simulated system.

It is easy to see that constraining the size of a system in some way is useful, typically disqualifying the first edge handling method for our systems of interest. Any system that does not inherently stick to itself (e.g. a solid or particles with artificially constrained distances) will eventually evaporate due to thermal motion.

While all three methods have some benefits and drawbacks, the consensus seems to be that in most simulations the last option "periodic boundaries" is the best option. Periodic boundaries have the benefit of keeping the system at a fixed size, without the drawback of introducing wall interactions where there should be none. More explicitly, there are no boundary effects at all, so even for small systems the macroscopic "boundary to bulk" ratio is realistic.

In this work we will mostly use the periodic boundary conditions. An interesting issue with periodic boundary conditions is described in the chapter on polymer networks. It will explain why the classification of a system as a network gets highly complicated by the introduction of periodic boundaries and give a method for overcoming these complications.

1.4 Dissipative Particle Dynamics

The Dissipative Particle Dynamics (DPD) method was introduced in 1992 by Hoogerbrugge and Koelman [7]. Their model was improved in 1995 when Español and Warren [8] put it on a solid theoretical foundation by getting the thermodynamic equilibrium properties right. The method itself is a mesoscopic particle method that is well-suited for multi-component fluid simulations, and specifically ones where the dynamics itself are interesting. With mesoscopic we mean in this context that the simulations capture the fluctuations with small scale physics, but individual particles making up the system are meant to represent many atoms or molecules. An oft overlooked property of the DPD model is that it has a relatively simple equation of state. This simple equation of state allows a theorist to work out analytical solutions to some problems that can be checked with computer simulations in a relatively simple way. A simpler model like an ideal gas, will also have a simpler equation of state that fails to describe things like phase transitions. On the other hand a more complex model like Molecular Dynamics with a Lennard-Jones potential results in an equation of state that describes phase transitions, but is very complex. The DPD model and corresponding equation of state fall in the middle of these, it describes for example mixing and de-mixing, but still has a simple equation of state.

With these properties in mind, it is useful to explain the actual method in a bit more detail. The method consists of three parts, a set of forces that enforce the correct static thermodynamic properties, an integration scheme

that takes care of the correct time evolution of the system and finally a set of boundary conditions. The forces in DPD are usually given as three separate contributions. First the conservative force f_c

$$\vec{f_c} = \begin{cases} a_{ij}(1 - \frac{r_{ij}}{r_c})\hat{r_{ij}} & \text{if } r_{ij} < r_c \\ 0 & \text{if } r_{ij} > r_c \end{cases}$$

Here a_{ij} is a function of the particle types involved in a particle pair interaction. The distance between particles i and j is r_{ij}, the force cut-off radius r_c and $\hat{r_{ij}}$ is the unit distance vector between i and j. Notice that, unlike with most potentials used for molecular simulations, the force does not go to infinity at small separations. The repulsion between DPD particles is soft, making it possible to integrate the equations of motion with a longer time-step. The a_{ij} parameter can be used to tune the interaction between different fluids in the system. Phase transitions in the system occur because of this force, setting it to 0 reduces the system to an ideal gas. Second we have the random force, that together with a dissipative force acts as the system thermostat. The random force $\vec{f_r}$ consists of a tunable magnitude σ, a distance dependent weight ω_R and a randomly chosen variable from a Gaussian distribution with unit variance Θ_{ij} chosen to be

$$\vec{f_r} = \begin{cases} \sigma\omega_R(r_{ij})\Theta_{ij}\hat{r_{ij}} & \text{if } r_{ij} \leq r_c \\ 0 & \text{if } r_{ij} > r_c \end{cases}$$

Finally the dissipative force $\vec{f_d}$ force has the following form:

$$\vec{f_d} = \begin{cases} -\gamma\omega_D(r_{ij})(\hat{r_{ij}} \cdot \vec{v_{ij}})\hat{r_{ij}} & \text{if } r_{ij} \leq r_c \\ 0 & \text{if } r_{ij} > r_c \end{cases}$$

Here γ is a tunable friction coefficient and ω_D is a distance dependent weight. The random and dissipative forces balance the system to be at a temperature $k_B T = \frac{\sigma^2}{2\gamma}$ if $\omega_D(r_{ij}) = \left(\omega(r_{ij})\right)^2$. For simplicity the weight function $\omega_R(r_{ij})$ is usually chosen to be

$$\omega_R(r_{ij}) = \begin{cases} 1 - \frac{r_{ij}}{r_c} & \text{if } r_{ij} \leq r_c \\ 0 & \text{if } r_{ij} > r_c \end{cases}$$

The total force on a single particle i is now given by

$$\vec{f_i} = \Sigma_{i \neq j} \left(\vec{f_c}(r_{ij}) + \vec{f_r}(r_{ij}) + \vec{f_d}(r_{ij}) \right)$$

Note that although this is a double summation, all three force components have a limited range, so the simulation remains manageable even for large amounts of particles.

Now the equations of motion also have to integrated. There are some options here, but for strict DPD using the random and dissipative forces a scheme that has no velocity dependent forces is useful. A modified velocity-Verlet scheme is typically used. Here the equations of motion are integrated as follows, using g as a shorthand for the calculated forces based on the new particle positions:

$$\vec{r}_i(t+\Delta t) = \vec{r}_i(t) + \Delta t\vec{v}_i(t) + \frac{1}{2}(\Delta t)^2\vec{f}_i(t) \tag{1.1}$$

$$\vec{v}_i^*(t+\Delta t) = \vec{v}_i(t) + \lambda\Delta t\vec{f}_i(t) \tag{1.2}$$

$$\vec{f}_i(t+\Delta t) = g(\vec{r}_i(t+\Delta t), \vec{v}_i^*(t+\Delta t)) \tag{1.3}$$

$$\vec{v}_i(t+\Delta t) = \vec{v}_i(t) + \frac{1}{2}\Delta t(\vec{f}(t) + \vec{f}_i(t+\Delta t)) \tag{1.4}$$

Here the velocity v_i^* is a guess for the actual velocity at $t+\Delta t$. A suggested optimal value for this the value of λ is 0.65.

1.5 Lowe Andersen thermostat

The Lowe Andersen thermostat is a modification of the original DPD algorithm [9]. It recognises that the dissipative and random forces in DPD basically combine to act as a thermostat. While conceptually not much different, it modifies this thermostat. Discarding the conservative force, it then yields a good representation of an ideal gas that resolves hydrodynamics correctly on a scale larger than the typical inter-particle distance. As we will see, however, it can also be applied where conservative forces are present.

The DPD thermostat is replaced by a stochastic mechanism that acts directly on velocities, the forces are implied. In spirit, this idea is taken from the Andersen thermostat. The Andersen thermostat can be written as

$$\vec{v_i^*}(t) = \begin{cases} \vec{v_i}(t) & \Gamma\Delta t < \zeta_1 \\ \vec{\epsilon} & \Gamma\Delta t \geq \zeta_1 \end{cases} \tag{1.5}$$

meaning that every iteration with a time-step Δt, the velocity $\vec{v}$ of particle i is replaced by ϵ with a chance $\Gamma\Delta t$. ζ_1 is a uniform random number between 0 and 1. Here ϵ is sampled from a Boltzmann distribution for a temperature k_BT:

$$\vec{\epsilon} = \sqrt{\frac{k_BT}{m_i}}[\zeta_2, \zeta_3, \zeta_4] \tag{1.6}$$

with ζ_2, ζ_3 and ζ_4 being random numbers from a Gaussian distribution of unit variance and 0 mean. The particle mass is given by m_i.

As will be shown in Chapter 2 the Andersen approach has a negative effect on the auto-correlation function of the particle velocities. The negative effect implies artificially slowed down dynamics. An alternative approach is to do the thermostatting operation on particle pairs. It is then possible to conserve momentum (an essential requirement to recover correct hydrodynamics). Every simulation time step, with a probability $\Gamma\Delta t$ for each particle pair with a relative distance smaller than r_c, their relative velocities $\vec{v}$ are replaced with one from a Boltzmann distribution with the desired system temperature $k_B T$ with a chance based on the interaction frequency Γ and time-step Δt:

$$\begin{aligned} \vec{v}_i^*(t) &= \begin{cases} \vec{v}_i(t) & \Gamma\Delta t < \zeta_1 \\ \vec{v}_i + \frac{\mu_{ij}}{m_i}\left(\lambda - \left(\vec{v}_i - \vec{v}_j\right)\cdot\hat{\sigma}_{ij}\right)\hat{\sigma}_{ij} & \Gamma\Delta t \geq \zeta_1 \end{cases} \\ \vec{v}_j^*(t) &= \begin{cases} \vec{v}_j(t) & \Gamma\Delta t < \zeta_1 \\ \vec{v}_j - \frac{\mu_{ij}}{m_j}\left(\lambda - \left(\vec{v}_i - \vec{v}_j\right)\cdot\hat{\sigma}_{ij}\right)\hat{\sigma}_{ij} & \Gamma\Delta t \geq \zeta_1 \end{cases} \end{aligned} \tag{1.7}$$

Here $\hat{\sigma}_{ij}$ is the unit separation vector $\hat{\sigma}_{ij} = (\mathbf{r}_i - \mathbf{r}_j)/\left|\mathbf{r}_i - \mathbf{r}_j\right|$, m_i and m_j the masses of particles i and j respectively, μ_{ij} $(= m_i m_j/(m_i + m_j))$ the reduced mass of the pair and λ a stochastic variable $\lambda = \zeta_2\sqrt{(k_B T/\mu_{ij})}$ where ζ_2 is again a random number from a Gaussian distribution with zero mean and unit variance.

Integration of the equations of motion is simpler than DPD since the forces are not velocity dependent. We get the new velocity $\vec{v}_{\text{new}}$ from the old velocity $\vec{v}_{\text{old}}$ from the conservative force $\vec{f}$ and time-step size Δt as

$$\vec{r}_{\text{new}} = \vec{r}_{\text{old}}\Delta t + \frac{\vec{f}_{\text{prev}}\Delta t^2}{2} \tag{1.8}$$

$$\vec{v}_{\text{new}} = \vec{v}_{\text{old}} + \frac{\Delta t\vec{f}_{\text{prev}}}{2} \tag{1.9}$$

We can see the Lowe-Andersen method as a best of both worlds approach; on one side it has the stability of a velocity based algorithm, on the other side it preserves hydrodynamics to a large extent.

1.6 Nomenclature

A clear distinction has to be made between the various terms describing parts of the techniques mentioned here. Historically, the DPD method is used to refer to the combination of the pair potential, a thermostat and an integration scheme. Since we will adapt some parts of the method later on, we will try to refer to the individual parts separately. "The DPD method" will be the combination of the potential, the thermostat and the integration scheme. The most often referred to modification will be the Lowe-Andersen thermostat.

There is no such thing as "The Lowe-Andersen method" so we will always specify if we use just the thermostat and associated integration scheme, or a combination of the thermostat and the potential. Using the Lowe-Andersen thermostat in absence of a pair potential is effectively a way of modelling an ideal gas with implicitly correct hydrodynamics.

1.7 Outline of this thesis

This work will describe improvements of existing algorithms and develop some new ones as well. The common theme is that they can all be applied in some way to solving polymer related problems. In Chapter 2 we will consider the use of the Lowe-Andersen thermostat in Molecular Dynamics systems. This is a useful method for realistically, but in a simple way, including the thermal and dynamical effects of solvent surrounding polymers. We show how it alleviates some problems associated with other alternative approaches and address issues of implementing it efficiently. In Chapter 3 we develop and apply a barostat, commonly used in Molecular Dynamics systems to maintain a set pressure, to coarse grained DPD systems. This extends the methodology of Chapter 2 to the case where the pressure and temperature are specified. This is normally the case under experimental conditions. The simulations show that it is a robust way of doing this and confirm predictions for the phase behavior of the coarse grained model.

In Chapter 4 we consider an important class of polymer problem, the formation of networks. Network formation produces solid-like materials, such as gels, with highly variable properties. Specifically, we address a commonly overlooked problem when simulating such objects in systems using the periodic boundary conditions desirable in a simulation. How do we identify when we have or have not got a network? We introduce a new method to resolve this problem that we then apply to study deformation in a model polymer gel. Then in Chapter 5 we use the techniques developed in Chapters 2 and 3 to simulate the collapse of a polymer in a super critical solvent. A theoretical model for describing the collapse and expansion of a polymer in a symmetric solvent predicts counterintuitively that the collapsed state in this regime contains more solvent than the expanded state. The simulation model allows us to quantitatively test this prediction. In Chapter 6 we introduce a simple method that will enable simulations of model polymers in a specific type of flow field - an extensional flow - in particle systems using periodic boundary conditions. This type of flow is relevant because it is used in microfluidic experiments to study the deformational and elastic properties of long polymers, for example DNA. Finally in Chapter 7 we consider an unusual aspect of the theory of polymers. These often predict their properties in fractional as

well as integer spacial dimensionalities. To test these predictions a method of simulating polymers in non-integer dimensions is required and here we try to develop such a method.

Chapter 2

Applications of the Lowe-Andersen thermostat

Being able to control your simulation can be a complex process. The closer a property is to the actual building blocks, the easier it typically is. Restraining a single particle not to move is easy for example. On the other hand there are the large scale observables like temperature, viscosity and heat capacity, all some form of ensemble averages. For some of the observables we are only interested in their values, but for some we want to be able to choose a specific value, just like with a real experiment.

Temperature is one of the things we want to control. Since the properties of a system can vary wildly, an experimenter usually controls the temperature in one way or another. While not always a concious decision, it is easy to see that many experiments of all sorts would result in very different situations if the temperature is off by a factor 10. As an example, try to measure the density of ice at 2700 degrees Kelvin, this will be problematic.

Real experiments use a thermostat of some sorts when the temperature has to be controlled. If the actual temperature does not have to be accurate up to a few degrees, performing the experiment in a room at the right temperature will suffice. For more control we could place the experimental set-up in a large liquid bath with the right temperature. The zeroth law of thermodynamics works in our favour here. If the surroundings of our experiment are large enough and heat transfer is fast enough, our experimental set-up will have a temperature close to that of the environment. Keeping this coupling principle in mind makes it easier to understand the mechanisms used to control the temperature in a simulation.

Given the definition of temperature $k_B T$ in our typical particle system with N particles each with mass m_i and velocity v_i, $k_B T = \frac{1}{3N} \sum_i m_i v_i^2$, one approach can be to simply alter the velocities of all particles at given times.

Simply rescaling is the simplest (and most naive) way. A simple argument is that in a particle system the temperature is not the exact same value at each instant in time, it fluctuates around this value.

A more physical approach is to try to couple the system to a larger system with the correct temperature. This is the approach taken by most actually used thermostats like the Andersen, Nose-Hoover and Berendsen thermostats [10–12]. Even more alternative approaches with their drawbacks and benefits are reviewed by Hunenberger [13].

This chapter investigates a relatively new scheme (the Lowe-Andersen thermostat) that is relatively simple, but has smaller drawbacks than alternative simple schemes. First we show the benefits of the Lowe-Andersen scheme, then we introduce two modified versions of this scheme and finally we investigate a problem introduced by the pairwise interaction in this method.

2.1 The Lowe-Andersen thermostat in MD simulations

2.1.1 Introduction

Specifying the temperature in Molecular Dynamics (MD) involves using a "thermostat" that in some way couples the system to an external heat bath. The simulation then samples the Canonical (NVT) Ensemble. Here we show that a recently proposed method [9], the Lowe-Andersen thermostat, has some advantages over existing methods.

Thermostats can be categorised as either local or global. Local thermostats dissipate energy on a spatially localised scale whereas global thermostats dissipate energy uniformly in the system. Generally one may regard the former as preferable because this is usually more realistic and (in principle) allows local temperature control. For example, using a local thermostat at the solid surface reproduces the correct diffusive behaviour of absorbents in carbon nano-tubes [14]. The simplest local thermostat is the Andersen thermostat [10]. It works by having particles undergo "bath" collisions. A bath collision involves assigning the particle a new velocity taken from the Maxwellian distribution (for the set temperature). Each particle has a probability $\Gamma\Delta t$ per time-step Δt of undergoing a bath collision. Here Γ is the collision frequency. This method satisfies detailed balance so the equilibrium quantities are correct. A similar method solves a Langevin equation of motion for the particles [15]. This replaces the discrete bath collisions in the Andersen method with a dissipative friction force, but no longer necessarily satisfies detailed balance. Both methods have one notable drawback. They significantly perturb the dynamics of the system relative to the unthermostatted case [5]. For the Andersen method, unless the collision frequency is low, the rate of diffusion of the particles is reduced. This is inefficient because it means that

configurational phase space is only sampled slowly. On the other hand, high collision frequencies are preferable for efficient thermostatting. Thus, there is a trade-off between these two competing requirements.

An elegant "global" approach is the "extended system" methodology, developed by Nosé [11] and Hoover [16]. One advantage of this method is that it does not significantly perturb the dynamics of the equilibrium system. For example, in a Lennard-Jones fluid the diffusion coefficients calculated in a system coupled to a Nosé-Hoover thermostat hardly differ from those calculated in the unthermostatted system [5]. However, as well as being global it has the additional disadvantage of not satisfying Galilean invariance. That is, centre of mass motion, unless explicitly corrected for, is treated as an increase in the temperature. This is problematic in non-equilibrium simulations [17].

Stoyanov and Groot [18] have recently proposed a method that attempts to construct a local, Galilean invariant, Nose-Hoover thermostat. This combines a Nosé-Hoover approach with a Lowe-Andersen thermostat [9]. Here we consider just the latter. Its origins lie in the field of mesoscopic simulation [19]. A technique called "Dissipative Particle Dynamics" (DPD) was developed by Hoogerbrugge and Koelmans [7]. Español and Warren [8] put the method on a firm theoretical foundation by showing that DPD basically consists of particles interacting with a generic soft potential, coupled to a thermostat. The thermostatting part of the algorithm is not dependent on the soft potentials so in principle it can also be used with the more realistic potential used in MD. This is indeed the case [17]. Further, the DPD thermostat actually has some nice properties. It is local, Galilean invariant and conserves linear and angular momentum. One issue however is how exactly to solve the DPD stochastic equations of motion [20]. Using a simple Verlet type scheme [21] can lead to serious errors in the equilibrium properties of the system unless a very small time-step is used [22, 23]. Several more complicated algorithms were proposed [24–26] to mitigate this problem. Notably, Pagonabarraga *et al.* pointed out that in order to satisfy detailed balance an iterative procedure was necessary [26]. Lowe suggested that rather than modify the method of solving the DPD equations one might instead modify the method [9]. He proposed using the Andersen methodology but, instead of thermalizing the velocity of individual particles, thermalizing the relative velocity of pairs of particles. The resulting algorithm has DPD's positive features (locality, Galilean invariance and momentum conservation) but, using a simple Verlet scheme to solve the equations of motion, satisfies detailed balance. Peters showed that in the short time-step limit the two methods are equivalent [27]. The Lowe-Andersen thermostat can, however, use a much larger time-step and still reproduce equilibrium properties accurately. It can therefore be viewed as a computationally more efficient method of implementing the DPD thermostat. As with the DPD thermostat, the Lowe-Andersen thermostat should work equally well

in MD simulations. In this section we compare simulations of a Lennard-Jones fluid using the Andersen and Lowe-Andersen methods. There are reasons to believe that, if implemented efficiently, the latter should perturb the dynamics of the system to a lesser extent. We begin by explaining why this should be the case and how one should best configure the thermostat in MD. The simulations we describe then test to what extent this methodology represents an improvement on the Andersen method.

2.1.2 Comparison of the Andersen and Lowe-Andersen methods

The original Andersen thermostat proceeds by first integrating the equations of motion, using for example a velocity Verlet algorithm [28]. In a second step the particles have a probability $\Gamma\Delta t$ of undergoing a "bath" collision. Here Γ is the bath collision frequency and Δt the time-step. Bath collisions involve taking a new velocity from the Maxwell distribution, so the new velocity of particle i, $\vec{v}_i^*(t)$, is

$$\vec{v}_i^*(t) = \begin{cases} \vec{v}_i(t) & \Gamma\Delta t < \zeta_1 \\ \vec{\epsilon} & \Gamma\Delta t \geq \zeta_1 \end{cases} \tag{2.1}$$

where the random vector $\vec{\epsilon}$, in terms of the temperature T, Boltzmann constant k_B and particle mass m_i, is

$$\vec{\epsilon} = \sqrt{\frac{k_B T}{m_i}}\,[\zeta_2, \zeta_3, \zeta_4] \tag{2.2}$$

with ζ_1 a uniform random number between 0 and 1, and ζ_2 to ζ_4 independent random numbers taken from a Gaussian distribution of unit variance. Andersen showed that under these operations the Canonical distribution is invariant and that it is a valid Monte-Carlo scheme (semi-detailed balance is satisfied so long as the first step conserves total energy) [10]. For the Lowe-Andersen thermostat one considers pairs of particles located within a distance R_T of each other. A bath collision then involves taking a new relative velocity for the two particles from the Maxwellian for relative velocities. To conserve angular momentum this operation is only performed on the component of the relative velocity parallel to the line of centres. Further, the new relative velocity is imposed in such a way that linear momentum is conserved. A bath collision then takes the form

$$\begin{aligned} \vec{v}_i^*(t) &= \begin{cases} \vec{v}_i(t) & \Gamma\Delta t < \zeta_1 \\ \vec{v}_i + \frac{\mu_{ij}}{m_i}\left(\lambda - \left(\vec{v}_i - \vec{v}_j\right)\cdot\hat{\sigma}_{ij}\right)\hat{\sigma}_{ij} & \Gamma\Delta t \geq \zeta_1 \end{cases} \\ \vec{v}_j^*(t) &= \begin{cases} \vec{v}_j(t) & \Gamma\Delta t < \zeta_1 \\ \vec{v}_j - \frac{\mu_{ij}}{m_j}\left(\lambda - \left(\vec{v}_i - \vec{v}_j\right)\cdot\hat{\sigma}_{ij}\right)\hat{\sigma}_{ij} & \Gamma\Delta t \geq \zeta_1 \end{cases} \end{aligned} \tag{2.3}$$

Here $\hat{\sigma}_{ij}$ is the unit separation vector $\hat{\sigma}_{ij} = (\vec{r}_i - \vec{r}_j)/\left|\vec{r}_i - \vec{r}_j\right|$, m_i and m_j the masses of particles i and j respectively, μ_{ij} $(= m_i m_j/(m_i + m_j))$ the reduced mass of the pair and λ a stochastic variable $\lambda = \zeta_2\sqrt{(k_B T/\mu_{ij})}$. The procedure is carried out sequentially for each pair and the velocity appearing on the right hand side is always the current value (i.e. it can itself be a post-collisional value).

The equations of motion are integrated in two stages. First the "pre-collision" step

$$\vec{x}_i(t + \Delta t) = \vec{v}_i(t) + \frac{1}{2}\Delta t^2 \vec{f}(t) \tag{2.4}$$

$$\vec{v}_i(t + \frac{1}{2}\Delta t) = \vec{v}_i(t) + \frac{1}{2}\vec{f}_i(t) \tag{2.5}$$

then the forces acting on all particles are recalculated with the new positions and velocities and the velocity update scheme 2.3 is applied. Finally the velocity is updated again

$$\vec{v}_i(t + \Delta t) = \vec{v}_i(t + \frac{1}{2}\Delta t) + \frac{1}{2}\vec{f}_i(t + \Delta t) \tag{2.6}$$

.

To compare the extent to which the two methods might affect the dynamics of the system we can first look at the short time behaviour of the velocity autocorrelation function (VACF). The VACF, $C(t)$, is defined as

$$C(t) = \langle \vec{v}(0) \cdot \vec{v}(t) \rangle \tag{2.7}$$

it can in turn be related to the diffusion coefficient, D, via the Green-Kubo relation

$$D = \frac{1}{d}\int_0^\infty C(t)dt \tag{2.8}$$

where d is the dimensionality of the system. The VACF can be expanded as a Taylor series to give

$$C(\Delta t) = \frac{3k_B T}{m} + \frac{\Delta t}{m}\langle \vec{v}(t) \cdot \vec{f}(t) \rangle + \mathcal{O}\Delta t^2 \tag{2.9}$$

where $\vec{f}$ is the force acting on particles with mass m. For the thermostatted system the force can be decoupled into two independent contributions, one arising from the inter-particle forces, $\vec{f}^{\mathrm{I}}$ and one due to the thermostat, $\vec{f}^{\mathrm{T}}$. In a system with continuous potential, it follows from time reversal symmetry that $\left\langle \vec{v} \cdot \vec{f}^{\mathrm{I}} \right\rangle = 0$. On the other hand, the action of a bath collision for the Andersen thermostat corresponds to a force

$$\vec{f}^{\mathrm{T}} = \frac{m_i}{\Delta t}(\vec{\epsilon} - \vec{v}) \tag{2.10}$$

and for the Lowe-Andersen thermostat

$$\vec{f}^{\mathrm{T}} = \frac{\mu_{ij}}{\Delta t}\left(\lambda - \left(\vec{v}_i - \vec{v}_j\right)\cdot\hat{\sigma}_{ij}\right)\hat{\sigma}_{ij} \tag{2.11}$$

If we look at the VACF after one time-step for the Andersen thermostat this gives

$$C(\Delta t) = \frac{3k_BT}{m_i}\left(1 - \Gamma\Delta t\right) \tag{2.12}$$

Restricting ourselves to a one component system (all masses equal), for the Lowe-Andersen thermostat we have

$$C(\Delta t) = \frac{3k_BT}{m_i}\left(1 - \frac{\Gamma\Delta t}{6}\right) \tag{2.13}$$

Given that $\Gamma\Delta t \leq 1$ we see that the Andersen thermostat in the worst case reduces the VACF to zero after one time-step. This is clearly part of the reason it strongly suppresses diffusion. On the other hand for the Lowe-Andersen thermostat, in the same limit $C(\Delta t)/C(0) = 5/6$. This analysis shows that while the Lowe-Andersen method does perturb the dynamics at very short times, it does so to a lesser extent than the Andersen method.

At longer times one also expects a difference. The Lowe-Andersen method conserves momentum whereas the Andersen method does not. One therefore expects only the former to preserve hydrodynamic behaviour in the system. It is well know that the longer time behaviour of the VACF in a fluid is well described by continuum hydrodynamic theory [29]. A method that preserves hydrodynamics should affect the long-time dynamics to a much lesser extent than a method that does not. In the case of the Lowe-Andersen thermostat there is however a proviso. The thermostat makes a contribution to the instantaneous stress in the system and hence increases the viscosity relative to the non-thermostatted case. In some cases this is desirable [9] but not if one wishes the thermostat to minimally perturb the dynamics of the system. For the one component system an estimate for the extra contribution to the viscosity is [9]

$$\eta^T \sim \frac{\pi\rho^2 {R_{\mathrm{T}}}^5\Gamma}{75m} \tag{2.14}$$

where ρ is the density. Unless η^T is small compared to the viscosity of the unthermostatted system the increased total viscosity will slow the dynamics of the system. This is not what we want so clearly we should aim to minimise η^T. Given that optimal thermostatting requires that we maximise Γ the interaction radius for the thermostat is the only parameter we are free to vary (the other quantities are intrinsic to the fluid). However, the number of possible thermostat collisions a particle can experience per time-step itself depends on

R_T. We should therefore aim to make R_T as small as possible while still satisfying the constraint that, on average, there are enough collisions per time-step to efficiently thermostat the system.

2.1.3 Numerical test of the algorithms

To test the relative effect of the Lowe-Andersen and Andersen thermostats we have simulated a fluid of particles interacting through a Lennard-Jones pair potential. Using the usual reduced units to describe this system [28], we considered five state points. Three are at a reduced temperature $T^* = 2.0$ (in the supercritical fluid regime) at moderate to high density $\rho^* = 0.1, 0.4$ and 0.8. Two are in the liquid regime; $T^* = 0.8, \rho^* = 0.7$ and $T^* = 0.722, \rho^* = 0.8442$. The latter is near to the triple point. In order to compare the two methods we need to match the bath collision frequency. As noted above, for the Lowe-Andersen method this will depend on the interaction radius used for the thermostat and we wish to minimise this quantity.

We therefore chose the minimum value of R_T that gives a prescribed collision rate. In practice we achieve this by, at each state point, carrying out a simulation during which we calculate the collision rate for the Lowe-Andersen thermostat as a function of R_T. This simply requires one simulation to calculate the average number of particle pairs within hypothetical radius R^T of each other. Once we have this information, we can determine from the plot of Γ versus R_T the minimum value of R_T that gives the same collision frequency as the Andersen thermostat simulation with which we are comparing. In figure 2.1 we show as an example the collision frequency as a function R_T at the state point $T^* = 0.8, \rho^* = 0.7$.

Obviously, for small R_T where the thermostat interaction radius is smaller than the collision diameter there will be no thermostat collisions. However once this distance is exceeded the frequency rapidly increases. Thermostat radii for which the actual number of collisions match the choice of $\Gamma\Delta t$ for the five state points are given in table 2.1.

In all cases R_T is only fractionally greater than the collision diameter. Using a time step of $\Delta t^* = 0.001$ we calculated the temperature, the mean potential energy per particle U^* and the pressure P^*. The values are also in tabulated table 2.1, along with the values calculated by Johnson ***et al.*** [30]. Repeating the simulations with $\Delta t^* = 0.0005$, the results, to the accuracy quoted, were indistinguishable. As the data in table 2.1 show, the method clearly gives the static properties correctly.

Turning to the dynamics, in figure 2.2 and 2.3 we have plotted the mean-squared displacement as a function of time for simulations using our method and Andersen's method.

This is the same calculation described by Frenkel and Smit to illustrate

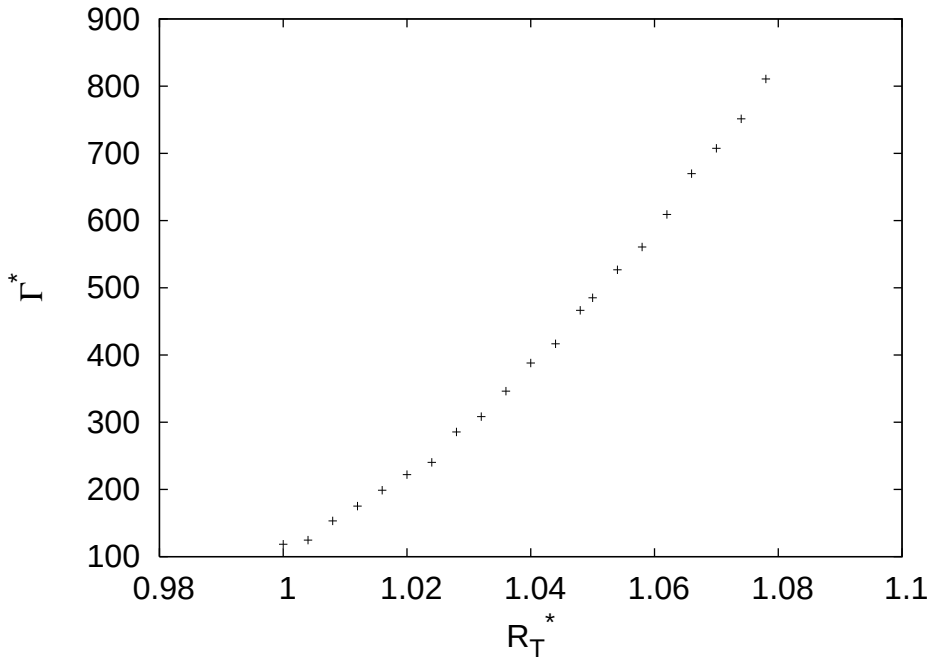

Figure 2.1: Collision frequency Γ^* as a function of the thermostat interaction radius R_T^* at a state point T^*=0.8, ρ^*=0.7.

the limitations of the Andersen thermostat [5]. The figure shows that for $\Gamma \leq 20$ the dynamics are little perturbed by our thermostat whereas there is already a pronounced reduction in the rate of increase of the mean squared displacement (which is proportional to the diffusion coefficient) using the Andersen thermostat. Moving to higher thermostatting rates and other state points, in figure 2.4 we have plotted the measured diffusion coefficients up to the maximum collision frequency the time step permits.

All diffusion coefficients have been scaled relative to their "true" value (that is, the value they have in the weakly thermostatted limit). We can clearly see that although there is a significant decrease in the diffusion for high collision frequencies for the Lowe-Andersen thermostat the effect is nowhere near as dramatic as for the Andersen thermostat. At $\Gamma = 1000$, for example, in the worst case (corresponding to the lowest density and hence the largest thermostat radius) the diffusion coefficient is reduced by 40%. In the best case (the highest density) only 20%. This should be compared with the Andersen method which, at this collision rate, typically reduces the diffusion coefficient by more than two orders of magnitude. This observation applies for all the state points we considered.

T^*	ρ^*	R_T	P^*	P^* [30]	U^*	U^* [30]
0.722	0.8442	1.036	0.41(1)	-	-5.860(5)	-
0.8	0.7	1.051	-0.50(1)	-0.521(5)	-4.852(5)	-4.851(2)
2.0	0.1	1.348	0.1806(5)	0.1802(2)	-0.640(5)	-0.643(2)
2.0	0.4	1.088	0.75(1)	0.742(3)	-2.431(5)	-2.435(3)
2.0	0.8	1.005	5.50(5)	5.453(7)	-4.538(5)	-4.543(1)

Table 2.1: Results for the internal energy U^* and pressure P^* compared with the results from Johnson [30]. R_T is the thermostat radius for which the number of collisions match that of an Andersen thermostat with the same collision probability. All values are in reduced units and the pressure and energy include tail corrections.

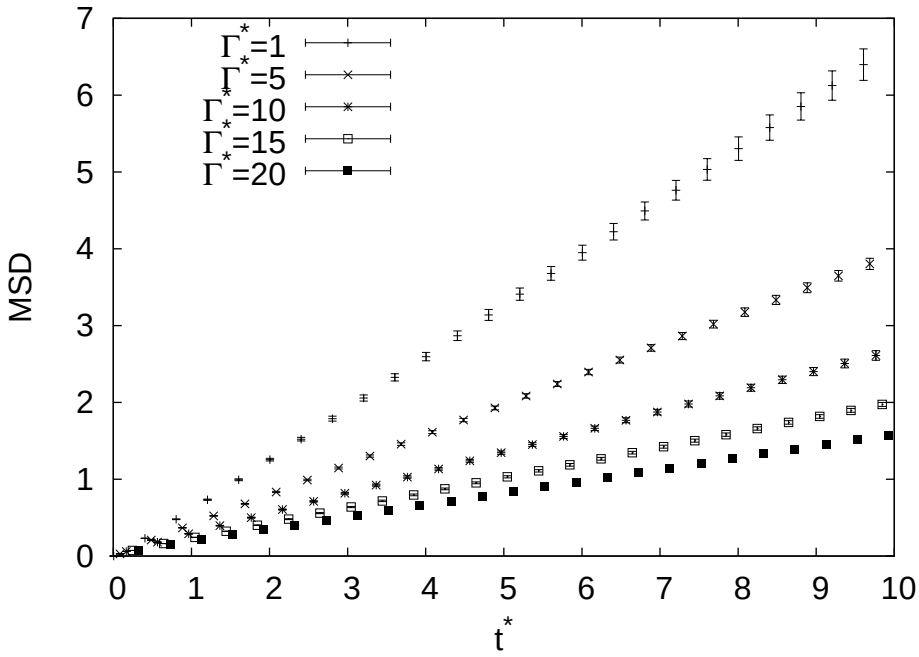

Figure 2.2: Mean square displacement for low collision frequencies Γ^* using the Andersen thermostat. The state point is the one considered in [5] (T^*=2.0, ρ^*=0.8442)

2.1.4 Conclusions

We showed that the Lowe-Andersen thermostat can be used in Molecular Dynamics simulations. The method has the nice features of the Andersen thermostat, locality and simplicity. Furthermore, the method is Galilean in-

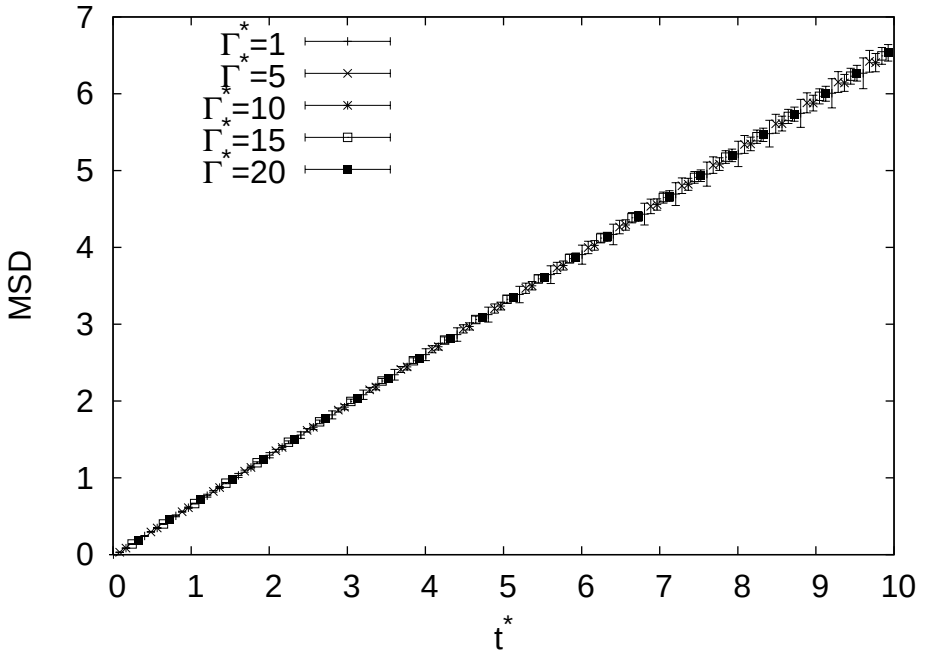

Figure 2.3: Mean square displacement for low collision frequencies Γ^* using the Lowe-Andersen thermostat. The state point is the one considered in [5] (T^*=2.0, ρ^*=0.8442)

variant and this, as Soddemann *et al.* pointed out [17], is advantageous in non-equilibrium simulations. Moreover, it has a significant advantage over the Andersen thermostat even for the equilibrium simulations considered here. At high thermostatting rates, where the system rapidly samples the canonical ensemble, it does not significantly slow the configurational dynamics. It therefore circumvents a known limitation of the Andersen approach. We should add that here we considered simulations of relatively dense fluids for which a thermostat radius somewhat larger than the particle diameter yields sufficient thermostat collisions. Minimising this radius is important because it minimises the extra contribution the thermostat makes to the viscosity of the system. If the system has a significantly lower (gas-like) density this will not be possible. In this regime the approach described by Stoyanov and Groot [18] is necessary. However, as the majority of MD simulations study the fluid (and indeed solid) regime, the simple method described here has widespread applicability.

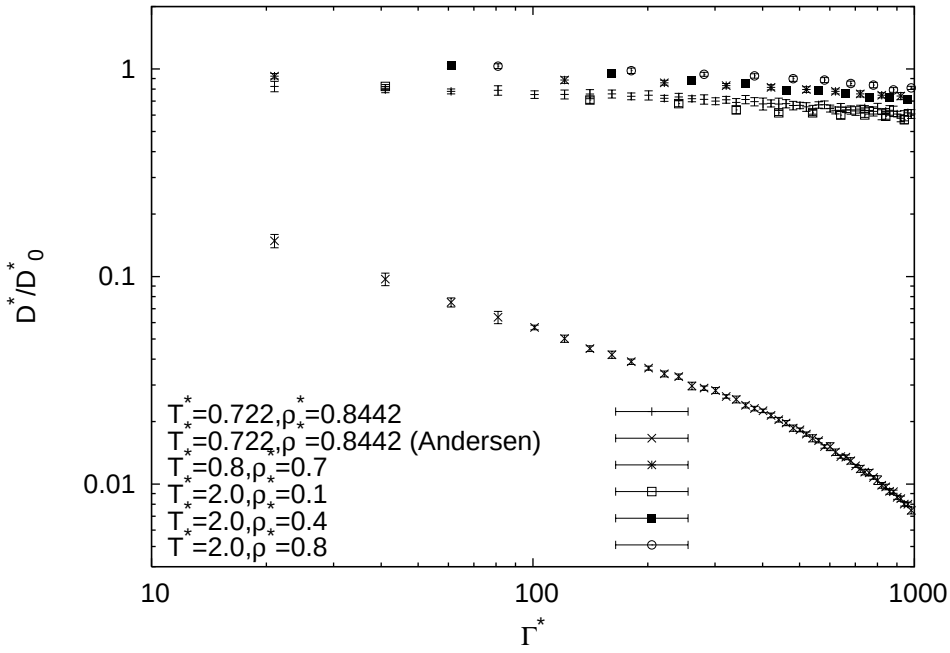

Figure 2.4: Diffusion coefficient D^* normalised by the true (unthermostatted) diffusion coefficient D_0^* as a function of the collision frequency Γ^*. Results for the Lowe-Andersen thermostats at all five state points are shown. The Andersen thermostat results at one state point (T^*=0.722, ρ^*=0.8442) are shown for comparison

2.2 An adapted thermostat

In the second section of this chapter we propose a simpler version and an alternative approach for the Lowe Andersen thermostat. First we will treat the simpler version. Also in this section wherever we present results and ommit units, we have chosen units such that the interaction radius $r_c = 1$, the temperature $k_B T = 1$ and particle masses are $m = 1$. This choice fixes the unit of time implicitly. The radius r_c is functionally identical to R_T from the previous section, but R_T is chosen through an iterative procedure, while r_c gets it's value by construction.

2.2.1 A simpler Lowe-Andersen scheme

The Lowe-Andersen thermostat is constructed in such a way that it conserves both linear and angular momentum in the pair collisions. It is relatively easy to see why conservation of linear momentum is required. We want a particle

system to have correct hydrodynamic behaviour at long length scales. So we must make assumptions that are analogous to the assumptions leading to the Navier-Stokes equations. These are conservation of mass, momentum and energy, so obviously we also want the thermostat to conserve linear momentum. Energy conservation will not be required as the thermostat acts as a coupling to an external heat bath.

From a theoretical point of view, microscopic conservation of angular momentum is not necessary to recover correct fluid like behaviour [31]. If the microscopic angular momentum is randomly oriented, then the stress tensor is symmetric and the angular momentum of a continuum fluid is no longer independent of the linear momentum equation. That is, the angular momentum equation of the fluid motion on long length-scales arises from the linear momentum equation. This was also confirmed by computer simulation [32]. Consequently, this requirement for the Lowe-Andersen thermostat is for many cases unnecessarily restrictive. If we remove the need for conservation of angular moment and apply this to the Lowe-Andersen thermostat, we can change the velocity update procedure to just reset the relative velocities and not the relative velocities along the lines of centre. So we change

$$\begin{aligned} \vec{v}_i^*(t) &= \begin{cases} \vec{v}_i(t) & \Gamma\Delta t < \zeta_1 \\ \vec{v}_i + \frac{\mu_{ij}}{m_i}\left(\lambda - \left(\vec{v}_i - \vec{v}_j\right)\cdot\hat{\sigma}_{ij}\right)\hat{\sigma}_{ij} & \Gamma\Delta t \geq \zeta_1 \end{cases} \\ \vec{v}_j^*(t) &= \begin{cases} \vec{v}_j(t) & \Gamma\Delta t < \zeta_1 \\ \vec{v}_j - \frac{\mu_{ij}}{m_j}\left(\lambda - \left(\vec{v}_i - \vec{v}_j\right)\cdot\hat{\sigma}_{ij}\right)\hat{\sigma}_{ij} & \Gamma\Delta t \geq \zeta_1 \end{cases} \end{aligned} \tag{2.15}$$

to

$$\begin{aligned} \vec{v}_i^*(t) &= \begin{cases} \vec{v}_i(t) & \Gamma\Delta t < \zeta_1 \\ \vec{v}_i + \frac{1}{2}\left(1 - \frac{\sqrt{6k_bT}\delta}{|\Delta\vec{v}_{ij}|}\right)\Delta\vec{v}_{ij} & \Gamma\Delta t \geq \zeta_1 \end{cases} \\ \vec{v}_j^*(t) &= \begin{cases} \vec{v}_j(t) & \Gamma\Delta t < \zeta_1 \\ \vec{v}_j - \frac{1}{2}\left(1 - \frac{\sqrt{6k_bT}\delta}{|\Delta\vec{v}_{ij}|}\right)\Delta\vec{v}_{ij} & \Gamma\Delta t \geq \zeta_1 \end{cases} \end{aligned} \tag{2.16}$$

where δ is a random Gaussian variable with mean 0 and standard deviation 1 and ζ_1 is again a uniform random variable between 0 and 1.

2.2.2 An alternative Lowe-Andersen scheme

We can also take yet another different approach to the Lowe-Andersen thermostat. Instead of rescaling the velocity along the line of centres, we can rescale the velocity component not in that direction. So instead of using 2.15, we can write the update scheme as follows. For simplicity we assume both particles have equal mass.

We start with defining the perpendicular velocity vectors

$$\vec{v}_{i\perp} = \vec{v}_i - \vec{v} \tag{2.17}$$
$$\vec{v}_{j\perp} = \vec{v}_j + \vec{v} \tag{2.18}$$

with

$$\vec{v} = \frac{1}{2}\left(\left(\vec{v}_i - \vec{v}_j\right) \cdot \hat{\sigma}_{ij}\right)\hat{\sigma}_{ij} \tag{2.19}$$

Now the relative perpendicular velocity vector is defined as

$$\vec{v}_\perp = \vec{v}_{i\perp} - \vec{v}_{j\perp} \tag{2.20}$$

With this we can write the velicoty update scheme as

$$\begin{aligned}
\vec{v}_i^*(t) &= \begin{cases} \vec{v}_i(t) & \Gamma\Delta t < \zeta_1 \\ \vec{v}_{i\perp} + \frac{1}{2}\vec{v}_\perp + \frac{\vec{v}_\perp}{|\vec{v}_\perp|}\sqrt{4k_B\overline{T}}\delta + \vec{v} & \Gamma\Delta t \geq \zeta_1 \end{cases} \\
\vec{v}_j^*(t) &= \begin{cases} \vec{v}_j(t) & \Gamma\Delta t < \zeta_1 \\ \vec{v}_{j\perp} - \frac{1}{2}\vec{v}_\perp - \frac{\vec{v}_\perp}{|\vec{v}_\perp|}\sqrt{4k_B\overline{T}}\delta - \vec{v} & \Gamma\Delta t \geq \zeta_1 \end{cases}
\end{aligned} \tag{2.21}$$

with δ a random Gaussian variable with mean 0 and standard deviation 1. Please note the factor 4, we are only re-thermalizing 2 degrees of freedom for 2 particles since we keep the component along the line of centers.

2.2.3 Results

We now have 3 "flavours" of the Lowe Andersen thermostat, the original one, a simplified version (that rescales relative velocities) and an adapted version (that works on the perpendicular part of the velocity). We will refer to the 3 thermostatting methods as "Lowe-Andersen", "Relative" and "Perpendicular" respectively. To examine the differences between the three methods we calculate the Schmidt number Sc defined as $Sc = \frac{\eta}{D}$ for some systems using exactly the same parameters and only changing the thermostat. The Schmidt number is a dimensionless constant that gives the ratio of the momentum diffusivity over the mass diffusivity.

For DPD particle systems a known problem is that the high Schmidt numbers required for the liquid regime ($O(10^3)$) are hard to achieve, and typical simulations are done in the gaseous regime ($O(1)$). So a thermostat that can increase the Schmidt number without changing the static properties is useful. A proper analysis of the behaviour of the Lowe-Andersen thermostat has already been performed [33], and the conclusion is that it is possible to achieve liquid like Schmidt numbers. However it is still useful to investigate the two

alternative schemes to see if they can reach the same Schmidt numbers at lower collision frequencies.

The value for the diffusion is calculated using the slope of the MSD function. The viscosity η is calculated using the method proposed by Backer [34]. In the Backer method we apply two small pressure gradients of equal magnitude but opposite directions to left and right half of the system. As long as the pressure gradient is small we can stay in the low Reynolds number ($\ll 1$) regime and use a known analytical solution for the resulting Poisseuille flow to directly estimate the dynamic viscosity.

As we can see from table 2.3 the increase of the Schmidt number is quite dramatic for all the chosen system parameters (as found in table 2.2). The static properties like the actual temperature and radial distribution function are not perturbed by using this different thermostat. We checked the static properties explicitly but there was no significant difference between the observed values.

Parameter set:	1	2	3
ρ	2.00	5.00	2.00
N_p	500	1000	500
Δt	0.05	0.05	0.05
T	1.0	1.0	1.0
Γ	10	10	1

Table 2.2: Parameter sets used, again all units are such that $k_B T = 1$, the unit of length is the thermostat cut-off radius and the density is the number density.

	Set 1			Set 2			Set 3		
Method	D	η	S_c	D	η	S_c	D	η	S_c
Lowe-Andersen	0.12	1.7	7.5	0.037	12	64	0.50	0.74	0.7
Perpendicular	0.035	7.7	111	0.022	52	477	0.24	1.17	2.4
Relative	0.030	9.7	160	0.022	63	571	0.21	1.45	3.4

Table 2.3: Schmidt numbers S_c for the three thermostats described, together with the observed dynamic viscosities η and diffusion coefficients D. For clarity the methods are ordered by increasing Schmidt number values.

While the diffusion is a bit lower for both the "perpendicular" and "relative" thermostats, the high increase in dynamic viscosity makes for a very significant increase in the Schmidt number.

2.2.4 Conclusion

In this section we introduced two new thermostatting schemes and performed some simple tests to validate their usage. We did not find any differences in static properties, only a large impact on the dynamic properties. As both schemes are also slightly simpler to use, they might make for good alternative candidates to the Lowe-Andersen thermostat in some cases where the artificially slowed down mass transfer dynamics (e.g. reduced diffusion) are an acceptable trade-off for the highly increased momentum transfer dynamics.

2.3 Thermostat interaction ordering

In this last section we will describe a flaw in the way the Lowe-Andersen thermostat works. Because the pairwise operation is not commutative, the ordering in which the operation is applied may matter as the operation introduces correlations between the particle pair it acts upon. We will give a few examples when these correlations become an issue and a simple way of working round this. Again, whenever we don't explicitly specify units in this section, they are chosen such that the thermostat interaction radius $r_c = 1$, the temperature $k_B T = 1$ and particle masses are $m = 1$. This choice fixes the unit of time implicitly.

2.3.1 Theoretical analysis

In the methods described in this work there are a small set of operations that are carried out on the system particles. These operations can be grouped into an integration scheme, continuous force operations and instantaneous operations.

All continuous forces considered are pairwise additive. This means that in a 3 particle system the forced exerted on particle 1 is the sum of the forces exerted by particle 2 in the absence of particle 3 and the force exerted by particle 3 in the absence of particle 2. This is a typical property of systems that are in the classical limit.

The integration schemes change a particle's position and velocity, but only as a function of the particle's own properties. This implies that the order in which particles are treated in the integration scheme and continuous force parts of the simulation code will not matter. One can update for example the particle positions in order of the particle number, sorted by position or completely at random.

For instantaneous operations there might be a problem, especially if the operation somehow correlates the new state of a particle with its old one and that of a different particle. Now the order of the operations carried out will

matter. As an example we will consider the operation A where we assign the average of property p to particle 1 and 2 for a system of 3 particles. Out of the 6 possible orders to do operation A we will pick 2, $A(1,2), A(2,3), A(3,1)$ and $A(2,3), A(1,2), A(3,1)$ and consider the final value for particle 1. Using the first order we will end up with the value $\frac{3}{8}p_1 + \frac{3}{8}p_2 + \frac{1}{4}p_3$ while doing it in the second order we get the value of $\frac{1}{4}p_1 + \frac{3}{8}p_2 + \frac{3}{8}p_3$.

While the calculation of A is a trivial example, it should be clear that similar problems may appear in simulations with operations like this. In this work we often use the Lowe Andersen thermostat. This is effectively an operation that instantaneously changes the velocity of a particle pair. In the low collision frequency limit the chance that a single particle gets thermostatted is relatively small, within the interaction volume $V_I = \frac{4}{3}\pi R_I^3$ of a single particle there are typically $N_I = \rho V_I$ particles. The chance P that a single particle then has more than 1 interaction is then

$$P_{>1} = 1 - (1 - \Gamma\Delta t)^{N_I} - \frac{(N_I - 1)!}{N_I!}(1 - \Gamma\Delta t)^{(N_I - 1)}\Gamma\Delta t$$

.

For most simulations done here $P_{>1}$ is close to 1 for the parameters used, so it is useful to consider the effect of ordered collisions on the system. As a model system to verify that the problem does indeed occur with the thermostat we put 3 particles in a box within their cut-off radius and ran a regular simulation where we calculated the velocity autocorrelation function 2.7 for all 3 particles individually. During the simulation we do not actually update the positions to keep all of them within their respective cut-off radii. As we can in figure 2.5, there is indeed a significant effect in this small model system. We also checked the temperature of the system, both the average for the entire system and per particle. In all cases the observed temperature was within the error margins of the set temperature. In figure 2.6 we also show the averaged velocity autocorrelations functions for the 3 particles in both cases. There appears to be no discernible difference between the two. This implies that system averages will not be directly influenced by the ordering effect.

If we also consider a slightly larger system with 5 particles (2.7,2.8) we see the same thing. Ordering collisions results in different velocity autocorrelation functions for the different particles. Another way of illustrating the ordering problem is by rewriting the Lowe Andersen thermostat in slightly simpler way where we do not conserve angular momentum, we will go into more detail on this in a later section. The simplest way to describe the thermostat is then

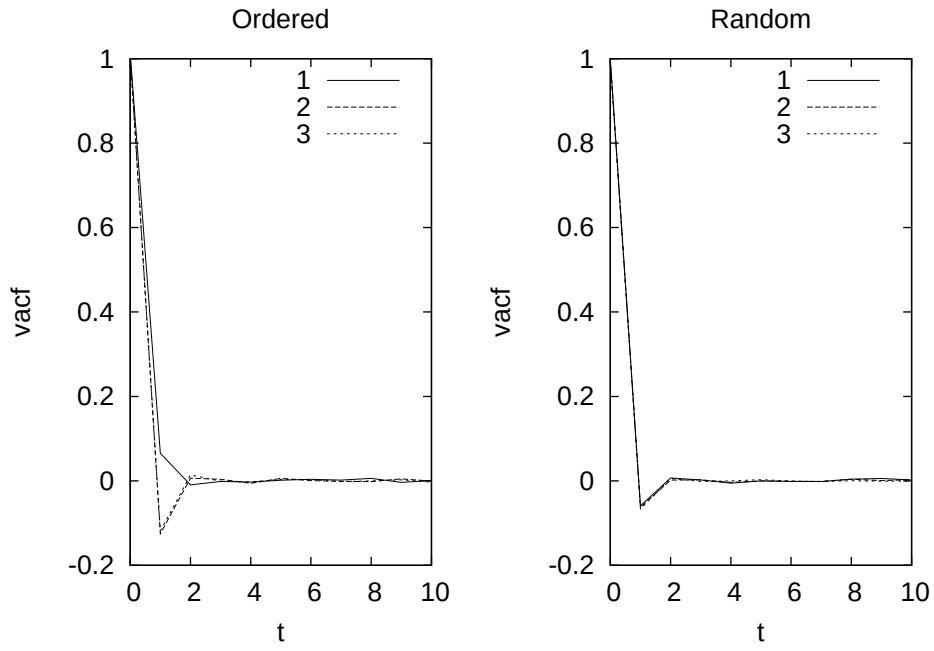

Figure 2.5: Normalized one component velocity autocorrelation functions for a small 3 particle system. The left one is for ordered collisions, the right one for random collisions.

$$\begin{aligned}
\vec{v}_i^*(t) &= \begin{cases} \vec{v}_i(t) & \Gamma\Delta t < \zeta_1 \\ \vec{v}_i + \frac{\vec{v_i}-\vec{v_j}}{2|\vec{v_i}-\vec{v_j}|} + \frac{\gamma(\vec{v_i}-\vec{v_j})}{2|\vec{v_i}-\vec{v_j}|} & \Gamma\Delta t \geq \zeta_1 \end{cases} \\
\vec{v}_j^*(t) &= \begin{cases} \vec{v}_j(t) & \Gamma\Delta t < \zeta_1 \\ \vec{v}_j - \frac{\vec{v_i}-\vec{v_j}}{2|\vec{v_i}-\vec{v_j}|} - \frac{\gamma(\vec{v_i}-\vec{v_j})}{2|\vec{v_i}-\vec{v_j}|} & \Gamma\Delta t \geq \zeta_1 \end{cases}
\end{aligned} \tag{2.22}$$

This scheme effectively rescales the relative velocity for two particles to be from a Gaussian distribution. The parameter γ is a random Gaussian variable for which the exact width is irrelevant for now, but the mean should be 0. We can rewrite the velocity replacement as a matrix operation M_{ij} that acts on the "super" velocity vector $(v_{ix}, v_{iy}, v_{iz}, v_{jx}, v_{jy}, v_{jz}) == (\vec{v_i}, \vec{v_j})$

$$M_{ij} = \frac{1}{2}\begin{pmatrix} 1-\frac{\gamma_{ij}}{\Delta_{ij}} & 1+\frac{\gamma_{ij}}{\Delta_{ij}} \\ 1+\frac{\gamma_{ij}}{\Delta_{ij}} & 1-\frac{\gamma_{ij}}{\Delta_{ij}} \end{pmatrix}$$

with $\Delta_{ij} = |\vec{v_j}-\vec{v_i}|$. For a 2 particle system this is a relatively trivial operation, but for an N particle system, it gets more complex. Since Δ_{ij} depends on the

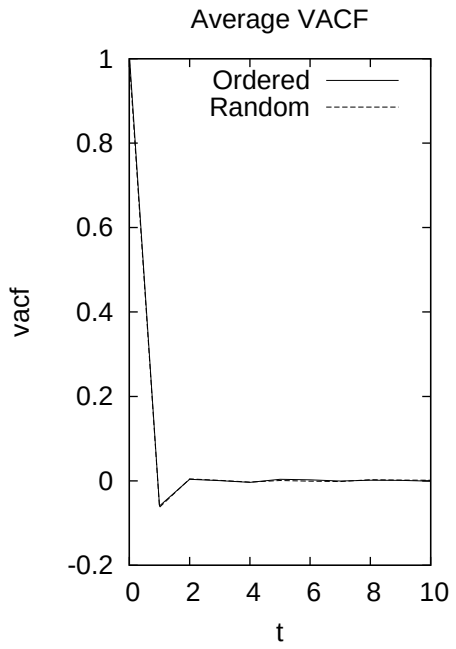

Figure 2.6: Normalized averaged velocity autocorrelation functions for the ordered and random interactions.

the vector the matrix operates on it is not possible to construct one single matrix that calculates all the new velocities in one go in this way. Instead we have to multiply the velocity super vector m times by m different matrices $M_{ij}^{(k)}$ with m the amount of particle interactions in a given time-step and k the k'th interaction. Each of these matrices $M_{ij}^{(k)}$ has elements m_{ab} given by

$$m_{ab} = \begin{cases} \frac{1}{2}\left(1 - \frac{\gamma_{ij}}{\Delta_{ij}}\right) & ((a = i \text{ and } b = i) \text{ or } (a = j \text{ and } b = j) \\ \frac{1}{2}\left(1 + \frac{\gamma_{ij}}{\Delta_{ij}}\right) & ((a = i \text{ and } b = j) \text{ or } (a = j \text{ and } b = i) \\ 1 & (a = b \text{ and } (a \neq i \text{ and } a \neq j) \\ 0 & \text{otherwise} \end{cases} \quad (2.23)$$

Each of these matrices $M_{ij}^{(k)}$ contains a unique Δ_k that only depends on the matrices $M_{ij}^{(l)}$ with $l < k$. We can analyse a single matrix $M_{ij}^{(k)}$ and get the eigenvalues to be 1 and $\frac{\gamma}{\Delta_k}$. This implies that a single collision leaves the velocity distribution invariant on average as γ and Δ_k both come from the same Gaussian distribution.

Now depending on the (arbitrary) ordering of pair collisions that determine

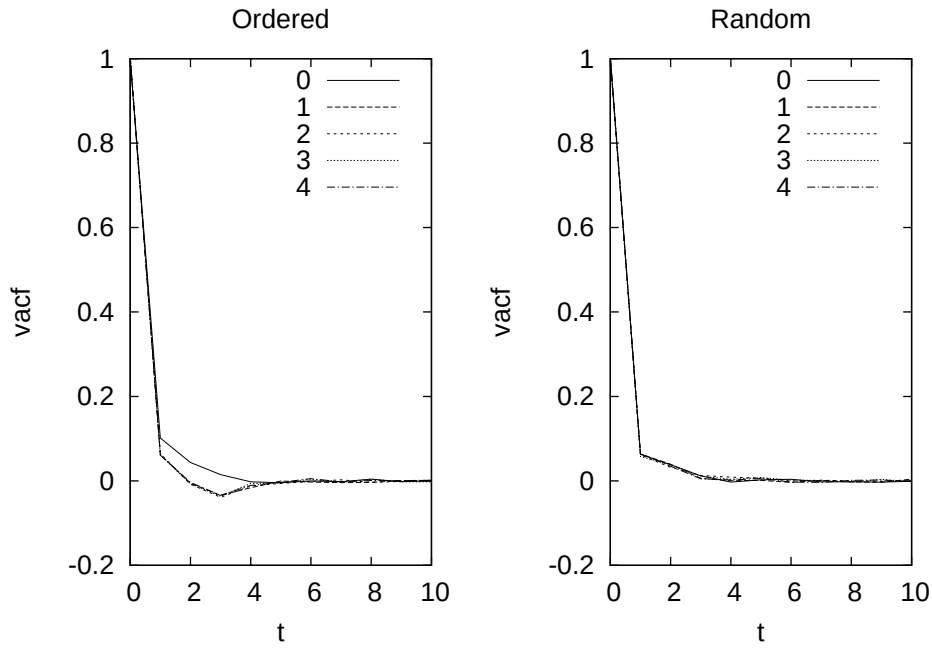

Figure 2.7: Normalized one component velocity autocorrelation functions for a small 5 particle system. The left one is for ordered collisions, the right one for random collisions. The only interactions done are 1-2, 2-3, 3-4, 4-5 and 1-5.

the application of the M_k's on the super velocity vector we can introduce correlations. We will not treat the analytical approach in much detail, as figure 2.5 already indicates that the effect can be significant.

Without actually modifying the Lowe-Andersen thermostat, as that has been proven to work sufficiently well in multiple systems, we can apply a symmetry argument to work around the ordered collision ordering. Assuming there is a measurable effect on the velocity autocorrelation function of a single particle, we want to make sure that each particle suffers from the ordering effect in the same way. If we do not have any details about the system implementation, the only way to enforce this is by having all collisions occur in a random order. Random ordering ensures that no particle *a* will always have a velocity update before particle *b*. Another argument why collisions should be random is by looking at the underlying process we try to model. Particles interact with each-other at random times that do not have to occur exactly at the discrete points in time where we describe the system. Only by treating the continuous range of events at fixed intervals do we introduce the artefacts described above.

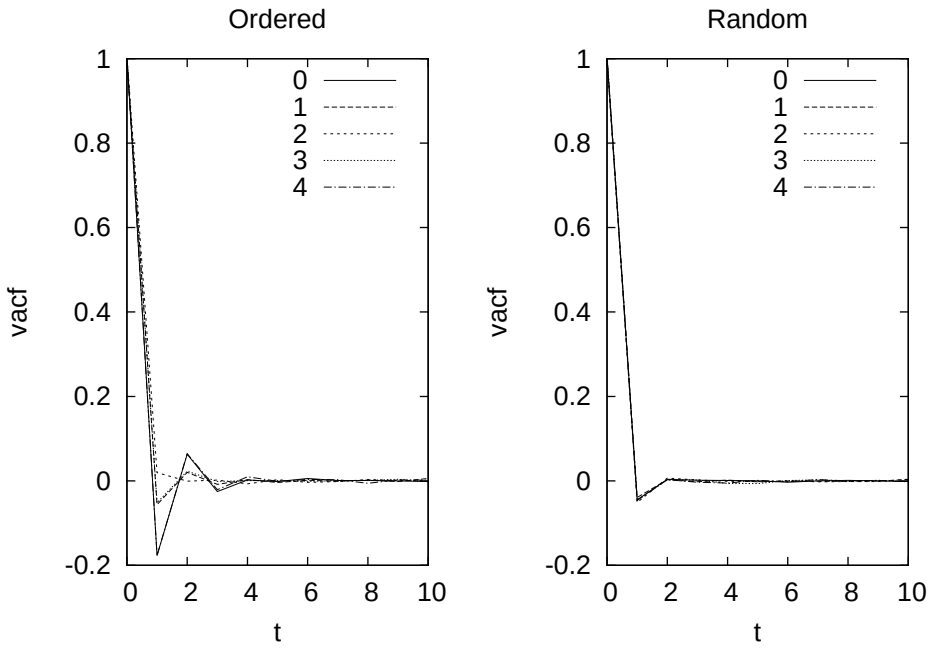

Figure 2.8: Normalized one component velocity autocorrelation functions for a small 5 particle system. The left one is for ordered collisions, the right one for random collisions. All 10 possible interactions are done.

To randomize the interactions, the following procedure can be followed. First decide if you will update the velocity of two particles due to a thermostat interaction (the $\Gamma \Delta t \geq \zeta_1$ case in equation 2.3). If so, instead of constructing new velocities, add two references to the particles as a tuple to a list and continue finding the next candidate pair. After you are done treating all the pairs in this way, apply a list randomization algorithm to the list of tuples. A simple and effective algorithm is "Algorithm 235: Random permutation" [35]. With the new ordering of pairs, go back and apply the velocity update scheme as usual (2.3 and assume $\zeta_1 = 0$).

A simple check on an existing system is to see if it contains particle collisions that are always in a specific order and a particle has on average more than 1 collision per time-step. An example of this would be a linear polymer chain in a solvent where the monomers making up the chain are always thermostatted pairwise from one end to the other with a high collision frequency. The first bead in the chain can then have a significantly higher diffusion coefficient than the other beads. The higher diffusion in turn may alter ensemble averages of the entire polymer. More precisely, consider the following straightforward implementation of a particle system. The particle system

consists of N_s solvent particles and 1 polymer consisting of N_p monomer particles linked together to form 1 polymer molecule. The simplest way of actually modelling this is by having $N = N_s + N_p$ particles numbered 1 to N. The first N_p particles are linked together to form the single polymer chain, so during force calculation some spring force is applied to all particles pairs $(i, i+1)$ with $i < N_p$. Pair interactions when done in the simplest way possible is to check for each particle i if there is a particle $j > i$ within the interaction range, and then possibly applying the interaction. With this straightforward implementation now consider particle 1, one head of the polymer chain. The thermostat interactions with particles 2 and 3 will always occur in the same order. As shown earlier this will decrease the effective diffusion of this particular monomer. Because the rest of the polymer chain is attached to the head monomer, the complete dynamics of the polymer chain may be altered. Even when the thermostat interactions are forbidden between particles making up the polymer and all thermostatting is provided through the solvent-monomer and monomer-monomer interactions the head particle with index 1 is (on average) always thermostatted before the particle with index 2, possibly still introducing a slow-down of the dynamics of particle 1.

In most systems however, the ordering issue should not occur. As the Lowe-Andersen thermostat is mostly applied to solvents and gasses, particles are not expected to be in each other's interaction radius for long times. Another simple test will be to change the time-step size while keeping the collision rate per time constant. Dynamic properties like the diffusion should not change in this case. Further research will have to quantify exactly how measurable the ordering effect is, but if a model is free of long-time systematic ordering we expect no issues relating to interaction ordering.

2.3.2 Parallel implementations of the Lowe Andersen thermostat

It is very common for scientific programs to run on multiple computers at once, this is usually referred to as parallel code. It is called parallel because multiple processing units are working on small pieces of the same problem in parallel.

There are a number of ways to distribute the work for a single problem over multiple computers. The most favourable and easy one is so called "farming". Farming means that the very same problem is solved multiple times with different starting points from the same ensemble and the measured observables are averaged over all simulations. For almost all of the experiments done in this work farming is a sufficient way to parallelize the workload. The obvious benefits are perfect scalability, i.e. twice the numbers of computers results in two times more usable data, cheap computing units (no expensive network is needed) and a trivial algorithm, just start the same simulations

multiple times with a different random number seed.

However, this is not the end of the story. There are many simulations that simply cannot be run in this farming way on a single computer. One might be interested in a simulation of a large system that evolves from one state to the other and takes a very long time on a single computer. Also the problem size might be so large that it does not fit in the computer memory. In both cases it might be worth it to change the algorithm in a such a way that a single simulation runs on multiple computers at once.

There are quite a number of ways to do this. A very intuitive one is domain decomposition. Here the relative position of elements in the simulation is used to group bits that interact often with each other together and have each computer work on such a group of elements. For a particle system one could slice the system into equal sized slabs and have each node work on all particles in one such slab. Interactions between two particles in different domains need to be dealt with explicitly by having two computers negotiate on the operations to perform on the particle pair. While domain decomposition might seem like a very good way of solving a problem in parallel it does have some drawbacks. In case of very long ranged interactions the computers spend most of their time communicating with other computers, this is a very slow operation even on the best performing networks available. This communication part will always cause the efficiency of doing a same problem in this way to be lower than by doing it with farming.

The last way of parallelizing we will consider here is doing it on a very low level. Start with the actual code used, look for the piece that the computer spends most of the time executing. Then try to split this piece of code in separate pieces that do not depend on each other. For example, there might be a code block where the values in a large array $\vec{x}$ are multiplied by some constant value, immediately followed by values in array $\vec{y}$ being multiplied by some constant value. One could then distribute the workload over 2 processors, one could multiply the values in array $\vec{x}$, the other in array $\vec{y}$.

After this short introduction to parallel code we will attempt to explain what the problem is with implementing the Lowe-Andersen thermostat in a parallel program and how we think this implementation can be done in an efficient way. The interactions caused by the thermostat are completely local at first glance. Just two particles interact only within a fixed distance, therefore domain decomposition should be a trivial task. This however, is not the case. Because a thermostat collision correlates the velocities of two particles, a particle that subsequently interacts with either of those two gets weakly correlated with the other of the pair. Taken to the extreme, particles that are at the largest possible distance in the system might still get correlated in a single iteration. This effectively means that the thermostatting algorithm cannot be efficiently implemented using domain decomposition in the case

where all the thermostat interactions have to be done in 1 very specific order. Luckily this is an extreme example, as we will show here.

Let us first treat a thermostat interaction between two particles as a connection between those two. So if two particles a and b both only interact with c, they are still (indirectly) connected. Using this approach we can group together all the interactions, in the extreme limit there will be just 1 group with all interactions.

We can make a very rough estimate of the transition to one connected group in terms of collision frequency and particle density. If we have a particle density ρ, a point in the system has on average $\frac{4}{3}\pi\rho r_{rc}^3$ possible interactions with neighbours. Out of those possible interactions, only a fraction $\Gamma\Delta t$ will actually interact. So the interaction density will be approximately $\frac{4}{3}\pi\rho r_c^3\Gamma\Delta t$. A very rough estimate of a "percolation threshold" is that it will only happen if the amount of interactions in a volume r_c^3 is higher than one. So a first order approximation of a requirement for efficient parallelization would be

$$\frac{4}{3}\pi\rho r_c^6\Gamma\Delta t < 1$$

Filling in some numbers that we typically use for simulations ($\rho \approx 3, \Delta t \approx 0.01, r_c^3 = 1$) we get roughly $\Gamma < 12$; this is a very reasonable condition in most simulations.

So typically it is possible to group the interactions together in a high number of groups. If the number of groups is larger than the number of computers this work can in theory be done efficiently in parallel. We also have to remember the collision ordering randomization requirement from the previous section. The conclusion is that the order of interactions must be randomized. For a large system randomization can be mistaken to randomize all of the interactions. This is not directly necessary. If two groups of particles do not interact at all in a given iteration, there is no need to treat the two groups in any particular order, so you only have to randomize interactions within the groups themselves. So grouping allows you to split the workload for treating different groups over multiple computer nodes.

Using a grouping approach it should be clear that efficient parallelization of the thermostat is directly related to networks and percolation mentioned in Chapter 4.1. Way above the percolation threshold the interactions are all long ranged, only below this threshold it is possible to evenly distribute the workload for the thermostat over all computers. We have not actually implemented a parallel version of the Lowe-Andersen thermostat in our simulation code. The reason for not building a parallel implementation is, as mentioned earlier, that in many cases you are actually interested in ensemble averages. By simulating multiple instances of the same system in different configurations in parallel you can get the desired results without the need for complex parallel

computation methods. Only when you're interested in a long time evolution of a single process, say the mixing dynamics of two initially separated fluids you may want to parallelize your simulation code.

Chapter 3

DPD simulations at constant pressure

3.1 Introduction

Specifying the pressure in particle simulations involves using a "barostat" that in some way couples the system to an infinite external system with a fixed pressure. The system then samples the Isobaric-Isothermal (NPT) ensemble. Here we will show a way of sampling the NPT ensemble in DPD simulations using a technique similar to the way it is often done in Molecular Dynamics. We will then use this method to test a simple 2 component equation of state for a mixed DPD fluid.

Since most real experiments are done at constant pressure, temperature and number of particles, this is often the most logical choice for a simulation as well. Unfortunately typically particle methods by themselves have a constant volume, energy and number of particles. Sampling in another ensemble requires an explicit modification of the method. In Chapter 2 we have studied ways of keeping the temperature of a system fixed. Here we will implement a way of also keeping the pressure constant in a DPD simulation.

3.2 Dissipative Particle Dynamics

The Dissipative Particle Dynamics (DPD) method was introduced in 1992 by Hoogerbrugge and Koelman [7]. Their model was improved in 1995 when Español and Warren [8] put it on a solid theoretical foundation by getting the thermodynamic equilibrium properties right. The method itself is a mesoscopic particle method that is well-suited for multi-component fluid simulations, and specifically ones where the dynamics itself are interesting. By mesoscopic we mean in this context that the simulations capture the fluctuations with small

scale physics, but individual particles making up the system do represent many atoms or molecules. An oft overlooked property of DPD is that it has a relatively simple equation of state. This simple equation of state allows a theorist to work out analytical solutions to some problems that can be checked with computer simulations in a relatively simple way. The simple equation of state can be placed in between the overly simple ideal gas equation of state and the relatively complex Lennard-Jones equation of state. Both equations of state also have a corresponding simulation technique, but the ideal gas fails to exhibit phase transitions, while a Lennard-Jones equation of state can have over 32 terms making it very cumbersome to work with. In contrast, a simple DPD equation of state has only two terms while still exhibiting phase transitions.

With these properties in mind, it is useful to explain the actual method in a bit more detail. The method consists of 3 parts, a set of forces that enforce the correct static thermodynamic properties, an integration scheme that takes care of the correct time evolution of the system and finally a set of boundary conditions. The forces in DPD are usually given as 3 separate parts. First the conservative force f_c

$$\vec{f_c} = \begin{cases} a_{ij}(1 - \frac{r_{ij}}{r_c})\hat{r_{ij}} & \text{if } r_{ij} < r_c \\ 0 & \text{if } r_{ij} > r_c \end{cases}$$

Here a_{ij} is a function of the particle types involved in a particle pair interaction. The distance between particles i and j is r_{ij}, the force cut-off radius r_c and $\hat{r_{ij}}$ is the unit distance vector between i and j. The a_{ij} parameter can be used to tune the interaction between different fluids in the system. Phase transitions in the system occur because of this force, setting it to 0 reduces the system to an ideal gas. Second we have the random force, that together with a dissipative force acts as the system thermostat. The random force $\vec{f_r}$ consists of a tunable magnitude σ, a distance dependent weight ω_R and a randomly chosen variable from a Gaussian distribution with unit variance Θ_{ij} chosen to be

$$\vec{f_r} = \begin{cases} \sigma\omega_R(r_{ij})\Theta_{ij}\hat{r_{ij}} & \text{if } r_{ij} \leq r_c \\ 0 & \text{if } r_{ij} > r_c \end{cases}$$

Finally the dissipative force $\vec{f_d}$ force has the following form:

$$\vec{f_d} = \begin{cases} -\gamma\omega_D(r_{ij})(\hat{r_{ij}} \cdot \vec{v_{ij}})\hat{r_{ij}} & \text{if } r_{ij} \leq r_c \\ 0 & \text{if } r_{ij} > r_c \end{cases}$$

Here γ is a tunable friction coefficient and ω_D is a distance dependent weight. The random and dissipative forces balance the system to be at a temperature $k_B T = \frac{\sigma^2}{2\gamma}$ if $\omega_D(r_{ij}) = \big(\omega(r_{ij})\big)^2$. For simplicity the weight function $\omega_R(r_{ij})$ is

usually chosen to be

$$\omega_R(r_{ij}) = \begin{cases} 1 - \frac{r_{ij}}{r_c} & \text{if } r_{ij} \leq r_c \\ 0 & \text{if } r_{ij} > r_c \end{cases}$$

The total force on a single particle i is now given by

$$\vec{f}_i = \Sigma_{i \neq j} \left(\vec{f}_c(r_{ij}) + \vec{f}_r(r_{ij}) + \vec{f}_d(r_{ij}) \right)$$

. Note that although this is a double summation, all three force components have a limited range, so the simulation remains manageable even for large amounts of particles.

Now the equations of motion also have to integrated. There are some options here, but for strict DPD using the random and dissipative forces a scheme that has no velocity dependent forces is useful. A modified velocity-Verlet scheme is typically used. Here the equations of motion are integrated as follows, using g as a shorthand for the calculated forces based on the new particle positions:

$$\vec{r}_i(t + \Delta t) = \vec{r}_i(t) + \Delta t \vec{v}_i(t) + \frac{1}{2}(\Delta t)^2 \vec{f}_i(t) \quad (3.1)$$

$$\vec{v}_i^*(t + \Delta t) = \vec{v}_i(t) + \lambda \Delta t \vec{f}_i(t) \quad (3.2)$$

$$\vec{f}_i(t + \Delta t) = g(\vec{r}_i(t + \Delta t), \vec{v}_i^*(t + \Delta t)) \quad (3.3)$$

$$\vec{v}_i(t + \Delta t) = \vec{v}_i(t) + \frac{1}{2}\Delta t(\vec{f}(t) + \vec{f}_i(t + \Delta t)) \quad (3.4)$$

Here the velocity v_i^* is a guess for the actual velocity at $t + \Delta t$. A suggested optimal value for this the value of λ is 0.65.

Finally there is the question of how to handle particles near the edge of a simulation box. This is an area of active research, as it is apparently quite difficult to simulate a wall that has all the expected properties correctly. Fortunately we are only interested in bulk properties of the system, so we choose to use simple periodic boundaries. This means that if at any point a particle in a system with with some pre-defined dimensions moves outside of the simulation box, the particle is translated to the other side of the system. Forces are also calculated over the boundary if that results in a shorter distance.

For simplicity in the rest of this chapter, we take the force cut-off radius r_c to be 1, the particle masses to be 1 and the temperature in units such that $k_B T = 1$. This implicitly fixes the unit of time as well.

3.3 Barostat

While DPD by itself already contains a thermostat (the random and dissipative force), it has no way of controlling the pressure. Therefore many papers report

on results acquired with a canonical ensemble. Since DPD has a very simple equation of state, one could theoretically calculate the density required for a certain pressure and do the simulations at that state-point. There are however some drawbacks to this approach, it can only be done for a one component system. Many DPD simulations are however done with multiple species. Then an iterative procedure is needed to get the right density, or ideally a way to control the pressure.

There are several methods available to perform constant pressure Molecular Dynamics and Monte Carlo simulations. The dynamic approach involves coupling the dynamics of the system to those of a hypothetical piston. These methods have indeed been used to develop constant pressure DPD algorithms. For example, Trofimov et al. [36] used a DPD system coupled to an Andersen barostat [10] while Jakobsen [37] used a Langevin barostat [38]. The problem with using a dynamic method is that the same problem arises as does with constant volume DPD simulations. Namely, that to ensure that detailed balance is satisfied it requires an iterative algorithm. Incorporating the barostat dynamics just represents an additional degree of complexity in an iterative scheme. The Lowe-Andersen thermostat modifies the DPD method so that a simple (non-iterative) scheme for integrating the equations of motion satisfies detailed balance. Here, in the same spirit, we propose a method for constant pressure simulations.

The Lowe-Andersen thermostat satisfies detailed balance because it is a valid Monte-Carlo scheme (or vice versa). The idea here is to combine this with a Monte Carlo constant pressure algorithm, involving small changes in the size of the simulation box. Specifically we use the method of McDonald, described in [5], Chapter 5. This method mixes normal (constant volume) moves and volume changes. Since the Lowe-Andersen thermostat generates, in Monte Carlo terms, valid constant volume displacements the combined method will satisfy detailed balance even with a simple algorithm to update the equations of motion.

Our implementation attempts to update the system volume periodically in the following way. A new volume V' is chosen based on the current volume V by $\ln V' = \ln V + \frac{\sigma - \frac{1}{2}}{10}$. Where as usual σ is a uniform random number in the interval [0..1]. All particles positions are then rescaled to fit inside the new volume, and the new total energy E' is calculated. If our system contains "molecules", e.g. polymers, we only translate the centre of mass of the molecule and do not simply change the individual particle positions. With the new energy and volume values we then calculate the acceptance term accept for the Monte Carlo scheme as

$$\text{accept} = \min\left(1, e^{-\frac{1}{k_B T}((E'-E)+P(V'-V)-(N+1)k_B T \ln \frac{V'}{V})}\right).$$

Then we generate another uniform random number σ in the interval [0..1] and accept the new system volume if $\sigma <$ accept, or we revert to the previous volume otherwise. In the acceptance rule E refers to the (potential) energy of the configuration, P to the pressure and V to the volume, quantities denoted with a prime are the possible new values if the configuration is accepted. Note that N is the number of "molecules" and not strictly the number of particles.

3.4 Results

3.4.1 Constant pressure

To verify if this barostatting method works, we ran a number of DPD simulations with various potentials, temperatures and densities and measured the system pressure. As can be seen in figure 3.1, for all tried configurations the system pressure is equal to the external pressure within the statistical error associated with particle systems. Note that many of the shown configurations are phase separated systems with 2 components with varying densities. This is a good indication the method will also work for even more complex systems.

3.4.2 Mixing and demixing

We also compared the system densities with the densities predicted by a DPD equation of state for a 2 component system. One of the assumptions for this equation of state is that the system is not demixed in two phases with different densities. Many of our configurations are for phase separated systems, so we need a way to filter those configurations without visual inspection.

We can use the structure factor $S(\vec{k})$ for this, this is the Fourier transform of the density-density correlation. This is a quantity that is directly measurable in real life scattering experiments. There are multiple definitions, but one we can use is

$$S(\vec{k}) = \frac{1}{N}\Sigma_i\Sigma_j e^{i\vec{k}\cdot\vec{r}_{ij}} \tag{3.5}$$

where $\vec{k}$ is a wave vector. In a typical experiment one could send a beam with a certain wave vector into a crystal sample, measure the scattered beam and reconstruct the crystal structure from it. The important thing is that if we know the structure factor as a function of $\vec{k}$ we have a good indication of the amount of "structure" on certain length scales (hence the name structure factor). Since we are not doing real experiments but computer experiments, we have to take a slightly different approach. We already know all the particle positions, we can define certain wave vectors and calculate a structure factor, but in general this will not give us much new information. In our case it is useful though. Since the structure factor gives us an indication of the amount of structure

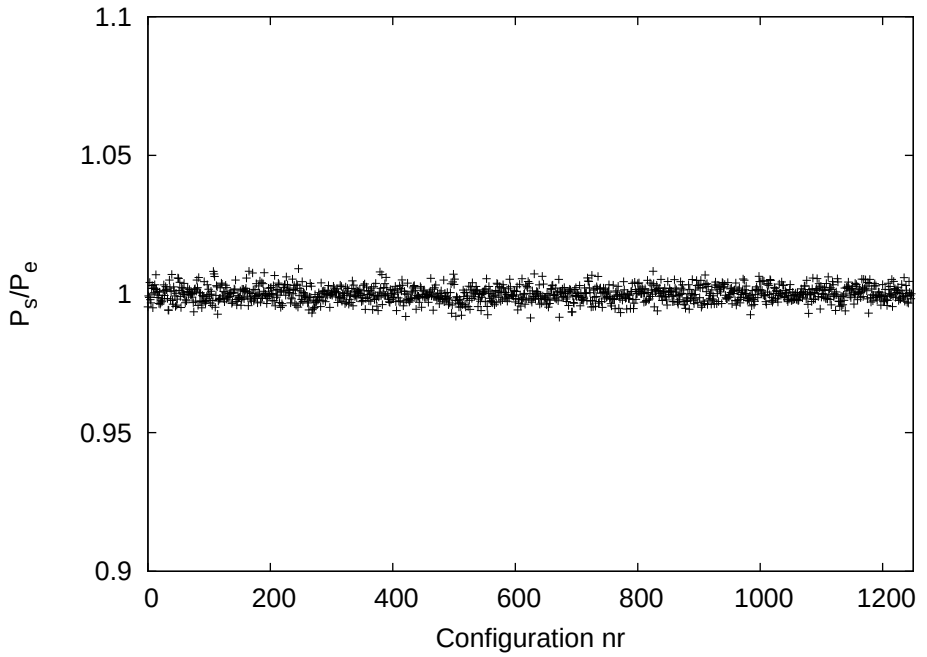

Figure 3.1: Measured pressure P_s normalized to the applied pressure P_e for all permutations of the following parameters: $a_{00}, a_{01}, a_{11} \epsilon [15, 16..20]$ and $P \epsilon [1, 2, 4, 8, 16, 32, 64]$. All configurations consisted of 333 particles of type 0 and 667 particles of type 1. The temperature was set to $1 k_B T$ and all configurations were sampled 800 times during 8000 time steps with $\Delta t = 0.02$ after 2000 equilibration steps.

on certain length scales, we can use it to check whether the solvent particles have structure on length scales much larger than their own size. If there is structure on length scales much larger than the particle size, the solvent is demixed.

We can rewrite the structure factor to

$$S(\vec{k}) = \frac{1}{N} \Sigma_i (\cos^2(\vec{k} \cdot \vec{r_i}) + \sin^2(\vec{k} \cdot \vec{r_i})).$$

(See appendix 10.2). Take note that since we are working with a finite size, periodic system, we should take care that the wave vector always "fits" in the box. We can easily accomplish this by using only vectors in the three principal directions with length $\frac{i2\pi}{L}$ with $i \epsilon [1, 2, 3...]$ and L being the box length. In our case we only use the structure factor value for the largest $\vec{k}$ that will fit in the simulation box, $\vec{k}^*$. So for a (cubic) simulation box with length L, $\vec{k}^*$ will be one of $\{\frac{2\pi}{L}\hat{e}_x, \frac{2\pi}{L}\hat{e}_y, \frac{2\pi}{L}\hat{e}_z\}$. For accuracy we take the average of the three

values $S(\vec{k}^*)$. A secondary benefit of taking the average is that for laminar phase separation, we don't accidentally measure the structure factor parallel to the separating planes.

In this way we can simply verify if the system is phase separated by considering the one component structure factor $S(\vec{k}^*)_{00}$ where we only take into account the particles of one type. We do not take the full structure factor that considers all the particles, as this would not show phase separation effects when the densities for the two particles types are identical. In theory this should be a good measure for the amount of separation for the 2 component system. The value of $S(\vec{k}^*)_{00}$ will be close to 0 for a mixed system, similar to a one component system. This is because a mixed system has no observable structure/density fluctuations on long length scales. It will not be exactly 0 because of finite size effects and statistical errors in the measurement. For a demixed system, the value of $S(\vec{k}^*)_{00}$ will be larger than the structure on the small length scales, which is close to the number density for the particle type we're measuring, ρ_0. As can be seen in figures 3.3, 3.2 and 3.4, the exact value for which we should decide through this procedure if a system is demixed or not is open to interpretation, but something close to ρ_0 is a good estimate.

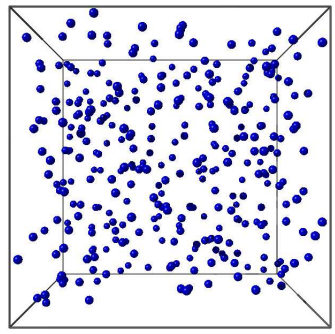

Figure 3.2: Particle system where $S(k^*) \ll \rho_0$

To verify if this is a good choice we can also attempt to estimate the mixing and demixing from the free energy differences between the two states. We start by using the following 2 component equation of state with $\rho = \rho_0 + \rho_1$:

$$P = \rho k_B T + \alpha(a_{00}\rho_0^2 + 2a_{01}\rho_0\rho_1 + a_{11}\rho_1^2), \tag{3.6}$$

and the one component equation of state as found by Groot and Warren:

$$P = \rho k_B T + \alpha a \rho^2. \tag{3.7}$$

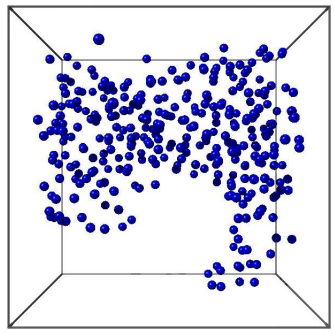

Figure 3.3: Particle system where $S(k^*) \gg \rho_0$

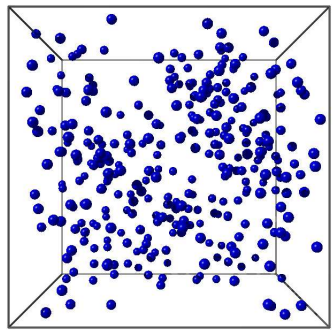

Figure 3.4: Particle system where $S(k^*) \approx \rho_0$

In both equations α is a fitted constant, that is usually found to be 0.101, and a_{ij} is the interaction strength between species i and j. The two-component equation of state is a trivial extension to two components that reduces to the one-component equation of state when $a_{00} = a_{01} = a_{11}$.

For both components we solve the predicted densities ρ for the mixed (ρ_m) and demixed (ρ_d) states $\rho_{m0}, \rho_{m1}, \rho_{d0}$ and ρ_{d1}. Then we calculate the chemical potentials of the 4 cases to be

$$\mu_{m0} = 2\alpha\left(a_{00}\rho_{m0} + a_{01}\rho m1\right) + k_B T \log \rho_{m0}, \tag{3.8}$$

$$\mu_{m1} = 2\alpha\left(a_{00}\rho_{m0} + a_{01}\rho m1\right) + k_B T \log \rho_{m1}, \tag{3.9}$$

$$\mu_{d0} = 2\alpha a_{00}\rho_{0d} + k_B T \log \rho_{0m}, \tag{3.10}$$

$$\mu_{d1} = 2\alpha a_{11}\rho_{1d} + k_B T \log \rho_{1m}, \tag{3.11}$$

and finally use these chemical potentials to calculate the free energy difference between the mixed and demixed case to be

$$\Delta G = N\left(\mu_{m0} + \mu_{m1} - \mu_{d0} - \mu_{d1}\right). \tag{3.12}$$

If $\Delta G > 0$ we expect the system to demix. We can now compare the estimated

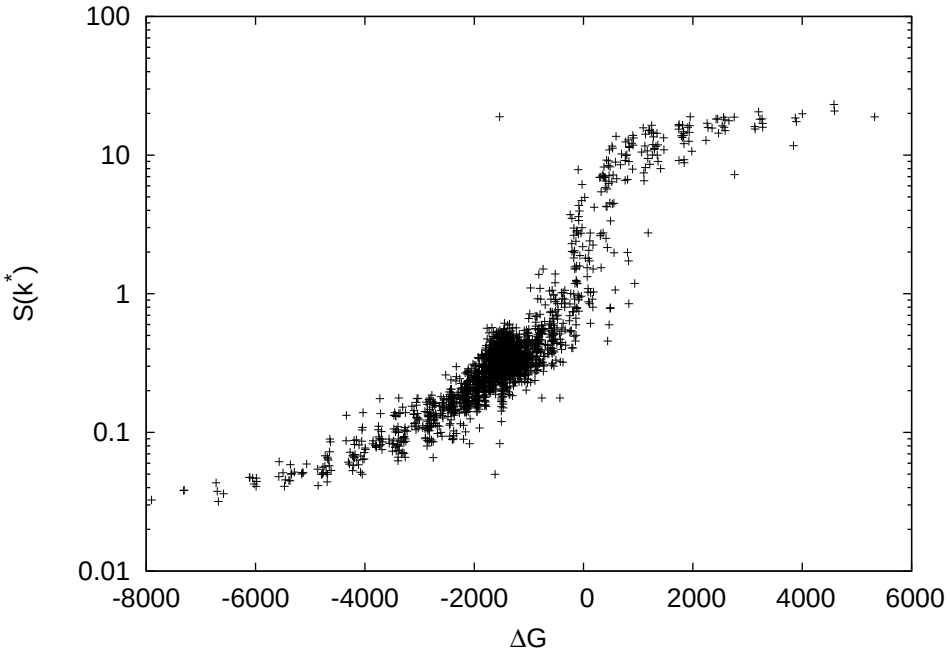

Figure 3.5: Comparison of the free energy difference between the mixed and demixed states ΔG predicted from equation 3.6 with the observed long range structure factor value $S(\vec{k}^*)$.

ΔG with $S(\vec{k}^*)$. As we can see in figure 3.5 there is a direct correlation between the two quantities and choosing $S(\vec{k}^*) > \rho_0$ is indeed a good choice, number densities fluctuate in the various configurations but are typically order 1. Please note that the equation of state is still an estimate only valid for $\rho >> 1$. So our predictions for when the system demixes are not exact. The observed results for the densities at many state points, including an indication if the configuration is phase separated can be found in figure 3.6.

3.4.3 A better equation of state

Remembering the two component equation of state 3.6, we can now compare the measured densities from our simulations with those predicted by the equation of state. Note that when we remove the points corresponding to the

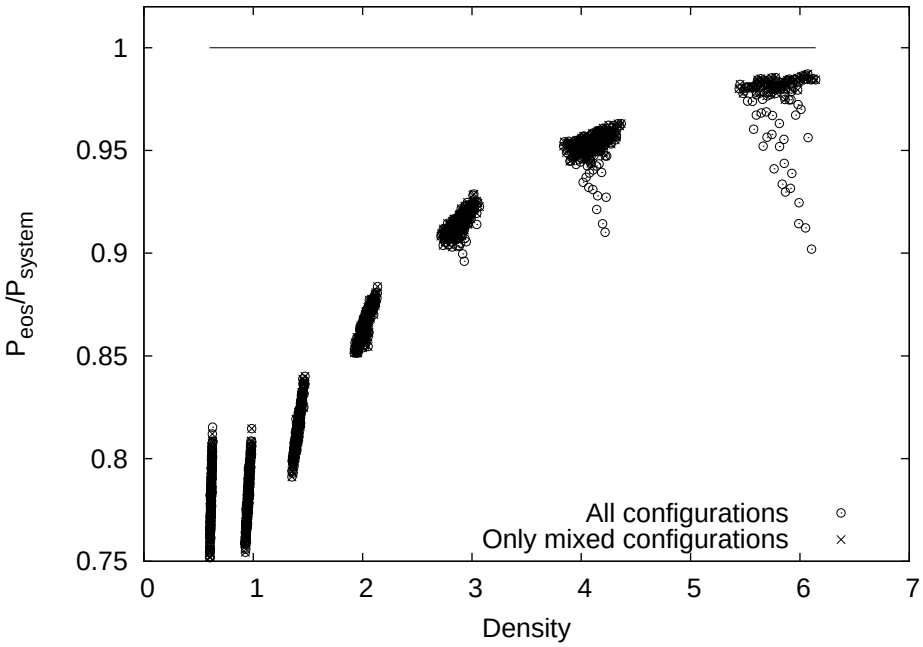

Figure 3.6: Comparison of the theoretical two-component equation of state with the observed densities at fixed pressures. All the simulations were done with the same settings mentioned in figure 3.1. The circles represent configurations that were phase separated, the crosses represent configurations that are mixed. The vertical axis is presented as the predicted pressure P_{eos} normalized to the observed pressure P_{system}. The leftmost blob is at $P_{system} = 1$, the second from the left at $P_{system} = 2$, the third at $P_{system} = 4$ etc.

demixed state and the low densities, we should in principle be able find a function that matches the remaining data points. The original DPD equation of state for a single component has been shown already by Groot and Warren, but only in the limit of ρ going to infinity and more specific $\rho > 3$. The two component equation has the same properties, but we should in principle be able to fix these limits to acquire a better equation of state. To do this we multiply 3.6 with a virial type of equation $1 + \frac{C_1}{\rho} + \frac{C_2}{\rho^2} + \frac{C_3}{\rho^3} + ...$ Here the values of C_i are a priori unknown constants that have to be determined by fitting to experimental data. Combining the virial expansion with the equation of state we get

$$P = (1 + \frac{C_1}{\rho} + \frac{C_2}{\rho^2} + \frac{C_3}{\rho^3} + ...)(\rho k_B T + \alpha(a_{00}\rho_0^2 + 2a_{01}\rho_0\rho_1 + a_{11}\rho_1^2)). \quad (3.13)$$

	1	2	3
C_1	-0.199(2)	-0.389(2)	-0.390(3)
C_2		0.150(8)	0.153(5)
C_3			-0.001(2)

Table 3.1: Fitting coefficients for 3.13. The columns indicate the highest order coefficient used for fitting. There is no significant difference between 2 and 3, so fitting up to and including C_2 should be sufficient.

We can then fit this equation to the data we have, again using just the mixed state data, to get the values listed in table 3.1. As we can see using up to C_2 is good enough to get a good equation of state.

3.4.4 Conclusion

We have been unable to find references to the usage of the barostat described in [5] in a DPD system. After building this, in combination with the Lowe-Andersen thermostat, it works as expected without spurious time-step effects. We then applied it to a two-component system and used it to verify a proposed mean field two component equation of state for DPD. The equation of state is usable to roughly predict if a system will be mixed or demixed. We can also extend the equation of state slightly using a lower order virial expansion and find that with just three terms we can quite accurately describe the system also for low ($\rho < 3$) densities. The major benefit of the barostatted DPD system will be in situations where a simple analytical equation of state is required that still describes complex behaviour. As mentioned earlier, DPD and its equation of state are about the simplest step up from an ideal gas, but the added richness you achieve with multiple species and the simple equation of state make it an ideal tool to allow computer experimentalists to work together with theorists closely on multi-phase problems. Usually an equation of state must be calculated for a given system, but for DPD it can be specified to a very good approximation.

Chapter 4

Network detection in particle systems

4.1 Introduction

Polymer networks form when polymer molecules are linked together in some way. There are three types of this "cross-linking" [39]. Firstly, chemical bonds, which are essentially permanent. Secondly, physical links such as hydrogen bonds or van der Waals attraction. These are often weak and not thermodynamically stable so the pattern of linking is not fixed. Furthermore, the connections between polymer molecules are easily broken by the application of external forces. Thirdly, topological links called entanglements where the polymer molecules are constrained by being wrapped around each-other. These are essentially permanent but evolve dynamically as the point of the "knot" moves along the chain. There is also the possibility that more than one of these cross-linking types exists in some systems. Given this, and the fact that the the linking mechanisms themselves have very different characteristics, materials containing polymer networks display a vast range mechanical and dynamic of behaviour. Another important factor in determining their properties is the proportion of space occupied by the network. Gels typically consist of a low density of polymers dispersed in a liquid (up to 99% liquid by weight). At the other ender of the spectrum, the polymer density of rubbers is much higher such that the polymers occupy all space. For the purpose of this chapter we will not distinguish between the two and just call them networks.

Networks are typically formed in a process where monomers or chains are linked together by a chemical or physical process. This effectively transforms the loose elements into larger connected structures, as the degree of cross linking increases. At some point the connected structure occupies the entire macroscopically observed system. There is a point where the system only

just has a structure that is the size of the system. A "spanning" structure exists. The macroscopic physical properties of the system change at this point, which is usually referred to as the gel point, the vulcanization threshold or the percolation point, depending on the context. The simplest change is that below the threshold the system behaves like a fluid, above it like a solid.

There are numerous publications on experimental work regarding polymer networks since materials of this type have interesting elastic properties that make them very useful in industrial and medical applications. Rubbers, being a types of polymer networks, deform under stress, but regain their original shape when the stress is removed. Gels find extensive use in pharmaceuticals, cosmetics and food. For the rational design of materials, required for a given application, knowledge and prediction of the properties are crucial. The seminal work of Flory [40] and de Gennes [2] laid the foundation for much progress in theoretical description of polymer network systems [41]. Theoretical research focuses mainly on the physical properties and amongst many other things on the collapse and expansion of a network, similar to the single polymer case discussed in other chapters of this work. However there is still a large gap between the experimental and theoretical work. From a practical point of view it is extremely hard, if not impossible, to create networks that are as ideal as theoreticians would like them to be for their theoretical models to hold and vice versa the models are too complicated to solve if a greater degree of realism is included. One approach in bridging this gap is with computer simulations. Typical simulations can effectively "solve" theories that are too complicated to solve analytically or create model systems that are much more ideal and "measurable" than real systems to test the validity of theoretical assumptions.

In the chapter we will address the following problem. In a simulation, how do we know when we do or do not have a network that spans the system. In the small, by macroscopic standards, systems that we can practically simulate different realizations of a given model system under fixed conditions will inevitably give both system spanning and non-spanning structures. However, if we calculate physical properties by averaging over all realizations and include those which do not have a spanning structure we will obviously get spurious results compared to a macroscopic system for which all realizations have a spanning structure. For example if we calculated the bulk modulus and included all realization, those that are liquid-like (do not contain a spanning structure) would clearly contribute unrealistically low values. Similarly, if we looked at the dynamic response they would contribute liquid-like (finite viscosity) behaviour, rather than the solid-like behaviour of spanning system. The obvious way to alleviate this problem is to exclude realizations that do not contain a spanning structure. However, to do so we need to identify which case we have.

There are a number of simulations techniques that deal directly or indirectly with detecting polymer networks. One of the older analogous problems that people have tried to solve with simulations is the percolation problem [42,43]. Consider a grid completely filled randomly with elements of either type a and b where the density of a is ρ_a. One can then attempt to calculate at what value of ρ_a neighbouring elements of type a form a connected path from one side of the system to the other. This problem was first posed in 1957 by Broadbent and Hammersley [44] and the first effective simulation technique that could simulate this problem in an efficient way was proposed in 1976 by Hoshen and Kopelman. [45,46]. We will later refer to this algorithm as 'HK76'. Later the HK76 method was modified to work off lattice [47].

Another technique we can use in polymer network simulations is the so-called burning algorithm developed in 1984 by Herrmann [48]. The procedure is similar to forest fire propagation, hence the name. This technique can be utilized to analyse polymer network structures, it can find the stress bearing backbone, identify loops and a few other properties [49]. People have employed these network analysis techniques to two types of network simulations [50–52]. The first type are typically atom-scale simulations where individual monomers are explicitly treated and have been typically done far above the vulcanization threshold [53–55]. They employ the burning algorithm to analyse their system and effectively categorize every single monomer. The other type of simulations try to model the system with a "phantom" network. This is typically a set of connected elements that does have the same connectivity as a real network but lacks excluded volume interactions and hence ignores entanglements and knots. While this may seem like an oversimplification, such models still capture many of the properties of such a system in a realistic way.

One of the biggest differences between a real system and a simulated system is the size of the sample. Simulated systems can not nearly be as large as real systems. A big simulated system containing thousands or even million monomers and be measured on a microsecond scale [56], while a real sample the size of a bucket will contain 20 orders of magnitude more monomers and a patient scientist can perform measurements that take years. On the one hand, this means that any effect on length scales larger than the simulation box cannot be captured. On the other it poses a serious problem with boundary conditions. Since in real systems the boundaries are effectively "infinitely" far away measuring bulk properties is easy, but this cannot be done in simulations. The fraction of "bulk material" over "close to the edge" material is close to 1 in a real system, and close to 0 in a simulated system. Two common solutions to this problems are either taking a small enclosed system or use periodic boundary conditions. Neither of the two solutions is perfect, but periodic boundaries do have some benefits over a closed system.

One is that periodic systems are more isotropic than confined systems. There are no effective edges, so there is no location dependent effect from an edge. In a way, they have an infinite size. The second is that although periodic systems do not allow for fluctuations to occur on lengths larger than the box length, a confined system with the same dimensions has a small system size. In addition, periodic systems inherently have translational invariance. This is a requirement for momentum conservation, an important consideration if one is interested in dynamics. A more thorough treatment on the difference between confined and periodic systems can be found in Allen and Tildesly [28].

Addressing the problem of identifying spanning structures in a confined systems is relatively straightforward. It just requires identifying whether there is a connected path between the confining boundaries. For periodic systems, which are essentially infinite in size, this is not the case, as we will see. In simulations using periodic boundary conditions in studies of system with densities far above the percolation threshold all realizations will yield spanning structures. However, as we will see near the gel point this is not the case. For the remainder of the chapter we will consider cross-links as simply the notion that one point is connected to another in some way. The nature of the connection is not considered. This applies most intuitively for covalent bond cross-linking, but in principle could be applied for other mechanisms.

4.2 The problem with resolving the problem

Probably the first problem you may run into when constructing a network is detecting if the system is actually a network. In the thermodynamic limit, the gel transition/vulcanization threshold is at a very specific density, but for a single small configuration this does not hold. Now say you are constructing networks close to the gel transition point but only want to take an ensemble average over the ones that are actually a network, how do you find out if a configuration should be taken into account. The problem is very clear in a very large real system, if you can trace a path from one side to the other, i.e. the system has structure on the largest possible length scale, then it is a network. One way of approaching this problem is by looking at order parameters, Puertas does this with a 'de-mixing' order parameters [57], but these methods all consider ensemble averages that do not specifically distinguish between network and non-network configurations. So they do find the transition point, but only for the average configuration, not for each single instance.

To illustrate the problem, for which we will propose an algorithm to solve, consider an almost empty periodic system with just a single long polymer chain. Percolation theory would tell us that this system will be a network if one end is touching one side of the box while the other end is touching the

other side of the box (we will refer to this as a "spanning cluster" from here on). Obviously a single linear chain is not a network from a conceptual point of view as can be seen in figure 4.1. A simple correction to this would be to require that the chain is connected to itself across the boundary. At first glance this might be the complete solution to the problem, but as we can see in 4.1 and 4.2 there are configurations that are obviously a network but are falsely identified by either of the two criteria.

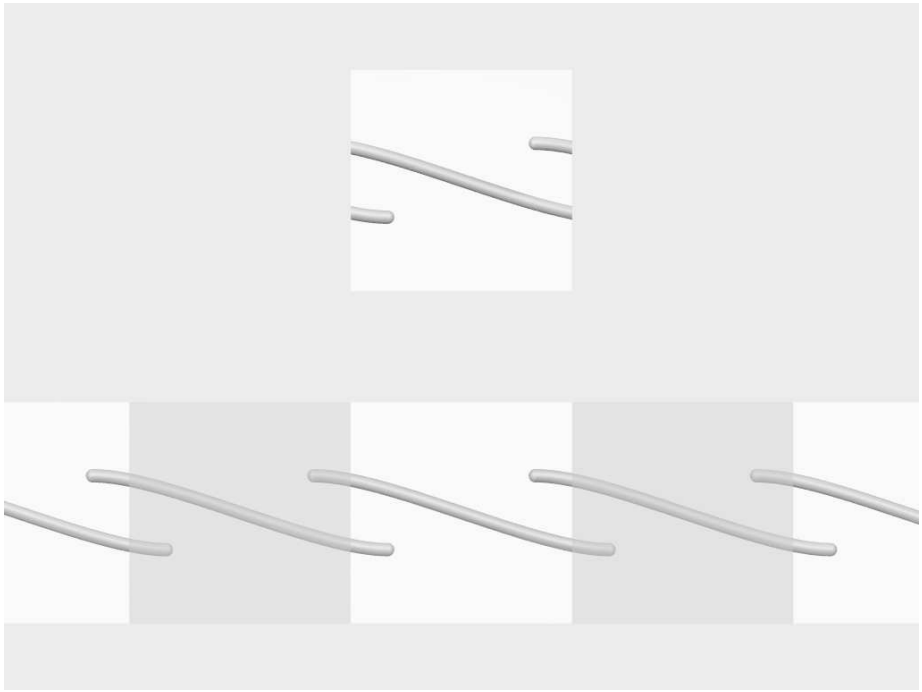

Figure 4.1: Probably the most obvious case where the percolating cluster condition fails. The object is obviously not a network, yet it spans an entire periodic box.

Taking 4.2 as a starting point, we can try to solve the network problem by explicitly correcting for these exceptions. Unfortunately there are an infinite number of possible exceptions (see 4.4) so a more elegant approach is needed. While most of the exceptions only occur in a very small subset of possible configurations, we will see later on that for typical simulation system sizes they do occur relatively often.

As a first naive approach one could consider simply expanding the simulation box to a much larger box. If the system is a network, it will not relax and the potential energy will continue increasing, i.e. it would take an infinite amount of energy to scale the system to infinity. There are a few issues with

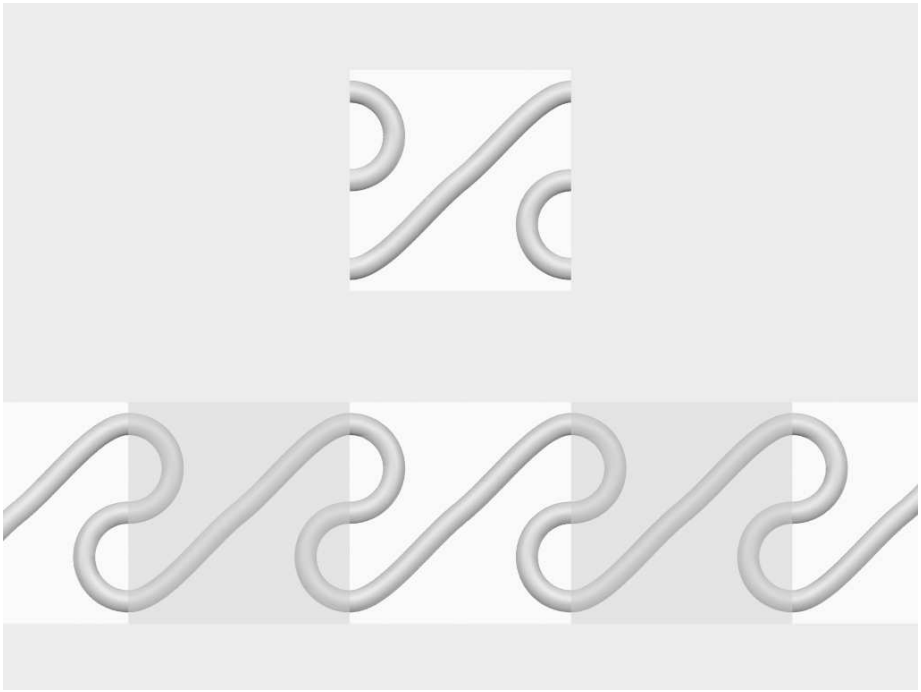

Figure 4.2: The simplest case where the self-connected condition fails. We can easily see that the object connects to periodic images of itself, yet the largest object in a single box does not connect to itself. This can change if we choose a different arbitrary periodic box location as we see in figure 4.3.

this however. Simply rescaling the system will not work, any rescaled system with at least one pair potential that goes to infinity for long distances will satisfy this condition, while not always being a network. To overcome this the system would have to be re-sized in small steps, where after each step the system would be allowed to equilibrate for some time. It is easy to see that this is a computationally expensive operation. Also it is wrong in a way. The system will have to be a network in all 3 directions, yet the energy will still go up to infinity if the system is a "slab" or even a "rod". Also it seems counter intuitive to explicitly perturb a system if all we are interested in is a topological property of the system. So ideally we would like to have a method that will correctly classify a system as network or not without explicitly resolving one or more of the odd cases but treat all of them with minimal computational effort. Other research has failed to notice this issue, for example [58] simply treats all configurations that have one cluster touching both ends of the simulation box as a network configuration. Others do not even explicitly

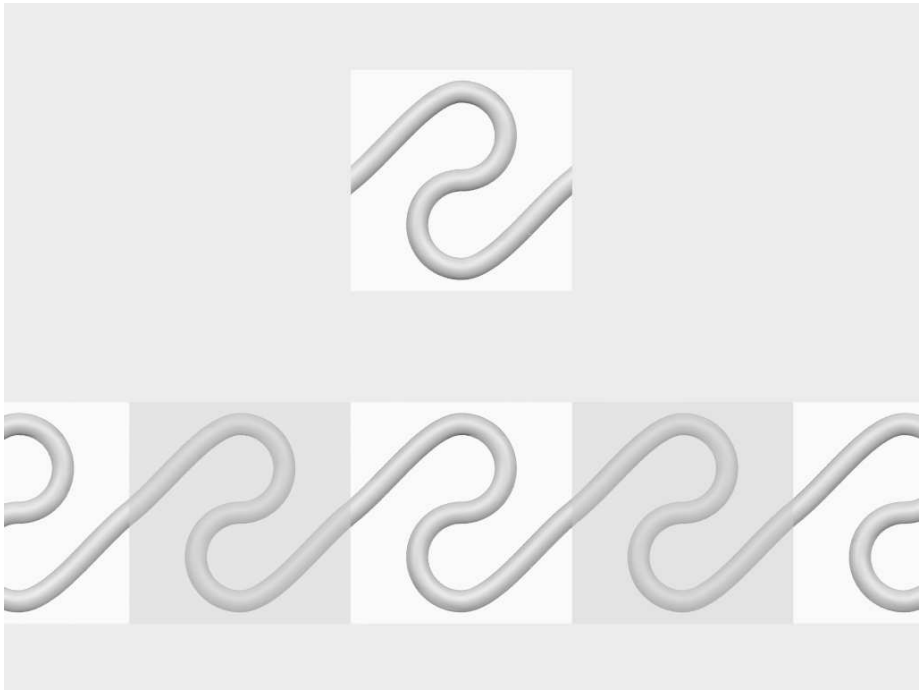

Figure 4.3: A simple case where all the network criteria agree, but only because we chose the right boundary for a periodic box, see 4.2 for a different choice causing them to disagree.

mention what they take as the condition or if they use periodic boundaries or not [59–61].

4.3 The algorithm

We propose the following algorithm to verify if a set of connected points in a periodic system is a network or not. The physical idea behind this algorithm is the requirement that if one stretches the box, the enclosed particle system stretches along and can not relax to freely moving bits. So

$$\lim_{L_i \to \infty} \sum_{i=1}^{N} \sum_{j=1}^{N} U_{ij} = \infty$$

for all directions $L_i \in \{L_x, L_y, L_z\}$ with U_{ij} the potential energy contribution for particle pair ij.

For a system to be a network we use the following three definitions.

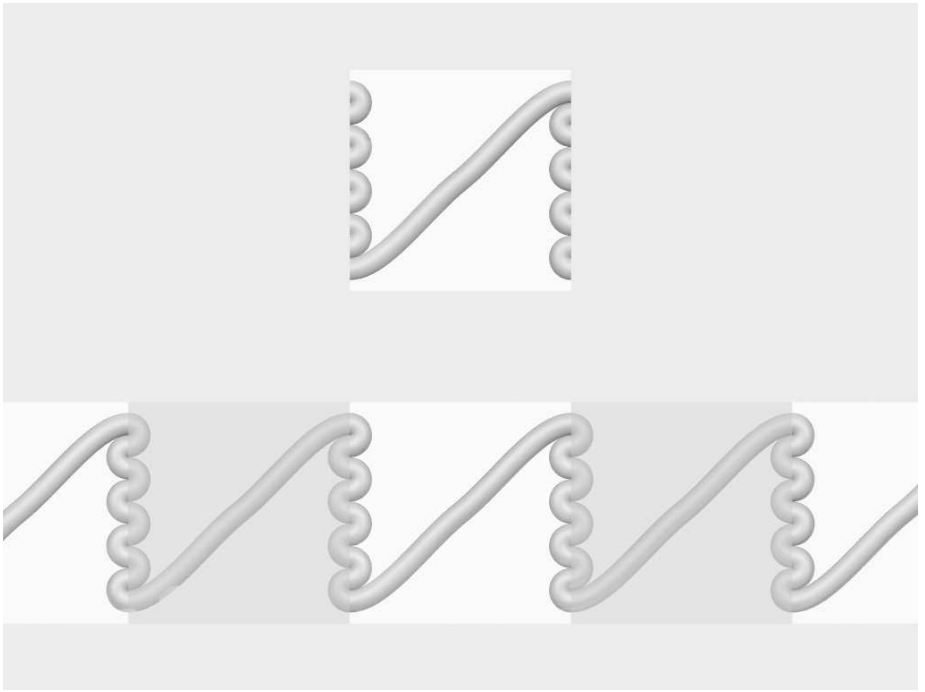

Figure 4.4: A more complex example where our algorithm and the self-connected condition disagree. See how explicitly trying to correct for specific cases will not work because there are can always be more exceptions to such rules.

D1. A cluster is the collection of all mononers that are directly or indirectly connected to each other, even across periodic boundaries in all but one unit direction.

D2. A cluster contributes to the network if there is a path to a different periodic image of itself using the discarded inter cluster connections.

D3. A system is a network if there are clusters contributing to the network for all n perpendicular directions in an n-dimensional system.

For definition D1, take note that this means that for an n-dimensional system, there are n possible clusterings. Each of these clusterings will be used to detect if the system is a network in the corresponding direction where the connections across periodic boundary conditions are not take into account. Taking figure 4.4 as an example, there will be 2 clusterings, one for the horizontal and one for the vertical direction. In the horizontal direction, we discard the connections crossing the left and right boundaries of the unit cell, so we get 9 clusters. As can be seen in the bottom half of the image,

those 9 clusters will eventually be considered a network in the horizontal direction because you can trace a clear path to one of the periodic images, see the further explanation of D2 below. In the vertical direction, we discard the connections crossing the top and bottom boundaries of the unit cell, of which there are none, so we get one cluster. This one cluster does not connect to itself across the top and bottom boundaries of the unit cell, so we don't consider this system a network in the vertical direction.

Definition D2 specifically mentions a connection to a periodic image of itself. This is the only way to guarantee that a part of a network is connected to another part infinitely far away. To see why, consider the case where none of the clusters connects to a periodic image of itself. If there are n clusters, the longest chain one can construct would span n periodic boxes, so the system has no structure on length-scales longer than that. Now the only way to get a longer (infinite) chain is by allowing a subset of clusters to be re-used. However if you reuse a cluster, it immediately follows that since there is a path between them, there must also be a path to opposite sides of it in both directions because of periodic boundary conditions, hence the definition for D2. For the algorithm to work the system has to be described as a set of point locations $\vec{r}_i$ and links l_{ij} connecting points i and j in a box with dimensions $\{L_x, L_y, L_z\}$. The algorithm works in two stages, both performed in all unit directions $\{\hat{e}_x, \hat{e}_y, \hat{e}_z\}$, so for a 3 dimensional box we would treat the $(1,0,0)$, $(0,1,0)$ and $(0,0,1)$ directions.

The need to think of this problem in terms of periodic images further away than the next nearest unit cell is illustrated in figure 4.5. It should be easy to see that this approach of constructing a system that only connects to itself three unit cells away can be extended to any arbitrary distance.

In the first stage of the algorithm all connections l_{ij} that cross a periodic boundary in one direction are dropped, we refer to these as l^*_{ij}. For our polymer-like systems this means that if the shortest path between $\vec{r}_i$ and $\vec{r}_j$ crosses the box boundary perpendicular to $\hat{e}_x$ we will discard it in this stage (assuming that we are in the first pass that considers $\hat{e}_x$). The monomers are then tagged with a unique identifier ("id") for each group that forms a cluster using an off lattice HK76 [45,46] adaptation as follows. There are many way of doing this, but the easiest (and most inefficient) way can be summarized as: Start by assigning each monomer a unique numerical id, then iterate over all connected monomer pairs and assign both of them the highest id if the id's differ. Stop iterating when after a pass over all pairs not a single id was changed.

Our implementation is slightly more involved but much more efficient, we will detail it in appendix 10.1, the approach is similar to the original HK76 implementation. The biggest change is to use two lists of pointers, the first points into the second and the second pointing into itself. Entries in the first

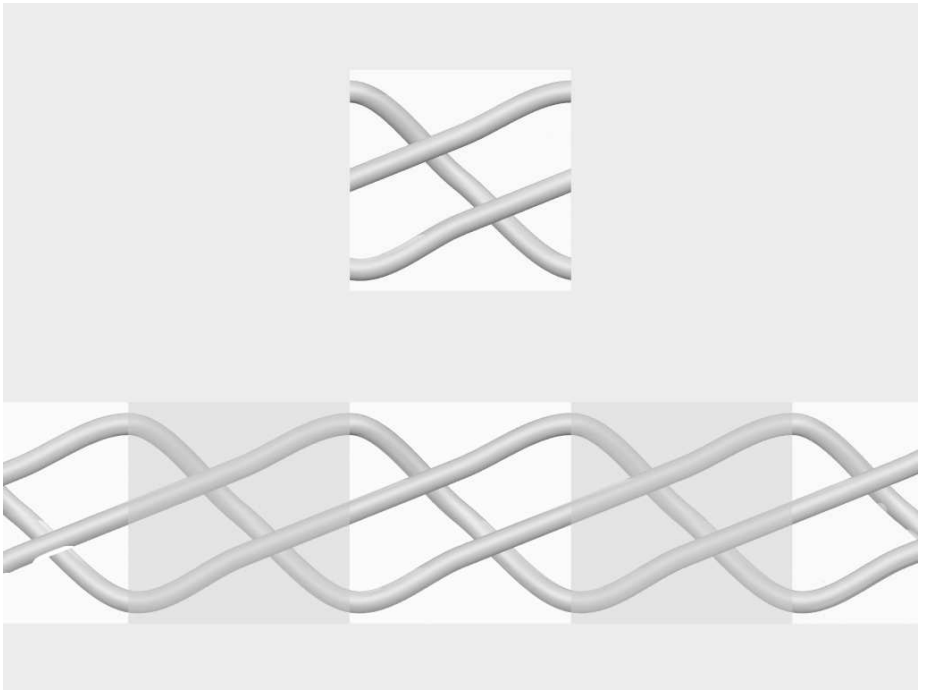

Figure 4.5: A simple case where we can easily see that we have a network, but that an element only first connects to a periodic image of itself after skipping 2 periodic boxes. While configurations are rare in typical grid-like percolation systems, they occur more often in system with large objects.

list are coupled one on one to particles, the pointer into the second array indicates to what cluster the point belongs. By allowing the second array to point into itself, you can join groups if required later, and with a consistent scheme all particles that are in the same cluster will get the same unique index. The benefit of this scheme is that it allows analysis of the entire cluster with a single pass over the connections.

At this stage in the algorithm the first network test can be performed. All monomers with the same id belong to a single cluster C_i. We will always have a number of clusters N_c smaller than the initial number of monomers. For each cluster C_i test if one of the dropped connections l^*_{ij} would connect a cluster to itself. This can be done by testing each l^*_{ij}, if $\vec{r}_i$ and $\vec{r}_j$ have a distance $\vec{r}_i - \vec{r}_j > \frac{L_x}{2}$ and both share the same id the cluster with this id connects to itself across a periodic boundary. This would immediately mean the systems satisfies definition D2.

If no network is detected, the second stage should be performed. In the second stage we first remove all "dead end" clusters. These are clusters that

would only have connections to one other cluster on one side through the dropped connections l^*_{ij}. The dead end removal should be done recursively since a dead end removal might introduce new dead ends. This step is optional, but it reduces the next step to a simpler problem. The result will be N_r ($\leq N_c$) clusters. We then construct a square matrix S_{lm} of size N_r where each element indicates if there would be a connection between clusters with id l and id m if there were no dropped links l^*_{ij}. An arbitrary but working choice of values is 0 for no connection, 1 if l connects to m from left to right and -1 if l connects to m from right to left. If there is an ambiguity in assigning a matrix element value it means that definition D2 is satisfied and there is a network in this direction.

Now we reach the final part in the algorithm. Here we re-use the burning algorithm [48] mentioned earlier. The primary purpose of the burning algorithm for us is to check if two points in a set of points and connections are connected through some path. This is done by recursively flagging points in the set directly connected to already flagged points, starting with one arbitrary point in the set. If the iterative process is carried out for long enough, any point that has a path to the original point will be flagged. Points that have no possible path to the original point will not be flagged no matter how long the iterative process takes. A secondary use is to keep a bit more information next to the flag that will allow us to resolve if clusters do or do not satisfy condition D2.

Put differently, the full final part of the algorithm is now as follows. We pick the first cluster $l = 1$ and analogous to the burning algorithm set this cluster "on fire" at a single certain periodic image identified by $q_1 = 0$. Then iteratively we "ignite" all clusters directly connected to burning ones that have not burned before, put out burning ones and assign all new burning clusters a value of $q_a = q_b + S_{ab}$. Here index a is the index of the newly burning cluster and b the one that set it on fire. Effectively this is the distance from the original burning cluster in periodic box units given the "arbitrary" values -1, 0 and 1.

A few things can happen now, the most important one is a collision. In a collision two clusters attempt to set the same other cluster on fire or a cluster tries to set another one on fire that has already burned. If a collision can result in multiple possible values of q then this implies that definition D2 is satisfied, otherwise we can ignore the collision. After a number of iterations there will be no fires left, in that case set one of the clusters that has not yet burned on fire and repeat the current stage of the algorithm. At some point all clusters will have burned, if there have been no collisions resulting in definition D2, the system is not a network.

As implied at the start, the full algorithm should be done 3 times for all 3 directions, only if definition D2 is satisfied in all 3 directions the system is a network. To summarize, we now distinguish 3 possible tests related to

percolation/network detection in a given direction:

T1. Percolating; There is a cluster spanning the entire system in one direction

T2. Our new method; There is a cluster connected to its periodic image

T3. Self connected; There is a cluster directly connected to its periodic image across a boundary

The first test will falsely classify systems as a network when they are not, the third will falsely classify systems as non-networks while they are networks and our method will not give any false classifications.

4.4 Justification

Our new algorithm is a combination of the off lattice HK76 algorithm [45, 46] and the burning algorithm [48]. Both have been proven to be efficient and correct so we will not discuss the validity and speed of these parts. When implementing these algorithms there are multiple approaches possible, we noticed that implementing both a fool-proof but slow and a complex but faster version was often useful to verify the correct implementation of the complex one.

The biggest change we have made is taking the relative positions of elements into account. Relative positions play a role when using periodic boundary conditions. In a confined system the HK76 algorithm should be used to tag spanning clusters and the burning algorithm can be used to analyse single clusters.

The validity of our new algorithm depends on the validity of the last stage, the burning of the clusters. Here the main assumption is that the system satisfies definition D2 if, and only if, there are two or more possibilities for the value of r_i somewhere in that part of the algorithm. It is easy to see that this is true. Assume a configuration with N elements in a periodic box that has uniquely determined values for r_i. This means that there is a single unique distance between two elements. The system can therefore be a maximum of N periodic box lengths long. This implies that the system cannot be a network since a network should have an infinite size.

Now assume that a system that has an ambiguous value for an element x, say a or b. This means that the element is both a distance a and b from the starting element with $r_1 = 0$. Because of periodicity, and the fact that $S_{ij} = -S_{ji}$, there will also be a path from element x to images of the starting element at $r_1 = a - b$ and $r_1 = b - a$. This again implies two paths from the starting element at $r_1 = a - b$ to x at $a - b + a$ and $a - b + b$. Now it should be clear that using this construction one can find a path connecting

x to periodic images of itself at infinite distance, thereby satisfying definition D2.

4.5 Application on known solution

Unfortunately it is very hard to give actual proof that a complex algorithm works in all cases. Although we believe that the previous section justifies this to be the case, we will not attempt to rigorously show that the algorithm will give the correct answer for all possible configurations no matter how exotic. We will however attempt to justify that the algorithm is correct by applying it to a problem with a well known solution.

Turning to basic percolation, theory gives us many sample systems to test. A percolation system consists of connected objects, and the problem is finding the object or connection density where the system allows a path from one end to the other. The best way to think of a percolating system is to think of it as a maze, if you can go from one end to the other, it is percolating, if you can not, it is not percolating. Now the more holes you make in walls, the more likely it becomes that you can go from one end of the maze to the other. Keeping with the maze metaphor, consider a maze that is infinitely large. For a very high hole density you can always find a path from one end to the other, for a very low hole density you can not. There exists a very specific density that is the cross-over density between the two situations.

Now a simple model of this maze is with a cubic lattice. Each point on the lattice is either filled or empty, and the lattice spacing is 1. Two lattice points are only connected to the four nearest neighbours. If we then fill lattice points at random on a very large grid, we should find that some grids will have a path from one side to the other, and some do not. Looking at the density of filled lattice points, we should see that at least for very large grids we will hardly find any configurations with a path from one end to the other if the density is below 0.3116, and almost only configurations with such a path for densities above 0.3116 [62].

As we will consider relatively small systems, we do not expect to reproduce this number very accurately, but it should be close. We generated a large number of configurations with point densities close to 0.3116. For each of these configurations we tested if T1, T2 and T3 hold. Then for a given density we calculated what fraction of configurations with that density meets T1, T2 or T3. The result can be found in figures 4.6 and 4.7. An example configuration that meets T1 but not T2 or T3 is given in 4.8.

All the results presented for T1, T2 and T3 are for the same set of configurations. So any difference in the fractions meeting T1, T2 or T3 are a systematic difference, and not due to statistical noise. If, for example, T2 is

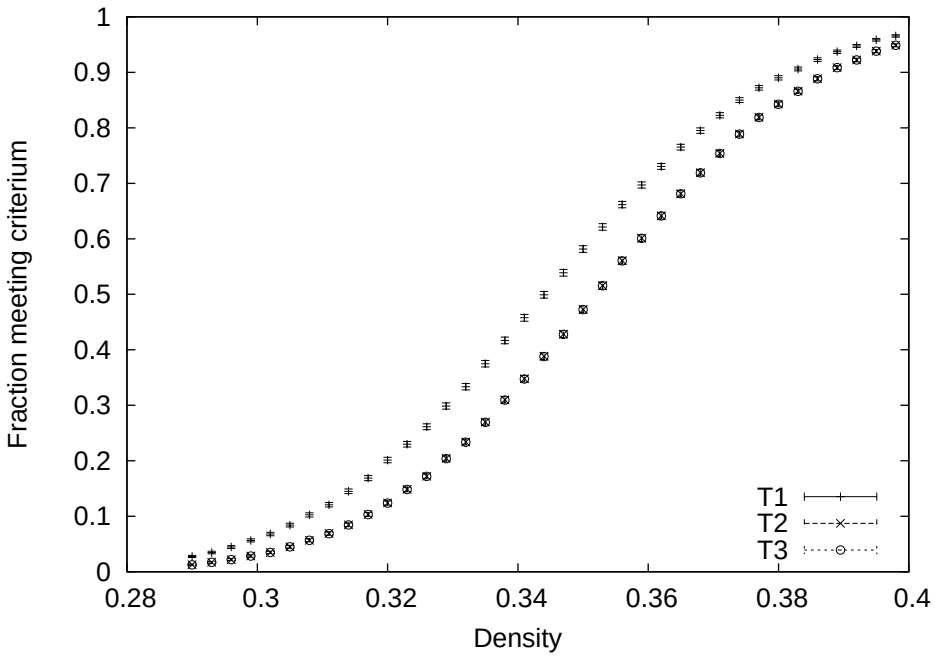

Figure 4.6: Fraction of generated configurations meeting criteria T1, T2 or T3. Note that T2 and T3 almost overlap. The density is the number density based on a length scale of 1, for points randomly placed on a grid. Each configuration was randomly generated on a 10x10x10 periodic grid.

met in f_{T2} of the cases and T3 in f_{T3}, certainly a fraction $f_{T2} - f_{T3}$ of the generated configurations caused disagreement between T2 and T3. Of course the numerical values $T2$ and $T3$ are still subject to statistical errors associated with ensemble averaging, but repeating the experiment many more times will never cause the sign of the difference to change.

An interesting observation here is that the difference between the three resulting data sets, particularly T2 and T3 is quite small. Nonetheless, T1 obviously over estimates to fraction of percolating networks and so falsely classifies non-percolating systems as percolating. The simple conclusion may be that our method is overly complex, relative to T3 but for polymer systems where we want to apply our proposed method this difference is much more pronounced as we will see in the next section. We should also point out that there are a few configurations where T3 indicates that a systen is not percolating whereas our condition T2 shows that in fact it is. We do not try to verify the value of 0.3116 in detail as the purpose of this work is not to find the exact value of the critical density, but to correct for false labelling of configurations due to finite size effects. In fact in the thermodynamic limit of

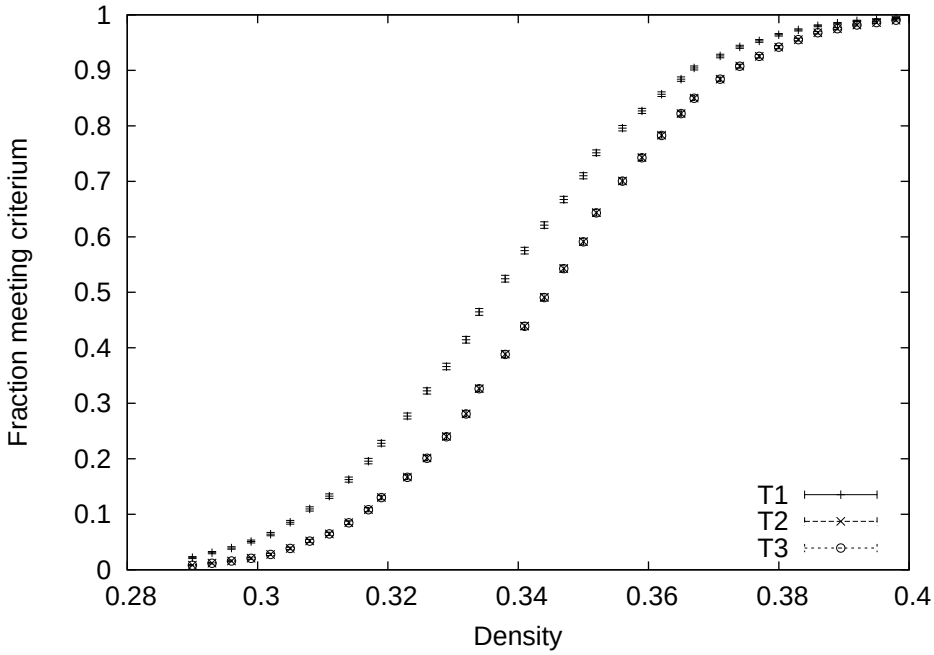

Figure 4.7: Fraction of generated configurations meeting criteria T1, T2 or T3. Note that T2 and T3 almost overlap. The density is the number density based on a length scale of 1, for points randomly placed on a grid. Each configuration was randomly generated on a 12x12x12 periodic grid.

an infinitely large grid all three methods should yield the same step function. It should be clear though that our simulations do not contradict this value. We also give the fraction of configurations meeting T2 for varying system sizes in 4.9. From that dataset we find that the transition between non-network and network is indeed steeper for the larger system. This is a clear indication that the existence of a region of densities that allow both network and non-network configurations is a finite size effect. Nonetheless, in simulations we are always dealing with finite sized systems, so the false classification is not trivial

4.6 Application on polymer network

4.6.1 Polymer network model

As a second test, we try to create the simplest possible network model with polymer chains. Such a network consists of a set of linear polymers that are then interconnected at some points along the chains. If we then assume that

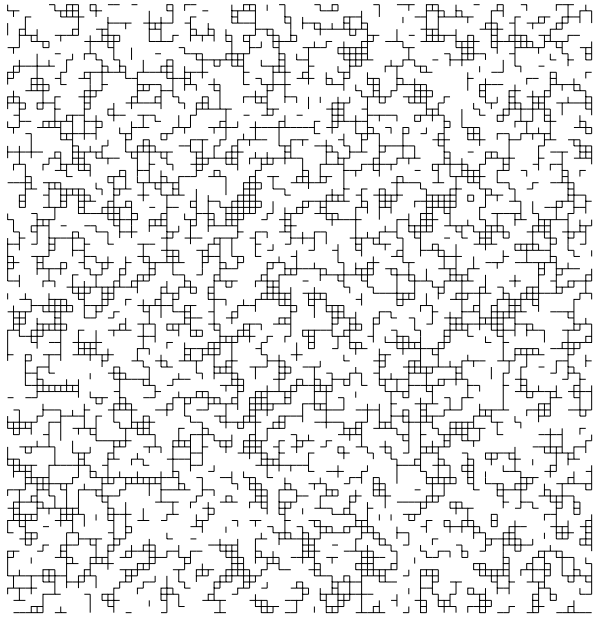

Figure 4.8: Example of a grid where T1 is true but T2 is not. We leave it as an exercise to the reader to spot why.

a typical distance along a chain between two interconnection points is larger than a persistence length (depending on the type of polymer this can be as few as 20 repeat units) we believe we can represent a network with a set of ideal chains. The actual interconnections are created in a system filled with a set of ideal chains such that two monomers on different chains are connected only if they are within a certain distance and with a certain probability. The system is created in a periodic box. For the initial configuration of the chains, statistically representative ideal chain conformations are generated and then located randomly in the simulation box. This is somewhat artificial as the chains are "ghost chains", in that they have no interaction with each-other other than the cross-linking. This model does, however, capture the entropic effects of deforming the polymers and forms the basis for many theoretical studies [63, 64]. During a dynamic simulation we model the connections between monomers as harmonic springs.

4.6.2 Static simulations

We first performed simulations to determine critical parameters for our simple network generation algorithm. Since we would like to generate configurations

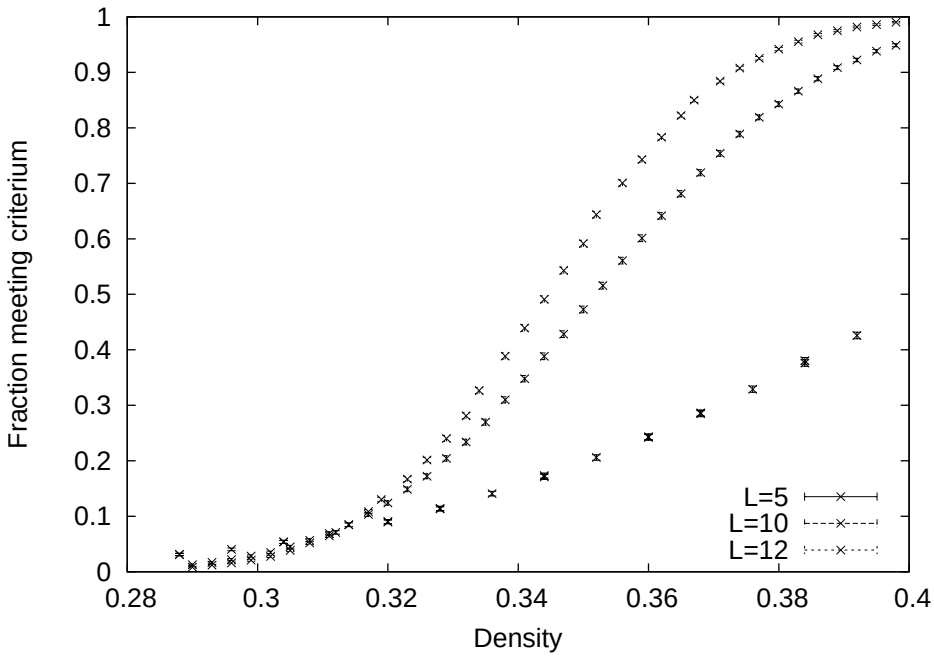

Figure 4.9: The fraction of generated configurations that meet condition T2 for varying densities and system sizes. As the system gets larger, the curve becomes steeper as we expect.

close to the critical density (gel point) for the system we will have to estimate this first. An analytical approach proved to be very complex, but fortunately the model is very cheap to test numerically and we only need it to obtain get a rough estimate of the critical density. To get this estimate, we repeatedly generate two random chains with 20 monomers with their centers of mass distance set explicitly to d_{cm}. We choose the unit of length to be the the Kuhn length for the connections. As we are not doing any dynamics, there is no need to specify a unit of time, but for simplicity we choose particle masses to 1, so the number density and mass density are the same. We then calculate how many connections we expect between the chains based on the same (simple) model we will use to actually generate the network later on. In our simulation we choose to link two monomers together with a probability of 0.1 if they were within a distance 1.0. By repeatedly doing this for various values of d_{cm} we obtain a numerical estimate of how likely it is that two chains are connected if they are a distance d_{cm} apart. As we can see in 4.10, the distance for where we expect 1 connection between two chains is approximately 1.8. We can now use this length to estimate a range in where we expect the critical density to be. One approach is to treat the individual polymers as

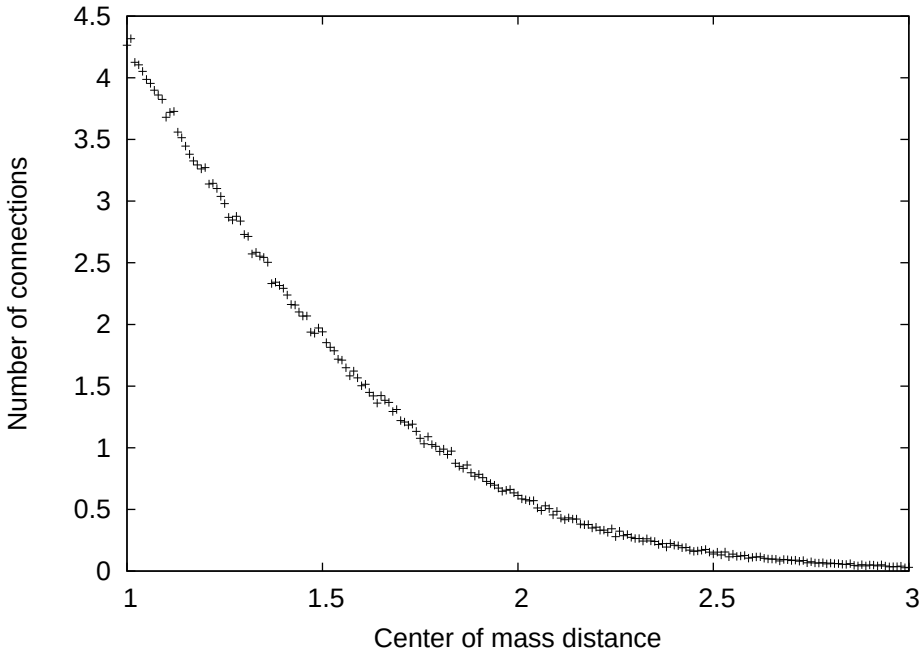

Figure 4.10: The amount of connections between two chains in our network model for a given centre of mass distance.

spheres with a diameter $d = 1.8$ ($r = 1.8/2 = 0.9$), the monomer density in the spheres is then $\rho_m = \frac{20}{\frac{4}{3}\pi 0.9^3} = 6.55$. For randomly placed overlapping spheres there is a known critical density. This density is given by the "Swiss cheese" model [65] and is about 0.289. The effective monomer density for the whole system is then given by simply multiplying this critical density with the density of monomers inside the spheres to yield $\rho_c = 6.55 \cdot 0.289 = 1.9$. The sphere approximation is not the best approximation for a polymer chain as they can form connections at much larger distances than 1.8 so we expect this to be more an upper bound. Another approximation is to simply use a dimensional argument, if the typical interaction length-scale is 1.8 and we normalize to this, we obtain a density of $\rho_m = \frac{20}{1.8^3} = 3.43$. Again we use the "Swiss Cheese" model to translate our monomer density to a critical monomer density, giving $\rho_c = 3.43 \cdot 0.289 = 1.0$. We now have a density range with a lower bound of 1.0 and an upper bound of 1.9 where we expect to find the critical density. Or more specifically for our simulations, the transition between non-network to a network. Based on the critical density estimate, we generated configurations at different densities. (ρ between 1.0 and 1.9, and various N between 1000 and 9000, 1000 samples per point). Then we determined the fraction of generated configurations that are networks.

To show why having a critical look at the network detection algorithm is important we again tested if a system is a network by using the self-connected and the percolating criteria. In figure 4.11 the conditions T1, T2 and T3 are validated for the same random configurations, for a system size of 7000 monomers with varying densities. While with the grid simulations in figures 4.6 and 4.7 the results for T2 and T3 were very close, they are significantly different for the polymer network. As noted above, although the curves are within the statistical errors the fact that the results are generated from the same configurations means that the difference between them, for this set of configurations is exact. There is no doubt that condition T3 fails to recognise a significant proportion of configurations as being networks when in fact they are. Since the two criteria differ in that T3 identifies clusters that are connected over distances of many periodic box lengths, this indicates that this is more commonly the case for the model polymer network than for the grid model. In other words, the clusters that form in this case are more rarefied, in that they are self connected on significantly longer length scales. As we also generated configurations for different values of N we have many graphs similar to 4.11, each showing the network transition as function of density. Next to this we can also calculate the dependency of the network transition as a function of the system size (number of monomers). Figure 4.12 shows that our largest systems do not yet converge to a "critical" density, but that our initial estimate of somewhere between 1.0 and 1.9 will be much

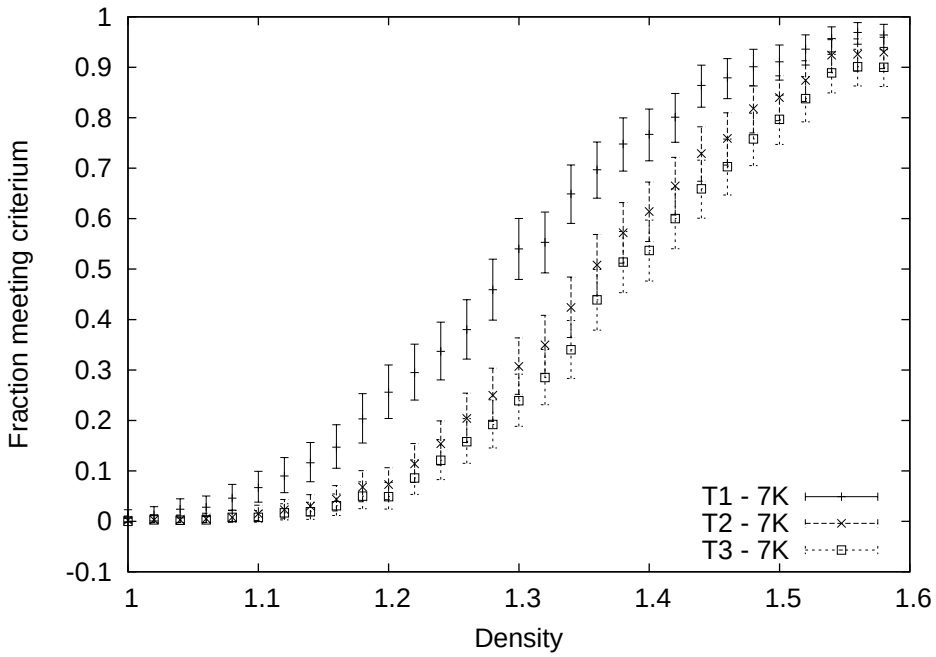

Figure 4.11: Fraction of generated configurations meeting criteria T1, T2 or T3. The density is the monomer density, where the length scale is based on the fixed bond length of 1. A single configuration consists of 7000 monomers. Note that the lines for T2 and T3 are now visually separated, in contrast to fig 4.6 and 4.7

closer to 1.0 than 1.9.

To display the results of those calculations coherently in 2 graphs we have put the fractional difference between T2 and T1 and T3 and T2 respectively in 4.13. As can be seen from these results the differences between the three methods can be very large, with up to 25% of the configurations being incorrectly classified as networks, or 9% being missed as valid networks depending on the choice of detection algorithm. We also did a similar analysis for the grid case in the previous section, but there the results were much less pronounced, especially for the T2-T3 difference, in much less than 1% of the cases do those two disagree. A simple explanation for the difference in results for the grid case and the network case is that the objects in the grid case are much smaller. As we are effectively studying a finite size effect, the effect should be smaller if the ratio of the object size to the system size is smaller.

Now there are two arguments to use our method over the T1 and T3 criteria. First we claim that our method always correctly identifies a network. Second our method can be used for periodic systems, in fact it is only relevant

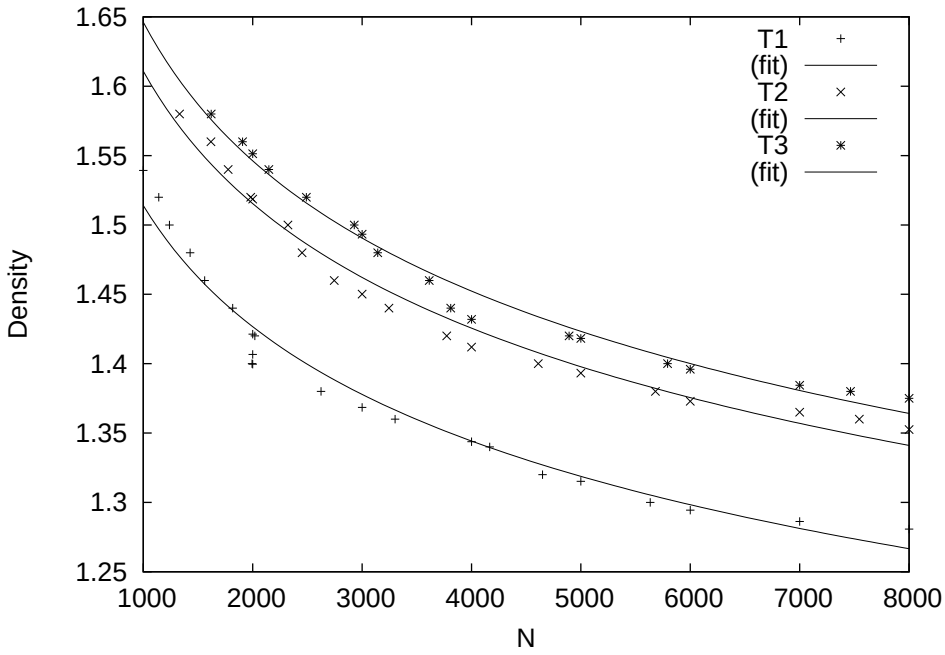

Figure 4.12: System density where half the generated systems are a network. The T3 condition clearly fails to recognize a fraction of the systems as a network while the T1 condition labels systems as networks incorrectly. The original data is relative noisy so three lines are fitted through the results. As a fitting function we choose a power law $\rho = aN^b$.

for periodic systems since it will reduce to the self connected check in confined systems. We feel that especially the last argument is the relevant one. Using periodic boundaries effectively removes all boundary effects in the system and is the best way to estimate properties of the system on a length scale larger than the actual box size. Even if the self-connected condition or the percolating condition is used in a periodic system, there is this arbitrary choice of the system boundary.

4.6.3 Dynamic simulations

The static simulations give some insight in the topological effects of varying system parameters like density, but the generated configurations are not equilibrium configurations. To get access to various equilibrium configurations so we can measure some ensemble averages, we have to do a time-evolution of the system. For time evolution we only have to add a thermostat and an integration scheme to the already described model, as this already includes

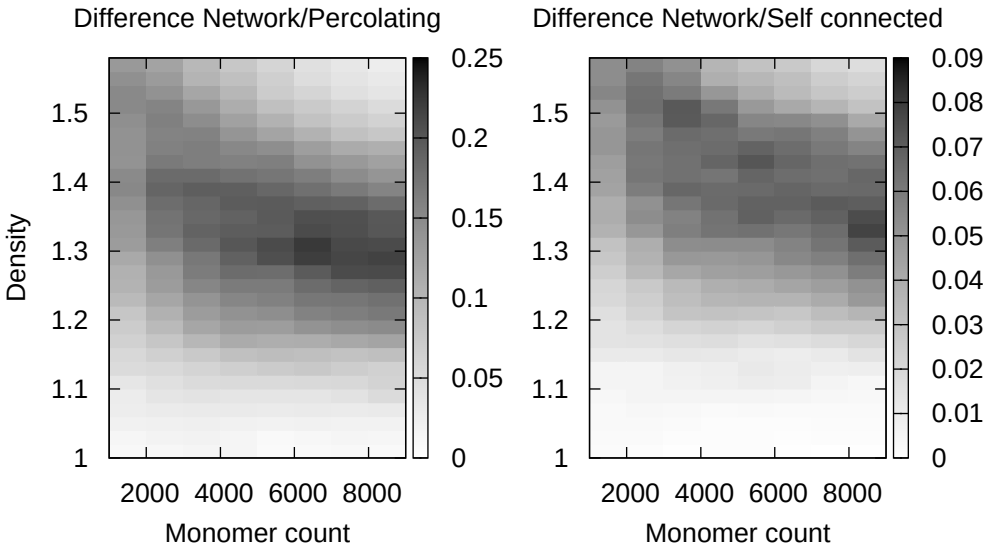

Figure 4.13: The fractions of incorrectly classified networks. We tested the classification algorithms on a system containing random Gaussian chains with Kuhn length 1 and 20 beads each. On the x-axis is the system size in total number of monomers, on the y-axis the monomer density. In the left graph we see the fraction of falsely classified networks, on the right the fraction of missed networks.

harmonic springs for the connections. For the thermostat and integration scheme we choose the Lowe Andersen thermostat as described in Chapter 2. During simulations the volume V, number of particles N and temperature T are kept constant.

We then generate a configuration with a density around the gel-point and verify that it is actually a network using the T2 condition. This configuration is then simulated for X time-steps while at the same time we calculate the structure factor. At the end of the simulation the same configuration is rescaled uniformly with a factor α to fit in a slightly larger box with volume $\alpha^3 V$. For this new volume we allow the system to equilibrate then calculate the structure factor again.

The results from repeating the procedure above multiple times are all rescaled to the original length $L = V^{\frac{1}{3}}$. As we can see from the results for this procedure, shown in figure 4.14, the stretching of the network is clearly not affine. The very low length scale structure factor and the large scale struc-

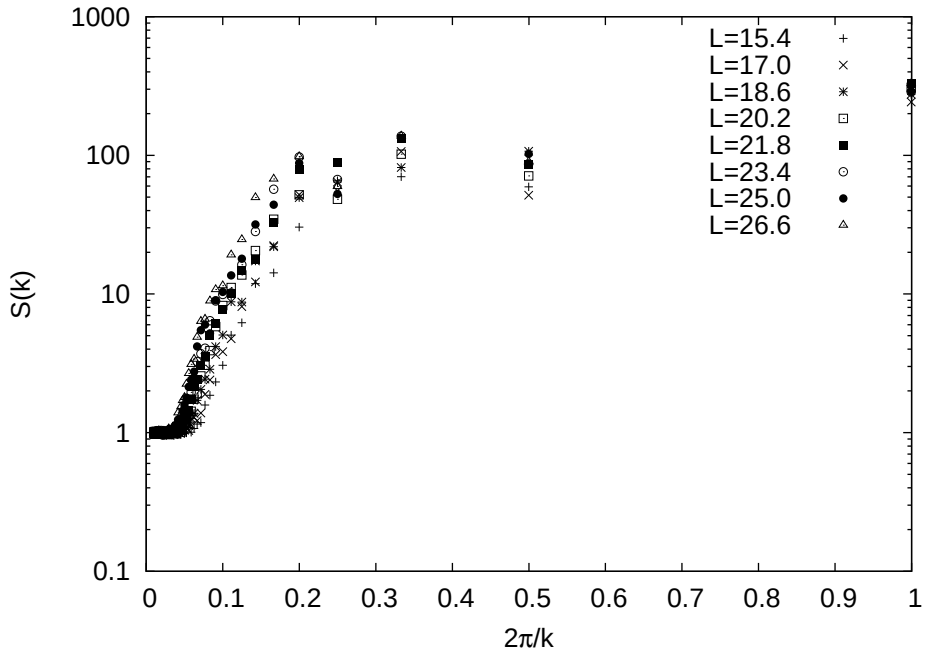

Figure 4.14: The structure factor for a single configuration calculated at various densities. The length is rescaled back to the simulation box size. The box length L given in the caption.

ture factor both go to the same value, but for intermediate length scales the structure factor for the more stretched configurations is significantly higher. Our results show that, at least near the gel point, even the simple ghost chain model yields a non-affine deformation. Non-affine deformation has been proposed as an explanation for the non-linearity of the elasticity of polymer networks [66]. Further, it is also frequently observed experimentally [67–69]. Experimentally though, it is often difficult to exclude the possibility that this is due to the presence of other cross-linking mechanisms, for example entanglements. Interestingly, experimental results show that hydrogels [69], for which the density is so low that this is unlikely to be the case, also deform non-affinely. Our simulations correspond most closely to an idealized model of these materials. The fact that they still show non-affine deformation supports the idea that, in reality, it is simply the inhomogeneity of the network that is responsible, rather than other possible explanations such as polymer stiffness or the nature of the cross linking.

Calculating the pressure for this system is complex, but we can calculate the potential energy for all the volumes. The results for this can be found in 4.15. Since calculating the pressure involves taking the derivative with respect

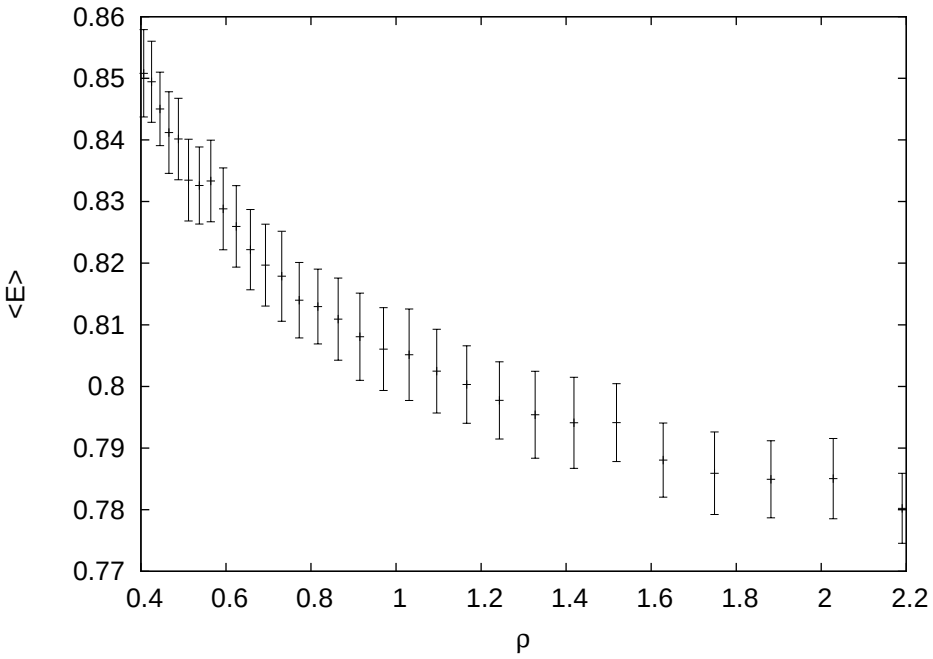

Figure 4.15: The average potential energy per particle (not per bond) for various system densities.

to the inverse density, clearly the results are of insufficient accuracy to make the pressure calculation meaningful. More extensive simulations of even this simple system are required. We regard this therefore as simply a "proof off principle".

For the configuration used in the dynamic simulations we counted the number of links per monomer and generated a histogram out of this, the results can be found in figure 4.16. We also generated various configurations with single monomers following the same interlinking procedure as for the polymer case. The goal was to get a configuration that has about the same number of links and is also a network. For this we had to choose a higher monomer density.

4.7 Computational efficiency

Our algorithm is partially based on already existing algorithms. For the clustering phase we use an off-lattice variant of the HK76 algorithm, this is known to scale linearly with the number of objects in the system. The next phases where we use the information from the clusters to determine if

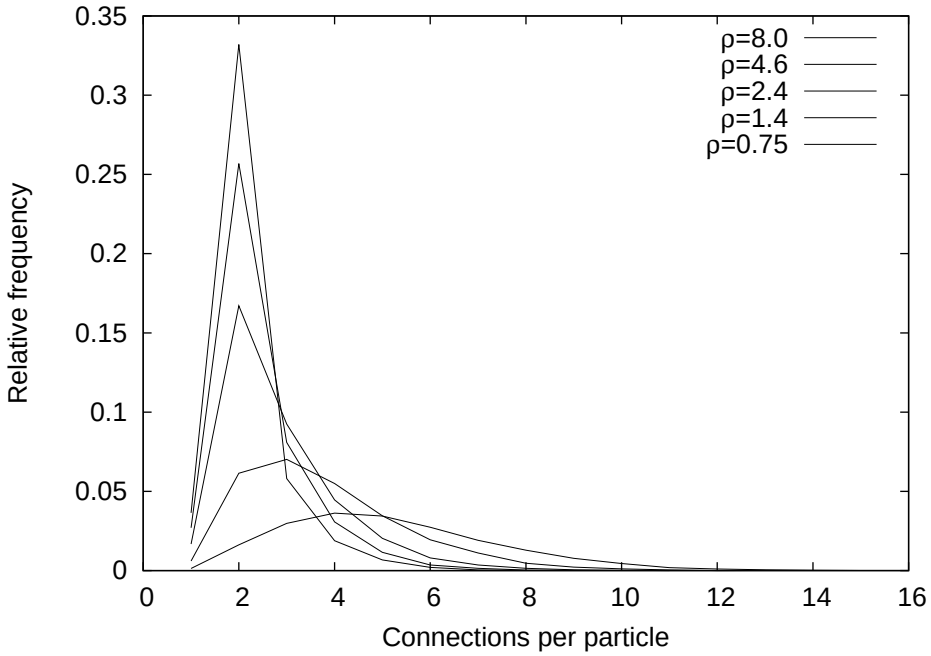

Figure 4.16: The dynamic network simulations were done with a configuration generated at a density of $\rho = 2.4$. This graph shows the distribution of the amount of connections a single monomer has to other monomers for various densities. Below the critical density it is peaked around 2 (the interchain connections), and above the critical density the distribution becomes broader.

a system is a network scale differently. Because the dead end removal is recursive, it scales as $O(N_c^2)$ at worst. This is however not typical scaling, normal scaling will be much closer to $O(N_c)$. Last is the burning phase, this is linear in the number of clusters.

To verify all this, we measured the wall clock time t for the grid simulations for two densities at various system sizes. This includes both the generation of configurations and the analysis thereof. One density ($\rho = 0.5$) was above the critical density ($\rho = 0.3116$), the other below ($\rho = 0.25$). As can be seen in 4.17 the scaling is not exactly linear, but very close. For these settings we get a scaling that can be fitted very well to a power law $t = aN^b$ with an exponent $b = 1.13$. The value for a is not really relevant as it captures the amount of iterations per data-point (100 here), the computer specifications, the compiler efficiency and so on. Whether this is relevant of course depends

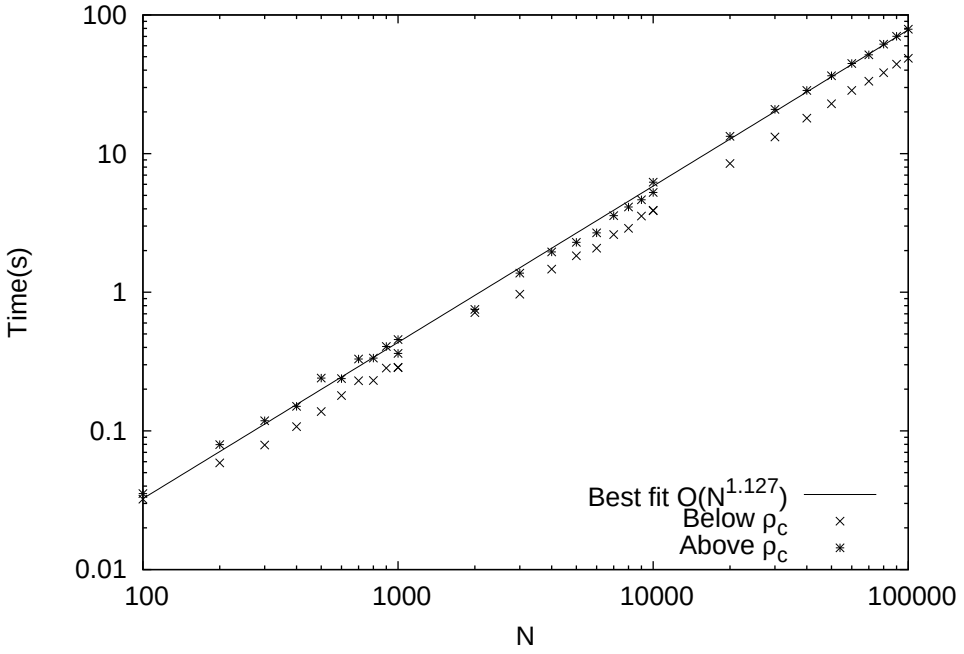

Figure 4.17: The wall clock time required to generate and analyse 100 grid configurations at a density below the critical density ($\rho = 0.25$) and above the critical density ($\rho = 0.5$). The scaling is unfortunately not linear, but the fitted exponent (1.127) is probably acceptable for most use cases.

on how much time is also necessary for other parts of a give calculation, for example network generation or solving for the dynamics. Given that the scaling of our algorithm is so close to linear, it is unlikely in most case to result in a significant overhead, particularly for more complex model networks.

4.8 Conclusions

We have adapted the original HK76 algorithm to detect polymer networks in off-lattice periodic systems. While our new algorithm might give little insight into the physics behind polymer networks we think it is an essential tool into the study of critical parameters of polymer network models. The algorithm gives a definitive answer if one wants to know if a periodic system is a network and thus cancels the uncertainty related to the network detection method. Using the method we showed that if we only take configurations that are networks into account, near the gel point even networks of ghost chains deform non-affinely.

One interesting thing with our method is that we expected a similar method to have been described in literature already. We were unable to find any alternative approaches to address the problems described in this chapter. Only after performing the tests on the simple lattice did we find that our new method gives almost always the same answer as the self-connected condition does. Given that percolation simulations are typically done on very large systems it is easy to see that the choice of either our new algorithm or any of the other two does not necessarily matter much, especially the self-connected condition agrees with our method most of the time for those systems. Similarly, far above or far below the percolation threshold all methods yield the same results (all configurations being networks in the first case and not networks in the second). Only when decreasing the system's size near the percolation threshold and introducing more complex objects, especially "stringy" objects, does our method show its true strength. This limits its applicability, but since its computational complexity is slightly higher than performing a self-connected check, we think that our method should find a place in any simulation that treats relatively small systems with "stringy" objects close to the percolation threshold. Another notable observation is that the accuracy of a detection method depends on the system being studied. For more complex systems this might be different again, so our method could be used to test whether simpler detection methods are adequate on a given system.

We also think this algorithm might be useful in other systems like protein simulations or zeolites. In protein systems it is often necessary to remove loose water molecules trapped inside the protein. Our algorithm can detect those loose pockets of water if we treat the hydrogen bonds as polymer bead connections. Zeolite systems can be thought of as large numbers of interconnected pockets through which molecules can diffuse. Our algorithm can be used to detect if particles are allowed to diffuse infinitely far in all directions.

We are aware that there may be more efficient algorithms to perform one of the sub tasks in the proposed algorithm. Since a typical polymer simulation with static bonds will only need to use this algorithm once at initialization

time, computational efficiency is not as critically important as other parts of a simulation.

Chapter 5

The nature of a collapsed solvated polymer

The application of the Lowe-Andersen thermostat described in this chapter is based on a theoretical model developed by Dreischor and Lowe in [70]. The model and its derivation are reproduced in a slightly modified form in sections 5.1 and 5.2.

5.1 Introduction

The temperature and pressure dependence of the conformational properties of a polymer are linked to its interaction with the solvent. These interactions also determine when dissolving the polymer is, or is not, possible. As it states in the review paper by Williams et al. [71], in 1953 Flory [1] wrote: "The better the solvent, the greater the "swelling" of the molecule. Conversely the poorer the solvent, the smaller the molecule." To a large extent this chapter addresses how generally true this second statement is. Flory is describing here the evolution of the average polymer dimensions as the effective pair interaction between monomers, mediated by the solvent, changes. Significantly, his work showed that generally at low enough temperature this interaction becomes attractive and the otherwise spindly expanded polymer "coil" collapses into a dense spherical blob, the "globule". The first experimental observation of this coil to globule transition was probably the denaturation of proteins on heating. Here, the protein goes from its compact active state at low temperatures to an inactive coil state at higher temperatures. In fact, the important role of the coil to globule transition in configurational changes of proteins, such as folding and unfolding, is an active are of research to this day [72, 73]. Generally the coil-to-globule transition is experimentally observed by measuring the polymer radius of gyration, while varying the temperature. Slagowski, Thai

and McIntyre were the first to report measurements of the coil to globule transition via measurements of the radius of gyration of polystyrene solvated in cyclohexane [74]. In the globular state the size l of the chain is seen to scale with the molecular weight M as $l \propto M^{1/3}$ as opposed to $l \propto M^{0.58}$ where the effective monomer interaction is repulsive. These two regimes are delineated by the theta temperature, at which the effective interaction is zero and the polymer behaves like an ideal chain, with $l \propto M^{\frac{1}{2}}$

The original theoretical model for these configurational properties is the lattice model based on Flory-Huggins equation of state, often referred to as the mean field model [75]. It models the polymer and solvent as a fully occupying a lattice. Because of this it considers the incompressible (or infinite pressure) limit. Here we give the derivation as presented by De Gennes [76]. The approach then proceeds as follows. A chain made up of N_m monomers is modelled as a spherical domain of volume V containing a homogeneous mixture of monomers and solvent (the "polymer domain"). Outside this domain there is pure solvent. The Free energy of mixing per lattice site according to the Flory-Huggins theory is given by

$$\frac{\Delta F}{n k_B T} = (1 - \phi_m) \ln(1 - \phi_m) + \chi \phi_m (1 - \phi_m) \tag{5.1}$$

where n is the total number of lattice sites in the system, ϕ_m is the monomer volume fraction in the polymer domain and χ is the well known Flory-Huggins parameter, which is always inversely proportional to the temperature, T. This parameter determines the "goodness" of the solvent at a given temperature. If the polymer is expanded, the fraction of space occupied by the monomers, ϕ_m, is small so we can expand this expression as

$$\frac{\Delta F}{n k_B T} \simeq (\chi - 1)\phi_m + \frac{1}{2}\phi_m(1 - 2\chi) + \frac{1}{6}\phi_m^2 \ldots \tag{5.2}$$

or alternatively

$$\frac{\Delta F}{k_B T} \simeq (\chi - 1)N_m + \frac{1}{2}v_L(1 - 2\chi)N_m\rho_m + \frac{1}{6}v_L^2 N_m \rho_m^2 \ldots \tag{5.3}$$

Here we have made use of the fact that $\phi_m = v_L \rho_m$, where ρ_m is the monomer density and v_L the lattice site volume. The connectivity of the chain is only taken into account in the form of an additional contribution to the free energy due to changes in the volume occupied by the polymer V relative to the volume occupied by the equivalent ideal chain $V_0 \propto b^3 N_m^{3/2}$, where b is the root mean separation between connected monomers. According to Flory the change in the free energy due to a change in the polymer volume, ΔF_v, is

$$\Delta F_v = \frac{3}{2}\left(\alpha^2 - 1\right) - \ln\left(\alpha^3\right) \tag{5.4}$$

Here $\alpha = (V/V_0)^{1/3}$ is the expansion factor. Minimizing the free energy with respect to α, using $\rho_m = N_m/(\alpha^3 V_0)$. yields the result

$$\alpha^5 - \alpha^3 - \frac{v_L^2 N_m^3}{3\alpha^3 V_0^2} = v_L \left(\frac{1}{2} - \chi\right) \frac{N_m^2}{V_0} \tag{5.5}$$

At high temperatures $\chi < 1/2$, and the chain is expanded, such that lengths scale as $l \propto N_m^{3/5}$. As the temperature is lowered χ becomes larger. When $\chi = 1/2$ the right hand term in equation 5.5 is zero and the chain will have a size proportional to its ideal chain size, where lengths scale as $l \propto N_m^{1/2}$. The temperature for which this happens, is known as the theta temperature Θ.

Below this temparature, the assumption that ϕ_m is small is invalid. However, using the full Flory-Huggins equation of state, the limit of zero temperature yields a polymer volume $V = N_m v_L$ and there is no solvent within the polymer domain, only the monomers of the polymer itself. The Flory model thus predicts that below the theta temperature a polymer collapses and expels the solvent. Polymer collapse can therefore be seen as a type of de-mixing of the polymer and the solvent. Indeed, as equation 5.1 shows, the coil to globule transition is linked to the mixing and demixing of the polymers themselves. Generally collapse will be accompanied by de-mixing of the polymers themselves from the solvent.

The coil to globule transition has also been studied more recently with computer simulation, often to test the assumptions of the mean field theory. Although in some cases an explicit solvent was used [77–85], in most cases the solvent was integrated out by using a simple potential such as the Lennard-Jones potential to describe the interactions between the monomers [86–110]. Alternatively integral equation theory has been used to calculate effective solvation potentials [111–113]. Further, rigorous calculations yielding exact results through a complete enumeration of polymer configurations have been applied to study the collapse transition [114–116]. These calculations, however, are limited to very short chains and again neglect the explicit effect of the solvent. In most cases a reasonable qualitative agreement between the mean field theory and simulation was found. Although progress has been made, it still remains computationally quite demanding to simulate the collapse of polymers in dense liquid solvents because of the amount of solvent required.

For this reason theoretical descriptions are useful to guide us. However, the Flory approach, or any approach that assumes incompressibility, are of limited applicability. There are two main reasons for this. Firstly, they yield only one theta temperature at a given pressure (except for the special case where χ is zero). In practice there can be as many as three. There is always the one the liquid phase that the Flory theory predicts, because here the incompressibility assumption holds to a good approximation. However,

further coil to globule transitions can occur at elevated temperatures where the solvent is a gas or a super critical fluid. Interestingly, in the supercritical regime as the temperature is raised above the theta temperature there is a second coil to globule transition. So, this time the polymer goes from being expanded to collapsed upon a small increase in the temperature. These effects at elevated temperature are not irrelevant either. The "almost mystical" [117] properties of supercritical solvents make them very interesting for applications in processing. Spice extraction and coffee decaffeination are two examples where this technology is used. Understanding the nature of the coil to globule transition in this regime, is therefore of practical relevance. Secondly, the theta temperatures generally depend on the pressure - even in the liquid phase. Incompressible models apply only in the high pressure limit, so only a model that does not assume incompressibility can tell us something about the effects of pressure.

Dreischor and Lowe [19,118] developed a simple off-lattice thermodynamic model for the size of an isolated solvated polymer, based on Flory's lattice model [19,118] that can account for the effects of compressibility of the solvent. This model indeed shows a far richer behaviour in terms of solvation effects, than is predicted by Flory's model. Using an appropriate equation of state this model, amongst other things, predicts under what conditions a coil to globule transition should occur free of the incompressibility assumption. These predictions are in good agreement with the results of computer simulation. Solving this model analytically requires that the fraction of space occupied by the monomers in the polymer domain is small. As such, it can predict where there is a globular state. It cannot, however, predict analytically its precise nature because for the collapsed state this assumption is invalid. Our aim in this chapter is to extend this model to predict the properties of a collapsed solvated polymer in the supercritical regime. The predictions of the model will be compared to the results of computer simulation. To facilitate this comparison we consider the simplest system. That is, the "best" solvent there is - a solvent composed of the monomers of the polymer itself.

5.2 Developing a simple thermodynamic model

The model derived by Dreischor and Lowe [118] follows in the spirit of Flory and views the solvated polymer as a spherical domain, containing a homogeneous mixture of monomers and solvent, surrounded by pure solvent. Rather than assuming incompressibility, the amount of solvent in the polymer domain is determined from the thermodynamic equilibrium condition, that the chemical potential of the solvent in both domains is equal. The interactions between the monomers and the solvent in both phases are included by taking an ap-

propriate equation of state for the monomer fluid/solvent mixture. As with the Flory theory, the connectivity of the monomers is only taken into account in the form of an additional contribution to the free energy due to changes in the volume occupied by the polymer V relative to the volume occupied by the equivalent ideal chain V_0 (using equation 5.4). We then write the total free energy of the system as

$$G(\alpha) = N_s\mu_s + N_s\tilde{\mu}_m(\alpha) + \Delta G_v(\alpha) - k_B T \ln(V_{sys}) \tag{5.6}$$

with N_s the total number of solvent particles in the system, μ_s the chemical potential of the solvent, $\tilde{\mu}_m(\alpha)$ the excess chemical potential of the monomers, $\Delta G_v(\alpha)$ the configurational free energy given by equation 5.4, and V_{sys} the system volume. Here, we take the limit of infinite dilution ($V_{sys} \gg V$), so V_{sys} is independent of α. Because the monomers are part of a chain, they do not have an ideal contribution to the chemical potential. The polymer is one molecule. If the chain is expanded the monomer density ρ_m in the polymer domain is very low, such that we can write both the monomer and the solvent chemical potential using a Taylor expansion. The condition that the chemical potential of the solvent is the same within and without the polymer domain then implies a slight difference in density between the two domains,

$$\rho_s - \rho^0 = -\frac{u_{ms}\rho_m\rho^0}{1 + u_{ss}\rho^0} \tag{5.7}$$

where ρ^0 is the bulk solvent density and ρ_s the solvent density in the polymer domain. The coefficients u_{ms} and u_{ss} are

$$u_{ms} = (\partial\tilde{\mu}_m/\partial\rho_s)_T]_{\rho^0,0} = (\partial\tilde{\mu}_s/\partial\rho_m)_T]_{\rho^0,0} \tag{5.8}$$

$$u_{ss} = (\partial\tilde{\mu}_s/\partial\rho_s)_T]_{\rho^0,0} \tag{5.9}$$

Both generally depend on the solvent density and temperature but their values are specified given an equation of state. To this order in the monomer density the condition of equal chemical potential also implies that the condition for mechanical equilibrium. Namely, that the pressure is the same in both domains. By minimizing the free energy with respect to α, we then obtain the result derived in a previous publication [118], namely

$$\alpha^5 - \alpha^3 = v\frac{N_m^2}{V_0} \tag{5.10}$$

where v is the excluded volume parameter

$$v = u_{mm} - \frac{u_{ms}^2\rho^0}{1 + u_{ss}\rho^0} \tag{5.11}$$

and the coefficient u_{mm} is

$$u_{\mathrm{mm}} = (\partial\tilde{\mu}_{\mathrm{m}}/\partial\rho_{\mathrm{m}})_T\big]_{\rho^0,0}. \tag{5.12}$$

Again, this is a density and temperature dependent quantity specified by an equation of state.

For the idealized case of a polymer solvated by fluid of its own monomers (the "symmetric" case) the above results simplify and both the solvent density in the polymer domain and the excluded volume parameter can be written in terms of the dimensionless compressibility of the solvent κ, where

$$\kappa^{-1} = \frac{1}{k_{\mathrm{B}}T}\left(\frac{\partial P}{\partial \rho^0}\right) \tag{5.13}$$

This leads to

$$\rho_{\mathrm{s}} - \rho^0 = (\kappa - 1)\,\rho_{\mathrm{m}} \tag{5.14}$$

and

$$v = \frac{1-\kappa}{\rho^0} \tag{5.15}$$

The interesting thing about this is that v is negative, so the polymer wants to collapse, when $\kappa > 1$. This being the case $\rho_{\mathrm{s}} - \rho^0 > 0$, that is the density of the solvent around the polymer is *higher* than the bulk solvent density. This is a very different scenario than the mechanism predicted by the Flory model. There, the coil to globule transition is driven by a de-mixing of the solvent from around the polymer and the solvent is expelled as the polymer collapses to the globular state. Here, this analysis suggests, the solvent almost becomes too good a solvent and the collapse is favourable because it allows the polymer to "suck" more solvent into its domain. We use "suggests" here because the theory makes use of a Taylor expansion of the equation of state about the limit $\rho_{\mathrm{m}} \to 0, \rho_{\mathrm{s}} \to \rho^0$. However, as the polymer collapses ρ_{m} increases so this approach breaks down. All we can really say is that the initial state of the collapse must be accompanied by an increase in the amount of solvent around the polymer because in the initial stages ρ_{m} is still small.

Although we cannot solve the model analytically to determine the nature of the globule state, if we know the free energy as a function of the polymer size it is straightforward to minimize this function numerically. In terms of our model, the polymer is essentially treated as a droplet of size $r \sim V^{1/3}$. Unlike for the the case of an expanded polymer, we can no longer be sure that the pressure in both domains is equal if there is additional contribution to the free energy, in the form of a surface free energy ΔG_γ. This contribution is given by

$$\Delta G_\gamma = 4\pi r^2 \gamma \tag{5.16}$$

where γ is the surface tension. According to Laplace's law this is directly related to the pressure difference between the two domains via

$$\Delta P = \frac{2\gamma}{r} \tag{5.17}$$

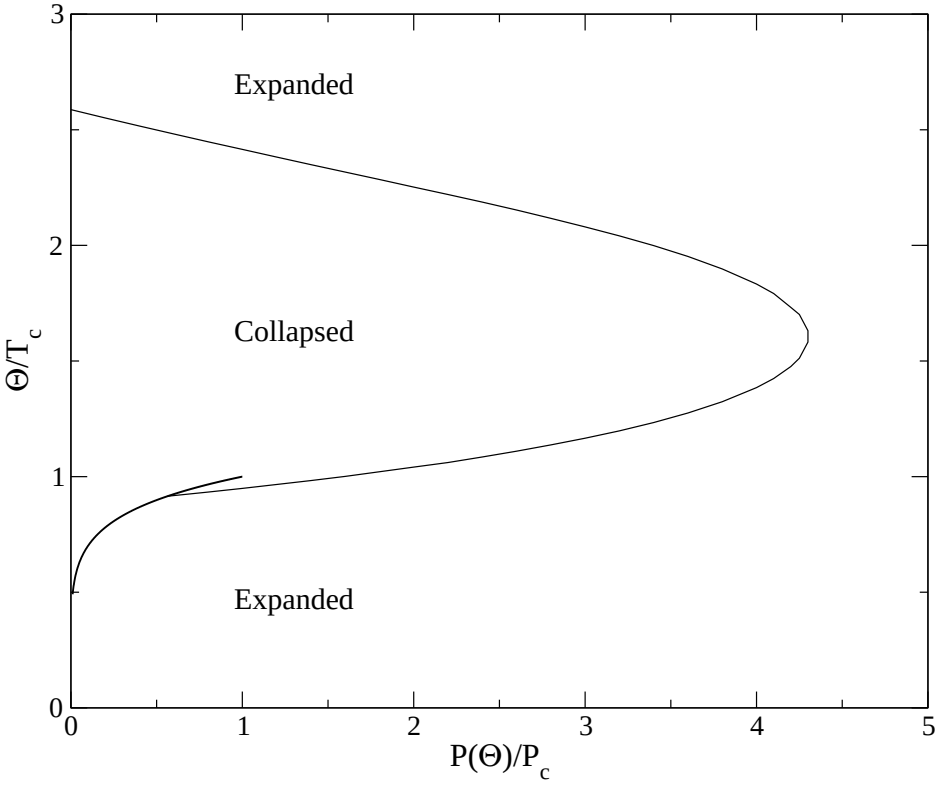

Figure 5.1: Expanded and collapsed region as a function of temperature and pressure as calculated from the Johnson equation of state. The thick solid line represent the (estimated) boiling points for the Lennard-Jones fluid.

We now have two routes to proceed. The first route is to treat this problem as we would treat a normal liquid droplet. We will assume the surface tension is a constant independent of the polymer size. Although this surface tension is not known, we can assume an infinitely long chain (meaning $r \to \infty$), such that the surface free energy is zero. Therefore, the pressure in both phases will be equal. We can thus find the size of the chain, by imposing that the pressure should be equal in the two phases, in addition to the condition for thermodynamic equilibrium (that the solvent chemical potential in both phases should be equal). Because of this additional condition, we are now

no longer free to minimize the free energy with respect to the size of the polymer. We will refer to this model as model 1. The second route we can take is to minimize the free energy of the system with respect to the size of the polymer, as we did in the case of the expanded chain, forgoing the condition that the pressure should be equal in both phases. In doing so we are in fact calculating a surface tension, which does not have to be independent of the polymer size. This model will be referred to as model 2. We will explore both these routes in the rest of this chapter and hopefully be able to conclude which of these is the more realistic, by comparing them to the results of computer simulation. Note that there is some ambiguity comparing simulation results with the theory because the concept of a polymer "volume" is somewhat ambiguous in a simulation. The approach that we used was to take for the theory a spherical volume with a radius of gyration equal to that calculated from the simulations.

5.3 Computational details

The theoretical model was derived using the Lennard Jones equation of state, so we will use that pair potential also in the simulations. The Lennard-Jones potential u_{lj} has the following form:

$$u_{lj}(r) = 4\epsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right]$$

The reason for choosing this potential is the existence of a known equation of state, so we can choose the smallest "typical" value for the cut-off radius often used in literature, in this case $r_c = 2.5\sigma$. Note that the calculations performed by Johnson were done at a cut-off radius of $r_c = 4.0\sigma$, so there will a small discrepancy between the theoretical model and the simulations. We have attempted to correct for this by working relative to the critical temperature and pressure. The configurational entropy is relatively insensitive to the potential, so we choose the simple harmonic potential

$$\vec{f}_{\text{spring}}(\vec{r_{ij}}) = -k d\vec{r_{ij}}$$

where $k = \frac{25}{6}T$. This spring constant k is chosen so that the Kuhn length for an ideal chain with this constant coincides with the minimum of the Lennard Jones potential. To look for the effects predicted by the theory we obviously need to include the solvent explicitly.

This problem will also give us an opportunity to test the new Lowe-Andersen thermostat and the constant pressure algorithm described in Chapter 3 on a slightly more involved problem. Remembering that the Lowe-Andersen thermostat does not suppress diffusion as much as the Andersen

thermostat, we expect it to sample phase space more efficiently. The barostat allows us to specify the state points we are studying in terms of the temperature and pressure. As such, we can study the collapsed state for points where the theories above predict that the collapsed state is solvent rich and test this prediction.

We ran simulations at different state points and chose one for which the solvent density of solvent in the collapsed polymer was predicted to significant. Because the simulations took a large amount of computer time, we choose the points based on 5.1 in the extreme region of the "Collapsed" region (i.e. bottom left, top left, extreme right, centre). The point we chose was $T = 0.85T_c$ and $P = 0.15P_c$. Under these conditions the polymer chain would collapse to a globule allowing for long chains to be simulated. However, since an expanded self avoiding chain has a size which scales with $R_g \propto N^{\frac{7}{12}}$, the amount of solvent required to fill a minimal box encapsulating this chain scales with $N_s \propto N^{\frac{21}{12}}$. Even in the best case, computational time scales linear with the amount of particles, so the computational effort required to calculate longer uncollapsed chains is almost quadratic. For a collapsed chain the size scales with $R_g \propto N^{\frac{1}{3}}$ and hence the amount of solvent needed as $N_s \propto N$ implying that the computational effort can scale linearly. Of course the computational time will never actually be linear due to cache effects and so on, but large chains are feasible with relatively little computational time.

Now there is the question of how to generate an initial configurations. The easiest way to generate a long self avoiding chain is by simply brute force. Start with the current chain and try to add one bead to the end, if that fails, start all over again with a one bead chain. Now such a configuration will not be a collapsed configuration. To get there we did a number of simulations for which we started with a very low density (and pressure) solvent and equilibrated the system at constant volume until the chain is collapsed. At this point the barostat (as described we used in Frenkel and Smit [5]) was enabled. It then slowly shrinks the box in such a way that it goes to the pressure we require. This effectively results in the desired equilibrium configuration, although the actual path of getting there was slightly un-physical. The main justification is that also for a low density system, where it is practical to simulate with a relatively large quantity of solvent, the polymer would collapse to a globule also when coupled to the barostat from the start. This approach makes the simulations, at the state-point we require, practical using an explicit solvent.

One important observation here was that the choice of the thermostat has a very major effect on the time needed for the chain to collapse. As can be seen in figure 5.2 the commonly used Andersen thermostat results in much slower dynamics. In fact, for the longer chains studied here the collapse is so slow

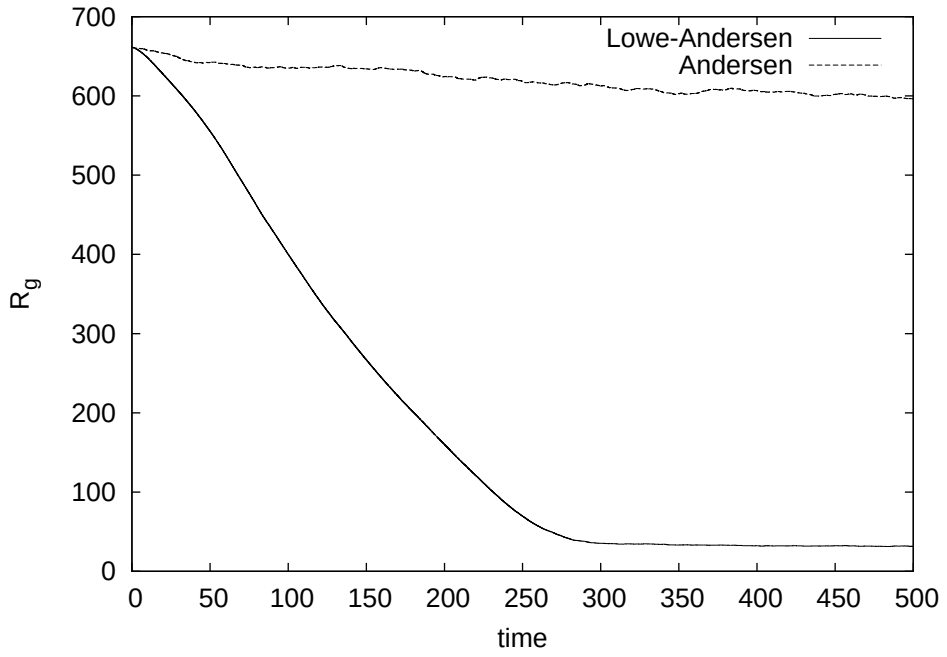

Figure 5.2: The radius of gyration of a chain consisting of 4096 particles over time. The lower value for the Lowe-Andersen reaches a dense packed spherical configuration on this time scale, whereas for the Andersen thermostat it does not.

that the simulations are almost impractical using it. The probable reason is that, because it is momentum conserving, the Lowe-Andersen thermostat preserves hydrodynamic behaviour. It is the hydrodynamic interactions between monomers that lead to the Zimm scaling of the polymer diffusion coefficient $D \propto 1/\sqrt{N}$. Without this hydrodynamic effect (as with the Andersen thermostat) the scaling is Rouse-like, $D \propto 1/N$ [119]. Obviously, for large N there is a big difference between the two and the hydrodynamics significantly speeds up the configurational dynamics of the polymer [120]. Interestingly, this effect has also been observed in the folding of proteins [121]. We are only interested in the static properties of the collapsed state here, but by using a more dynamically realistic methodology, at little additional overhead, we note that we can do this significantly more efficiently.

Once a collapsed configuration has been found, we can use that as a starting point from which we do the simulations to test the theory. The spherical symmetry of the problem allows us to map the data on just a single quantity, the distance to the centre of mass of the polymer. For the density this is a simple mapping, the pressure was slightly more complicated since it is not

by definition a local quantity. By dividing the system into spherical shells centred on the polymer centre of mass we could still obtain a "local" pressure. This way also allowed us to split the pressure into two parts. One radial part parallel to the vector pointing away from the centre of mass, and one tangential part perpendicular to this. We did this with this alternative equation for the radial virial pressure:

$$P_{//} = \frac{1}{3V}\left\langle \Sigma_{i<j}\frac{\vec{f}(\vec{r}_{ij})\cdot\vec{p}_i + \vec{f}(\vec{r}_{ij})\cdot\vec{p}_j}{2}\right\rangle$$

with p_i being the projection of r_{ij} on the unit length vector pointing away from the center of mass through particle i. The tangential virial pressure is then simply the difference between the normal virial pressure and the radial pressure:

$$P_T = P - P//$$

. The results of the simulations will be discussed in the next section.

5.4 Results

	Model 1 isolated/solvated	Model 2 isolated/solvated	Simulation isolated/solvated
ρ^0	0/0.019	0/0.019	0/0.019
ρ_m	0.748/0.592	0.527/0.253	$\sim$ 0.82/$\sim$ 0.65-0.66
ρ_s	0/0.141	0/0.288	0/$\sim$ 0.12-0.13
$\rho_m + \rho_s$	0.748/0.733	0.527/0.541	$\sim$ 0.82/0.78
ρ_s/ρ^0	-/7.468	-/15.22	-/$\sim$ 6.32-6.84

Table 5.1: The bulk solvent density ρ^0, monomer density ρ_m, solvent density ρ_s and total density $\rho_m + \rho_s$ inside the polymer globule at $T/T_c = 0.85$ and $P/P_c = 0.15$ for an isolated Lennard-Jones chain and a Lennard-Jones chain in a symmetric Lennard-Jones solvent as predicted by both models for the collapsed state and the results of simulation.

As was stated in the introduction, we will consider the simplest case of a collapsed solvated polymer, namely a polymer in a symmetric solvent (a polymer solvated by its own monomers). We opt to use the Lennard-Jones potential to describe the interactions between the monomers and the solvent

$$u(r) = 4\epsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right] \tag{5.18}$$

where σ is characteristic length and ϵ the characteristic energy of the particles. In order to make a direct comparison between our two models and the results of simulation, we will need to use an accurate equation of state for the Lennard-Jones fluid in the two models. We will use the equation of state developed by Johnson et al. [30]. This equation of state takes the form

$$\begin{aligned}\frac{P}{\rho k_B T} &= 1 + \left(a_1 + \frac{a_2}{T^{*1/2}} + \frac{a_3}{T^*} + \frac{a_4}{T^{*2}} + \frac{a_5}{T^{*3}}\right)\rho^* + \\ &\left(a_5 + \frac{a_6}{T^*} + \frac{a_7}{T^{*2}} + \frac{a_8}{T^{*3}} + \frac{a_9}{T^{*4}}\right)\rho^{*2} + \\ &\left(a_{10} + \frac{a_{11}}{T^*} + \frac{a_{12}}{T^{*2}}\right)\rho^{*3} + \frac{a_{13}}{T^{*2}}\rho^{*4} + \\ &\left(\frac{a_{14}}{T^{*2}} + \frac{a_{15}}{T^{*3}}\right)\rho^{*5} + \frac{a_{16}}{T^{*2}}\rho^{*6} + \left(\frac{a_{17}}{T^{*2}} + \frac{a_{18}}{T^{*3}}\right)\rho^{*7} + \\ &\frac{a_{19}}{T^{*3}}\rho^{*8} + \frac{\exp(-\gamma^2\rho^{*2})}{\rho^* T^*}\left\{\left(\frac{a_{20}}{T^{*2}} + \frac{a_{21}}{T^{*3}}\right)\rho^{*3}\right\} + \\ &\frac{\exp(-\gamma^2\rho^{*2})}{\rho^* T^*}\left\{\left(\frac{a_{22}}{T^{*2}} + \frac{a_{23}}{T^{*4}}\right)\rho^{*5} + \left(\frac{a_{24}}{T^{*2}} + \frac{a_{25}}{T^{*3}}\right)\rho^{*7}\right\} + \\ &\frac{\exp(-\gamma^2\rho^{*2})}{\rho^* T^*}\left\{\left(\frac{a_{26}}{T^{*2}} + \frac{a_{27}}{T^{*4}}\right)\rho^{*9} + \left(\frac{a_{28}}{T^{*2}} + \frac{a_{29}}{T^{*3}}\right)\rho^{*11}\right\} + \\ &\frac{\exp(-\gamma^2\rho^{*2})}{\rho^* T^*}\left\{\left(\frac{a_{30}}{T^{*2}} + \frac{a_{31}}{T^{*3}} + \frac{a_{32}}{T^{*4}}\right)\rho^{*13}\right\}\end{aligned} \tag{5.19}$$

where $\rho^* = \rho\sigma^3$ is the reduced density and $T^* = k_B T/\epsilon$ is the reduced temperature of the system. The parameters $a_1 - a_{32}$ can be found in [30]. The first thing we have to do, is calculate a phase diagram, to see in what regions of phase space the polymer is expanded or collapsed. We can obtain the theta temperature by setting the excluded volume parameter $v = 0$ in equation 5.10. The excluded volume parameter itself is obtained using equation 5.20. The resulting phase diagram is presented in figure 5.1. In this figure the theta temperature Θ is plotted as a function of the pressure. These quantities have been made dimensionless by the critical properties of the solvent (for a Lennard-Jones fluid, $T_c^* = 1.32$, $P_c^* = 0.13$). For this system there can be up to two theta temperatures at constant pressure. For the symmetric case the Flory model predicts that there is no theta temperature, and indeed the low temperature theta point in the liquid phase is absent for this model.

In order to obtain an idea what the globule state might look like, we pick a state point within the collapsed region ($T/T_c = 0.85$, $P/P_c = 0.15$). We then calculate the monomer and solvent densities within the polymer domain for both models. The results are presented in table 5.1. At this state point the

bulk solvent (the pure solvent phase in terms of the two models) is a dilute gas. Both models predict that the solvent density in the polymer domain is significantly higher than in the pure solvent phase. Model 2 predicts the largest difference in the solvent density. Both models predict that the total density in the polymer domain is liquid-like ($\rho > \rho_c$), where model 1 predicts the highest total density. We would like to note, that for both models the contribution of the configurational free energy ΔC_v to the total free energy is negligible. We also looked at the case of an isolated polymer chain, where the monomers interact with the Lennard-Jones potential. In this case both models predict the density of monomers in the polymer domain is approximately equal to the total density of particles predicted for the solvated chain.

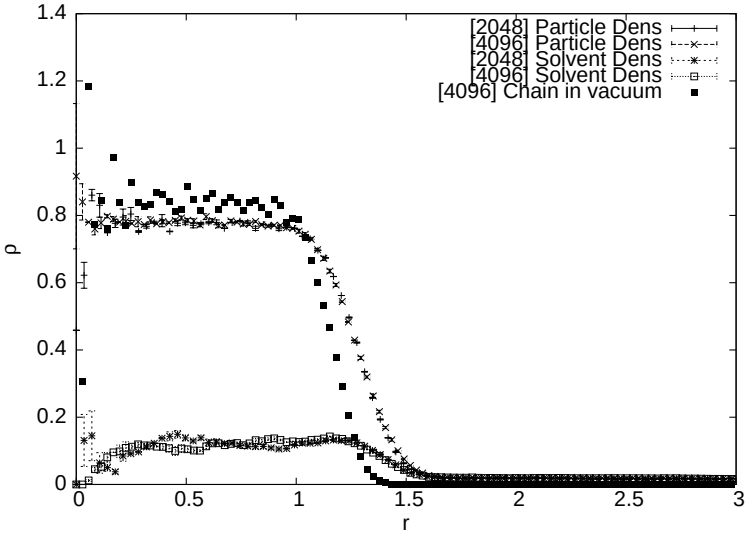

Figure 5.3: The solvent and total density ρ as a function of the distance from the center of mass r for a Lennard-Jones chain in a symmetric Lennard-Jones solvent at $T/T_c = 0.85$ and $P/P_c = 0.15$. The x-axis is in units of the polymer radius of gyration. The y-axis is in reduced Lennard-Jones units.

For comparison, we also performed simulations. These simulations were done at the same state point as was used for the model predictions ($T/T_c = 0.85$, $P/P_c = 0.15$) for a chain consisting of 2048 and of 4096 monomers. In the case of 2048 monomers, 4952 solvent particles were used, and in the case of 4096 monomers, 4904 solvent particles were used. In addition we also simulated a chain consisting of 4096 monomers in vacuum. The simulation results are shown in table 5.1, figure 5.3, and figure 5.4. The results indicate that the solvent density around the polymer is indeed significantly higher than in the bulk solvent. Figure 5.3 shows that if we plot the solvent density as

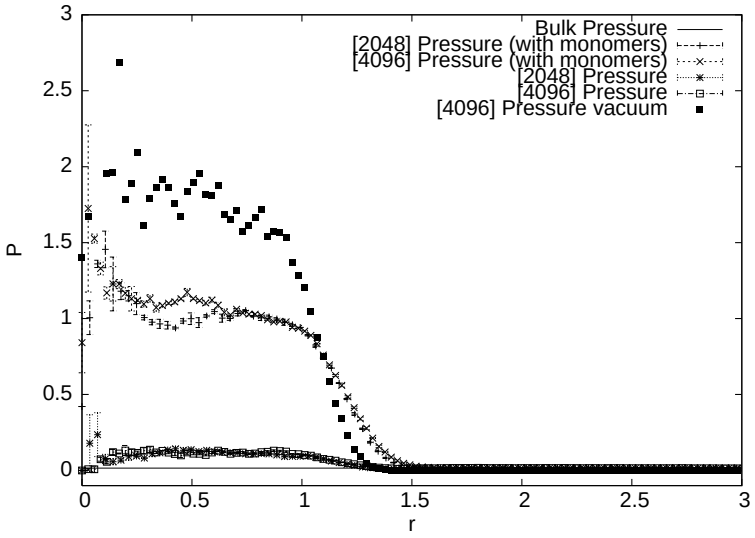

Figure 5.4: The solvent and total pressure P as a function of the distance from the center of mass r for a Lennard-Jones chain in a symmetric Lennard-Jones solvent at $T/T_c = 0.85$ and $P/P_c = 0.15$. The x-axis is in units of the polymer radius of gyration. The y-axis is in reduced Lennard-Jones units.

a function of the distance from the centre of mass of the polymer the density within a distance of approximately $r = 1$ in units of the polymer radius of gyration is roughly six to seven times higher than in the bulk solvent for both the chain of 2048 and 4096 monomers. In table 5.1 the results of the simulation are compared with the predictions of the two models. Although both models show the correct qualitative behaviour in terms of the solvent density, model 1 clearly outperforms model 2 in the quantitative prediction of the different densities, for both the chain in vacuum and the solvated chains. In addition to calculating the radial densities, we also calculated the radial pressure using the virial theorem. These results have been plotted in figure 5.4. The pressure is seen to increase significantly near the polymer centre of mass. In that respect both models are wrong. Although not shown in table 5.1 the pressure that we obtain from model 2 is negative ($P \approx -3.65$). This implies that the surface tension is negative and decreases with an increasing number of monomers (according to equation 5.17 $\gamma \sim -N_m^{1/3}$). For model 1 we had assumed the pressure around the polymer to be equal to the bulk solvent pressure.

5.5 Discussion

In this chapter we extended the model, previously developed for the size of a solvated polymer, to study the nature of the globule state. We derived two similar thermodynamic models. Both of these models treat the collapsed polymer essentially as a two phase problem. The first phase contains of a homogeneous mixture of monomers and solvent (the polymer domain). The second (bulk) phase contains pure solvent. The condition for thermodynamic equilibrium is that the chemical potential of the solvent is equal in both phases. In the first model it is then assumed that we can treat the polymer as we would a liquid droplet. There is a constant surface tension, such that as the number of monomers becomes very large the pressure in the two phases is equal. The polymer size is thus obtained by adding the condition that the pressure in both phases is equal. In the second model the polymer size is obtained by minimizing the free energy with respect to the size of the polymer. The interactions in both phases are described by an appropriate equation of state. In this chapter we considered a symmetric system, where the monomers and solvent particles interact via a Lennard-Jones potential.

Both models predict that the solvent density inside the polymer domain is significantly higher than the bulk solvent density. The total density of particles in the polymer domain (collapsed polymer plus solvent) is liquid-like. These predictions were confirmed by computer simulation. The simulations also show (see figure 5.1) that, as both theories require, the monomer density and solvent density in the collapsed polymer as a function of the dimensionless distance from the centre of mass are independent of the number of monomers in the polymer for a sufficiently long polymer. The functions are not, however, step functions as the theoretical models require. This is clearly a limitation of the theories. The first model gives the best quantitative agreement with simulation. These results contradict the predictions of the Flory model, which says that a polymer chain always expels the solvent as it collapses. The fact that the Flory-Huggins model predicts this, should be attributed to its underlying assumption of incompressibility. The condition that the total density has to remain constant, by construction implies that as the monomer density increases the solvent density should decrease.

So, which of the two thermodynamic models is the better representation of reality? Although, the first model clearly gives the better prediction for the densities of monomers and solvent within the polymer globule, neither model gives correct (qualitative) predictions for the pressure. The simulations clearly show a significant increase in the calculated pressure within the polymer chain. The first model by construction says that the pressure within the polymer should equal the pressure of the bulk solvent, whereas the second model predicts a negative pressure. According to Sanchez the theta

temperatures of a polymer solution should correlate with the critical solution temperatures, a fact since confirmed by simulation [122,123]. This implies that the polymer globules will want to coalesce to form a bulk phase. The second model predicts a negative surface tension, which means that the polymers will prefer to stay in their globule state, rather than form a bulk phase. It is for this reason that we believe the first model is the better model, as it does not contradict the results of Sanchez and co-workers. To further improve the model, we need to obtain a better understanding of the surface effects on the properties of a collapsed polymer.

Nonetheless, the simulations confirm that the qualitative predictions of the models that the collapse mechanism for a polymer in a low density solvent is very different in nature to that at liquid-like solvents. The collapsed state is indeed solvent rich rather than being solvent free. This suggests that studying the coil to globule transition for these systems requires that an explicit solvent is included, however computationally inconvenient this is. In addition this chapter gave us a good practical test case for the Lowe-Andersen thermostat and the barostat. Notably, using the former, the collapse dynamics are so much faster, that using a non-momentum conserving thermostat of the Langevin type would have made these simulation impractical.

Chapter 6

Inducing regular flow patterns

Dissipative Particle Dynamics was introduced as a particle method to model the behaviour of fluids on length scales in between the atomic scale and the "continuum scale" where thermal fluctuations play no noticeable role. The beauty of this method lies in the inclusion of both thermal fluctuations and hydrodynamics. Both of course within certain limits, DPD will probably never be used to model anything larger than a millimetre, and getting the right rate constant for micelle formation is also out of reach.

One thing we might be interested in, that should in theory be possible with DPD, is the behaviour of small structures in small flows. There are many problems where we can use DPD to model the inherent flow in a system resulting simply from the system geometry, for example a droplet of species A falling through a bulk of species B. Now one might also be interested in systems where we can specify the flow itself. Unfortunately there are not many ways of doing this. The only relatively simple method is a shear flow using the Lees-Edwards boundaries. There also exist more complex methods for simulating elongational flow, but those methods are quite convoluted [124,125]. For example the Kraynik-Reinelt method only allows for a periodic elongational flow, not a constant over time flow. For comparison with experiment this is, however, an important case. The response of polymers to microfluidic flows of this type are used to probe their elastic behaviour [126–128]

From basic fluid dynamics we know that on small enough length-scales any type of flow is a linear combination of an elongational/rotational and a shear flow. Given that the shear flow problem is already solved adequately through Lees-Edwards boundaries, we can focus on finding a time-independent flow set-up that has the required elongational flow as a steady state in (at least a large part of) the system. The important characteristic of elongational flow is the linear velocity gradient in the direction of the flow $\frac{\partial v_x^{\rightarrow}}{x} = \epsilon$. It would be useful to be able to construct a flow with this property to

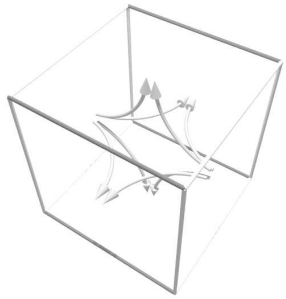

Figure 6.1: Schematic of the desired elongational "3D" flow with 4 flows going towards and 2 going away from the stationary point.

verify certain theoretical predictions or to study breakup behaviour of complex structures.

A very common approach in particle models is to model the system as a periodic system, meaning particles that go out on one side of the domain come in at the other side. Combining this with the requirement of a certain flow will have us carefully consider conservation of mass and momentum since we cannot simply use inflow and outflow conditions like in regular CFD. We can also specify 2 different types of elongational flow, one that is basically the 2D flow stacked on top of itself many times (fig 6.3) and the 3D flow that would be 4 streams flowing in and 2 out or the other way around (type A in figure 6.1 and type B in figure 6.2). We will refer to these two as 2D and 3D flow respectively, even though both of course represent a 3D system.

Since both of these flows do not directly allow them to be present in a periodic system, we introduce 8 of these in a 2x2x2 way in a single simulation box. For the 2D flow we can simply stack identical flows on top of each other and alternate the flow direction in the other 2 directions for each unit. For the 3D flow we alternate type A and type B such that the 6 nearest neighbours of a cell are of the other type.

This construction satisfies at the very least the mass conservation constraint for the periodic system. Now we only need to come up with a way to induce this regular flow pattern. Unfortunately we cannot do this in a way which is as elegant as the Lees-Edwards boundary condition for which one directly enforces a velocity. We do it by applying a force to the fluid locally at certain places. While it will be clear immediately that this will not allow direct control of the exact flow pattern, we will show that this method is good

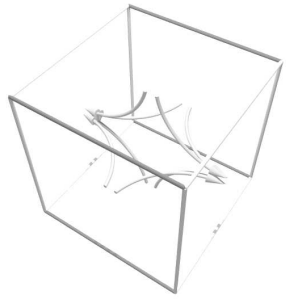

Figure 6.2: Schematic of the desired elongational "3D" flow with 4 flows going towards and 2 going away from the stationary point.

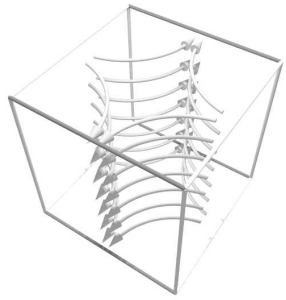

Figure 6.3: Schematic of the desired elongational "2D" flow

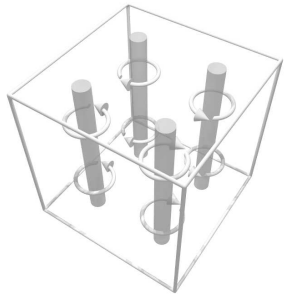

Figure 6.4: Location and direction of "rotors" required to set up a periodic 2D elongational flow

enough for the relevant problems.

6.1 The methods

Since we are working with a mesoscale method, we will use the analogy of a microfluid array. This is about the right length-scale and the periodicity of an array is a good approximation of the periodic boundary conditions.

Using this analogy we can introduce tiny rotors into the system that stir the fluid. In our simulation box we could do this by applying a force to small bits of the fluid, assuming the rotors are small. For our desired 2D flow pattern we could insert 4 long cylindrical rotors. Similarly we can set up 8 rotors to achieve a 3D elongational flow as can be seen in fig 6.5.

The rotors themselves can be built in many different ways, but we choose the following simple approach for the 2D flow. We define cylindrical regions as shown in figure 6.4. The cylinders are all aligned, in our method all parallel to the z-axis $\hat{e}_z$. The regions are spaced evenly and the cylindrical "rotor" region has a radius R_r. Now for each particle i inside radius R_r of a rotor we apply a force $\vec{f}_i$ given by

$$\vec{f}_i = m_r \hat{e}_z \times \vec{dr}_i$$

where $\vec{dr}_i$ is the shortest distance vector to the axis of a cylinder, and m_r is the interaction strength of the rotor, that we can choose arbitrarily. The sign of m_r should alternate for the rotor regions in a checker-board style.

As the force applied in the rotor regions is not guaranteed to sum to zero we are introducing an artificial drift into the system. To avoid this drift we

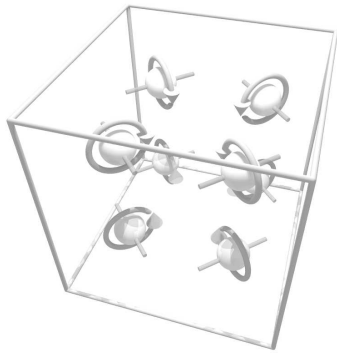

Figure 6.5: Location and direction of "rotors" required to set up a periodic 3D elongational flow, notice how one gets 8 unique stationary points per unit cell in this way. Of those 4 have 4 ingoing and 2 outgoing directions and 4 have 2 ingoing and 4 outgoing directions.

keep track of the net total force $\vec{f}_n$ applied in the rotor regions and apply a force $-\frac{1}{N}\vec{f}_n$ to all N particles making up the system to avoid a drift.

For the 3D elongational flow we follow an analogues procedure, but instead of defining 4 cylindrical regions, we define 8 spherical regions with radius R_r. The 8 rotor regions in the cubic box with length L are located at $\{(\frac{1}{2} \pm \frac{1}{4})L, (\frac{1}{2} \pm \frac{1}{4})L, (\frac{1}{2} \pm \frac{1}{4})L\}$. The rotor rotation axis are aligned in one of 2 directions $\hat{e}_r$, $\{1, 0, 1\}$ and $\{1, 0, -1\}$, when viewed parallel to the y-axis again all in a checker-board fashion. For each particle i within distance R_r of a rotor with a rotation axis $\hat{e}_r$ we apply a force $\vec{f}_i$ given by

$$\vec{f}_i = m_r \hat{e}_r \times \vec{dr}_i.$$

Here $\vec{dr}_i$ is the distance vector to the center of the rotor region and m_r is again the interaction strength of the rotor. The sign of m_r is alternated such that the effective force follows the directions as given in 6.5.

6.2 Results

A good test for the usefulness of our method will be by examining the flow velocity along the axes that go through the stagnation points. The common theoretical approach to treat elongational flow is by assuming that $\frac{\partial v_i}{\partial i} = \epsilon i$, so it would be useful if our flow has that property. We set up a cubic system with length $L = 30$ containing 27000 identical particles and used the 2D flow

rotors. Units are chosen such that the unit of length is the force interaction radius, $k_BT = 1$ and particle masses are 1. The choice for the rotor force magnitude was such that the largest average local flow does not become much larger than the typical thermal velocity. With an iterative approach we find that for $m_r = 0.2$ we get a maximum average flow magnitude of about 2.8, not too much larger than relatively high thermal fluctuations when $k_BT = 1$. The interesting properties are now the x and y components of the average flow velocity (remember the axes of the rotors are parallel to the z-axis). Because the flow velocity has no expected dependence on the z coordinate, we average the particle velocities for all z-coordinates in equal sized bins defined along the x- and y-axes. As we can see in figure 6.6 the average flow has the qualitative properties we expect, there is a clear symmetry, the stagnation points are clearly visible and the area around the stagnation points follows the pattern we expect. For a more quantitative analysis we can also take a section of the flow profile along the x-axis at $y = 15$ (meaning we effectively include the average velocity for $14 < y < 16$. In figure 6.7 we see that the flow along this axis also follows the quantitative constraint that the x-velocity is linear with the x-coordinate.

6.3 Conclusion

While not as elegant as Lees-Edwards boundaries, we have found a relatively simple and straightforward method to construct a time-independent elongational flow. The method will only generate a usable flow in a part of the system, but in this part the flow has the desired properties and does not fluctuate over time as other methods do.

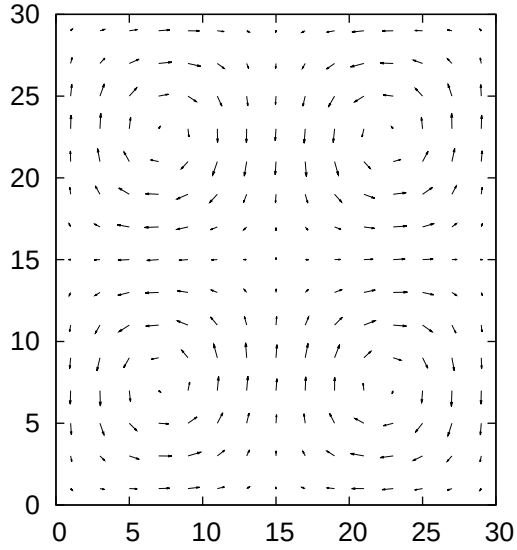

Figure 6.6: Flow field for the 2 dimensional rotor approach. The system consists of a square box of size 30 containing 27000 particles. The external force is applied within a range of the 4 axes where $x = 15 \pm 7\frac{1}{2}$ *and* $y = 15 \pm 7\frac{1}{2}$*. The DPD force parameter was set to 10, the temperature to* $1k_BT$ *and the system was simulated over 10000 time-steps with* $\Delta t = 0.02$*. The largest average velocity was 2.8, all velocity vectors have been rescaled such that the largest has length 1. We choose the rotor interaction strength to be* $m_r = 0.2$*.*

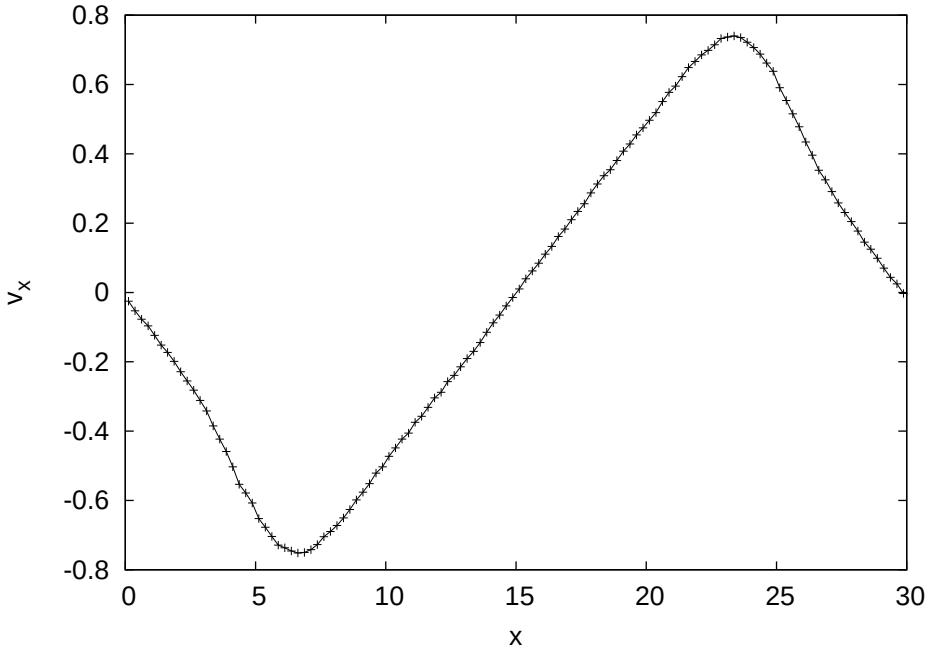

Figure 6.7: The same data as described in 6.6, but now only the x component of the velocity visualized only along the axis where y=15. Between $x = 7.5$ and $x = 22.5$ is clearly the straight line characteristic for elongational flow.

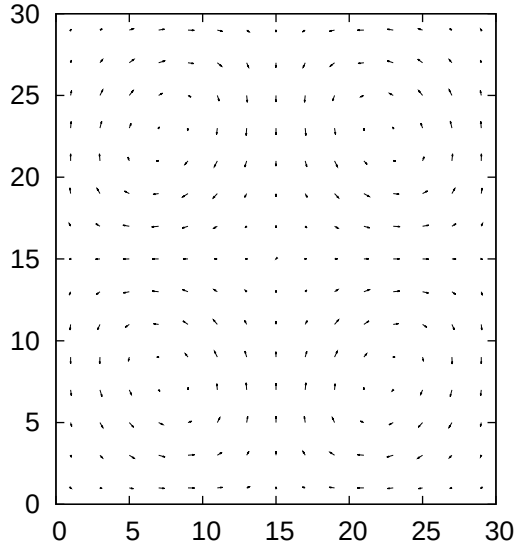

Figure 6.8: Flow field for the 3 dimensional rotor approach. The system consists of a square box of size 30 containing 27000 particles. The external force is applied within a range 6 of the 8 axes described in the text. The DPD force parameter was set to 10, the temperature to $1k_BT$ *and the system was simulated over 10000 time-steps with* $\Delta t = 0.02$*. The largest average velocity was 1.3, all velocity vectors have been rescaled such that the largest has length 1. We choose the rotor interaction strength to be* $m_r = 0.05$*. Note that we are looking along one of the axis here, just like in 6.6, but in this case there is a velocity dependence along this axis, so the graph doesn't show all the information.*

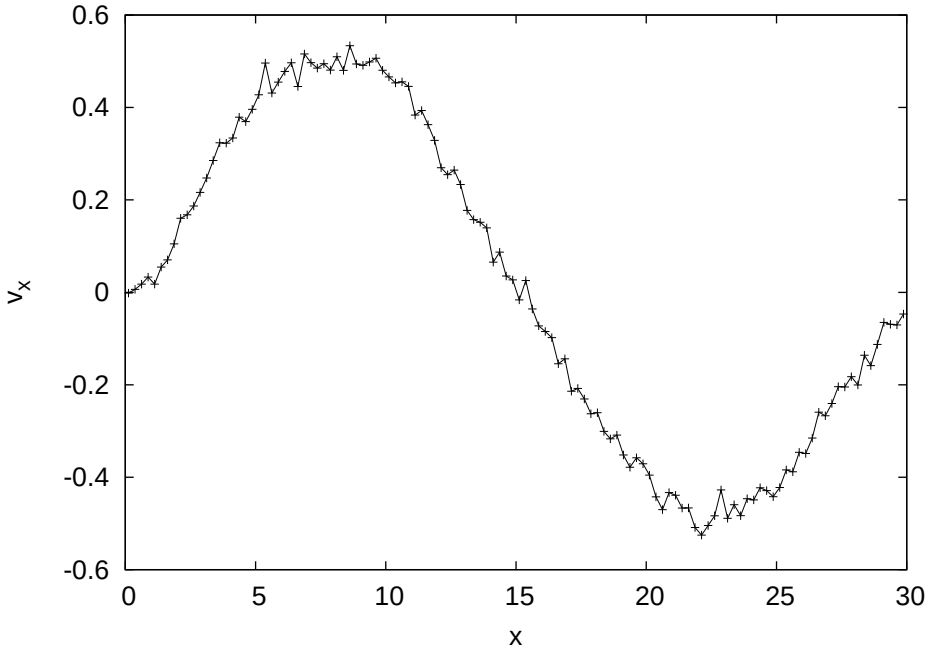

Figure 6.9: The same data as described in 6.8, but now only the x component of the velocity visualized only along the axis where $y = 15 \pm 1$ and $z = 15 \pm 1$. The data is noisier than the data in figure 6.7 since it is an average over a smaller volume. The shape of the flow is more sinusodial than piecewise linear, but is still relatively straight in the center region.

Chapter 7

Polymer scaling behaviour in non-integer dimensions

7.1 Introduction

We are generally comfortable with the idea of different spatial dimensionalities, d. For example, a one dimensional system lies on a line (described by one set of coordinates), a two dimensional system on a plane (described by two sets of coordinates) and a three dimensional system (described by three sets of coordinates) is that space that we commonly experience. Extending this to higher dimensionalities, described by an increasing number of coordinates, is mathematically (if not conceptually) straight forward. As an example, the integral I_d of some radially symmetric function $f(r)$ over a hyperspace volume V of radius R is

$$I_d = dC(d)\int_0^R f(r)r^{d-1}dr \tag{7.1}$$

with

$$C(d) = \frac{\pi^{d/2}}{\Gamma(1+d/2)} \tag{7.2}$$

Notice that in the above we could in principle treat d as a continuous variable. That is, also take non-integer values of d. This would correspond to the odd concept of a system described by a fractional number of Cartesian coordinates. Nonetheless, several quantities when considered in hyperspace yield an expression that "advertises" such an extension to non-integer dimensions in this way [129]. The question is; does this extension to fractional dimensionalities have any physical meaning, or is it simply one of an infinite number of possible interpolation formulas between integer spatial dimensions? After all, we could always augment this type of procedure by introducing an additional function that vanished for integer dimensions and still recover the correct expression

for integer dimensions. With this in mind, Stillinger attempted to derive an axiomatic, rather than phenomenological, basis for non-integer dimensions.

The relevance of this to polymers is as follows. The Flory result for the scaling exponent, ν, of a polymer with excluded volume (this is a chain where the monomers have a finite volume and cannot overlap, see Chapter 5) can be derived in general spatial dimensions yielding

$$\nu = \frac{3}{d+2} \tag{7.3}$$

for $1 \leq d \leq 4$, and $\nu = 1/2$ for $d \geq 4$. In the one dimensional case the scaling of the end to end vector $\vec{R}_e$ is trivial ($\langle R_e^2 \rangle^{\frac{1}{2}} = N$)) with N the number of monomers and assuming the Kuhn length to be unity. The Flory model reproduces this correctly. Similarly, the two dimensional case can also be solved analytically ($\langle R_e^2 \rangle^{\frac{1}{2}} = N^{\frac{3}{4}}$ [130]). and agrees with the Flory result. It is also correct in its prediction that for $d \geq 4$ there is no "excluded volume" effect, because then $\nu = 1/2$ - the ideal chain result. One approach to improving the estimate of the exponent in three dimensions is to take advantage of this fact and write the exponent as a perturbation expansion from $d = 4$ in terms of a parameter $\epsilon = 4 - d$. Deriving this expansion involves generalizing integrals to non-integer dimensions using the form given in equation 7.1. This so called "epsilon expansion" yields to first order

$$\nu(\epsilon) = \frac{1}{2} - \frac{\epsilon}{16} \tag{7.4}$$

In fact, the first indication that the Flory result was not in general exact came from the observation that it yields an incorrect coefficient of 1/12 for the second term in the ϵ-expansion [131, 132]. Consequently, its prediction for the most practically important case of three dimensions, that $\nu = 3/5$ is incorrect.The best current numerical estimate is that $\nu = 0.5876$ [133]

Clearly, the concept of non-integer dimensions is essential to this approach for deriving the scaling exponent. Numerical solutions to the problem provide predictions for the exponent in all dimensionalities, as do other closed expressions for the exponent based on alternative approaches to that of Flory [134, 135]. In lower dimensions the excluded volume affect is much stronger. That is short chains are more expanded and closer to the scaling limit, making it easier to determine the correct value of the exponent. An obvious way to test these predictions, rather than the brute force approach of simulating ever longer chains in three dimensions, is to develop a methodology for simulating them in fractional dimensions. For example, one could look for the lower critical dimensionality (the scaling exponent reaching unity for a dimensionality between $d = 1$ and $d = 2$) predicted by these theories.

The best known subject linked in a way to non-integer dimensional spaces is probably the group of objects known as fractals. These typically consist of subset of points from an integer dimensional space that is generated through some recursive rules. Trying to generate polymer chains in such spaces gives scaling exponents for even the most trivial case that are very different from the behaviour in the integer dimensional spaces in which the fractal is generated (See figures 7.1 and 7.2). This is related to the well studied phenomena of anomalous diffusion. We think that although this has its uses in fields of research such as diffusion in porous materials, it will not be of much use for problems related to theoretically predicted scaling exponents of polymers in fractional (as opposed to fractal) dimensions, because there is no demonstrable link to the mathematical procedures used when considering polymers in fractional spatial dimensionalities.

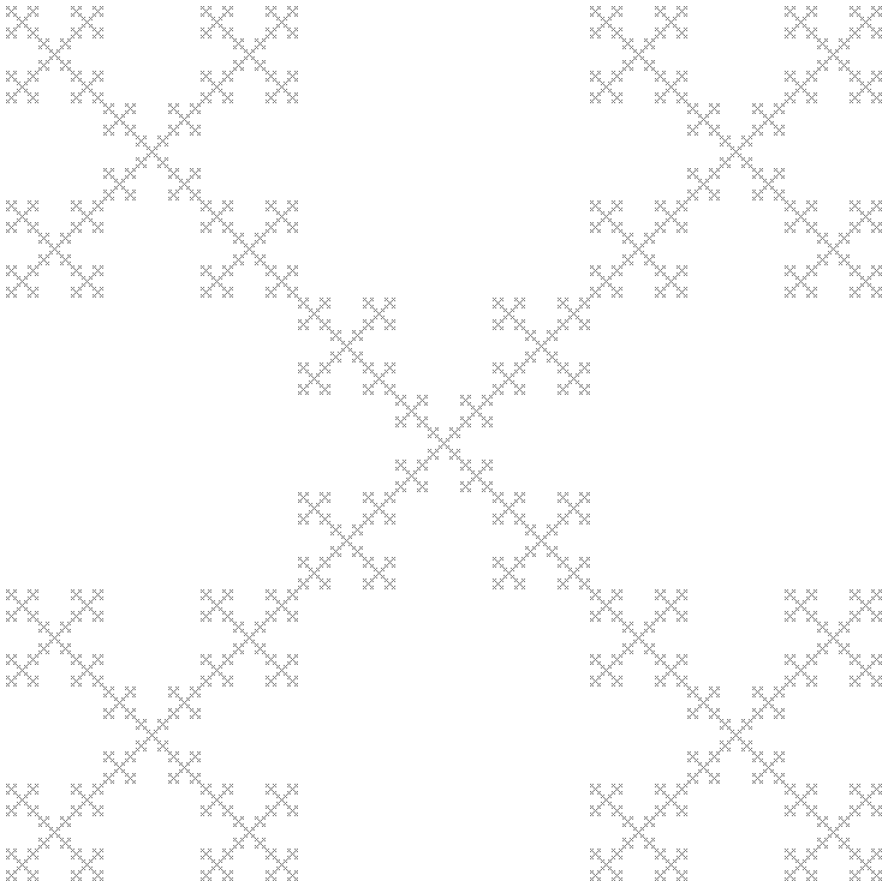

Figure 7.1: Fractal on which we generated random ideal chains.

Other people have already attempted to solve just this problem in non-

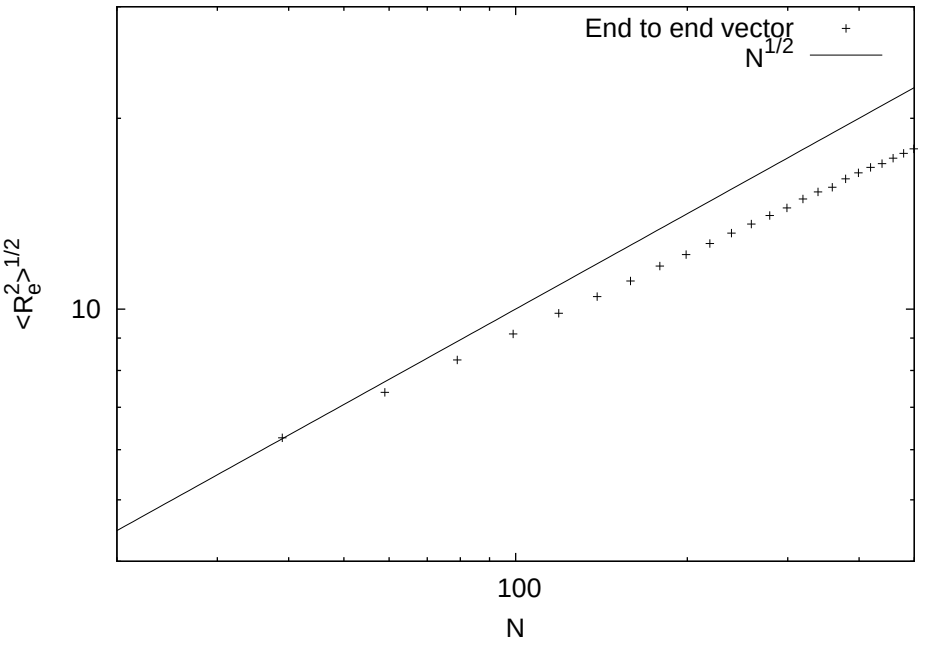

Figure 7.2: End to end vector scaling for an ideal chain constrained to be on a fractal with a dimensionality of 1.46. The best fit to this data gives a scaling exponent of 0.416(1). This is very different from the 0.5 that it should be in all integer dimensions.

integer dimensions using re-normalization theory. We will approach it using a different method based on the proposal of Stillinger for the treatment of non-integer dimensional spaces [129].

7.2 Non-integer dimensional spaces

Stillinger starts by giving 5 axioms that should apply to a non-integer dimensional space $\int_D$. Most are very similar to the axioms relating to regular integer dimensional spaces, but we will list all of them here.

A1 $\int_D$ is a metric space.

A2 $\int_D$ is dense in itself.

A3 $\int_D$ is metrically unbounded.

A4 For any two points $y, z \in \int_D$, and any $\epsilon > 0$ there exists an $x \in \int_D$ such that:

(a) $r(x,y) + r(x,z) = r(x,z)$;

(b) $|r(x,y) - r(x,z)| < \epsilon r(y,z)$.

A5 The following equation holds:

$$\int dx_0 \exp(-\Sigma_{j=1}^{n} \alpha_j r_{0j}^2) = \left(\frac{\pi}{\tau}\right)^{D/2} \exp\left(-\frac{1}{\tau}\Sigma_{j<k=1}^{n} \alpha_j \alpha_k r_{jk}^2\right)$$

with $\tau = \Sigma_{j=1}^{n} \alpha_j$

Here α is some arbitrary variable, r_{ij} is the distance between point i and j and finally n is the number of points considered. The last axiom can be thought of as a more generalized version of the well known Gaussian integral $\int d\vec{r} e^{(-\alpha r^2)} = \frac{\pi}{\alpha}^{\frac{D}{2}}$.

There are some consequences related to defining a space in this way. Most importantly, it does not allow us to write coordinates of a point in the regular vector way, we can only work with distances.

Stillinger then derives from axiom A5 an expression for integration weights. The easiest way to see where and how one needs those is from the volume of a sphere/circle in D dimensions. Knowing that we can only work with distances, we need a function that gives the amount of space at a distance R from a given point (the integration weight W_1), if we then integrate this function from 0 to R we get the amount of space embedded in such a volume. Stillinger gives $W_1 = \frac{2\pi^{D/2}}{\Gamma(D/2)} r^{D-1}$. Carrying out this integration, one gets $V(R,D) = \int_0^R W_1(r)dr = \frac{\pi^{D/2}R^D}{\Gamma(1+\frac{1}{2}D)}$. This indeed reduces to the well known results $2R$, πR^2 and $\frac{4}{3}\pi R^3$ for 1,2 and 3 dimensions respectively. Similarly, one can use the other integration weights W_n for integrals involving n distances, the expressions for W_n with higher n do however get very tedious to calculate.

Stillinger suggest that we use the following "2 center weight":

$$W_2(x_1, x_2 | r_{01}, r_{02}) = 2^{(D-3)}\sigma(D-1) r_{01} r_{02} r_{12}^{(2-D)} \Delta^{D-3}(r_{01}, r_{02}, r_{12})$$

with Δ being the area of a triangle with sides r_{01},r_{02} and r_{12} that can be calculated using a Cayley-Menger determinant [136]. A shorthand is used for σ, defined as $\sigma(D) = \frac{2\pi^{\frac{D}{2}}}{\Gamma(\frac{D}{2})}$. This indeed seems to be the right expression for the area of the overlap between two circles/spheres shells separated a distance r_{12} and with an inner radius of r_{01} and r_{02} and an outer radius of $r_{01} + \delta r_1$ and $r_{02} + \delta r_2$.

Since our main interest in this work is with polymers, we will first attempt to interpret the weight functions such that they are useful for generating valid configurations in D dimensions. Consider a configuration with 3 points x_0, x_1

and x_2 and only two known distances r_{01} and r_{12}. In all integer dimensions the distribution for the 3rd one can be calculated, for example the simple case where $r_{01} = r_{12} = 1$. In 1 dimension there are two possible solutions, $r_{12}\epsilon\{0, 2\}$. Both should be sampled with equal weight if we want to sample such configurations. For 2 dimensions we can derive the expression by realizing that the angle θ between the two known lines is uniform. The unknown distance is then $r_{12} = 2\sin(\theta/2)$, solving for θ we get $\theta = 2\sin^{-1}\frac{r}{2}$. If we then take the derivative with respect to r_{12} we get $\frac{1}{\sqrt{1-\frac{r^2}{4}}} = \frac{\partial\Theta}{\partial r}$.

We can take a similar approach for the 3 dimensional case (which we will not use further on). The angle between the two known lengths is uniform in $\cos\Theta$, which in this case is equal to the dot-product (ω). The resulting unknown distance r_{12} is now $\sqrt{2-2\omega}$, solving for ω we get $\omega = 1 - \frac{r^2}{2}$. Again taking the derivative we get $\frac{\partial\omega}{\partial r_{12}} = r$. This implies that if we take W_2 in any dimension $d > 1$, we can use it to generate a distribution from which we need to sample.

Now the next problem is generating 4 point configurations. Stillinger mentions for W_2 that the Cayley-Menger determinant should be 0 if no triangle can be formed. This is equivalent to saying that it should be 0 if it is not real and positive. We should also assume that if a triangle can be formed in any dimension $D > 1$, we can also form a tetrahedron in $D > 2$ and an n-dimensional simplex in $n > (D-1)$ dimensions.

There is a good argument for this that follows directly from Stillinger's formalism. Remembering that the integral $\int_0^a x^n dx$ is only defined for $n >= -1$ we can analyse W_n. W_n needs to be integrable to be able to use it as a distribution of lengths. So we should check for which values of D W_n is integrable . For W_1 we have $W_1 = r^{(D-1)} * f(D)$, with f being some irrelevant function. This is only integrable for $D > 0$. For the trimer case with bond-lengths 1 we can write $W_2 = r(4r^2 - b^4)^{(\frac{D-3}{2})}f(D)$. This is only integrable on $0 \leq r$ for $D > 1$. Similarly for W_3 there is a multiplication with a Cayley-Menger determinant raised to the power $D-4$. This determinant takes the form of a polynomial of at most order 4 in any of its variables. Using the same argument used for W_2 we come to the conclusion that W_3 can only be integrable for $D > 2$. Now our assumption is that any W_n is only integrable for $D > (n-1)$.

Working from this assumption we see that all the higher order weights W_n have relatively simple forms in dimensions $D \leq 2$, namely the values for which the Cayley-Menger determinant associated with the simplex in n dimensions is exactly 0. If we then work only between 1 and 2 dimensions we can see that the actual equation for configurations with more than 3 points is irrelevant, they should only be chosen such that the volume of a simplex spanned by those points is 0.

7.3 Application

We will apply Stillinger's approach to a single polymer chain in vacuum. To simplify matters we will only consider two models of such a chain, the ideal chain with fixed bond-lengths and an excluded volume chain where the fixed bond-length is equal to the monomer diameter. The reason for this is mostly computational efficiency later on.

Now let us first consider a trimer. We can place the first monomer x_0 anywhere we want in our fractional dimensional space. The second one is then added at a random point on the spherical shell with radius 1 having x_0 as centre point. The last monomer is now randomly put on the spherical shell with radius 1 having x_1 as centre point taking care that for multiple configurations the distribution w_2 is correctly sampled. An easy way to do this is simply construct a distance matrix r_{ij}, set the diagonal to 0, the first off diagonal to 1 and the second off diagonal to a random number sampled from the distribution W_2. Working only with distances also avoids explicitly constructing a position, for which we have no clear definition in a fractional dimensional space.

Generating a random number from W_2 can be done by integrating W_2 between 0 and 2, inverting it and applying that function to random numbers from a uniform distribution between 0 and 1. This procedure can be extended to chains of arbitrary lengths. There is however some subtlety involved that is not directly clear from Stillinger's formalism. All the weights W_n he mentions contain a multiplication with a Cayley-Menger determinant. The problem is that this restricts the domain for which W_n is valid in dimension $d \leq n-1$ because the volume of a d dimensional object in $d-1$ dimensions or less should be 0. If we interpret this as constraints on the system, we can generate longer chains.

For the ideal chain in $1 < d \leq 2$ dimensions we use the following two constraints: C_2 for a configuration with 4 points and 5 known distances, the 6th and last one then has 2 possible values, and C_1 for a configuration with 5 points and 9 known distances, the 10th and last one then has one single unique value. When applying C_2, we pick one of the two solutions at random with equal probability.

To summarize, we generate a single chain configuration by calculating every element in a distance matrix r_{ij} in this way:

$$r_{ij} = \begin{cases} 0 & |i-j| = 0 \\ 1 & |i-j| = 1 \\ \hat{W}_2 & |i-j| = 2 \\ C_2(r_{i,i+1}, r_{i,i+2}, r_{i+1,i+2}, r_{i+1,i+2}, r_{i+1,i+3}, r_{i+2,i+3}) & |i-j| = 3 \\ C_1() & |i-j| > 3 \end{cases}$$

where $\hat{W}_2$ is a random number generated from the distribution W_2, C_2 is the constraint using all the known distances between x_i, x_{i+1}, x_{i+2} and x_j (assuming $i < j$) and C_1 is the constraint using all the known distances between x_i,x_{j-3},x_{j-2}, x_{j-1} and x_j (assuming $i < j$).

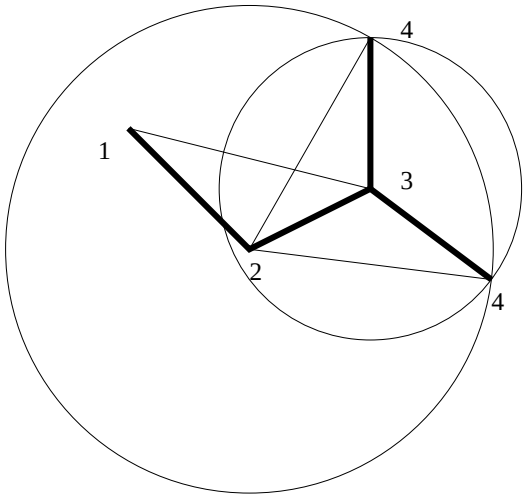

Figure 7.3: C2 constraint. For a 4 points configuration the last distance can be one of two values (or one degenerate). The thick line represents the configuration, 4 can be placed on either of the two given locations. The thin straight lines are distances we already fixed. The large circle represents the distance we have generated between 2 and 4, the small circle represents the (unit) distance between 3 and 4.

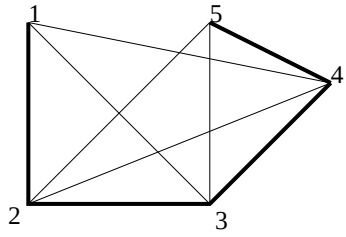

Figure 7.4: C1 constraint. For a 5 points configuration the last distance can be only one unique value. The thick line represents the configuration, the thin lines are the distances we have already generated.

There are at least two approaches for applying the constraints C_1 and C_2. One is finding the roots for the 4th and 3rd order polynomials resulting from the Cayley Menger determinants for a 4 and 3 dimensional volume respectively. The roots for C_2 always consist of 2 positive roots (possibly degenerate)

and the solution to C_1 always has just one positive degenerate solution (see figure 7.4). Another approach is constructing actual vector coordinates in 2 dimensions for all the points and using circle intersections for finding the solutions to C_2 and C_1 (see figure 7.3).

7.4 Experimental details

As in all the chapters in this work, the actual experiments are computer simulations. For our setup to generate non-integer dimensional polymer chains there is no reference code or even similar code available, so we will attempt to describe through some code snippets the full procedure.

We start with some helper functions and definitions.

```
double d; // Dimensionality of the system

double sigma()
{
  return 2 * pow(PI, d/2)/exp(gamma(d/2));
}

double vol(double a, double b, double c)
{
  return 0.25 * sqrt(2 * (a*a*b*b + a*a*c*c+b*b*c*c)
    - (a*a*a*a) - (b*b*b*b) - (c*c*c*c));
}

double w2(double r01, double r02, double r12)
{
  return pow(2.0, d-3) * sigma(d-1) * r01 * r02
    * pow(r12,2-d) * pow(vol(r01,r02,r12),d-3);
}

double randf()
{
  return (double)rand() / (double)RAND_MAX;
}

double c2(double r02, double r13)
{
  // Randomly pick on of the two options
  if(rand() \%2 == 0)
  {
```

```
    return (sqrt(2 + r02 * r02 * r13 * r13 - r02 *
              sqrt(4 - r02 * r02) * r13 *
              sqrt(4 - r13 * r13)) /
            (sqrt(2.0)));
  }
  else
  {
    return sqrt(1 + (0.5 * r02 * r02 * r13 * r13)
      + 0.5 * r02 * r13 * sqrt(4 - r02 * r02)
      * sqrt(4-r13*r13));
  }
}
double rw2()
{
  // Look-up on a pre-calculated table.
  // Table is calculated by integrating w2 with fixed
  // values r01=1 and r12=1, normalizing it and then
  // doing an inverse look-up from a uniform random number.
}
```

This calculates the unique missing distance if we have a 5x5 symmetric distance matrix with only (0,4) and (4,0) missing. We tried to do this directly by explicitly putting in the 0 value solution to a Cayley-Menger determinant with those two values as unknowns. However the expression we retrieved for this from Mathematica was over 130 lines of obfuscated c code that proved to be unstable due to round-off errors.

```
double c1(double y01, double y02, double y03,
          double y12, double y13, double y14,
          double y23, double y24, double y34)
{
  double r04[4],dr[4], min;
  int i,mini;
  double a,b,dz,e,f;
  // This one tries to find 0-4, do two sub problems and result
  // common solution:
  // first 0 2 3 4
  a=y02*y02;
  b=y03*y03;
  dz=y23*y23;
  e=y24*y24;
  f=y34*y34;
  r04[0]=(b*dz-dz*dz+b*e+dz*e-b*f+dz*f+a*(dz-e+f)
```

```
    -Sqrt(a*a+(b-dz)*(b-dz)-2*a*(b+dz))
    *Sqrt(dz*dz+(e-f)*(e-f)-2*dz*(e+f)))/2*dz;
  r04[1]=(b*dz-dz*dz+b*e+dz*e-b*f+dz*f+a*(dz-e+f)
    +Sqrt(a*a+(b-dz)*(b-dz)-2*a*(b+dz))
    *Sqrt(dz*dz+(e-f)*(e-f)-2*dz*(e+f)))/2*dz;
  //then 0 1 2 4
  a=y01*y01;
  b=y02*y02;
  dz=y12*y12;
  e=y14*y14;
  f=y24*y24;
  r04[2]=(b*dz-dz*dz+b*e+dz*e-b*f+dz*f+a*(dz-e+f)
    -Sqrt(a*a+(b-dz)*(b-dz)-2*a*(b+dz))
    *Sqrt(dz*dz+(e-f)*(e-f)-2*dz*(e+f)))/2*dz;
  r04[3]=(b*dz-dz*dz+b*e+dz*e-b*f+dz*f+a*(dz-e+f)
    +Sqrt(a*a+(b-dz)*(b-dz)-2*a*(b+dz))
    *Sqrt(dz*dz+(e-f)*(e-f)-2*dz*(e+f)))/2*dz;

  dr[0]=fabs(r04[0]-r04[2]);
  dr[1]=fabs(r04[0]-r04[3]);
  dr[2]=fabs(r04[1]-r04[2]);
  dr[3]=fabs(r04[1]-r04[3]);
  mini=-1;
  min=100000;
  // Directly comparing doubles will always fail
  // due to round-off. We just take the pair with
  // the lowest distance.
  for(i=0;i<4;i++)
    {
      if(dr[i]<min)
        {
          min=dr[i];
          mini=i;
        }
    }
  return Sqrt(r04[2+(mini%2)]);
}
```

And the actual program that we abbreviated by not doing self avoiding walks and generating just a single configuration, such functionality will be left as an exercise to the reader. When doing self avoiding walks, keep in mind that a very easy optimization is by only generating values from rw2() that fall

in the $[d..2]$ range when using spheres with diameter d.

```
int main(int argc, char **argv)
{
  // dimensionality, not checked for <1..2] range.
  d = atof(argv[1]);

  // r is a square matrix of size np, where
  // np[i][j] is the distance between particle i and j.

  // Distances to self
  for(i=0;i<np;i++)
  {
    r[i][i]=0;
  }

  // Rigid chain
  for(i=0;i<(np-1);i++)
  {
    r[i][i+1]=1;
  }

  // Stillinger's expression for 3 point distance
  // with 2 knowns (1,1)
  for(i=0;i<(np-2);i++)
  {
    r[i][i+2]=rw2();
  }

  // Get one of the (at most ) two allowed
  // distances in 1 < d <= 2
  for(i=0;i<(np-3);i++)
  {
    r[i][i+3]=c2(r[i][i+2],r[i+1][i+3]);
  }

  // All remaining points in the distance matrix are
  // now uniquely determined. Calculate each of them,
  // diagonal by diagonal, inward out.
  for(i=4;i<np;i++)
  {
    for(j=0;(j+i)<np;j++)
```

```
    {
      f=j;
      g=i+j;
      r[j][i+j] = c1(
        r[f  ][g-3], r[f  ][g-2], r[f  ][g-1],
        r[g-3][g-2], r[g-3][g-1], r[g-3][g  ],
        r[g-2][g-1], r[g-2][g  ], r[g-1][g  ]);
     }
   }

   // The end to end distance is now available
   // in r[0][np-1];
}
```

7.5 Results

Using the method described in the previous section we can generate two types of chains, ideal chains and excluded volume chains. Every generated configuration can be accepted as an ideal chain, while only chains where each distance is larger than 1 is a valid excluded volume chain.

In all integer dimensions we know that the following relation for the end to end distance R_e of an ideal chain holds:

$$\langle R_e^2 \rangle = N$$

We tested this for different dimensionalities between 1 and 2 and found that this relation holds very accurately.

It is interesting to note again that for the same problem on fractals, this relation does not hold (this is known as "anomalous diffusion"). Some end to end vector distributions can be found in figure 7.2.

If we consider excluded volume chains, the probability of generating a valid configuration is extremely small even for relatively short chains. One easy way to generate much longer chains is by generating random numbers from w_2 but only ones that are between 1 and 2. This way we can generate quite long chains for dimensionalities close to 1. Doing this we discover that the scaling of the end to end vector is different from the predictions of both Dreischor and Lowe and re-normalization group theory. For all dimensionalities $2 > d > 1$ the scaling seems to go to the two dimensional result $\langle R_e^2 \rangle \sim N^{3/4}$, only the "length-scale" where this happens gets longer for lower dimensionalities (Fig 7.5). Only in the limit of $d \to 1$ do we recover the 1 dimensional scaling $\langle R_e^2 \rangle \sim N$.

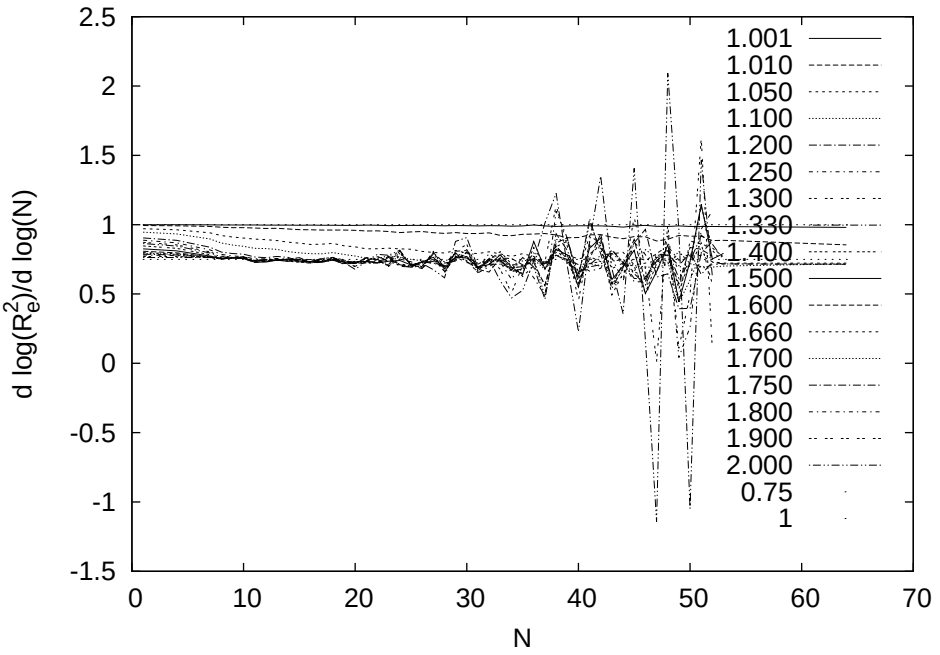

Figure 7.5: Scaling exponent of d-dimensional chains.

7.6 Conclusion

It seems that within Stillinger's framework it is possible to model polymer chains in non-integer dimensions. Interestingly enough the results do not agree with known predictions for scaling behaviour in these cases. One slightly disturbing observation is that the behaviour of both an ideal and an excluded volume chain approach that of a 2-dimensional chain for all dimensionalities between 1 and 2. Stillinger already notes something similar when he says "It leads to the striking conclusion that the number of mutually perpendicular lines can exceed the dimension of space, specifically when $2 > D > 1$". This can lead to two different conclusions. The first might be that the method is flawed. We assumed that, as Stillinger implies, a triangle "fits" in any dimensionality larger than 1, a tetrahedron only fits in a dimensionality larger than 2 etc. This assumption basically gives the method an integer number of independent degrees of freedom leading to integer dimensional behaviour.

The other conclusion (which we think is the relevant one) is that non-integer dimensionality is not "naturally" defined. We believe though, that this is the right approach for dealing with non-integer dimensional problems. As Stillinger shows, the formalism allows one to reproduce many results that

have the dimensionality as a continuous variable in the solution. Under the presumption that the formalism itself is correct, we now claim that at least for polymer physics, all physics in non-integer dimensions are equal to those in the next-highest integer dimension. While this might be a slightly disappointing conclusion, the computational effort required to get a certain degree of accuracy is less for certain non-integer dimensional cases. It might also open the door to new and more efficient Monte-Carlo schemes, since there are multiple ways of choosing a valid 3 particle pair configuration that we have shown are equivalent when you are only concerned with the scaling behaviour.

Chapter 8

Summary

This research has focused on relatively simple models that all relate to polymer behaviour. Polymers occur in many shapes and sizes, from extremely large and complex DNA molecules to relatively simple ones found in household items encountered daily. One of the difficulties in studying polymer behaviour is describing their behaviour in a rigorous set of equations. Due to the large variations in topology and the complexity of the object itself there is much to be gained from studying the behaviour of polymers through computer simulations. In a way this is still a way of analytical analysis, but on a scale that is untractable on a blackboard or piece of paper. Computer simulations provide a large toolbox that can be used to investigate polymer behaviour in very controlled circumstances. This toolbox provides means to control and measure each and every aspect, from the mass of an individual atom to measurements of the long time-scale diffusive behaviour. In this work we have taken some of the existing tools from this toolbox and attempted to improve them, added some additional tools and combined some existing tools in way not described in literature before.

Chapter two can be split into three parts. The first part takes an existing tool, the Lowe-Andersen thermostat, and combines this with another complex tool: Molecular Dynamics. The use of a thermostat in these computer simulations is to explicitly control the temperature, very analogous to the heating system in a house. Existing thermostats have some known drawbacks, they can be very complex or slow down the system dynamics. The Lowe-Andersen thermostat has the advantage that it does not slow down the dynamics very much, while still being a very simple tool to use. Molecular Dynamics provide a straightforward way of simulating sets of molecules interacting on very small length and time scales. At the time of writing nobody had applied the Lowe-Andersen thermostat to a Molecular Dynamics system. We did so and analysed the slowing down of dynamics by analysing the effect on the dif-

fusion. Think of this as the speed with which an ink drop discolours a glass of clean water, if the diffusion is too low, the ink spreading speed will be too slow. We found that for the Lowe-Andersen thermostat we could still very effectively control the temperature, without affecting the diffusion significantly. This makes it a very good drop-in replacement for the oft-used Andersen scheme, that is almost as simple to use, but slows down dynamics very significantly.

The second part of Chapter 2 focuses on an easily overlooked possible problem while using the Lowe-Andersen thermostat. As this tool operates on pairs of particles we should know what the effect is of applying the tool two two different pairs in which one particle is shared. It turns out that if the pair operations are performed frequently on certain pairs in the same strict order, there is a measurable artificial dependence on the order of operations. We show that this is easily resolved by randomizing the operations and explain if and how this is usable in very large computer simulations that are done by multiple computers in parallel. Lastly, Chapter 2 suggests two small changes to the Lowe-Andersen thermostat, both simplifying the scheme a bit. The net effect of the simplified schemes is also that it allows easier simulation of "water-like" conditions.

The third chapter is again an application of two tools in a novel way. This time we take a method for simulating multi-component fluids on time and length-scales a bit larger than the previous chapter and combine it with a known technique for keeping the system pressure constant. A key benefit of the simulation method (Dissipative Particle Dynamics, DPD) is that it has a very simple, yet non-trivial equation of state. Because of the simple equation of state, some predictions can be made about the mixing behaviour of a simple two-component mixture at a given temperature and pressure. As DPD has no inherent way of controlling the pressure, we took an existing method that had only been used in Molecular Dynamics simulations and successfully combined both. The net result is a good agreement between the theoretical prediction and the observed system behaviour.

The next chapter is the result of the initial research done on polymer networks. Polymers are made up of small building blocks linked together, when the polymer chains are also linked together the entire structure can become a network. There exist many tools to do computer simulations on polymer networks, but one thing that showed up in our earlier simulations is that we need to be able to distinguish between something that is a network and something that is not. Many existing methods avoid this problem by keeping away from the "cross-over" state, so there is no need to explicitly check it. Also we could not find anybody who had done a rigorous analysis of this problem, especially for the type of boundary conditions we use. It turns out that the solution is not very straightforward and it is easy to pick

a wrong method that either overestimates or underestimates if a system is a network. The best analogy that explains the problem is one using wallpaper that depicts spaghetti. Assume the wallpaper pattern is perfectly aligned, so you get identical copies of the same "unit pattern" from left to right. Now you might wonder what happens when you pull one of the spaghetti strands, do you get just a short piece, or is there a strand that you can keep pulling on indefinitely because it connects to a copy of itself on one of the next sections of wallpaper. Being able to pull indefinitely is the property that makes the spaghetti a network (strictly speaking only in the left-right direction on our wallpaper). This work describes a method of analysing a single non-repeating piece of the wallpaper and give a claim about whether pulling indefinitely is possible.

The following chapter is the result of collaborative research performed with M. Dreischor. By applying a rigorous analysis of a more complex equation of state we found that the theory predicts that a long polymer chain solved in a solution consisting of its own monomers would be collapsed, forming a tiny droplet. To verify this prediction we performed a long time-scale simulation using Molecular Dynamics to investigate the behaviour of said polymer. It turned out that indeed the chain collapses, and interestingly enough it does so by encapsulating a significant amount of solvent into the collapsed state, making it truly a tiny droplet that would precipitate if it were a real system.

Next is a short chapter that tries to solve a relatively complex problem of creating a flow pattern with some specific properties with a simple method. The idea again has a simple analogy, say you're baking a cake and you have two sets of mixers (making for 4 rotors), we have a way of setting up a flow in the cake dough that follows some desirable properties.

The final chapter again involves polymers, but this time we pursued a new approach to attempt to get another clue into finding the scaling exponent for an excluded volume chain in 3 dimensions. There exist some predictions that give this exponent as a continuous function of the dimensionality, but as can be imagined it is hard to verify such a prediction in anything other than 1, 2 or 3 dimensions. We did attempt to measure the scaling exponent for dimensionalities in between 1 and 2 dimensions by rigorously applying a set of tools that had been though up by Stillinger in 1977. He thought of a way of describing non-integer dimensional spaces that appear to be sufficient to describe simple polymer chains as well. By only treating distances between sets of points we figured out how to generate random polymer chain statistics (both ideal and self-avoiding) in dimensions between 1 and 2. By measuring the average end to end distance for many different configuration we validated that for ideal chains we do indeed get apparently valid configurations, but the self-avoiding chains are a bit more complex. What we see is that for any dimensionality slightly above 1, the end to end scaling for long polymer

chains follows the behaviour of two-dimensional chains. We conclude that our approach does not give additional clues for the value of the 3 dimensional scaling exponent as apparently in the formalism we used the longer length-scale behaviour of objects is always the same as that in the next integer dimensionality.

Chapter 9

Samenvatting

Mijn onderzoek heeft zich geconcentreerd op vrij eenvoudige modellen die polymeergedrag beschrijven op kleine tijd- en lengteschalen. Polymeren hebben veel verschijningsvormen: van complexe DNA-moleculen tot alledaagse huishoudelijke artikelen. Het beschrijven van polymeergedrag door middel van vergelijkingen is op zijn zachtst gezegd lastig. Door de grote variatie in verschijningsvormen en complexiteit valt er veel kennis te halen uit het bestuderen van polymeergedrag door middel van computersimulaties. Je zou computersimulaties nog steeds kunnen beschouwen als een theoretische analyse, maar dan op een schaal die onhaalbaar is op een schoolbord met een krijtje. Computersimulaties bieden een grote doos gereedschap die je kunt gebruiken om polymeren te bestuderen in perfect controleerbare omstandigheden. Met deze doos gereedschap kun je alles in een simulatie controleren en meten, van de massa van een individueel deeltje tot de diffusie op lange tijdschalen. In dit werk hebben we een aantal gereedschappen genomen uit de standaard set gereedschap en gepoogd ze te verbeteren. Ook hebben we een aantal nieuwe gereedschappen gemaakt en een aantal op nieuwe manieren gecombineerd.

Hoofdstuk twee bestaat uit twee delen. Het eerste deel gebruikt een bestaand stuk gereedschap, de Lowe-Andersen thermostaat, en combineert die met een ander complex stuk gereedschap, "Molecular Dynamics". Een thermostaat in computersimulaties wordt gebruikt om de temperatuur in een simulatie in te stellen, analoog aan een huisthermostaat. Bestaande thermostaten hebben allerhande problemen; ze zijn heel ingewikkeld of vertragen het gesimuleerde systeem. De Lowe-Andersen thermostaat heeft het voordeel dat hij het systeem niet heel erg vertraagt en ook relatief simpel in het gebruik is. Molecular Dynamics biedt daarnaast een mechanisme om grote hoeveelheden moleculen op kleine lengte- en tijdschalen te simuleren. Tijdens het onderzoek bleek dat nog niemand deze twee technieken met elkaar had gecombineerd. Dit deden we dus en we bekeken de vertraging van het

systeem door de diffusie nauwkeurig te meten. De analogie hier is het laten vallen van een druppel inkt in een groot glas water. Als de diffusie laag is, is de geobserveerde verspreiding van de inkt langzaam. We vonden dat als we de Lowe-Andersen thermostaat gebruikten we nog steeds nauwkeurig de systeemtemperatuur konden kiezen, zonder de diffusie significant te verlagen. Bij elkaar maakt dit de Lowe-Andersen thermostaat een goede vervanging voor de veelgebruikte Andersen thermostaat. De Andersen thermostaat is ook simpel in gebruik, maar heeft als nadeel dat hij het systeem significant verlaagt.

Het tweede deel van hoofdstuk 2 bekijkt een gemakkelijk over het hoofd te zien probleem met de Lowe-Andersen thermostaat. Deze thermostaat werkt op paren van deeltjes, en we willen graag weten wat er gebeurt als je de thermostaat tegelijk op twee paren toepast die een deeltje gemeenschappelijk hebben. Het blijkt dat als de volgorde van het toepassen van de thermostaat altijd vastligt er een meetbare kunstmatige afhankelijkheid hiervan ontstaat in meetbare eigenschappen van de deeltjes. Door de paren in een willekeurige volgorde te behandelen lossen we dit probleem relatief simpel op. Ook stellen we voor hoe dit probleem opgelost kan worden in hele grote simulaties die op meerdere computers tegelijk plaatsvinden. Daarnaast stellen we nog twee aanpassingen voor voor de Lowe-Andersen thermostaat waardoor hij net iets simpeler wordt. Deze aanpassingen zorgen ervoor dat de thermostaat ook inzetbaar is om simulaties te doen voor vloeistoffen die dynamisch gezien erg lijken op water.

Het derde hoofdstuk is wederom een toepassing van twee gereedschappen op een nieuwe manier. Deze keer bekijken we een bestaande methode om meer-componenten vloeistoffen te simuleren op langere lengte- en tijdschalen en combineren we deze met een bestaande techniek om de druk in een systeem constant te houden. Een groot voordeel van deze techniek (Dissipative Particle Dynamics, DPD) is dat het een heel simpele, niet-triviale toestandsvergelijking heeft. Door deze simpele toestandsvergelijking kunnen we voorspellingen maken over hoe een twee-componenten vloeistof zich gedraagt bij een gegeven druk en temperatuur. De DPD methode heeft geen ingebouwde manier om de druk in te stellen, dus gebruiken we een bestaande methode afkomstig uit Molecular Dynamics simulaties en combineren we beiden met succes. Het uiteindelijke resultaat is dat we een goede overeenstemming vinden tussen de theoretische voorspellingen en het waargenomen gedrag van het twee-componenten systeem.

Het volgende hoofdstuk is het resultaat van de eerste stappen van ons onderzoek naar polymeer netwerken. Polymeren zijn normaal opgebouwd uit kleine eenheden die aan elkaar vast zitten als een ketting. Als deze kettingen onderling ook nog eens aan elkaar zitten kan de gehele structuur een netwerk vormen. Er bestaan al meerdere technieken om polymeernetwerken

te simuleren, maar we constateerden al vrij snel uit dat het heel lastig is om onderscheid te maken tussen systemen die wel en geen netwerk zijn. Veel onderzoek loopt niet tegen dit probleem aan omdat de polymeren al zo verstrengeld zijn dat het sowieso netwerken vormen, maar als de verstrengeling "op het randje" is heb je iets nodig dat kan bepalen of je systeem wel of geen netwerk is. Bij het doorzoeken van de literatuur kwamen we geen onderzoek tegen dat dit probleem rigoreus had opgelost, zeker niet met ons type randvoorwaarden. Het blijkt dat de oplossing niet geheel triviaal is. In het grensgebied is het vrij makkelijk om de vraag of een systeem wel of niet een netwerk is verkeerd te beantwoorden. Om het probleem te verduidelijken, stel je een behang voor waarom spaghetti wordt afgebeeld. Het behang is netjes aangebracht, dus op de naden loopt het patroon ononderbroken door, en er is duidelijk een herhaling zichtbaar van links naar rechts over de banen. Stel je nu voor dat je aan een van de slierten spaghetti begint te trekken. Onze stelling is nu dat het afgebeelde systeem een netwerk is als er een sliert is waaraan je kunt blijven trekken als de muur oneindig ver doorloopt. De analogie is dat als je kan blijven trekken, de niet herhalende eenheid op een stuk behang een netwerk voorstelt (wel alleen in de horizontale richting). Ons werk geeft nu een wiskundige beschrijving over hoe je een zo'n niet herhalende eenheid kan analyseren en kan concluderen of deze eenheid wel of niet een netwerk vormt.

Het volgende hoofdstuk is het resultaat van een samenwerking met Menno Dreisschor. Door stug doorrekenen met een complexe toestandsvergelijking vonden we dat de theorie voorspelt dat een lange polymeerketen opgelost in zijn eigen bouwblokjes (de monomeren) in sommige omstandigheden in elkaar krimpt en een klein druppeltje gaat vormen. Om deze voorspelling te toetsen, deden we een langdurige simulatie gebruikmakend van Molecular Dynamics. Het blijkt dat dit inderdaad gebeurt, en interessant genoeg krimpt de keten niet alleen ineen, maar zuigt hij daarbij ook een flinke hoeveelheid monomeren op en vormt daarmee een klein druppeltje dat in een echt systeem neer zou slaan al ware het een regendruppel.

Het een-na-laatste hoofdstuk is een kort hoofdstuk waarin we een methode voorstellen om in deeltjessimulaties een kenmerkend type stroming te veroorzaken. De methode is het best uit te leggen met wederom een analogie. Stel je voor dat je een taart bakt en een grote kom met deeg hebt. Onze methode legt uit hoe je met een tweetal mixers in een groot deel van het deeg de juiste stroming kan veroorzaken.

Het laatste hoofdstuk gaat het wederom om polymeren, maar deze keer proberen we informatie te vinden over de schalingsexponent voor een polymeerketen in drie dimensies die zichzelf niet mag overlappen. Er zijn verschillende voorspellingen voor deze schalingsexponent die hem voorspellen als continue functie van de dimensionaliteit. Helaas is het lastig om deze voor-

spelling te toetsen buiten een, twee en drie dimensies. We hebben gepoogd om de schalingsexponent ook te meten in niet heeltallige dimensies. Hierbij hebben we gebruik gemaakt van het werk van Stillinger gedaan in 1977. Hij heeft een methode bedacht om niet-heeltallig-dimensionale ruimtes te beschrijven, en zijn methode zou voldoende informatie moeten bevatten om ook polymeerketens te beschrijven. Zijn methode werkt door het alleen beschrijven van afstanden tussen punten. Absolute locaties worden expliciet niet gebruikt. Binnen deze methode leggen we uit hoe je polymeerketens kan maken die zichzelf wel of niet mogen overlappen. Als we dit dan daadwerkelijk doen zien we dat voor elke dimensionaliteit net hoger dan 1, het schalingsgedrag voor een 2-dimensionale keten volgt. We concluderen dat onze aanpak geen nieuwe inzichten geeft over het vastleggen van de 3-dimensionale schalingexponent van een niet overlappende keten. Dit omdat met onze aanpak elke keten die zich bevindt in een niet heeltallig-dimensionele ruimte zich op lange lengteschalen gedraagt als een keten in een eerstvolgende heeltallige ruimte.

Chapter 10

Appendix

10.1 Efficient clustering in unstructured networks

Here we list a piece of C++code that can efficiently group together a set of unstructured points. We only know the point coordinates and we have a set of point index pairs representing links. The code refers in many places to the color of a point. Initialy this is set to a unique index, each greater than the total number of particles, afterwards all points connected together share the same color. The code has been slightly rewritten to increase readability, for example the std::map⟨T,U⟩::find() == std::map⟨T,U⟩::end() construction might be unclear to people not familiar with C++so this has been changed to psuedocode "contains()".

The algorithm works by effectively keeping track of two arrays of colors. The first is on the points themselves, the second is the colormap. The values in the first array are pointers into the second array. Each point in the second array points to the left or nowhere. If it points to the left, that indirection should be followed recursively until it points nowhere. If a point in the second array points nowhere, that index in the array is the final cluster index.

The first stage constructs the two arrays, the second stage performs the final cluster index lookup for each point.

```
void Detector::ColorLoop(vector<CPoint> Points, vector<CLink> Links)
{
  map<int, int> ColorMap;
  map<int, int> ColorMapMap;
  int ClusterCount = 1;

  // Do a single pass over all links, construct a map that contains
  // redirections to eventually the correct cluster index.
  for (vector<CLink>::iterator iLink = Links.begin();
       iLink != Links.end(); ++iLink)
  {
```

```
  int Color1 = m_Points[iLink->P1()].GetColor();
  int Color2 = m_Points[iLink->P2()].GetColor();
  int MaxColor = Color1 > Color2 ? Color1 : Color2;

  if (Color1 < Color2)
    m_Points[iLink->P1()].SetColor(Color2);
  else
    m_Points[iLink->P2()].SetColor(Color1);

  // None found
  if (!ColorMap.contains(Color1) && !ColorMap.contains(Color2))
  {
    ColorMap[MaxColor] = ClusterCount;
    ClusterCount++;
  }
  // Only Color2 found
  if (!ColorMap.contains(Color1) && ColorMap.contains(Color2))
  {
    if (Color1 > Color2)
      ColorMap[Color1] = ColorMap[Color2];
  }
  // Only Color1 found
  if (ColorMap.contains(Color1) && !ColorMap.contains(Color2))
  {
    if (Color2 > Color1)
      ColorMap[Color2] = ColorMap[Color1];
  }
  // Both found, conflict, resolve by going all the way down
  // and merging to lowest value
  if (ColorMap.contains(Color1) && ColorMap.contains(Color2))
  {
    int Lowest1 = ColorMap[Color1];

    while (ColorMap.contains(Lowest1) && (Lowest1 != ColorMap[Lowest1]))
      Lowest1 = ColorMap[Lowest1];

    int Lowest2 = ColorMap[Color2];

    while (ColorMap.contains(Lowest2) && (Lowest2 != ColorMap[Lowest2]))
      Lowest2 = ColorMap[Lowest2];

    if (Lowest1 < Lowest2)
      ColorMap[Lowest2] = Lowest1;
    else
      ColorMap[Lowest1] = Lowest2;
  }
}

// For each point hop through the map to get to the actual index
// for its cluster
for (vector<CPnt>::iterator iPoint = m_Points.begin();
```

```
       iPoint != m_Points.end(); ++iPoint)
  {
    if (ColorMap.find(iPoint->GetColor()) != ColorMap.end())
    {
      int NewColor = ColorMap[iPoint->GetColor()];

      while (ColorMap.contains(NewColor) && NewColor != ColorMap[NewColor])
        NewColor = ColorMap[NewColor];

      iPoint->SetColor(NewColor);
    }
  }
}
```

10.2 Structure factor calculation

Calculating the static structure factor can be a time consuming task if not done properly. The structure factor is given by

$$S(\vec{k}) = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{N} e^{(i\vec{k}\cdot\vec{r_{ij}})}.$$

A careful observer will notice the double summation, something that you typically try to avoid in computer programs since it is an indication of at least $\mathcal{O}(n^2)$ complexity.

Fortunately there is a neat way of rewriting the equation so that it becomes linear in the amount of particles. We start by writing $S(\vec{k})$ as

$$\begin{aligned}
S(\vec{k}) &= \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1} N e^{i\vec{k}\cdot(\vec{r_i}-\vec{r_j})} && (10.1)\\
&= \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{N} e^{(i\vec{k}\cdot\vec{r_i})} e^{(i\vec{k}\cdot\vec{r_j})} && (10.2)\\
&= \frac{1}{N}\sum_{i=1}^{N} e^{(i\vec{k}\cdot\vec{r_i})}\sum_{j=1}^{N} e^{(-i\vec{k}\cdot\vec{r_j})} && (10.3)\\
&= \frac{1}{N}\sum_{i=1}^{N} \cos(\vec{k}\cdot\vec{r_i}) + i\sin(\vec{k}\cdot\vec{r_i})\sum_{j=1}^{N} \cos(-\vec{k}\cdot\vec{r_j}) + i\sin(-\vec{k}\cdot\vec{r_j}) && (10.4)\\
&= \frac{1}{N}\sum_{i=1}^{N} \cos(\vec{k}\cdot\vec{r_i}) + i\sin(\vec{k}\cdot\vec{r_i})\sum_{j=1}^{N} \cos(\vec{k}\cdot\vec{r_j}) - i\sin(\vec{k}\cdot\vec{r_j}) && (10.5)
\end{aligned}$$

and now since all the $\sin(..)\cos(..)$ terms vanish because they directly cancel each other out and all the $\sin(\vec{k}\cdot\vec{r_i})\cos(\vec{k}\cdot\vec{r_j})$ with $i \neq j$ are on average 0 we get

$$S(\vec{k}) = \frac{1}{N}\sum_{i=1}^{N}\cos^2(\vec{k}\cdot\vec{r_i}) + \sin^2(\vec{k}\cdot\vec{r_i}) \tag{10.6}$$

which is a single summation.

Now the second thing to keep in mind when calculating structure factors is that the length associated with $\vec{k}$ has to fit an integer number of times in the system box. An abbreviated version of our actual implementation is shown here. This code is easily extended to handle multi species structure factors.

```
void StructureFactor(int Mode, SParameters Prmtrs, SParticle* pPrts)
{
  static double Sk[3][SIZE]; // SIZE = 100
  static int updates = 0;
  double Kv[3][SIZE];
  switch(Mode)
  {
    case MODE_INIT:
      for (int i = 0; i < SIZE; i++)
      {
        Sk[0][i] = Sk[1][i] = Sk[2][i] = 0;
        for (int dim = 0; dim < 3; dim++)
          Kv[dim][i] = (double)(i + 1) * (2 * PI / Prmts.boxsize[dim]);
      }
      break;
    case MODE_UPDATE:
      updates++;
      for (int dim = 0; dim < 3; dim++)
      {
        for (int ki = 0; ki < SIZE; ki++)
        {
          double a = 0, b = 0;
          for (int i = 0; i < Prmts.NP; i++)
          {
            a += cos (Kv[dim][ki] * pPrts[i].Pos[dim]);
            b += sin (Kv[dim][ki] * pPrts[i].Pos[dim]);
          }
          Sk[dim][ki] += a*a + b*b;
        }
```

```
      }
      break;
    case MODE_FINALIZE:
      for (int i = 0; i < SIZE; i++)
        for (int dim = 0; dim < 3; dim++)
          Sk[sim]i] /= (double)(updates * Prmtrs.NP);
      ...// Sk now contains correct structure factor
      break;
  }
}
```

Chapter 11

Dankwoord

Het is af, eindelijk! Nooit meer "hoe is het met je boekje", "wanneer ga je promoveren" en "moet je niet aan je boekje werken?" In een eerste versie van dit dankwoord schreef ik "Ik ben nog geen vader en ook nog niet getrouwd, dus ik zou het afkrijgen van dit werk willen verheffen tot mooiste moment van mijn leven.". Alle drie deze observaties zijn achterhaald door de feiten, maar het voltooien van dit werk is zeker een mooi moment. Maar laat me beginnen bij het begin. Allereerst dank voor Piet Iedema voor het mogelijk maken van mijn promotietraject en de begeleiding aan het begin. Ook natuurlijk dank aan mijn promotor Peter Bolhuis die het mij mogelijk maakte dit proces to voltooien. Also Christopher, many thanks to you. You taught me how to do science, write science and think science. Verder nog dank aan alle andere staf die me veel geleerd hebben over wetenschap, Evert-jan, Gooitzen, Gadi en iedereen die ik vergeten ben.

Ik denk te kunnen stellen dat de echte toegevoegde waarde van dit promotietraject het leren kennen van alle collega AiOs was. Het zijn er vrij veel, maar in alfabetische volgorde allereerst Abdon. Bedankt voor de al ruim 11 jaar dat we collega's zijn, denk je aan bier, denk je aan Abdon. Arjen, bedankt voor alle leuke muziek waar je me mee kennis hebt laten maken. Bas, je bent met stip op 1 de meest fanatieke levensgenieter die ik ken. David, volgens mij heb jij het doorbikkelen op de kaart gezet, bedankt voor het helpen afronden van dit werk! Dirk, bedankt voor het zetten van alle koffie. Edith, bedankt dat je ook in Diemen woonde. Elske, bedankt voor het herpopulariseren van de 70ies. Francesco, thanks for teaching me more about Italy. Jantien, bedankt voor alle gezelligheid op B6.23 en je toevoeging van geluidseffecten aan alle conversaties. Jarek, thanks for teaching me more about Poland. Jasper, bedankt voor het niet publiceren van niet nader te noemen barfotos. Jan-Willem, bedankt voor het introduceren van "denk". Jocelyne, bedankt voor alle vriendelijke gesprekken bij de koffie. Jochem, bedankt voor het eindeloze

scherphouden van iedereen. Marieke, bedankt voor de gezelligheid. Nicole, bedankt voor het wakker houden van iedereen. Nils, thanks for teaching me more about Germany. Menno, bedankt voor alle hulp. Ranieri, bedankt voor alle leuke tech-demos die je hebt laten zien. Rini, bedankt voor de wondere wereld van micellen. Sven, thanks for learning me more about Germany. Tim, bedanktjes! Vincent, bedankt voor het populariseren van De Gennes.

Naast de inhoudelijke steun bij het voltooien van dit werk is er ook morele support nodig. Hiervoor wil ik al mijn vrienden bedanken die me regelmatig uitgedaagd hebben om uit te leggen wat ik in godsnaam nou eigenlijk deed. Wederom in alfabetische volgorde. Bruce, bedankt dat je ook een science-fanboy bent. Karel, bedankt voor al je artistieke inbreng. Marc, bedankt voor het populariseren van megalomanie als een sociaal geaccepteerd verschijnsel. Paul, bedankt voor al je beestjeskennis en het proeflezen van dit werk. Rick, bedankt voor alle interessse en het regelmatige gamen. Rutger, bedankt voor alle keren gamen. Steven, bedankt voor meer dan op deze pagina past. Werner, bedankt voor alle vette bokken.

Mijn familie wil ik natuurlijk ook bedanken. Pa, ma, Jan-Willem, hoewel ik nooit helemaal heb uit kunnen leggen wat ik nou deed aan de UvA zijn jullie altijd heel aanmoedigend geweest. Bedankt voor alle steun.

Zoals in elk dankwoord eindig je met de belangrijkste mensen. Toen ik 11 jaar terug aan mijn onderzoek begon had ik niet durven dromen dat ik hier twee mensen mag bedanken. Marlies, bedankt voor al je begrip, liefde en aandacht. Bedankt voor het, na lange dagen en korte nachten, zonder mopperen 's avonds alleen op de bank zitten terwijl ik aan m'n boekje schreef. Zonder jou had ik dit nooit afgemaakt :-*. Iris, het is moeilijk om te te beschrijven hoe blij ik word van het zien van jouw lach. Terwijl ik dit schrijf zit je naast me en sabbelt op een knuffel. Ik hoop dat je dit ooit leest en zal begrijpen hoe inspirerend je bent.

Bibliography

[1] P.J. Flory. *Principles of polymer chemistry*. Cornell University Press, 1953.

[2] P. G. De Gennes. *Scaling concepts in polymer physics*. Cornell university press, 1979.

[3] H. Yamakawa. *Modern theory of polymer solutions*. Harper & Row (New York), 1971.

[4] B. H. Zimm. Dynamics of polymer molecules in dilute solution: viscoelasticity, flow birefringence and dielectric loss. *The Journal of Chemical Physics*, 24:269, 1956.

[5] D Frenkel and B Smit. *Understanding molecular simulation: from algorithms to applications*. Elsevier (formely published by Academic Press), 1996.

[6] J.E. Jones. On the determination of molecular fields. ii. from the equation of state of a gas. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, 106(738):463–477, 1924.

[7] P.J. Hoogerbrugge and J.M.V.A. Koelman. Simulating microscopic hydrodynamic phenomena with dissipative particle dynamics. *Europhysics Letters*, 19(3):155, 1992.

[8] P. Espanol and P. Warren. Statistical mechanics of dissipative particle dynamics. *Europhysics Letters*, 30(4):191, 1995.

[9] C.P. Lowe. An alternative approach to dissipative particle dynamics. *Europhysics Letters*, 47(2):145, 1999.

[10] H.C. Andersen. Molecular dynamics simulations at constant pressure and/or temperature. *The Journal of Chemical Physics*, 72(4):2384, 1980.

[11] S. Nosé. A unified formulation of the constant temperature molecular dynamics methods. *The Journal of Chemical Physics*, 81:511, 1984.

[12] H. J. C. Berendsen, J. P. M. Postma, W. F. van Gunsteren, A. DiNola, and J. R. Haak. Molecular-dynamics with coupling to an external bath. *The Journal of Chemical Physics*, 1984.

[13] P.H. Hünenberger. Thermostat algorithms for molecular dynamics simulations. In *Advanced Computer Simulation*, pages 105–149. Springer, 2005.

[14] S. Jakobtorweihen, M.G. Verbeek, C.P. Lowe, F.J. Keil, and B. Smit. Understanding the loading dependence of self-diffusion in carbon nanotubes. *Physical Review Letters*, 95(4):044501, 2005.

[15] G.S. Grest and K. Kremer. Molecular dynamics simulation for polymers in the presence of a heat bath. *Physical Review A*, 33(5):3628, 1986.

[16] W.G. Hoover. Canonical dynamics: Equilibrium phase-space distributions. *Physical Review A*, 31(3):1695, 1985.

[17] T. Soddemann, B. Dünweg, and K. Kremer. Dissipative particle dynamics: A useful thermostat for equilibrium and nonequilibrium molecular dynamics simulations. *Physical Review E*, 68(4):046702, 2003.

[18] S.D. Stoyanov and R.D. Groot. From molecular dynamics to hydrodynamics: A novel galilean invariant thermostat. *The Journal of Chemical Physics*, 122:114112, 2005.

[19] C.P. Lowe, A.F. Bakker, and M.W. Dreischor. Novel methods in soft matter simulation. *Lecture Notes in Physics*, 640:39, 2004.

[20] P. Nikunen, M. Karttunen, and I. Vattulainen. How would you integrate the equations of motion in dissipative particle dynamics simulations? *Computer Physics Communications*, 153(3):407–423, 2003.

[21] R.D. Groot and P.B. Warren. Dissipative particle dynamics: Bridging the gap between atomistic and mesoscopic simulation. *The Journal of Chemical Physics*, 107:4423, 1997.

[22] B. Hafskjold, C.C. Liew, and W. Shinoda. Can such long time steps really be used in dissipative particle dynamics simulations? *Molecular simulation*, 30(13-15):879–885, 2004.

[23] L.J. Chen, Z.Y. Lu, H.J. Qian, Z.S. Li, and C.C. Sun. The effects of lowe–andersen temperature controlling method on the polymer properties in

mesoscopic simulations. *The Journal of Chemical Physics*, 122:104907, 2005.

[24] T. Shardlow. Splitting for dissipative particle dynamics. *SIAM Journal on Scientific Computing*, 24(4):1267–1282, 2003.

[25] W.K. Den Otter and J.H.R. Clarke. A new algorithm for dissipative particle dynamics. *Europhysics Letters*, 53(4):426, 2001.

[26] I. Pagonabarraga, M.H.J. Hagen, and D. Frenkel. Self-consistent dissipative particle dynamics algorithm. *Europhysics Letters*, 42(4):377, 1998.

[27] E.A.J.F. Peters. Elimination of time step effects in dpd. *Europhysics Letters*, 66(3):311, 2004.

[28] M.P. Allen and D.J. Tildesley. *Computer simulation of liquids*. Oxford university press, 1989.

[29] B.J. Alder and T.E. Wainwright. Decay of the velocity autocorrelation function. *Physical Review A*, 1(1):18, 1970.

[30] J.K. Johnson, J.A. Zollweg, and K.E. Gubbins. The lennard-jones equation of state revisited. *Molecular Physics*, 78(3):591–618, 1993.

[31] Ronald L Panton. *Incompressible flow*. John Wiley & Sons, 2006.

[32] Ingo O Götze, Hiroshi Noguchi, and Gerhard Gompper. Relevance of angular momentum conservation in mesoscale hydrodynamics simulations. *Physical Review E*, 76(4):046705, 2007.

[33] V. Symeonidis, G.E. Karniadakis, and B. Caswell. Schmidt number effects in dissipative particle dynamics simulation of polymers. *The Journal of Chemical Physics*, 125, 2006.

[34] J.A. Backer, C.P. Lowe, H.C.J. Hoefsloot, and P.D. Iedema. Poiseuille flow to measure the viscosity of particle model fluids. *The Journal of Chemical Physics*, 122, 2005.

[35] Richard Durstenfeld. Algorithm 235: random permutation. *Communications of the ACM*, 7(7):420, 1964.

[36] S.Y. Trofimov, E.L.F. Nies, and M.A.J. Michels. Constant-pressure simulations with dissipative particle dynamics. *The Journal of Chemical Physics*, 123:144102, 2005.

[37] A.F. Jakobsen. Constant-pressure and constant-surface tension simulations in dissipative particle dynamics. *The Journal of Chemical Physics*, 122:124901, 2005.

[38] S.E. Feller, Y. Zhang, R.W. Pastor, and B.R. Brooks. Constant pressure molecular dynamics simulation: the langevin piston method. *The Journal of Chemical Physics*, 103:4613, 1995.

[39] H.H. Winter and F. Chambon. Analysis of linear viscoelasticity of a crosslinking polymer at the gel point. *Journal of Rheology*, 30:367–382, 1986.

[40] P.J. Flory. Theory of elasticity of polymer networks. the effect of local constraints on junctions. *The Journal of Chemical Physics*, 66(12):5720–5729, 1977.

[41] F. Gittes and F.C. MacKintosh. Dynamic shear modulus of a semiflexible polymer network. *Physical Review E*, 58(2):1241–1244, 1998.

[42] P.J. Reynolds, H.E. Stanley, and W. Klein. Large-cell monte carlo renormalization group for percolation. *Physical Review B*, 21(3):1223–1245, 1980.

[43] M.S. Watanabe. Percolation with a periodic boundary condition: The effect of system size for crystallization in molecular dynamics. *Physical Review E*, 1995.

[44] S.R. Broadbent and J.M. Hammersley. Percolation processes. I. Crystals and Mazes. In *Proceedings of the Cambridge Philosophical Society*, volume 53 of *Proceedings of the Cambridge Philosophical Society*, pages 629–641, July 1957.

[45] J. Hoshen and R. Kopelman. Percolation and cluster distribution. i. cluster multiple labeling technique and critical concentration algorithm. *Physical Review B*, 14(8):3438, 1976.

[46] J. Hoshen, M.W. Berry, and K.S. Minser. Percolation and cluster structure parameters: The enhanced hoshen-kopelman algoritm. *Physical Review E*, 56(2):1455–1460, 1997.

[47] A. Al-Futaisi and T.D. Patzek. Extension of hoshen-kopelman algorithm to non-lattice environments. *Physica A: Statistical Mechanics and Its Applications*, 321(3):665–678, 2003.

[48] H.J. Herrmann, D.C. Hong, and H.E. Stanley. Backbone and elastic backbone of percolation clusters obtained by the new method of'burning'. *Journal of Physics A: Mathematical and General*, 17(5):L261, 1984.

[49] F Babalievski. Cluster counting: the hoshen–kopelman algorithm versus spanning tree approaches. *International Journal of Modern Physics C*, 9(01):43–60, 1998.

[50] S.J. Barsky and M. Plischke. Order and localization in randomly cross-linked polymer networks. *Physical Review E*, 53(1):871–876, 1996.

[51] O. Farago and Y. Kantor. Entropic elasticity of phantom percolation networks. *Europhysics Letters*, 52(4):413–419, 2000.

[52] S. Lay, J.U. Sommer, and A. Blumen. Comparison of structural properties of different polymer network types as obtained by computer simulation. *The Journal of Chemical Physics*, 110(24):12173–12182, 1999.

[53] E.R. Duering, K. Kremer, and G.S. Grest. Structural properties of randomly crosslinked polymer networks. *Progress in Colloid & Polymer Science*, 90:13–15, 1992.

[54] E.R. Duering, K. Kremer, and G.S. Grest. Structure and relaxation of end-linked polymer networks. *The Journal of Chemical Physics*, 101(9):8169–8192, 1994.

[55] G.S. Grest, K. Kremer, and E.R. Duering. Kinetics and relaxation of end crosslinked polymer networks. *Physica A: Statistical Mechanics and Its Applications*, 194(1):330–337, 1993.

[56] S.J. Marrink, J. Risselada, and A.E. Mark. Simulation of gel phase formation and melting in lipid bilayers using a coarse grained model. *Chemistry and physics of lipids*, 135(2):223–244, 2005.

[57] A.M. Puertas, M. Fuchs, and M.E. Cates. Simulation study of nonergodicity transitions: Gelation in colloidal systems with short-range attractions. *Physical Review E*, 67:031406, 2003.

[58] J.C.G. Gimel, D. Durand, and T. Nicolai. Transition between flocculation and percolation of a diffusion-limited cluster-cluster aggregation process using three-dimensional monte carlo simulation. *Physical Review B*, 51(17):11348–11358, 1995.

[59] Y. Liu and R.B. Pandey. Computer-simulation studies of kinetic gelation. *Physical Review B*, 55(13):8257, 1997.

[60] E. Del Gado, L. de Arcangelis, and A. Coniglio. A percolation dynamic approach to the sol-gel transition. *Journal of Physics A: Mathematical and General*, 31(8):1901, 1998.

[61] H.J. Herrmann, D. Stauffer, and D.P. Landau. Computer simulation of a model for irreversible gelation. *Journal of Physics A: Mathematical and General*, 16(6):1221, 1983.

[62] H.G. Ballesteros, L.A. Fernández, V. Martin-Mayor, A.M. Sudupe, G. Parisi, and J.J. Ruiz-Lorenzo. Scaling corrections: site percolation and ising model in three dimensions. *Journal of Physics A: Mathematical and General*, 32(1):1, 1999.

[63] T. Alfrey and W.G. Lloyd. Network polymers. i. theoretical remarks. *Journal of Polymer Science*, 62(173):159–165, 1962.

[64] W.G. Lloyd and T. Alfrey. Network polymers. ii. experimental study of swelling. *Journal of Polymer Science*, 62(174):301–316, 1962.

[65] C.D. Lorenz and R.M. Ziff. Precise determination of the critical percolation threshold for the three-dimensional Şswiss cheseŤ model using a growth algorithm. *The Journal of Chemical Physics*, 114:3659, 2001.

[66] M. Rubinstein and S. Panyukov. Nonaffine deformation and elasticity of polymer networks. *Macromolecules*, 30(25):8036–8044, 1997.

[67] W. Gronski, R. Stadler, and M. Maldaner Jacobi. Evidence of nonaffine and inhomogeneous deformation of network chains in strained rubber-elastic networks by deuterium magnetic resonance. *Macromolecules*, 17(4):741–748, 1984.

[68] A. Basu, Q. Wen, X. Mao, T.C. Lubensky, P.A. Janmey, and A.G. Yodh. Nonaffine displacements in flexible polymer networks. *Macromolecules*, 44(6):1671–1679, 2011.

[69] Q. Wen, A. Basu, P.A. Janmey, and A.G. Yodh. Non-affine deformations in polymer hydrogels. *Soft Matter*, 8(31):8039–8049, 2012.

[70] M.W. Dreischor. *The Physics of Polymer Chains; Returning to Simple Theory*. PhD thesis, Universiteit van Amsterdam, november 2005.

[71] C. Williams, F. Brochard, and H.L. Frisch. Polymer collapse. *Annual Review of Physical Chemistry*, 32(1):433–451, 1981.

[72] K. Kamata, T. Araki, and H. Tanaka. Hydrodynamic selection of the kinetic pathway of a polymer coil-globule transition. *Physical Review Letters*, 102(10):108303, Mar 2009.

[73] C.D. Snow, E.J. Sorin, Y.M. Rhee, and V.S. Pande. How well can simulation predict protein folding kinetics and thermodynamics? *Annual Review of Biophysics and Biomolecular Structure*, 34:43–69, 2005.

[74] E. Slagowski, B. Tsai, and D. McIntyre. The dimensions of polystyrene near and below the theta temperature. *Macromolecules*, 9(4):687–688, 1976.

[75] P.J. Flory and W.R. Krigbaum. Thermodynamics of high polymer solutions. *Annual Review of Physical Chemistry*, 2:382–402, 1951.

[76] P.G. De Gennes. Collapse of a polymer chain in poor solvents. *Journal de Physique Lettres*, 36(3):55–57, 1975.

[77] M. Dijkstra, D. Frenkel, and J.P. Hansen. Phase separation in binary hard-core mixtures. *The Journal of Chemical Physics*, 101:3179, 1994.

[78] M. Dijkstra and D. Frenkel. Evidence for entropy-driven demixing in hard-core fluids. *Physical Review Letters*, 72(2):298, 1994.

[79] J.K.C. Suen, F.A. Escobedo, and J.J. de Pablo. Monte carlo simulation of polymer chain collapse in an athermal solvent. *The Journal of Chemical Physics*, 106:1288, 1997.

[80] V.V. Vasilevskaya, P.G. Khalatur, and A.R. Khokhlov. Conformation of a polymer chain near the solvent critical region. ii. monte carlo simulation. *The Journal of Chemical Physics*, 109:5119, 1998.

[81] J.M. Polson. Entropic collapse transition of a polymer in a solvent with a nonadditive potential. *Physical Review E*, 60(3):3429, 1999.

[82] J.M. Polson and M.J. Zuckermann. Simulation of heteropolymer collapse with an explicit solvent in two dimensions. *The Journal of Chemical Physics*, 113:1283, 2000.

[83] J.M. Polson and M.J. Zuckermann. Simulation of short-chain polymer collapse with an explicit solvent. *The Journal of Chemical Physics*, 116:7244, 2002.

[84] C.F. Abrams, N.K. Lee, and S.P. Obukhov. Collapse dynamics of a polymer chain: Theory and simulation. *Europhysics Letters*, 59(3):391, 2002.

[85] J.M. Polson and N.E. Moore. Simulation study of the coil-globule transition of a polymer in solvent. *The Journal of Chemical Physics*, 122:024905, 2005.

[86] N. Karasawa and W.A. Goddard III. Phase transitions in a polymethylene single chains from monte carlo simulated annealing. *The Journal of Chemical Physics*, 92:5828, 1988.

[87] I. Szleifer, E.M. OŠToole, and A.Z. Panagiotopoulos. Monte carlo simulation of the collapse-coil transition in homopolymers. *The Journal of Chemical Physics*, 97:6802, 1992.

[88] A. Milchev, W. Paul, and K. Binder. Off-lattice monte carlo simulation of dilute and concentrated polymer solutions under theta conditions. *The Journal of Chemical Physics*, 99:4786, 1993.

[89] T.A. Kavassalis and P.R. Sundararajan. A molecular-dynamics study of polyethylene crystallization. *Macromolecules*, 26(16):4144–4150, 1993.

[90] P. Grassberger and R. Hegger. Simulations of three-dimensional θ polymers. *The Journal of Chemical Physics*, 102:6881, 1995.

[91] P. Grassberger. Pruned-enriched rosenbluth method: Simulations of θ polymers of chain length up to 1 000 000. *Physical Review E*, 56(3):3682, 1997.

[92] A.M. Rubio, J.J. Freire, J.H.R. Clarke, C.W. Yong, and M. Bishop. Characterization of the theta state and transition curves of off-lattice three dimensional chains. *The Journal of Chemical Physics*, 102:2277, 1995.

[93] J. Ma, J.E. Straub, and E.I. Shakhnovich. Simulation study of the collapse of linear and ring homopolymers. *The Journal of Chemical Physics*, 103:2615, 1995.

[94] G. Tanaka and W.L. Mattice. Chain collapse by atomistic simulation. *Macromolecules*, 28(4):1049–1059, 1995.

[95] G. Tanaka and W.L. Mattice. Chain collapse by lattice simulation. *Macromolecular theory and simulations*, 5(3):499–523, 1996.

[96] M. Wittkop, S. Kreitmeier, and D. Göritz. The collapse transition of a single polymer chain in two and three dimensions: A monte carlo study. *The Journal of Chemical Physics*, 104:3373, 1996.

[97] Y. Zhou, M. Karplus, J.M. Wichert, and C.K. Hall. Equilibrium thermodynamics of homopolymers and clusters: Molecular dynamics and monte carlo simulations of systems with square-well interactions. *The Journal of Chemical Physics*, 107:10691, 1997.

[98] J.P.K. Doye, R.P. Sear, and D. Frenkel. The effect of chain stiffness on the phase behaviour of isolated homopolymers. *The Journal of Chemical Physics*, 108:2134, 1998.

[99] W. Hu. Structural transformation in the collapse transition of the single flexible homopolymer model. *The Journal of Chemical Physics*, 109:3686, 1998.

[100] H Noguchi and K Yoshikawa. Morphological variation in a collapsed single homopolymer chain. *The Journal of Chemical Physics*, 109:5070, 1998.

[101] V.A. Ivanov, W. Paul, and K. Binder. Finite chain length effects on the coil–globule transition of stiff-chain macromolecules: A monte carlo simulation. *The Journal of Chemical Physics*, 109:5659, 1998.

[102] Q. Liao and X. Jin. Formation of segmental clusters during relaxation of a fully extended polyethylene chain at 300 k: A molecular dynamics simulation. *The Journal of Chemical Physics*, 110:8835, 1999.

[103] A. Irbäck and E. Sandelin. Monte carlo study of the phase structure of compact polymer chains. *The Journal of Chemical Physics*, 110:12256, 1999.

[104] P.Y. Lai. Histogram monte carlo simulation on phase transitions in single-polymer systems. *Macromolecular theory and simulations*, 8(4):382–390, 1999.

[105] H. Liang and H. Chen. First-order transition of a homopolymer chain with lennard-jones potential. *The Journal of Chemical Physics*, 113:4469, 2000.

[106] V.A. Ivanov, M.R. Stukan, V.V. Vasilevskaya, W. Paul, and K. Binder. Structures of stiff macromolecules of finite chain length near the coil-globule transition: A monte carlo simulation. *Macromolecular theory and simulations*, 9(8):488–499, 2000.

[107] S. Fujiwara and T. Sato. Structure formation of a single polymer chain. i. growth of trans domains. *The Journal of Chemical Physics*, 114:6455, 2001.

[108] W. Paul and M. Müller. Enhanced sampling in simulations of dense systems: The phase behavior of collapsed polymer globules. *The Journal of Chemical Physics*, 115:630, 2001.

[109] F. Calvo, J.P.K. Doye, and D.J. Wales. Collapse of lennard-jones homopolymers: Size effects and energy landscapes. *The Journal of Chemical Physics*, 116:2642, 2002.

[110] L. Huang, X. He, Y. Wang, H. Chen, and H. Liang. Phase transitions of short chains at ultralow temperature. *The Journal of Chemical Physics*, 119:2432, 2003.

[111] M.P. Taylor and J.E.G. Lipson. Collapse of a polymer chain: A born–green–yvon integral equation study. *The Journal of Chemical Physics*, 104:4835, 1996.

[112] H.H. Gan and B.C. Eu. Influence of the solvent on the conformation of a chain molecule. *The Journal of Chemical Physics*, 109:2011, 1998.

[113] H.H. Gan and B.C. Eu. Integral equation theory of single-chain polymers: Comparison with simulation data for hard-sphere and square-well chains. *The Journal of Chemical Physics*, 110:3235, 1999.

[114] M.P. Taylor. Configurational statistics for isolated square-well chain molecules: exact results for short chains. *Molecular Physics*, 86(1):73–85, 1995.

[115] M.P. Taylor. Collapse transition of isolated lennard-jones chain molecules: Exact results for short chains. *The Journal of Chemical Physics*, 114:6472, 2001.

[116] M.P. Taylor. Collapse transition for isolated square-well chain molecules: The exact density of states for short chains. *The Journal of Chemical Physics*, 118:883, 2003.

[117] H.G. Schwartzberg and M.A. ed Rao. *Biotechnology and food process engineering*, volume 5. CRC Press, 1990.

[118] C.P. Lowe and M.W. Dreischor. The size of a polymer in a symmetric solvent. *The Journal of Chemical Physics*, 122:084905, 2005.

[119] M. Doi and S.F. Edwards. *The Theory of Polymer Dynamics Clarendon*. Oxford, 1986.

[120] N. Kikuchi, J. F. Ryder, C. M. Pooley, and J. M. Yeomans. Kinetics of the polymer collapse transition: The role of hydrodynamics. *Physical Review E*, 71(6):061804, Jun 2005.

[121] T. Frembgen-Kesner and A.H. Elcock. Striking effects of hydrodynamic interactions on the simulated diffusion and folding of proteins. *Journal of Chemical Theory and Computation*, 5(2):242–256, 2009.

[122] I.C. Sanchez. Phase transition behavior of the isolated polymer chain. *Macromolecules*, 1979.

[123] G. Luna-Bárcenas, J.C. Meredith, I.C. Sanchez, K.P. Johnston, D.G. Gromov, and J.J. de Pablo. Relationship between polymer chain conformation and phase boundaries in a supercritical fluid. *The Journal of Chemical Physics*, 107:10782, 1997.

[124] AM Kraynik and DA Reinelt. Extensional motions of spatially periodic lattices. *International journal of multiphase flow*, 18(6):1045–1059, 1992.

[125] András Baranyai and Peter T Cummings. Steady state simulation of planar elongation flow by nonequilibrium molecular dynamics. *The Journal of Chemical Physics*, 110:42, 1999.

[126] G. Juarez and P.E. Arratia. Extensional rheology of dna suspensions in microfluidic devices. *Soft Matter*, 7(19):9444–9452, 2011.

[127] M. Nakano, H. Kurita, J. Komatsu, A. Mizuno, and S. Katsura. Stretching of long dna molecules in the microvortex induced by laser and ac electric field. *Applied Physics Letters*, 89(13):133901–133901, 2006.

[128] J. Wang and C. Lu. Single molecule λ-dna stretching studied by microfluidics and single particle tracking. *Journal of Applied Physics*, 102(7):074703, 2007.

[129] F.H. Stillinger. Axiomatic basis for spaces with noninteger dimension. *Journal of Mathematical Physics*, 18:1224, 1977.

[130] B. Nienhuis. Exact critical point and critical exponents of o (n) models in two dimensions. *Physical Review Letters*, 49:1062–1065, 1982.

[131] Z. Alexandrowicz. Self-avoiding walks of continuous spatial dimensionality. *Physical Review Letters*, 50(10):736, 1983.

[132] P.G. de Gennes. Exponents for the excluded volume problem as derived by the wilson method. *Physics Letters A*, 38(5):339–340, 1972.

[133] N. Clisby. Accurate estimate of the critical exponent ν for self-avoiding walks via a fast implementation of the pivot algorithm. *Physical review letters*, 104(5):055702, 2010.

[134] C.P. Lowe and M.W. Dreischor. Simplicity and scaling-size of a real polymer in three (or any) dimensions. *arXiv preprint arXiv:0809.3666*, 2008.

[135] J. Goettker-Schnetmann. O (n)-invariant hierarchical renormalization group fixed points by algebraic numerical computation and ϵ-expansion. *arXiv preprint cond-mat/9909418*, 1999.

[136] D.M.Y. Somerville. *An Introduction to the Geometry of n Dimensions*. Dover Publications, Inc., New York, 1958.