Molecular Simulation of Nucleation in Polymers

Thomas J. Wicks, MMath.

Thesis submitted to The University of Nottingham
for the degree of Doctor of Philosophy

September 2015

Dedicated to Ian Wicks

Abstract

We develop several new algorithms using molecular simulation to investigate the nucleation barrier of a single, freely-jointed polymer chain. In the first part of the thesis, we use a free particle model to develop a new biasing technique, which uses an automated feedback mechanism to overcome the poor sampling of crystal states in a thermodynamic system. Our feedback technique does not require any prior knowledge of the nucleation barrier and enables good representative sampling of all available states of interest. In the second part of the thesis, we simulate the nucleation barrier of the single, freely-jointed, square-well chain. We use our feedback technique and parallel tempering with a non-standard temperature distribution to overcome poor sampling of crystal states and configuration mixing issues respectively. We also provide some comparative analysis of different choices of configurational order parameters for the single chain. Finally, we apply stretching to the chain to approximate flow-induced crystallisation and investigate the effect of different degrees of stretch on the nucleation barrier.

We verify the quality of our simulation with careful monitoring of several criteria, including the acceptance ratios of configuration swaps between simulations with adjacent temperatures, evolution of the energy traces as a result of configuration swaps between tempering levels, and ensuring effective de-correlation of configurations through reptation moves. Our simulations provide strong reproducible results for the base, the peak and beyond the peak of the barrier for the quiescent and stretched single chain. We observe a remarkably strong effect of modest stretching on the nucleation barrier for a single chain, which can potentially lead to dramatic effects on the nucleation rate. Our simulation code has been made publicly available, with details provided in an appendix.

Acknowledgements

I would like to thank my supervisors Richard Graham and Jonathan Wattis for their advice, guidance and patience over the last four years. I found the endless hours of meetings, discussions and coding sessions infinitely valuable to the development of my understanding of my model and the underlying mathematics.

Many thanks also to Joe Collis for our brief but fruitful discussions about parallel programming, which turned out to be essential for unlocking the capabilities of my simulation. Thank you to Philip Paine for being a sounding board for my often random musings and complaints about my research over the years we shared an office. Of course, I can't forget to express my gratitude for the welcome distraction of completing nearly every online Only Connect Wall together too!

Many thanks also to Tom Bennison for offering vast amounts of coding advice and resources throughout my project. Thanks also for being the second half of an effective teaching double-act for many years. This has provided me with countless valuable experiences and skills that have supplemented my research studies.

Thank you to Andrea Blackbourn, Dave Parkin, Helen Cunliffe and all the other support staff in the School of Mathematical Sciences for their tireless hard work in supporting me and all the other research students.

I cannot emphasise enough how grateful I am to my friends and fellow PhD students in the school for all the tea room breaks, cake club meetings, film nights and many many more social outings that have kept me going over the years! I would particularly like to thank my close friends Sunny Modhara, Áine Byrne, Ria Symonds and Lorna Wilson for their tireless support and advice, particu-

larly in the last few months of my PhD.

Thank you to my Mum and late Dad, who have never stopped believing in me in my many years in education. My Mum will be even happier now that I've got a job!

Finally, my biggest thank you goes to my girlfriend Charlotte, who has been there every step of the way. Thank you for putting up with the lonely evenings when I've been working late and dealing with my absent mindedness in the final stages of writing up.

Contents

CHAPTER 1

Introduction

1.1 Overview

The aim of this project is to devise a set of algorithms using molecular simulation to study the nucleation barriers of free particles and single polymer chains. In addition, we wish to observe the effect of stretching the chain on its free energy barrier. To obtain accurate and reproducible results for the free energy barrier from our simulations, we develop new simulation techniques to overcome the issues of poor sampling of rare crystal states and poor mixing in simulations of connected particles. In addition to developing the new techniques, we provide evidence that our simulations are effectively exploring and sampling the state-space of interest by carefully setting several criteria that are monitored throughout the simulation.

1.1.1 Outline of introduction

The introduction and literature review is divided into three sections. The first section deals with the methods for modelling the liquid-solid phase transition. The second section concerns popular techniques for simulation. Finally, the third section contains details on the modelling of polymers, particularly their nucleation. Collectively, these are the three areas that are needed for the model being developed in this thesis for simulating the nucleation of polymer chains.

In the **crystallisation** section, we explore the current methods for modelling the nucleation process, starting with classical nucleation theory and how this

has been developed, extended and even abandoned in the recent literature. We begin from elementary thermodynamics to develop governing equations for a thermodynamic system in a variety of ensembles using statistical mechanics. The liquid-solid transition is harder to model than the gas-liquid, since we must consider the particular crystal structure of the resulting solid after the transition. This motivates us to consider work on defining order parameters to distinguish crystal structures in both atomic and polymeric structure, so order parameters appear in both this section and in the discussion of polymers. Several different order parameters from the literature are discussed, along with our preference and the reasons for our choice.

Having established our model for nucleation, we proceed to a discussion of **simulation techniques** for systems of particles. One can perform a simulation of a thermodynamic system simply to obtain averages of quantities, so this is the starting point in our investigation. We discuss various different ensembles in which to conduct our simulations and the costs and benefits of each. We motivate our use of the Metropolis Monte Carlo method and outline how it is implemented. Since we are resolving the free energy barrier for nucleation, we need to sample the incredibly rare nucleation events. Thus, we need biasing to obtain sufficient sampling of rare states. The methods of doing this from the literature are discussed along with our chosen procedure, though the algorithmic details regarding our implementation of the bias are omitted until the relevant chapters. We finally introduce the method of parallel tempering, which allows the simulation to explore the state-space sufficiently to achieve ergodicity, while achieving good sampling of rare crystal states.

Our final introductory section concerns the modelling of **polymers**. In order to model a generalised polymer, a common technique in the literature is the united atoms model. We describe this model, along with the corresponding force field to simulate the forces between the true atoms in the polymer before applying the model so that the results are physically meaningful. We go on to discuss various order parameters proposed and used in the literature that can be used to determine the crystal structure of polymer chains. Lastly, we introduce the concept of flow-induced crystallisation, since we wish to simulate nucleation from stretched chains and investigate how the resulting stretch of the polymers affects the free energy barrier for nucleation.

1.2 Crystallisation

1.2.1 Classical nucleation theory

The phenomenon of nucleation has been studied for almost a century, with Gibbs [1906] being the first to find that the stability of a phase is related to the work that needs to be done to create a critical nucleus of the new phase. The relevance of his work to nucleation was not apparent until the 1920's and 1930's when Volmer and Weber [1926] and Becker and Döring [1935] laid the foundations for what is now known as classical nucleation theory (CNT).

When discussing nucleation, it is important that we distinguish between *heterogeneous* and *homogeneous* nucleation. Heterogeneous nucleation is nucleation that is assisted by a heterogeneity in the system, such as the walls or an impurity, while homogeneous nucleation occurs due to spontaneous fluctuations in the pure unadulterated bulk phase. Although heterogeneous nucleation is a more common occurrence in everyday life, we restrict our studies to homogeneous nucleation, since it is here that there are still many discrepancies between theoretical and experimental results [Auer and Frenkel, 2001, Filion et al., 2010, Kawasaki and Tanaka, 2010, Filion et al., 2011], which makes it a much more interesting area.

Nucleation is the localised transition from one thermodynamic phase to another, for example, a small nucleus of crystallised water molecules forming as ice in a large body of water at low temperature. A successful nucleation event is when a nucleus of sufficient size to be thermodynamically stable is created. The addition of a single particle to the nucleus is not a rare event, but the sequence of additions of particles in order for the nucleus to reach critical size to grow spontaneously rather than redissolving is extremely rare [Auer and Frenkel, 2004]. The nucleation barrier is approximated by the Gibbs free energy of a spherical nucleus of radius R

$$\Delta G = \frac{4}{3}\pi R^3 \rho_s \Delta\mu + 4\pi R^2 \gamma, \tag{1.2.1}$$

where ρ_s is the number density of the bulk solid, $\Delta\mu < 0$ is a combination of the gain in potential energy and the loss in entropy upon crystallisation and γ is the free energy density of the liquid-solid interface [Auer and Frenkel, 2004]. Both

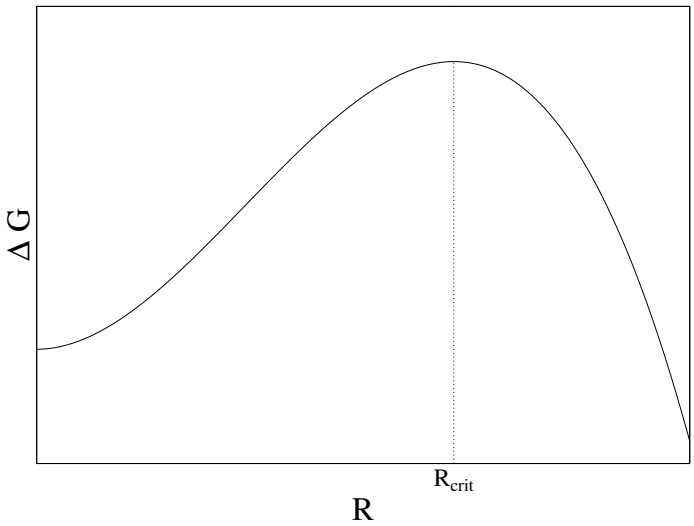


Figure 1.1: The Gibbs free energy, ΔG, as a function of R, the radius of the nucleus. See (1.2.1).

$\Delta\mu$ and γ are determined by the interactions of crystalline particles interact and how they pack in the solid and liquid phase.

The first term on the RHS of (1.2.1) represents the bulk free energy gained by increasing the volume of the nucleus, since the solid is more stable than the liquid. The second term is the free energy lost due to the increased surface tension at the interface between the solid nucleus and surrounding liquid. The form of (1.2.1) shows that nucleation is characterised by an energy barrier, which is plotted in Figure 1.1.

A simpler form of (1.2.1) in terms of the number of particles in the nucleus i is used in the classical nucleation paper by Turnbull and Fisher [1949], who give, assuming $i \propto R^3$ in (1.2.1), the expression

$$\frac{\Delta F}{k_B T} = -Ai + Bi^{2/3}, \tag{1.2.2}$$

where ΔF is the local free energy change associated with the formation of a region of a new phase, k_B is Boltzman's constant, T is the temperature and A and B are constants proportional to the quantities in the corresponding terms of (1.2.1). However, we shall proceed using (1.2.1), as we are interested in the microscopic interaction of particles.

The critical nucleus size for spontaneous growth is given by the maximum of ΔG. This gives a critical radius of $R_{crit} = 2\gamma/(\rho_s|\Delta\mu|)$, as marked in Figure 1.1.

The corresponding height of the energy barrier

$$\Delta G^* = \frac{16\pi}{3} \frac{\gamma^3}{(\rho_s |\Delta\mu|)^2}, \tag{1.2.3}$$

is the free energy required to form a critical nucleus. The probability of forming a critical nucleus depends exponentially on this free energy of formation [Auer and Frenkel, 2004]

$$P_c \propto \exp(-\Delta G^*/k_B T). \tag{1.2.4}$$

Therefore, we also have

$$\tau \propto \exp(-\Delta G^*/k_B T), \tag{1.2.5}$$

where τ is the nucleation rate. Since ΔG^* tends to be large for the systems we wish to examine, the probability of the formation of a critical nucleus will be very small, with the nucleation rate being correspondingly slow. This makes direct dynamic simulations of nucleation difficult to conduct, since unrealistically long simulations are needed for a nucleus to reach critical size.

1.2.1.1 Criticism of classical nucleation theory

We mentioned above the discrepancy between results predicted by CNT and experiments. These arise due to the many assumptions made by CNT. The first of these assumptions is that nucleation only depends on the macroscopic properties of the system, such as temperature and does not take into account the microscopic properties. Other assumptions, such as nuclei always being spherical and incompressible, meaning that the bulk solid number density, ρ_s does not change with the pressure p are also major simplifications that are not always reasonable. As a result, many attempts have been made to refine CNT [Reiss et al., 1997] and extend [McGraw and Laaksonen, 1996, 1997] and theoretical methods have been devised to go beyond the macroscopic description of CNT [Oxtoby and Evans, 1988].

The reason that it is so difficult to understand nucleation at the microscopic level is that it is an activated process, in that it must overcome an energy barrier, such as that in Figure 1.1, to go forward. In experiments, the probability of finding a nucleus at the top of the barrier is incredibly small, of the order 10^{-12} per cm^3 and the system will only stay at the top of the barrier for a matter of picoseconds to nanoseconds for atomic or simple molecular systems [ten Wolde,

1998]. Thus, nucleation is very infrequent and fast and so it is difficult to study these critical nuclei in an experiment. However, it is the structure and dynamics of these critical nuclei that play an essential role in nucleation, so computer simulations are a natural tool to study the nucleation process.

We wish to use molecular simulations to compute the energy barrier (such as that shown in Figure 1.1) for a three-dimensional system of N particles. This is a high-dimension simulation, as each of the particles has three degrees of freedom, giving $3N$ dimensions in total. In particular, we wish to resolve the barrier near its maximum in order to determine ΔG^*, the free energy required to form a critical nucleus and achieve spontaneous growth of the crystal.

1.2.2 Ensembles and potentials

For completeness, we derive the underlying thermodynamic laws that underpin the molecular simulation of nucleation. Though much of the material is not directly implemented into the simulation, we must understand the concept of an *ensemble*, which defines the fixed and varying thermodynamic variables within a simulation. We start with some thermodynamic nomenclature and outline the basic principles of thermodynamics to parametrise our model system of particles, whose nucleation is to be simulated.

1.2.2.1 The first law of thermodynamics

The first law of thermodynamics for a closed system can be stated as

$$\mathrm{d}U = \mathrm{d}\tilde{W} + \mathrm{d}Q, \tag{1.2.6}$$

where $\mathrm{d}U$ is the change in internal energy of the system, $\tilde{W}$ is the work done on the system and Q is the heat added to the system. Thus, the first law is a statement of conservation of energy in a thermodynamic system. The internal energy, U is our first *function of state*, which is a function that is only dependent on the current state of the system and is independent of the path taken to reach the current state. The first law is not particularly useful to us in this form, so we use the following definitions to rewrite $\tilde{W}$ and Q.

We assume that the system remains in *mechanical* and *thermal equilibrium*, meaning that the pressure and temperature are constant respectively. Since mechan-

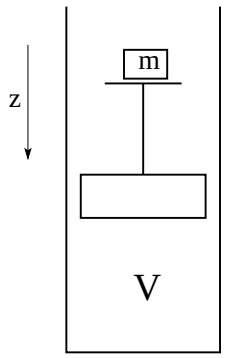


Figure 1.2: Diagram of a gas cylinder, where a mass m rests on a piston, which moves along the z-axis.

ical work is not a function of state, any changes made to the system must be *adiabatic*, meaning that the changes are infinitessimally small such that the system remains in equilibrium throughout the change. Furthermore, any changes made to the system must be *reversible*, i.e. reversing direction along the specified path will return the system to its initial state. As an example, take a gas cylinder with a piston at one end, which moves along the z-axis and has a mass m resting on top of it. A tiny mass δm is added to the piston so that it drops a small distance $\mathrm{d}z$. This process is repeated in order to make a quasistatic change to the volume, V, of the system. We have

$$\mathrm{d}V = -A\,\mathrm{d}z, \tag{1.2.7}$$

where A is the area of the piston cross-section. The force acting downwards is $-(m+\delta m)g\hat{\mathbf{z}}$, where g is the acceleration due to gravity. Thus, the work done on the system is

$$\begin{aligned} đ\tilde{W} &= (m+\delta m)g\,\mathrm{d}z, && \text{(1.2.8a)}\\ &= -\frac{(m+\delta m)g}{A}(-A\,\mathrm{d}z), && \text{(1.2.8b)}\\ &= -\frac{(m+\delta m)g}{A}\,\mathrm{d}V, && \text{(1.2.8c)} \end{aligned}$$

where đ denotes an inexact differential. The inexact differential reflects the fact that the work done is not a function of state, so is path dependent, meaning that one cannot simply integrate this expression between the initial and final positions of the piston to find the total work done. However, since the system is in mechanical equilibrium, mg/A is the pressure, p, of the gas in the cylinder. Also, if δm is negligible compared to m,

$$đ\tilde{W} = -p\,\mathrm{d}V. \tag{1.2.9}$$

We can now write (1.2.6) as

$$\mathrm{d}U = đQ_{\text{rev}} - p\,\mathrm{d}V, \tag{1.2.10}$$

where Q_{rev} is the heat that is added to the system in a reversible way. We are now in a position to define two more functions of state.

1.2.2.2 Enthalpy

We again use the fact that the system is in mechanical equilibrium to state that the pressure is constant and so $\mathrm{d}p = 0$. From (1.2.10),

$$\begin{aligned} đQ_{\text{rev}} &= \mathrm{d}U + p\,\mathrm{d}V, && \text{(1.2.11a)} \\ &= \mathrm{d}(U + pV), && \text{(1.2.11b)} \\ &= \mathrm{d}H, && \text{(1.2.11c)} \end{aligned}$$

where $H = U + pV$ is the *enthalpy* of the system and is measured in units of energy. Any change in H is equal to the heat added to the system provided there is no other form of work done on it.

1.2.2.3 Entropy

The second function of state that can be defined is the *entropy* S. When we reversibly add heat $đQ_{\text{rev}}$ to the system such that the temperature remains constant, the entropy is defined as

$$\Delta S = \int_i^f \frac{đQ_{\text{rev}}}{T}, \tag{1.2.12}$$

where i and f are the initial and final values of the heat energy in the system as a result of the change $đQ_{\text{rev}}$ and T is the temperature. There are several physical interpretations of the entropy of a system. For our purposes, we interpret it as a measure of the number of available states that the system can be in, given the values of all of the other thermodynamic quantities. This interpretation will become clearer in light of §1.2.2.6. Using (1.2.12), we obtain the differential relation

$$\mathrm{d}S = \frac{đQ_{\text{rev}}}{T}. \tag{1.2.13}$$

Substituting for Q_{rev} in (1.2.10), we can write the first law as

$$\mathrm{d}U = T\,\mathrm{d}S - p\,\mathrm{d}V, \tag{1.2.14}$$

which is Gibbs' equation [Bowley and Sanchez, 1999]. With this definition of entropy, we have a thermodynamic definition of the temperature

$$T = \left(\frac{\partial U}{\partial S}\right)_V, \tag{1.2.15}$$

i.e. the temperature is the change in internal energy with respect to the entropy at fixed volume. For example, with the appropriate scaling, the temperature is the amount of heat released to the environment when one bit of information is deleted from a computer (see §1.2.2.6 for what is meant by 'the appropriate scaling').

1.2.2.4 Chemical potential

Up to this point in the analysis of our thermodynamic system, we have not specified whether the number of particles is constant or not. If we begin with N particles and this remains constant throughout the experiment, the system is in *chemical equilibrium*. If this is not the case, then we are doing chemical work on the system and we define

$$\mu = \left(\frac{\partial U}{\partial N}\right)_{S,V}, \tag{1.2.16}$$

as the *chemical potential*, which has units of energy and is a measure of the energy required to add or remove particles from the system. With this definition, we finally have the appropriate form of the first law

$$\boxed{\mathrm{d}U = T\,\mathrm{d}S - p\mathrm{d}V + \mu\,\mathrm{d}N.} \tag{1.2.17}$$

This is our first *thermodynamic potential* that can be used to parametrise our system. There are several other potentials that can be obtained by taking appropriate *Legendre transforms* of (1.2.17) (see Appendix A). The commonest thermodynamic potentials are listed in Table 1.1. For example, if we wish to conduct an experiment in which the variables T, p and N are fixed, the appropriate potential to use is the Gibbs free energy, whose independent variables are T, p and N. Since T, p and N are constant, $\mathrm{d}G = 0$ and G is invariant for each state of the system throughout the experiment.

Table 1.1: Thermodynamic potentials and their independent variables.

Potential	Symbol	Formula	Variables
Internal Energy	U	$\int (T\,\mathrm{d}S - p\,\mathrm{d}V + \mu\,\mathrm{d}N)$	S, V, N
Helmholtz Free Energy	F	$U - TS$	T, V, N
Enthalpy	H	$U + pV$	S, p, N
Gibbs Free Energy	G	$U + pV - TS = H - TS$	T, p, N
Grand Potential	Φ	$U - TS - \mu N$	T, V, μ

Table 1.2: Intensive and extensive thermodynamic variables.

Intensive Variables	Extensive Variables
μ, p, T, ρ	U, F, H, G, Φ, S, V, N

1.2.2.5 Intensive and extensive variables

There is one final classification of thermodynamic variables to define. An *intensive* quantity is one which does not depend on the system size or the amount of material in the system. On the other hand, an *extensive* quantity is one which scales with the system size. Clearly, we prefer to deal with intensive properties, so that we can scale the system size arbitrarily. We therefore take ratios of extensive quantities to obtain a new intensive quantity. An example of this is taking the ratio of N and V, which are both extensive, to obtain the bulk number density $\rho = N/V$, which is intensive. Table 1.2 summarises the intensive and extensive nature of the variables we have discussed so far.

1.2.2.6 Statistical mechanics

The principles of statistical mechanics are laid out in Appendix B, which lead to Boltzmann's formula for the entropy,

$$S = k_\mathrm{B} \ln W, \tag{1.2.18}$$

where k_B is Boltzmann's constant. In (1.2.18), W stands for the German word for probability, *Wahrscheinlichkeit*, since W is linked to the probability that the system is in a particular state, or rather the number of ways that an observable state can be realised. To understand what this means, we define *microstates* and *macrostates*.

1.2.2.7 Microstates and macrostates

A *microstate* is a complete microscopic description of the quantum state of the system. In the context of our simulations, we define a microstate A as a vector, $\mathbf{r}^A$, of $\mathbf{r}_i^A$, the positions of each of the N particles in the system when the whole system is in state A, thus

$$\mathbf{r}^A = (\mathbf{r}_1^A, \mathbf{r}_2^A, \ldots, \mathbf{r}_N^A). \tag{1.2.19}$$

In practice, it is usually not possible to measure the complete microscopic description of a system, such as the potential energy of every particle, so we define a *macrostate*. A macrostate is a complete specification of the macroscopic quantities of the system, such as the temperature and pressure, which are possible to measure. A macrostate can be realised by many different microstates. In the context of statistical mechanics, a macrostate is characterised by a probability distribution of microstates over a statistical ensemble of systems with identical macroscopic quantities. Next, we discuss how to describe these statistical ensembles.

1.2.2.8 Microcanonical ensemble

If we define microstates using (1.2.19), we only consider states that are accessible given the macroscopic quantities that have been specified. When there are several microstates that have the same macroscopic properties, we say the macrostate is *degenerate*. We calculate the number of degenerate microstates W, where the system has equal probability $p = 1/W$ of being in any of them. The ensemble of degenerate microstates for a given set of macroscopic quantities is the *microcanonical ensemble*. This ensemble is often called the NVU ensemble, since the total number of particles N, volume V and total internal energy U are kept constant. In practice, it is difficult to keep U fixed, so we need a more realistic ensemble to work in.

1.2.2.9 Canonical ensemble

We take a similar ensemble of systems as in the microcanonical ensemble, but this time the energy of each system is not constant. Each system is in thermal

contact with the remainder, which collectively act as a heat bath. We choose a system, A, in which to conduct our simulation. For the chosen system, A, one can fix the temperature by exchanging heat energy with the heat bath surrounding it. This is called the *canonical ensemble*. Over time, system A and remaining heat bath reach thermal equilibrium, matching the temperature of the bath. Therefore, the canonical ensemble is often named the NVT ensemble, since N, V and T remain constant throughout the simulation.

The combined system, made up of system A and surrounding heat bath, has total internal energy U_T, which is constant since the combined system is thermally isolated. The combined system also has entropy $S = k_\mathrm{B} \ln W$, where W is the number of accessible microstates of the combined system.

The chosen system, A, has internal energy U_A and the heat bath has energy U_R, where

$$U_R = U_T - U_A. \tag{1.2.20}$$

The total number of accessible microstates of the combined system, $W(U_A)$, is a function of the energy of system A. The volume and number of particles in A are fixed, but the energy can vary, so

$$W(U_A) = W_A(U_A) \times W_R(U_T - U_A), \tag{1.2.21}$$

where $W_A(U_A)$ and $W_R(U_T - U_A)$ are the number of accessible microstates of system A and the heat bath respectively. If we choose a particular microstate of A, $\mathbf{r}_i^A$ with corresponding energy $U_A = E_i$, the total number of microstates states for the combined system is

$$W(E_i) = 1 \times W_R(U_T - E_i). \tag{1.2.22}$$

Using (1.2.15) along with Boltzmann's formula for S, (1.2.18), we define the temperature of the heat bath as

$$\frac{1}{k_\mathrm{B} T} = \left(\frac{\partial \ln W_R}{\partial U_R} \right)_V. \tag{1.2.23}$$

By definition, adding energy to the heat bath does not alter its temperature, so integrating (1.2.23) gives

$$W_R = \gamma e^{U_R / k_\mathrm{B} T}, \tag{1.2.24}$$

where γ is a constant of integration, and (1.2.22) becomes

$$W(E_i) = \gamma e^{(U_T - E_i)/k_\mathrm{B} T}. \tag{1.2.25}$$

Since each of the microstates are equally probable, the total number of accessible microstates of the combined system is

(1.2.26)

where the sum is over all microstates of system A. Now, the classical probability that the system is in state $\mathbf{r}_i^A$ is given by the number of microstates equivalent to $\mathbf{r}_i^A$ divided by the total number of accessible microstates:

$$p_i = \frac{W(E_i)}{N_{\text{total}}} = \frac{e^{-E_i/k_\text{B}T}}{\sum_j e^{-E_j/k_\text{B}T}} = \frac{1}{Z_{NVT}} e^{-E_i/k_\text{B}T}, \tag{1.2.27}$$

where

$$Z_{NVT} = \sum_j e^{-E_j/k_\text{B}T} \tag{1.2.28}$$

This is the Boltzmann distribution, where $\exp(-E_i/k_\text{B}T)$ is the Boltzmann factor and Z_{NVT} is the partition function associated with the canonical ensemble and is defined in (1.2.28). The Z is for the German *Zustandssumme*, which means 'sum over states.' Note that the distribution of states in system A have no dependence on the states of the heat bath. This means that we can use the heat bath to specify the temperature of system A without it having any influence on the resulting distribution of states. The partition function has far-reaching significance in thermodynamics, since if it is known it can be used to compute all thermodynamic quantities for a given system. See [Bowley and Sanchez, 1999] for how this can be achieved. Note that the partition function will have a different form depending on the chosen ensemble for the simulation.

A *thermodynamic potential* Ψ_{ens} is defined for a particular ensemble [Allen and Tildesley, 1989] as

$$\Psi_{\text{ens}} = -\log Z_{\text{ens}}, \tag{1.2.29}$$

where Z_{ens} is the partition function associated with the ensemble. The function Ψ_{ens} is minimised when the system is in thermodynamic equilibrium. One can use Table 1.1 to determine the appropriate potential. The correct potential with independent variable that correspond to the variables that have been fixed, since these potentials emerged on the basis that they do not vary during the experiment. Thus, $\mathrm{d}\Psi_{\text{ens}} = 0$ and the potential is minimised. In the case

of the canonical ensemble, the appropriate potential is F, the Helmholtz free energy, so we define

$$F = -k_B T \log Z_{NVT}. \tag{1.2.30}$$

For a derivation of this and an explanation of the emergence of the factor of $k_B T$, see [Bowley and Sanchez, 1999]. In the next section, we discuss how the Boltzmann distribution is used in the context of our simulations.

1.2.2.10 The Boltzmann distribution

Following (1.2.19), we define a microstate A of N particles, where particle i has position $\mathbf{r}_i^A$, and

$$\mathbf{r}^A = (\mathbf{r}_1^A, \mathbf{r}_2^A, \ldots, \mathbf{r}_N^A). \tag{1.2.31}$$

The microstate has a corresponding potential energy $U(\mathbf{r}^A)$ (see §1.3.2.1 for our reasons for dropping the kinetic part of the energy, or Allen and Tildesley [1989]). The calculation of $U(\mathbf{r}^A)$ is dealt with below in §1.2.3.1. The microstates are distributed according to the Boltzmann distribution (1.2.27), so the probability of the system being in state A, ρ_A, is given by

$$\rho_A = \frac{1}{Z_{NVT}} \exp\left(\frac{-U(\mathbf{r}^A)}{k_B T}\right), \tag{1.2.32}$$

where $\exp(-U(\mathbf{r}^A)/k_B T)$ is the Boltzmann factor and Z_{NVT} is the NVT partition function

$$Z_{NVT} = \int \exp\left(\frac{-U(\mathbf{r}^A)}{k_B T}\right) \mathrm{d}\mathbf{r}^A. \tag{1.2.33}$$

In the definition of the partition function, we have replaced the sum in (1.2.27) with an integral since we are in the continuum limit, so the integral is over all possible microstates that can realise the specified macroscopic quantities. In general, it is not possible to evaluate the partition function, Z_{NVT} because of the high dimension of the integral and because the integral is sharply spiked around regions of low energy. Thus, we eliminate Z_{NVT} by instead reformulating (1.2.32) as

$$U(\mathbf{r}^A) = -k_B T \ln(\rho_A) - k_B T \ln Z_{NVT} \tag{1.2.34}$$

end then computing the energy difference in moving from state A to B, which is given by

$$\Delta U_{AB} = U(\mathbf{r}^B) - U(\mathbf{r}^A) = -k_B T \ln\left(\frac{\rho_A}{\rho_B}\right). \tag{1.2.35}$$

Rearranging (1.2.35), we obtain

$$\frac{\rho_B}{\rho_A} = \exp\left(-\frac{\Delta U}{k_\mathrm{B}T}\right). \tag{1.2.36}$$

The expression of the ratio ρ_B/ρ_A in terms of ΔU_{AB} in (1.2.36) will be essential for deriving our governing equations for our simulations in §1.3.

1.2.2.11 Isothermal-isobaric ensemble

The final ensemble we describe is the *isothermal-isobaric* ensemble. This is similar to the canonical ensemble in that we have a system containing a constant number of particles N, with the temperature kept constant by a surrounding heat bath, but one of the walls is replaced with a piston. The piston changes the volume of the system in order to maintain a constant pressure p. This ensemble is often called the NpT ensemble to reflect the fact that the macroscopic quantities N, p and T are fixed throughout the experiment. The probability of the system being in state A is now given by

$$\rho_A = \frac{1}{Z_{NpT}} \exp\left(-\frac{U(\mathbf{r}^A) + pV}{k_\mathrm{B}T}\right), \tag{1.2.37}$$

where Z_{NpT} is the partition function associated with the isothermal-isobaric ensemble. Note that the term in the exponent is the thermodynamic enthalpy of the system $H = \langle U \rangle + p\langle V \rangle$ when averaged over all states. Also, the volume V is now a microscopic quantity, since it is allowed to vary according to the prescribed pressure p, which is now a macroscopic quantity. The partition function, Z_{NpT}, is

$$Z_{NpT} = \int_0^\infty \int \exp\left(-\frac{U(\mathbf{r}^A) + pV}{k_\mathrm{B}T}\right) \mathrm{d}\mathbf{r}^A \, \mathrm{d}V. \tag{1.2.38}$$

The associated thermodynamic potential here is the Gibbs free energy, G, given by

$$G = -k_\mathrm{B}T \log Z_{NpT}. \tag{1.2.39}$$

As with Z_{NVT} (1.2.28), which is used to compute the Helmholtz free energy in the canonical ensemble using (1.2.30), Z_{NpT} is very difficult to compute, so we use a similar process to that in §1.2.2.10 to eliminate it during our simulations. The details of this are given below in §1.3.2.9

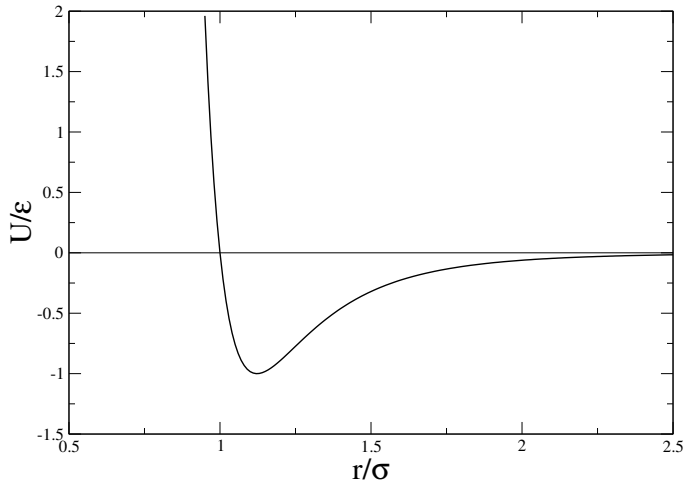


Figure 1.3: The Lennard-Jones potential (1.2.40). The r^{-12} term gives the repulsive core while the r^{-6} term gives the attractive tail. The potential is zero when $r/\sigma = 1$ and has a minimum of $U/\varepsilon = -1$ when $r/\sigma = 2^{1/6} \approx 1.12$.

1.2.3 Defining crystal order

In order to model the microscopic details of the nucleation process, we need to consider how individual particles interact. In addition, we must carefully define the crystal state of a given microstate $\mathbf{r}^A$ so that we can accurately describe the pathways to a critical nucleus.

1.2.3.1 Particle interaction

The simplest model for a crystal is one that is composed of spherical particles, which interact with each other according to a prescribed potential. Here we shall consider spheres that interact in accordance with the Lennard-Jones pair-potential [Allen and Tildesley, 1989, Chaikin and Lubensky, 1995]

$$U(r) = 4\varepsilon \left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6} \right], \tag{1.2.40}$$

where r is the distance between a pair of particles, ε is a measure of the attraction between them at their optimal separation and particles repel one another when they are a distance less than σ apart (see Figure 1.3). When assigning the energy of a microstate, $U(\mathbf{r}^A)$, in §1.2.2.10, we sum over all pair potentials (1.2.40) in the system in state A.

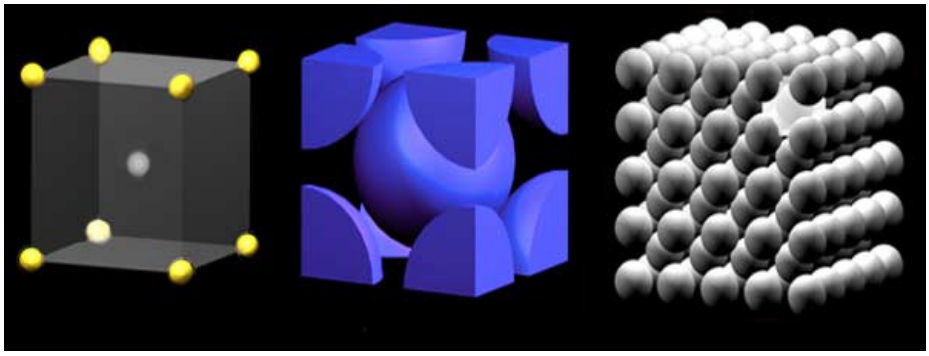


Figure 1.4: BCC structure. Image taken from NDT Education Resource Centre [2001a]. The left-most image shows the placement of spherical particles in the unit cell. The middle image is a close-up of the particles to illustrate their local relative arrangement. The right-most image shows the large lattice generated by repeating the unit cell in all directions.

1.2.3.2 Crystal structures

Though we use the Lennard-Jones model (1.2.40) for particle interaction in our system, we consider the packing of hard spheres, which is a good approximation for how Lennard-Jones particles pack together [ten Wolde et al., 1996a, ten Wolde and Frenkel, 1998].

When hard spheres come together to form a solid, they form a layered lattice structure, of which there are fourteen distinct types [see Chaikin and Lubensky, 1995]. We consider only three of these, namely body-centered cubic (BCC), face-centered cubic (FCC) and hexagonal close-packed (HCP). These structures are illustrated in Figures 1.4, 1.5 and 1.6. In each figure, the first two images from the left illustrate the unit cell for each structure, which then repeats in every direction in the lattice. The right-most image indicates the unit cell in a large lattice. Chaikin and Lubensky [1995] give a detailed description of these structures along with schematics of other types of lattices.

The BCC structure does not allow the spheres to minimise their separation, so crystals tend to have this form at high temperature, near to a change of phase. At lower temperatures, the FCC or HCP structure will be observed, since both of these allow the spheres to pack as closely as possible. The difference between FCC and HCP structures is the sequence in which layers of the lattice

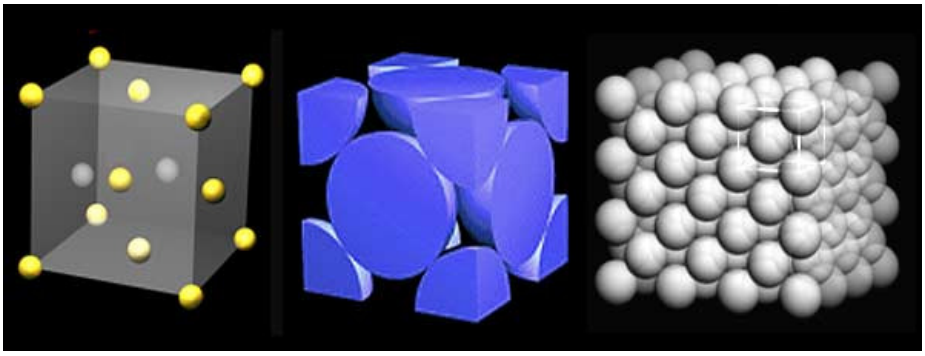


Figure 1.5: FCC structure. Image taken from NDT Education Resource Centre [2001a].

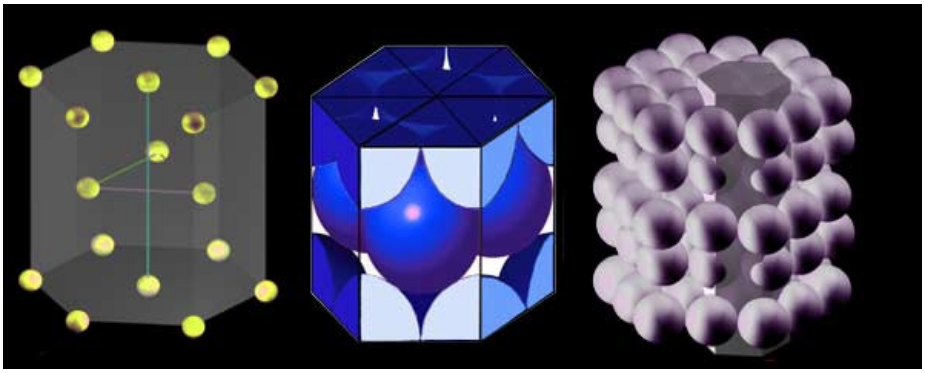


Figure 1.6: HCP structure. Image taken from NDT Education Resource Centre [2001a].

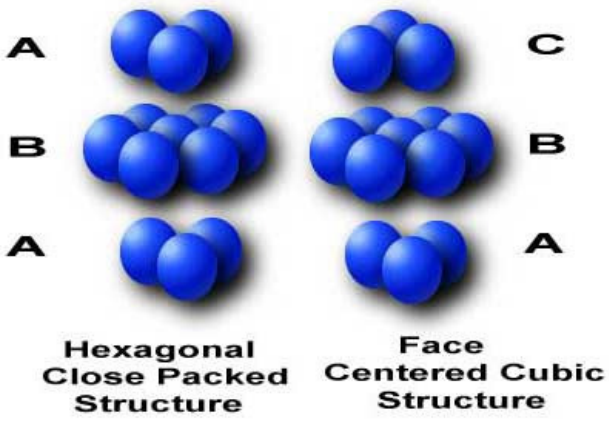


Figure 1.7: HCP layers are in an ABAB structure, while FCC layers are in an ABCABC structure. Image from NDT Education Resource Centre [2001b].

are packed. The HCP lattice has alternating layers of particles in two equivalent arrangements, while FCC has three, as shown in Figure 1.7. Since this is the only difference between HCP and FCC, a crystal can form with a combination of HCP and FCC structures. Thus, given a configuration of particles, we need a way of determining which regions are in a crystallised lattice, along with the corresponding lattice structure. Steinhardt et al. [1983], ten Wolde and Frenkel [1998], Auer and Frenkel [2004] and Filion et al. [2011] have addressed this question, suggesting an order parameter, which detects crystal structures by considering the orientation of vectors between the centres of closely-packed particles.

1.2.3.3 Order parameters

To measure crystal order in our system, we use one of the order parameters proposed by Steinhardt et al. [1983], which has since been used by many authors, such as ten Wolde et al. [1996a], who adapted the original parameters to identify crystal particles in the study of nucleation. The ideas in this paper were then developed by Auer and Frenkel [2004], with a detailed explanation of how to identify crystal particles. In addition to these studies, there are many examples in the literature of these order parameters being used, in particular, see van Duijneveldt and Frenkel [1992], ten Wolde et al. [1995, 1996b], Filion et al. [2011].

The first step in defining this order parameter is to determine how many near neighbours each particle in the system has. The near neighbours of particle i are defined as the set of $N_b(i)$ particles that are within a cut-off radius r_c of particle i. Steinhardt et al. [1983] specify a cut-off radius of $r_c = 1.2\sigma$, where σ is the Lennard-Jones length defined in (1.2.40). ten Wolde et al. [1996a], however, defines $r_c = 1.5\sigma$, since this corresponds to the first minimum in the radial distribution function, $g(r)$, in a FCC crystal of Lennard-Jones particles. As the minimum of the Lennard-Jones potential is located at $r = 2^{1/6}\sigma \approx 1.12\sigma$ (see Figure 1.3), it is sensible to choose a cut-off radius that is close to, but slightly larger than this value. Since authors' choice of r_c vary in the literature, we will choose a value in the interval $r_c \in [1.2\sigma, 1.5\sigma]$, where the choice depends on how permissive we wish to be when identifying near neighbours.

A *local* order parameter can be defined as

$$\overline{q}_{lm}(i) = \frac{1}{N_b(i)} \sum_{j=1}^{N_b(i)} q_{lm}(\mathbf{r}_j), \tag{1.2.41}$$

where

$$q_{lm}(\mathbf{r}) = Y_{lm}(\theta(\mathbf{r}), \phi(\mathbf{r})), \tag{1.2.42}$$

are spherical harmonics and $\theta(\mathbf{r})$ and $\phi(\mathbf{r})$ are the polar angles of the bond between particles i and j with respect to some reference coordinate system. ten Wolde [1998] shows that one can construct a *global* order parameter by a weighted average of the $\overline{q}_{lm}(i)$ over all N particles, giving

$$\overline{Q}_{lm} = \frac{\sum_{i=1}^{N} N_b(i)\overline{q}_{lm}(i)}{\sum_{i=1}^{N} N_b(i)}. \tag{1.2.43}$$

An additional order parameter, W_l, is defined in the original paper by Steinhardt et al. [1983] and also in the majority of subsequent papers that make use of $\overline{Q}_{lm}$ and is defined by

$$W_l = \sum_{\substack{m_1,\, m_2,\, m_3 \\ m_1 + m_2 + m_3 = 0}} \begin{pmatrix} l & l & l \\ m_1 & m_2 & m_3 \end{pmatrix} \overline{Q}_{lm_1} \overline{Q}_{lm_2} \overline{Q}_{lm_3} \tag{1.2.44}$$

where the term in brackets is a Wigner-3j symbol. The Wigner-3j symbol arises when computing coefficients for coupled angular momentum in quantum mechanics. The evaluation of the Wigner-3j symbol is complicated, but can be simplified using its cyclic permutation symmetry and othogonality properties. If $m_1 + m_2 + m_3 \neq 0$, then the symbol evaluates to zero. Otherwise, the algebraic formula used by Rotenberg et al. [1959] to calculate the general Wigner-3j symbol for integers j_i, m_i, is

$$\begin{aligned}
\begin{pmatrix} j_1 & j_2 & j_3 \\ m_1 & m_2 & m_3 \end{pmatrix} &= (-1)^{j_1 - j_2 - m_3} \\
&\times \left[\frac{(j_1 + j_2 - j_3)!(j_1 - j_2 + j_3)!(-j_1 + j_2 + j_3)!}{(j_1 + j_2 + j_3 + 1)!} \right]^{1/2} \\
&\times [(j_1 + m_1)!(j_1 - m_1)!(j_2 + m_2)!(j_2 - m_2)!(j_3 + m_3)!(j_3 - m_3)!]^{1/2} \\
&\times \left[\sum_k \frac{(-1)^k}{k!(j_1 + j_2 - j_3 - k)!(j_1 - m_1 - k)!(j_2 + m_2 - k)!} \right. \\
&\qquad \left. \times \frac{1}{(j_3 - j_2 + m_1 + k)!(j_3 - j_1 - m_2 + k)!} \right],
\end{aligned} \tag{1.2.45}$$

Table 1.3: Values of the global bond orientational order parameter Q_l and the global symmetry parameter W_l for simple crystal geometries. Data extracted from ten Wolde [1998].

	Q_4	Q_6	$\widehat{W}_4$	$\widehat{W}_6$
BCC	0.036	0.511	0.159	0.013
FCC	0.191	0.575	-0.159	-0.013
HCP	0.097	0.485	0.134	-0.012
Liquid	0	0	0	0

which is the Racah formula [Racah, 1942]. The sum in the last term of (1.2.45) is over integers k for which the arguments of the factorials within the sum are positive.

This parameter is referred to as a *symmetry* parameter and can be used to classify the crystal structure of a cluster of particles. The symmetry parameter W_l is a global order parameter, but one can construct the corresponding local symmetry parameter by replacing the $\overline{Q}_{lm_i}$ with $\overline{q}_{lm_i}$ in (1.2.44).

Both $\overline{Q}_{lm}$ and W_l are dependent on the choice of reference frame, so it is useful to construct rotationally invariant combinations of each parameter, thus

$$Q_l \equiv \left(\frac{4\pi}{2l+1} \sum_{m=-l}^{l} |\overline{Q}_{lm}|^2 \right)^{1/2}, \tag{1.2.46a}$$

$$\widehat{W}_l \equiv W_l \left(\sum_{m=-l}^{l} |\overline{Q}_{lm}|^2 \right)^{-3/2}. \tag{1.2.46b}$$

Table 1.3 lists some values of the global order parameters for the crystal geometries discussed in §1.2.3.2. It can be seen from the table that one can take different combinations of the order parameters to distinguish between different types of crystal order of clusters in the system. However, we are only interested in whether something has crystallised or not, so the best order parameter to use is Q_6, since this vanishes in the liquid phase and has relatively high values for all crystal structures, meaning there is clear distinction between the liquid and solid phases.

Global order parameters are useful in molecular dynamics simulations, since they are sensitive to the system becoming completely nucleated, so one can measure the time taken for this to occur. However, we wish to use a Monte

Carlo scheme to sample the entire nucleation barrier, meaning we need a parameter to measure the intermediate stages of nucleation. Thus, we extend the technique of using rotationally invariant global order parameters (1.2.46) to describe the crystallinity of the system to identifying solid-like particles and therefore clusters, giving a measure of the degree to which the system has nucleated. This measure of nucleation is desirable over the global scheme since it is not sensitive to the overall crystal structure of the cluster and simply allows us to determine how many particles have nucleated. We therefore define our local rotationally invariant order parameter for particle i as

$$q_l(i) \equiv \left(\frac{4\pi}{2l+1} \sum_{m=-l}^{l} |\bar{q}_{lm}|^2 \right)^{1/2}, \tag{1.2.47}$$

where $l = 6$, since we previously stated that Q_6 is the best order parameter for distinguishing between liquid and solid phases. It should be noted, however, that the local order parameters $q_6(i)$ are non-zero in the solid **and** liquid phases. The reason that the corresponding global order parameter Q_6 is zero in the liquid phase is because the $q_6(i)$ add up incoherently. Conversely, the $q_6(i)$ add up coherently in the solid phase, giving a non-zero Q_6. Therefore, it is the coherence of the local orientational order parameter that we shall use to identify solid-like particles.

To do this, we follow the method of Auer and Frenkel [2004], who define a normalised $(2l+1)$-dimensional complex vector $\tilde{\mathbf{q}}_6(i)$, which is assigned to each particle i, whose components are

$$\tilde{q}_{6m}(i) \equiv \frac{\bar{q}_{6m}(i)}{\left(\sum_{m=-6}^{6} |\bar{q}_{6m}(i)|^2 \right)^{1/2}}, \tag{1.2.48}$$

where $\bar{q}_{6m}(i)$ is defined in (1.2.41). We then define the dot product of the complex $\tilde{\mathbf{q}}_6$ vectors of neighbouring particles i and j as

$$d_6(i,j) = \mathbf{q}_6(i) \cdot \mathbf{q}_6(j) \equiv \frac{\sum_{m=-6}^{6} \tilde{q}_{6m}(i) \tilde{q}^*_{6m}(j)}{\left(\sum_{m=-6}^{6} |\tilde{q}_{6m}(i)|^2 \right)^{1/2} \left(\sum_{m=-6}^{6} |\tilde{q}_{6m}(j)|^2 \right)^{1/2}}, \tag{1.2.49}$$

where * denotes complex conjugate. We consider particles i and j to be connected if their dot product given by (1.2.49) exceeds a certain threshold d_c.

There does not seem to be a definitive value of d_c in the literature. For example, ten Wolde et al. [1996a] set $d_c = 0.5$. In our simulations, we use the stricter condition of Filion et al. [2010], who set $d_c = 0.7$, since we aim to simulate growth of large crystals, which can only grow from initial coherent nuclei. Furthermore, we are more rigorous in the way we define a solid-like particle. In a solid, almost all the $\tilde{\mathbf{q}}_6(i)$ are in phase with each other and so will add up coherently to produce a non-zero Q_6. However, some particles in a liquid will have at least one bond with a neighbour that is in phase, so these two particles would be considered to be connected according to our specified criteria. Therefore, we must have a secondary threshold of the number of connected neighbours a particle i must have to be identified as solid-like. To this end, we continue to follow the procedure of Filion et al. [2010] and define

$$\xi(i) = \sum_{j=1}^{N_b(i)} H(d_6(i,j) - d_c), \tag{1.2.50}$$

where H is the Heaviside step function and $d_c = 0.7$, so $\xi(i)$ is a count of the number of connected neighbours of particle i that are oriented in crystal order. For particle i to be considered solid-like, $\xi(i) > \xi_c$ must be satisfied. Filion et al. [2010] and Ni [2012] found that the height of the nucleation barrier does not depend on the value of ξ_c, but, for larger values of ξ_c, more particles are identified as non-crystalline. Hence, the critical nucleus size is larger for smaller values of ξ_c, meaning the free energy barrier is less steep and therefore easier to sample. Typical values for ξ_c found in the literature are in the interval $\xi_c \in [5, 9]$, so we set ξ_c to a value in this range. Figure 1.8 shows a 3D visualisation of a HCP crystal structure, which has been colour coded according to the value of $\xi(i)$ for each particle. This image was generated using the Visual Molecular Dynamics software [Humphrey et al., 1996].

Since the creation of large nuclei is a rare event, it is reasonable, particularly for large clusters, to assume that all solid-like particles belong to the same cluster. Using this assumption, we define n_C as

$$n_C = \sum_{i=1}^{N} H(\xi(i) - \xi_c), \tag{1.2.51}$$

which is a measure of the size of crystal nucleus that has formed in the system. The particles i with $\xi(i) \geq \xi_c$ are considered to be *crystalline*.

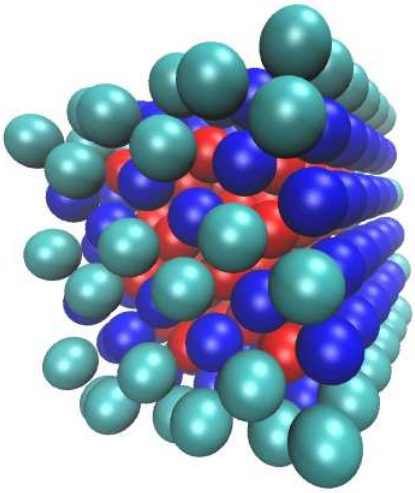


Figure 1.8: 3D image of a 6x6x6 HCP-packed lattice of Lennard-Jones atoms, generated using the Visual Molecular Dynamics software package [Humphrey et al., 1996]. Red atoms are fully crystalline ($\xi(i) = 12$), blue atoms are part crystalline, lying on the surface of the nucleus ($8 \leq \xi(i) \leq 11$) and green atoms are considered to be not part of the nucleus ($0 \leq \xi(i) \leq 7$).

1.3 Simulation techniques

1.3.1 Molecular dynamics vs. Monte Carlo

Simulation of particle interaction is largely carried out using one of two techniques: Monte Carlo (MC) and molecular dynamics (MD) [Allen and Tildesley, 1989]. Both techniques involve a simulation box. A simulation box is defined as a cube of side-length L, which contains all the particles in the system. Carrying out an MD simulation requires us to compute the forces acting on particles in the simulation, thus enabling us to determine the time-dependent behaviour of the configuration. Hence, properties such as the nucleation rate can be calculated. MC, however, is a static simulation which samples the state-space of a system with a Boltzmann distribution to determine its equilibrium properties. An MC simulation does not determine the dynamic evolution of a system, but samples the most likely states. An advantage of MC over MD is that radical, unphysical moves can be proposed that prevent the simulation from getting

"trapped" in a highly dense state, where particles have limited space to escape via physical moves.

The naive approach to simulating the nucleation process by conducting an MD simulation is to supercool the system, i.e. set the temperature to well below the freezing point of the substance, then simply wait for nuclei to appear [Mandell et al., 1976, Tanemura et al., 1977, Cape et al., 1981, Mountain and Brown, 1984]. An advantage of MD is that one can explore the energy landscape of a system, since the simulation is advanced by displacing particles according to Newton's laws of motion, effectively computing gradients in the energy landscape of the system. Auer and Frenkel [2004] have been successful in using MD to predict the crystallisation rates of systems of hard-sphere colloids. Long MC simulations achieve the same effect by proposing random moves of particles to explore the energy landscape, with the risk of proposing moves that are energetically unfavourable, such as a move that places one particle on top of another, which would be instantly rejected. However, we must conduct MC simulations, since we are using a discrete order parameter (1.2.51) to bias the sampling distribution of our simulations (see §1.3.3). We mitigate the poor exploration of the energy landscape by carefully choosing the types of moves that are proposed. In the case of free particle displacement, the maximum move size is tuned to achieve a desirable move acceptance ratio (§1.3.2.4) and in the case of polymer chains of connected particles, we implement a suite of different move types which are designed to explore the energy landscape (§1.4.4).

1.3.2 Metropolis Monte Carlo method

1.3.2.1 Motivation

In the canonical ensemble, we have N particles interacting in a simulation box of volume V at temperature T. The probability of the system being in a microstate A can be written as

$$\rho_A(\mathbf{r}^A, \mathbf{p}^A) \propto \exp(-\beta E(\mathbf{r}^A, \mathbf{p}^A)), \tag{1.3.1}$$

where $\mathbf{r}^A$ are the positions of each particle defined in (1.2.19), $\mathbf{p}^A$ are the momenta, $E(\mathbf{r}^A, \mathbf{p}^A)$ is the total energy of the system in state A and $\beta = 1/k_B T$ is the inverse temperature. The relation (1.3.1) arises from the fact that the states

are distributed according to the Boltzmann distribution derived in §1.2.2.6. The expectation of a quantity X of the system is then given by the weighted average over all possible states

$$\langle X \rangle = \frac{\int\int X(\mathbf{r}^A, \mathbf{p}^A)\rho(\mathbf{r}^A, \mathbf{p}^A)\,\mathrm{d}\mathbf{r}^A\,\mathrm{d}\mathbf{p}^A}{\int\int \rho(\mathbf{r}^A, \mathbf{p}^A)\,\mathrm{d}\mathbf{r}^A\,\mathrm{d}\mathbf{p}^A}. \tag{1.3.2}$$

The total energy can be decomposed into

$$E(\mathbf{r}^A, \mathbf{p}^A) = U(\mathbf{r}^A) + K(\mathbf{p}^A), \tag{1.3.3}$$

where $U(\mathbf{r}^A)$ and $K(\mathbf{p}^A)$ are the potential and kinetic energy respectively. Since MC simulations are static, we can only measure quantities X that are equilibrium properties of the system. Hence, X is time-independent and so only depends on r^A. Therefore, substituting (1.3.3) into (1.3.1), we can simplify (1.3.2), yielding

$$\begin{aligned}\langle X \rangle &= \frac{\int X(\mathbf{r}^A)\exp[-\beta U(\mathbf{r}^A)]\,\mathrm{d}\mathbf{r}^A \int \exp[-\beta K(\mathbf{p}^A)]\,\mathrm{d}\mathbf{p}^A}{\int \exp[-\beta U(\mathbf{r}^A)]\,\mathrm{d}\mathbf{r}^A \int \exp[-\beta K(\mathbf{p}^A)]\,\mathrm{d}\mathbf{p}^A}, \\ &= \frac{\int X(\mathbf{r}^A)\exp[-\beta U(\mathbf{r}^A)]\,\mathrm{d}\mathbf{r}^A}{\int \exp[-\beta U(\mathbf{r}^A)]\,\mathrm{d}\mathbf{r}^A},\end{aligned} \tag{1.3.4}$$

where the denominator is the partition function Z_{NVT} (1.2.33). Recall from §1.2.2.10 that $\exp[-\beta U(\mathbf{r}^A)]$ is the Boltzmann factor, so the quantity X is averaged over the Boltzmann distribution in (1.3.4). A simple approach to evaluating (1.3.4) is to generate some random configurations and compute the value of $X(\mathbf{r}^A)$ along with the corresponding weight $\exp[-\beta U(\mathbf{r}^A)]$. However, for large N this method is inefficient due to the low probability of selecting configurations with large (negative) energies, which are indicative of crystal states. In answer to this, Metropolis et al. [1953] suggested a method in which $\langle X \rangle$ is computed by generating a Markov chain of configurations via a sequence of *Monte Carlo moves*. In the case of free particles, a move consists of randomly displacing a randomly chosen particle. For polymer chains, we impose more sophisticated moves to account for the connectivity of particles (see §1.4.4). Having generated M steps of the Markov chain of configurations, we compute

$$\langle X \rangle = \frac{1}{M}\sum_{i=1}^{M} X(\mathbf{r}_i^A). \tag{1.3.5}$$

As a result, we have a way of sampling the most "important" parts of the phase space according to the Boltzmann distribution and computing the corresponding averaged quantity $\langle X \rangle$.

Before describing the details of the Metropolis Monte Carlo method, we consider two constraints on the simulation.

1.3.2.2 Ergodicity

Firstly, our simulation must be *ergodic*, which means that all possible states of the system are reachable by the simulation via a sequence of Monte Carlo moves. This is important when we define the boundary conditions (§1.3.2.7) and when biasing the distribution of states (§1.3.3), since we cannot completely forbid the system from entering into a state if it is physically possible. Furthermore, we must consider the ergodicity of the simulation when proposing Monte Carlo moves to be sure that all important states are visited during a simulation of reasonable length. In the case of free particle simulation, we monitor the acceptance ratio of particle displacements along with the ratio of proposals between adjacent crystal states n_C, where n_C is computed using (1.2.51). In the case of polymers, the monitoring of ergodicity is more subtle and is covered in §4.2.2.1.

1.3.2.3 Detailed balance

In addition, our MC simulations must obey detailed balance, which is the condition that, for states m and n, the flux of moves from state m to n must be equal to the reverse flux from n to m. That is,

$$P_m \pi_{mn} = P_n \pi_{nm}, \tag{1.3.6}$$

where P_k is the probability of being in state k, and π_{ij} is the transition probability from state i to j. The transition matrix must satisfy (1.3.6) along with the condition

$$\sum_n \pi_{mn} = 1. \tag{1.3.7}$$

The solution suggested by Metropolis et al. [1953] has two cases:

$$\pi_{mn} = \alpha_{mn}, \qquad P_n \geqslant P_m, \quad m \neq n, \tag{1.3.8a}$$

$$\pi_{mn} = \alpha_{mn}(P_n / P_m), \quad P_n < P_m, \quad m \neq n, \tag{1.3.8b}$$

where $\boldsymbol{\alpha}$ is a symmetric stochastic matrix ($\alpha_{mn} = \alpha_{nm}$). The probability of the configuration remaining in the same state is given by

$$\pi_{mm} = 1 - \sum_{n \neq m} \pi_{mn}. \tag{1.3.9}$$

The use of the probability is impractical, since it is hypothetical and can only be approximated by a long run of a simulation. Instead, we use (1.2.36) to express the quantity P_n / P_m in terms of the change in potential energy $\Delta U = U(\mathbf{r}^n) - U(\mathbf{r}^m)$ and reformulate (1.3.8) in terms of $U(\mathbf{r})$ to obtain

$$\pi_{mn} = \alpha_{mn}, \qquad U(\mathbf{r}^n) \leq U(\mathbf{r}^m), \quad m \neq n, \tag{1.3.10a}$$

$$\pi_{mn} = \alpha_{mn} \left(\exp\left[-\frac{\Delta U}{k_\mathrm{B} T} \right] \right), \quad U(\mathbf{r}^n) > U(\mathbf{r}^m), \quad m \neq n, \tag{1.3.10b}$$

$$\pi_{mm} = 1 - \sum_{n \neq m} \pi_{mn}. \tag{1.3.10c}$$

The transition probability (1.3.10a) ensures that moves that are energetically favourable will always be accepted, while (1.3.10b) means that moves that are energetically unfavourable are accepted according to the Boltzmann distribution. Finally, (1.3.10c), which is the probability of a configuration remaining in the same state, guarantees that π_{mn} satisfies (1.3.7).

1.3.2.4 Method outline

The simulation takes place in a cubic box with side-length L. We begin by computing the total energy of an initial configuration of Lennard-Jones particles, U, by summing over all pair-potentials (1.2.40). To implement the method, we define the underlying stochastic matrix $\boldsymbol{\alpha}$. The choice of $\boldsymbol{\alpha}$ is largely arbitrary, as long as $\alpha_{mn} = \alpha_{nm}$ so that detailed balance is adhered to. A proposed move from state m to n is defined as follows. Firstly, a particle i, with position $\mathbf{r}_i^m$, is picked with probability $1/N$ so that each particle has an equal chance of being chosen. We then propose to move the particle to the position $\mathbf{r}_i^n$, where $\mathbf{r}_i^n$ is within the region S, which is centred at $\mathbf{r}_i^m$ and has side-length δr_max. This is described in Figure 1.9 in two dimensions.

Throughout the simulation, the maximum move size, δr_max, is adjusted to control the convergence of the Markov chain, which is measured by the acceptance ratio of particle moves. There is some dispute in the literature over the optimal

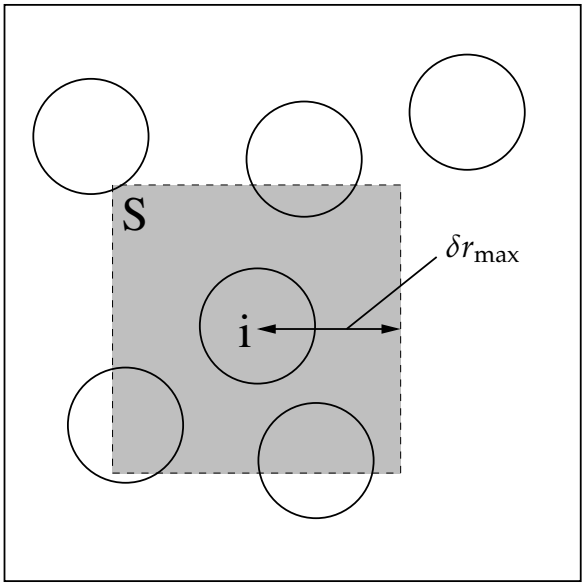


Figure 1.9: State m moves to state n when particle i moves to any point in the shaded region S with equal probability, $1/N_S$.

acceptance ratio, since there is no rigorous method for finding it. ten Wolde et al. [1996a] suggest a value as little as 25%, while many more agree on a value of 50% [van Duijneveldt and Frenkel, 1992, ten Wolde and Frenkel, 1998, Yi and Rutledge, 2009]. Since this is the value favoured by Allen and Tildesley [1989], whose algorithm ours is adapted from, we select 50% as our optimal acceptance ratio.

We define $\boldsymbol{\alpha}$ so that there is an equal probability of proposing to move particle i to every possible new position within S. Due to finite digit precision in a computer simulation, there are a large but finite number of new positions of particle i, N_S, so we can write α_{mn} as

$$\alpha_{mn} = 1/N_S, \quad \mathbf{r}_i^n \in S, \tag{1.3.11}$$

$$\alpha_{mn} = 0, \qquad \mathbf{r}_i^n \notin S, \tag{1.3.12}$$

so we can use (1.3.8) for the transition probability matrix.

Having attempted a move, we decide whether to accept or reject it depending on the relative energies of state m and n. To make this decision, we compute the change in potential energy $\Delta U_{mn} = U_n - U_m$, where U_k is the potential energy of the system in state k. If $\Delta U_{mn} \leqslant 0$, then $P_n \geqslant P_m$ and we use (1.3.10a). Correspondingly, if $\Delta U_{mn} > 0$, then $P_n < P_m$ and we use (1.3.10b). In summary,

moves from state m to n are accepted with probability $P_{\text{ac}}(m \to n)$, where

$$P_{\text{ac}}(m \to n) = \min[1, \exp(-\beta \Delta U_{mn})], \tag{1.3.13}$$

since this captures both the uphill and downhill energy moves.

In practice, when computing ΔU, it is not necessary to recalculate the whole configurational energy of each state, just the changes associated with moving particle i. Thus, we compute ΔU^i_{mn} by summing the energy of particle i with all other $N-1$ particles before and after the move, then find the difference

$$\Delta U^i_{mn} = \sum_{j=1}^{N} u(r^n_{ij}) - \sum_{j=1}^{N} u(r^m_{ij}), \tag{1.3.14}$$

where the sums exclude particle $j = i$; $u(r_{ij})$ is the pair-potential between particles i and j and r^k_{ij} is the length of the vector from particle i to j in state k. Furthermore, we truncate the potential calculation by only considering the explicit interaction of particle i with its neighbours within a cut-off radius r_c, where $r_c < L/2$. This reduces simulation time, since the evaluation of explicit particle interactions is computationally expensive. The contribution to the potential energy from atoms outside r_c is determined using a mean field correction, details of which are given in §1.3.2.6.

1.3.2.5 Pressure calculation

It is useful to compute thermodynamic averages during the simulation, since we can then compute equations of state and compare with experimental results to confirm that our simulations are physically realistic. Since we fix the volume in the canonical ensemble, we compute the pressure by following the method laid out in [Allen and Tildesley, 1989], using the virial theorem in the generalised equipartition form

$$\left\langle q_k \frac{\partial \mathcal{H}}{\partial q_k} \right\rangle = k_{\text{B}} T, \tag{1.3.15}$$

where q_k is a generalised coordinate, $\mathcal{H}$ is the Hamiltonian and the integer k ranges from 1 to $3N$, where N is the number of particles. Choosing Cartesian coordinates and using Hamilton's equations of motion, we see each derivative with respect to q_k in (1.3.15) can be written as the negative of a component of a force $\mathbf{f}_i$ on a particle i. Summing over N particles, where each particle has three

degrees of freedom

$$-\frac{1}{3}\left\langle \sum_{i=1}^{N} \mathbf{r}_i \cdot \mathbf{f}_i^{\text{tot}} \right\rangle = N k_{\text{B}} T, \tag{1.3.16}$$

where $\mathbf{f}_i^{\text{tot}} = \mathbf{f}_i + \mathbf{f}_i^{\text{ex}}$ is the sum of intermolecular forces, $\mathbf{f}_i$, and external forces, $\mathbf{f}_i^{\text{ex}}$, on particle i. The external forces are related to the external pressure p exerted on the system by the container walls

$$\frac{1}{3}\left\langle \sum_{i=1}^{N} \mathbf{r}_i \cdot \mathbf{f}_i^{\text{ex}} \right\rangle = -pV, \tag{1.3.17}$$

where V is the volume of the simulation box. We define the internal virial, $\mathcal{W}$, by

$$\mathcal{W} = \frac{1}{3}\sum_{i=1}^{N} \mathbf{r}_i \cdot \mathbf{f}_i = -\frac{1}{3}\sum_{i=1}^{N} \mathbf{r}_i \cdot \nabla_{\mathbf{r}_i} U, \tag{1.3.18}$$

where U is a potential independent of velocities and times. Thus, substituting (1.3.17) and (1.3.18) into (1.3.16), we obtain

$$pV = N k_{\text{B}} T + \langle \mathcal{W} \rangle. \tag{1.3.19}$$

The first term in (1.3.19) is the contribution to the pressure from the ideal gas limit, while the second is the contribution from particle-particle interactions. Since we are dealing with particle *pair* interactions, we express $\mathcal{W}$ in a form which is independent of the coordinate origin by writing $\mathbf{f}_i$ as the sum of forces $\mathbf{f}_{ij}$ on particle i due to particle j

$$\sum_{i=1}^{N} \mathbf{r}_i \cdot \mathbf{f}_i = \sum_{i=1}^{N}\sum_{j\neq i} \mathbf{r}_i \cdot \mathbf{f}_{ij} = \frac{1}{2}\sum_{i=1}^{N}\sum_{j\neq i} (\mathbf{r}_i \cdot \mathbf{f}_{ij} + \mathbf{r}_j \cdot \mathbf{f}_{ji}), \tag{1.3.20}$$

where the last term follows because the indices i and j are equivalent. An application of Newton's third law $\mathbf{f}_{ij} = -\mathbf{f}_{ji}$ allows us to write

$$\sum_{i=1}^{N} \mathbf{r}_i \cdot \mathbf{f}_i = \frac{1}{2}\sum_{i=1}^{N}\sum_{j\neq i} \mathbf{r}_{ij} \cdot \mathbf{f}_{ij} = \sum_{i=1}^{N}\sum_{j>i} \mathbf{r}_{ij} \cdot \mathbf{f}_{ij}, \tag{1.3.21}$$

where $\mathbf{r}_{ij} = \mathbf{r}_j - \mathbf{r}_i$. Finally, we use (1.3.21) to write the virial, $\mathcal{W}$, in terms of the intermolecular pair virial $w(r_{ij})$ as

$$\mathcal{W} = -\frac{1}{3}\sum_{i=1}^{N}\sum_{j>i} \mathbf{r}_{ij} \cdot \nabla_{\mathbf{r}_{ij}} U(\mathbf{r}_{ij}) = -\frac{1}{3}\sum_{i}\sum_{j>i} w(r_{ij}). \tag{1.3.22}$$

where

$$w(r) = r\frac{\mathrm{d}U(r)}{\mathrm{d}r}, \tag{1.3.23}$$

1.3.2.6 Mean field correction

We briefly mentioned the use of a mean field correction in §1.3.2.4 to reduce the number of pair potentials that need to be explicitly computed having moved particle i. If r_{ij} is the distance between the centres of particles i and j, then we carry out an explicit calculation of the potential if $r_{ij} < r_c$. The contribution from particles for which $r_{ij} \geqslant r_c$ is approximated using the mean field correction. The mean field correction for the potential energy is given by [Allen and Tildesley, 1989]

$$U_{\text{LRC}} = 2\pi N\rho \int_{r_c}^{\infty} r^2 U(r)\,\mathrm{d}r, \tag{1.3.24}$$

where N is the number of particles and $\rho = N/V$ is the bulk number density. The long-range correction for the pressure is given by

$$(pV)_{\text{LRC}} = -\frac{2}{3}\pi N\rho \int_{r_c}^{\infty} r^2 w(r)\,\mathrm{d}r = -\frac{2}{3}\pi N\rho \int_{r_c}^{\infty} r^3 \frac{\mathrm{d}U(r)}{\mathrm{d}r}\,\mathrm{d}r. \tag{1.3.25}$$

Using integration by parts, (1.3.25) can be simplified to

$$(pV)_{\text{LRC}} = U_{\text{LRC}} + \frac{2}{3}\pi N\rho r_c^3 U(r_c). \tag{1.3.26}$$

1.3.2.7 Periodic boundary conditions

A major issue with simulating particle interactions in a box with solid walls is that particles may gather at the surface of the box and experience different forces to those in the bulk. To overcome this we impose periodic boundary conditions, whereby the original box is repeated in all directions, forming an infinite lattice. The central box now has no walls and therefore no surface particles. If a particle moves across a boundary of the box, its image will appear at the opposite face. Figure 1.10 shows a two-dimensional representation of this. We must take care when imposing periodic boundaries, since they have not been found to have any effect on equilibrium thermodynamic properties, as long as the simulation box is sufficiently large and the potential that governs particle correlations is sufficiently short-ranged [Allen and Tildesley, 1989]. For example, we must avoid the situation where a crystal nucleus forms that is similar to the box size, in which case particles will interact with their own images.

A further consideration when imposing periodic boundary conditions is to further truncate the potential calculations in (1.3.14). Since we no longer have

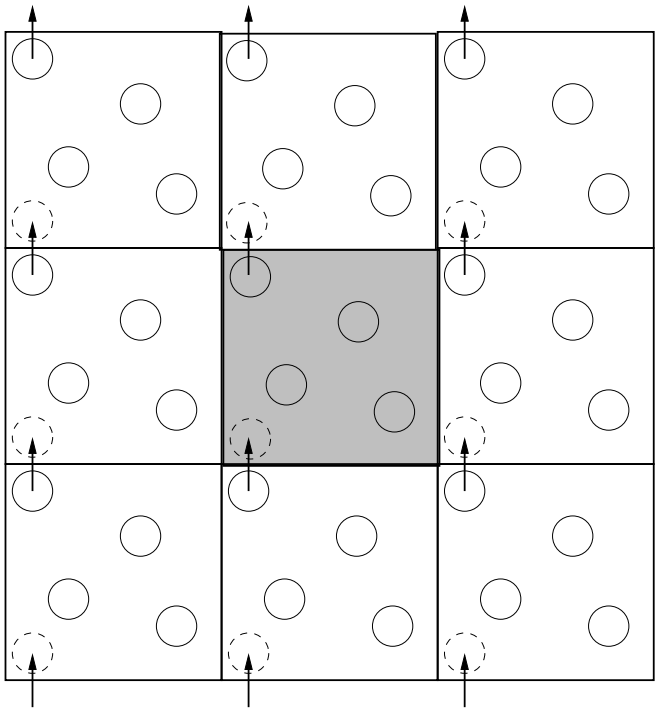


Figure 1.10: Periodic boundaries represented in 2D. Particles can enter and leave each box across each of the four edges.

physical boundaries on the box, we need to consider all pair-potentials of a particle with other particles in the central box, but also all their images, giving an infinite number of terms. We therefore use the minimum image convention, which is illustrated in two dimensions in Figure 1.11. We only consider interactions between particle i and particles contained within the dashed box, which has the same size and shape as the central box and is centred on i. We also impose the cut-off radius r_c as suggested in §1.3.2.4. Furthermore, it is essential to use (1.3.23) to compute the virial, $\mathcal{W}$, in terms of $\mathbf{r}_{ij}$ when imposing periodic boundary conditions, as there is no fixed coordinate origin when using the minimum image convention.

1.3.2.8 Initialisation

The particles in the system are assigned initial positions on a regular cubic lattice, with each particle being far enough apart from its nearest neighbours to be considered liquid by the order parameter described in §1.2.3.3.

An equilibration period, consisting of several (25,000-50,000 depending on sys-

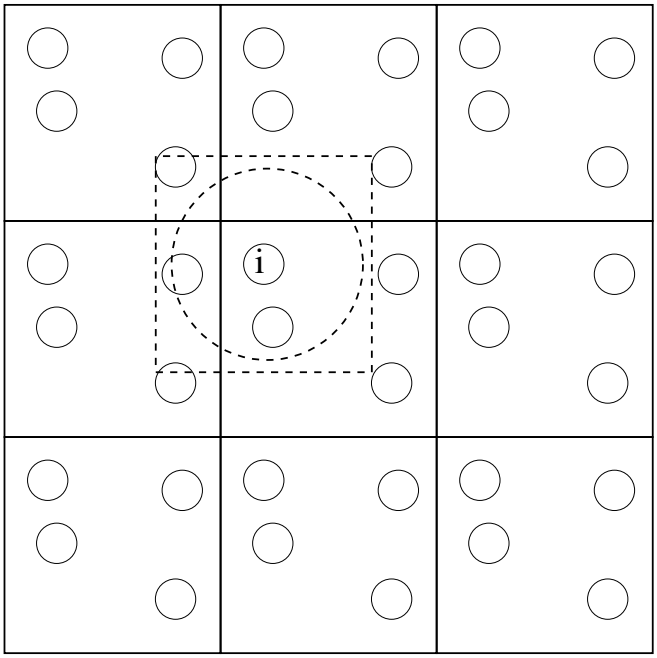


Figure 1.11: The minimum image convention. The central box contains four particles, as does the dashed box centred at particle i. The dashed circle has radius r_c.

tem size) Monte Carlo steps are carried out without collecting any statistics. This has the effect of "mixing" the particles and ensuring that the system is at thermodynamic equilibrium when we start to collect data.

1.3.2.9 Isothermal-isobaric ensemble

We conduct a simulation in the isothermal-isobaric ensemble by making one addition to the work already carried out. No changes need to be made to the boundary conditions or the mean field corrections, but we need to change the type of moves that can be carried out. Particle moves are the same as in the canonical ensemble and are accepted with exactly the same conditions. However, since we are now prescribing a pressure p rather than a volume, we now propose volume moves in order to maintain this pressure. A major advantage of the NpT ensemble over NVT is that we should not see the coexistence of two phases in the same simulation cell (i.e. a snapshot of a simulation), which must be actively avoided by selecting a sensible bulk number density ρ when simulating nucleation in the canonical ensemble.

Using (1.2.38), a configurational average of a quantity X over all possible states in the NpT ensemble is given by

$$\langle X \rangle_{NpT} = \frac{1}{Z_{NpT}} \int_0^\infty \mathrm{d}V \exp(-\beta p V) V^N \int \mathrm{d}\mathbf{s}^A X(\mathbf{s}^A) \exp(-\beta U(\mathbf{s}^A)), \quad (1.3.27)$$

where $\mathbf{s}^A$ is the set of scaled coordinates $\mathbf{s}^A = (\mathbf{s}_1^A, \mathbf{s}_2^A, \ldots, \mathbf{s}_N^A)$ such that

$$\mathbf{s}^A = L^{-1}\mathbf{r}^A, \quad (1.3.28)$$

where $L = V^{1/3}$, assuming that the simulation box is cubic with volume V. As V is no longer constant in this ensemble, we use scaled coordinates. The additional factor of V^N emerges from the volume element $\mathrm{d}\mathbf{r}$ and the fact that there are $3N$ degrees of freedom.

We implement the Metropolis Monte Carlo method in the same way as in §1.3.2.4, except that now the states have a limiting distribution proportional to

$$H = \exp(-\beta(U(\mathbf{s}^A) + pV) + N \log V) \quad (1.3.29)$$

as a result of (1.3.27). Thus, we accept particle moves or volume changes, which move the system from state m to n, with probability

$$P_{\mathrm{ac}}(m \to n) = \min[1, \exp(-\beta \Delta H_{mn})], \quad (1.3.30)$$

where

$$\Delta H_{mn} = \Delta U_{mn} + p(V_n - V_m) - N\beta^{-1} \log(V_n / V_m), \quad (1.3.31)$$

and ΔU_{mn} is given by (1.3.14). The first term of (1.3.31) is the change in potential energy due to the distance between particles changing either as a result of a particle move or because the volume of the box has changed and the distances rescaled. The second term is the work done against the pressure to change the volume and the third term accounts for the change in size of the domain of integration. We use the symbol ΔH_{mn} since the acceptance criteria is closely related to the change in enthalpy of the system as a result of the volume change. Note that if the volume does not change and we only move a particle, $\delta H_{mn} = \delta U_{mn}$ then we recover the same acceptance criteria as in the canonical ensemble. Finally, we remark that volume change moves are more computationally expensive than particle moves. This is because a volume change involves rescaling all particle positions and so we need to recalculate the configurational energy

rather than just some of it as a result of a single particle move. The computational cost of computing the new potential energy, U, after a volume change from state m to n can be significantly reduced when simulating a system of particles that exclusively interact through a pairwise energy, such as the Lennard-Jones potential. We maintain sums of the two terms in (1.2.40) and apply a scaling of the form

$$U_n = \left(\frac{L_n}{L_m}\right)^{-k} U_m, \tag{1.3.32}$$

to each of the terms, where $k = 12$ and $k = 6$ for the terms involving in (1.2.40) r^{-12} and r^{-6} respectively, and L_m and L_n are the side length of the box before and after the volume change, respectively. However, since the computational cost of a volume change is still higher than a particle displacement, we make volume change moves rare compared to particle moves.

1.3.2.10 Free energy barriers

The aim of the simulation described above is to simulate the nucleation of Lennard-Jones particles. There has been extensive work on this in the current literature in both the gas-liquid [ten Wolde and Frenkel, 1998, Vrabec and Hasse, 2002] and liquid-solid transition [van Duijneveldt and Frenkel, 1992, ten Wolde et al., 1996a, Auer and Frenkel, 2005, Filion et al., 2011]. In this thesis, we shall focus on the liquid-solid transition, since we are interested in the crystallisation of particles. In particular, we use the Metropolis Monte Carlo method to map out the nucleation barrier, similar to that predicted by classical nucleation theory in Figure 1.1. The particular free energy we refer to depends on the ensemble we are simulating in. In the canonical ensemble, it is the Helmholtz free energy, while in the isothermal-isobaric ensemble it is the Gibbs free energy.

In our simulations, we record the occupancy of each crystal state, where a crystal state is defined as the number of solid-like particles determined using (1.2.50). We can then use the relation [ten Wolde et al., 1996a]

$$\Delta F(n_C) = \text{constant} - k_B T \log[P(n_C)], \tag{1.3.33}$$

to determine the Helmholtz free energy, $\Delta F(n_C)$, as a function of cluster size, where $P(n_C)$ is the probability that the system has n_C crystalline particles and the constant is chosen such that $\Delta F(0) = 0$. We have used the notation ΔF

instead of F to emphasise that the free energy barrier represents the change in the free energy in moves from the zero-state to state n_C

1.3.3 Sampling rare events using biasing

It was stated in §1.3.2.1 that the Metropolis method is designed such that the system samples the "important" parts of the phase space according the Boltzmann distribution. Since nucleation is an incredibly rare phenomenon, states where there are many solid-like particles are not considered an "important" part of the phase space in a regular Metropolis simulation. Therefore, we need to bias the sampling distribution in favour of crystal states in order to sample the nucleation barrier. Having sampled the rare states in the biased distribution, we must have a method for expressing our results in terms of the original distribution so that they are physically meaningful. This is achieved in the following way [Allen and Tildesley, 1989, Frenkel and Smit, 1996].

Consider the general probability density function for the Boltzmann distribution in the canonical ensemble:

$$P_{\mathrm{o}}(\mathbf{r}) = \frac{\exp(-\beta U(\mathbf{r}))}{\int \mathrm{d}\mathbf{r}\, \exp(-\beta U(\mathbf{r}))}. \tag{1.3.34}$$

We introduce a positive-valued weight function $W(\mathbf{r})$, which is defined at the beginning of the simulation. We now have the general probability density function in the biased distribution

$$P_{\mathrm{w}}(\mathbf{r}) = \frac{W(\mathbf{r})\exp(-\beta U(\mathbf{r}))}{\int \mathrm{d}\mathbf{r}\, W(\mathbf{r})\exp(-\beta U(\mathbf{r}))}. \tag{1.3.35}$$

In practical terms, we modify the move acceptance probability of the unbiased Metropolis Monte Carlo simulation (1.3.13), such that moves from state m to n are accepted with probability

$$P_{\mathrm{ac}}^{\mathrm{w}}(m \to n) = \min\left[\frac{W_n}{W_m}\exp(-\beta \Delta U_{mn})\right], \tag{1.3.36}$$

where U_m and U_n are the values of the weight function in states m and n respectively.

Our objective is to define $\langle X \rangle_{\mathrm{o}}$, which is the average over P_{o} of a quantity of interest, $X(\mathbf{r})$, while sampling in the biased distribution P_{w} in our simulation.

Letting $\langle\cdot\rangle_{\mathrm{w}}$ denote an average over P_{w}, we can relate averages taken in the biased to distribution to the original distribution by

$$\langle X\rangle_{\mathrm{o}} = \frac{\langle X/W\rangle_{\mathrm{w}}}{\langle 1/W\rangle_{\mathrm{w}}}, \tag{1.3.37}$$

whose derivation is given in Appendix C. This amounts to evaluating the ratio of X/W and $1/W$ after each Monte Carlo step, then averaging both quantities over all steps to give the average of $X(\mathbf{r})$ in the original distribution P_{o}.

1.3.4 Parallel tempering

Parallel tempering is a well established method, first suggested by Swendsen and Wang [1986] and thoroughly reviewed by Earl and Deem [2005]. The idea of the technique is to simulate several replicas of the original system of particles at different temperatures, then to attempt exchanges between the replicas at regular intervals. Simulations at high temperatures explore the state-space well, but are unlikely to sample high crystal states. However, the low temperature systems will sample the high crystal states, while exchanges with high temperature systemes remove the risk of being trapped in low-energy states in the time-scale of the simulation, meaning that the overall simulation is ergodic.

Historically, there was slow uptake of parallel tempering, since for M replicas, one requires $\mathrm{O}(M)$ times more computational effort. However, with the development of large CPU clusters, one can simulate different replicas in parallel, leading to a vast improvement in the computational speed.

1.3.4.1 Acceptance criteria for replica swaps

We conduct a parallel tempering simulation with M replicas of the original system, each at a different temperature, with $T_1 < T_2 < \ldots < T_M$, where T_1 is the temperature of interest. The temperatures do not interact energetically, the partition function for this larger ensemble in the canonical ensemble is

$$Z_{NVT} = \prod_{i=1}^{M} \frac{1}{N!} \int \mathrm{d}\mathbf{r}_i \, \exp(-\beta_i U(\mathbf{r}_i)), \tag{1.3.38}$$

where N is the number of particles in each replica, $\mathbf{r}_i$ is the vector of positions of the particles in each replica, $\beta_i = 1/(k_{\mathrm{B}}T_i)$ is the reciprocal temperature and

U is the total potential energy. If the probability of performing a swap is equal for all conditions, then exchanges between ensembles i and j are accepted with probability P_{ac}, where

$$P_{\text{ac}} = \min\{1, \exp[(\beta_i - \beta_j)(U(\mathbf{r}_i) - U(\mathbf{r}_j))]\}. \tag{1.3.39}$$

Using this acceptance criterion, we guarantee that our simulation still obeys detailed balance. Strictly speaking, we should propose swap moves with a certain probability to maintain detailed balance, but it is sufficient to attempt swaps between replicas after a fixed number of Monte Carlo steps [Manousiothakis and Deem, 1999].

Kofke [2002] estimated the acceptance rate of the swaps, $\langle A \rangle$, and argued that this quantity is related to the entropy difference between replicas. Assuming an underlying Gaussian energy distribution in each system (see Figure 1.12), he found the average acceptance ratio to be given by

$$\langle A \rangle = \text{erfc}\left[\left(\frac{1}{2}C_V\right)^{1/2} \frac{1 - \beta_j/\beta_i}{(1 + (\beta_j/\beta_i)^2)^{1/2}}\right], \tag{1.3.40}$$

where C_V is the heat capacity at fixed volume. This shows that the acceptance rate depends on the likelihood that the replica sampling the higher temperature happens to be in the region of state-space that is important at the lower temperature. Given a temperature of interest T_1, this analysis is useful when considering the number of replicas simulated, M, and a sensible range of temperatures, $T_2, \ldots, T_M$. We need to ensure that T_M is large enough that the simulation is ergodic, but also that the T_i are sufficiently close that the acceptance ratio of swaps is sufficiently high.

1.3.4.2 Optimal choice of temperature range

There is much discussion in the literature regarding the optimal choice of a temperature range when conducting a parallel tempering simulation. The most common considerations are detailed in the review by Earl and Deem [2005]. As stated in the previous section, we need T_M sufficiently high that no replicas become trapped in local energy minima, while the number of replicas needs to be large enough to ensure that the acceptance ratio of swaps between adjacent replicas is sufficiently high. The simplest way to illustrate this is to consider

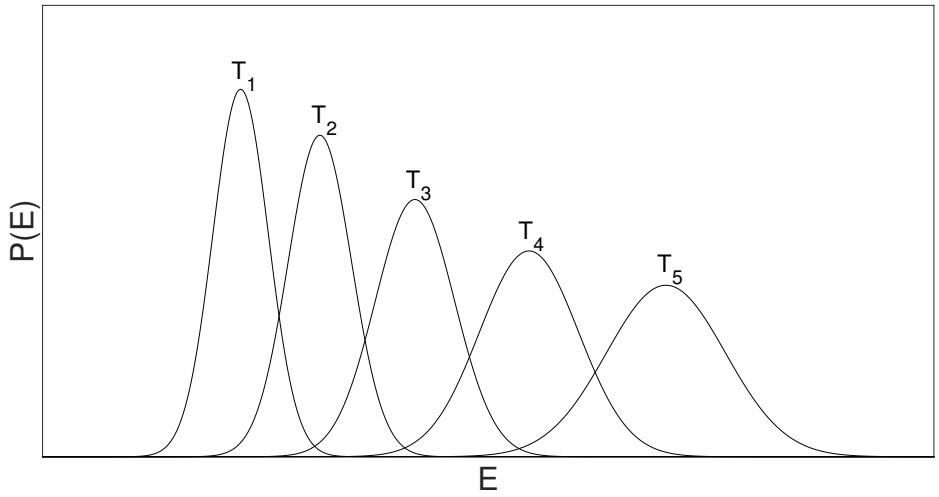


Figure 1.12: Energy histograms for a system at five different temperatures. Overlap of the adjacent histograms allows for acceptance of the configuration swaps.

the energy histograms of each replica, such as those plotted in Figure 1.12. We need enough replicas that the highest temperature has a large average energy, but the temperatures need to be spread so that there is good overlap of adjacent histograms.

Sugita and Okamoto [1999] and Kofke [2002] suggest using a geometric progression (T_i/T_j = constant) for systems in which C_V is constant across the temperatures, since this results in equal acceptance ratios for swaps between adjacent replicas. Using this implementation, each replica spends on average the same amount of time at each temperature. There have been several suggestions for further optimisation of the acceptance ratio of swaps by changing the intervals between each temperature, such as Sanbonmatsu and García [2002], who propose that a target acceptance ratio, A_{target}, be obtained iteratively by solving

$$A_{\text{target}} = \exp[\Delta\beta\Delta E], \tag{1.3.41}$$

where ΔE is the difference in the average energies of the systems in adjacent temperatures. This is similar to adjusting the maximum size of individual particle moves, δr_{max}, to achieve an acceptance ratio of 50%. [Schug et al., 2004, Rathore et al., 2005] have attempted similar iterative schemes for optimising the acceptance ratio and found empirically that an acceptance ratio of 20% al-

lows the simulation to sufficiently explore the available state-space, allowing the simulation to visit high crystal states while remaining ergodic. This means that it is difficult to achieve an acceptance ratio significantly higher than 20%, and doing so does not increase the performance of the simulation in terms of exploration of important states within a reasonable timescale. Furthermore, Kone and Kofke [2005] conducted a formal analysis of the selection of temperature intervals and found that the optimal acceptance rate is 23%, which is similar to the 20% found empirically by Schug et al. [2004] and Rathore et al. [2005].

In this thesis, we have no direct interest in formally optimising the acceptance rate of configuration swaps in our implementation of parallel tempering, we will fix the temperature interval at the beginning of the simulation and monitor the acceptance rate, ensuring it is approximately 20% or higher. To achieve this, we use the approach of Sugita and Okamoto [1999] and Kofke [2002], defining a temperature range $\{T_1, \ldots, T_M\}$, with $T_i/T_j = \text{constant}$.

1.4 Polymers

A polymer is a long connected chain of repeating monomers. A polymer melt is a collection of many long chain molecules. A common model for the generalised polymer is the Rouse model [Rouse, 1953], which describes the polymer as a set of beads connected by harmonic springs (see Figure 1.13). The beads can be modelled as Lennard-Jones particles as in the previous section. Simulating the interaction of polymer molecules has many different length scales. Namely, the distance between connected monomers in the chain, the distance between separate chains or particles further along the chain on the other side of a fold. The displacement of one of the particles in the chain exerts a force on its neighbours, which then exert forces on their neighbours and so on, so that the whole chain is moved along with the original particle.

1.4.1 Motivation for simulating a single chain

A single chain is the simplest polymer problem. Hence, the simulation of a single chain allows us to study the most fundamental problem of nucleation of connected objects.

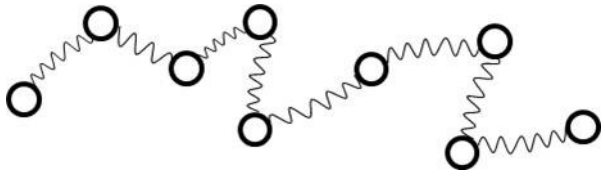


Figure 1.13: The Rouse model of a polymer chain. The polymer is modelled as a set of bead connected by harmonic springs.

Single chains have also been studied in experiments. Rastogi et al. [2005] carefully melted high-molecular-weight polyethylenes (UHMW-PE) at different heat rates to show that the number of entanglements in the melt can be reduced. They showed that it is feasible for crystals composed of single polymer chains to form. By reducing the entanglement, useful properties of the resulting polymer can be enhanced, namely the melt viscosity and drawability. Experiments were also carried out by Weber et al. [2007], who investigated the crystal structured formed by polyethylene nanoparticles. They found that crystallising nanoparticles with a small number of chains (around 14), with extreme undercooling (more than 100°C), forms very thin polymer crystals (approximately 6.3nm). Such crystals would be very useful in materials science, such as for the formation of very thin films and plastic coating.

Experiments were performed by Kawakami et al. [2004] on the polysacchiride dextran (a biomolecule), in which the molecule is subjected to oscillations using an atomic force microscope cantilever while being held under force-clamp conditions. The authors measure the response of the molecule to the forced "unfolding" to obtain measurements for the effective damping and elastic constant. Performing a simulation of a single polymer chain under stretch is analogous with such experiments involving the "unfolding" of biomolecules, and so are worth investigating.

1.4.2 United atoms model

Polyethylene is the simplest example of a polymer, so is often used as a basic model for simulations. Polyethylene consists of hydrogen and carbon atoms, with the chemical formula $(C_2H_4)_n$, illustrated in Figure 1.14. We use the united atoms model, which simplifies the structure of polyethylene by treating each unit of one carbon and two hydrogen atoms as a single bead. Each bead is

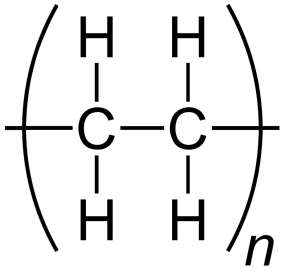


Figure 1.14: Polyethylene repeated unit.

connected to its two neighbours by an extensible rod of typical length l_0, which represent the chemical bonds. The geometry of the bonds between beads in the polymer chain is defined by three variables, l, θ and φ. If bond i connects particles i and $i+1$, we define each variable in the following way:

l: The length of the bond, given by the distance between the centres of particles i and $i+1$.

θ: The complement of the angle between bonds i and $i+1$.

φ: The torsion angle, given by computing the angle between the planes containing particles $\{i-2, i-1, i\}$ and particles $\{i-1, i, i+1\}$. This angle represents the rotation of bond i about its own axis.

The rods have a typical bond angle θ_0. See Figure 1.15 for an illustration of l_0 and θ_0 in the united atoms model. Figure 1.16 shows a physically realistic version of the representation of polyethylene, where the carbon-carbon bonds rotate about their axis. It is assumed that the carbon-hydrogen bonds maintain a fixed bond angle relative to the carbon-carbon bonds. The interaction between the pairs of hydrogen atoms bonded to each carbon atom are summarised by the united atoms model.

1.4.3 Force field

To implement the united atoms model, we must define an appropriate force field for the beads that emulates the chemical interaction between atoms in the physical chain. We shall use the force field originally proposed by Paul et al. [1995], which seems to be the accepted approach in the current literature, such as [Waheed et al., 2002, 2005, Yi and Rutledge, 2009, 2011].

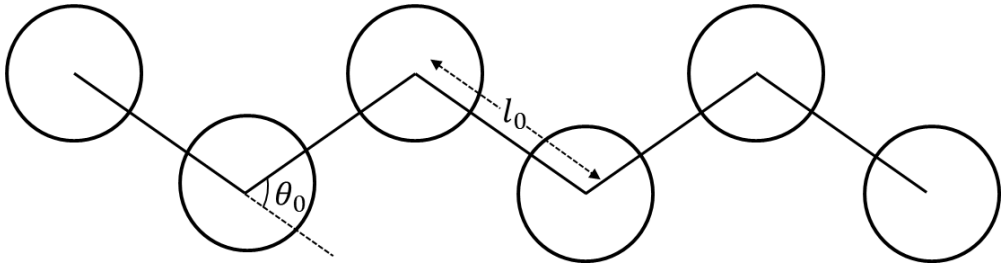


Figure 1.15: United atoms model of a polymer chain. Chains have typical bond length l_0 and typical bond angle $\pi - \theta_0$.

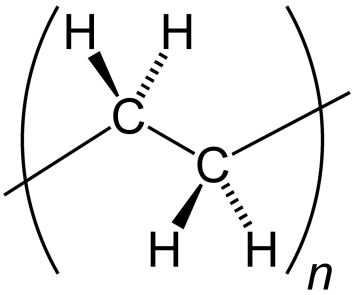


Figure 1.16: Polyethylene repeated unit, showing the effect of rotating the carbon-carbon bonds.

Our approach is to separate the forces between particles into the *bonded* and *non-bonded* interactions.

1.4.3.1 Non-bonded interactions

The non-bonded interactions are equivalent to the particle interactions studied previously in §1.2.3.1. The non-bonded potential applied to neighbouring particles i and j that are not chemically bonded ($j \neq i \pm 1$). Two common models for particle interaction are the Lennard-Jones potential and the square well potential.

1.4.3.1.1 Lennard-Jones potential

As in §1.2.3.1, the Lennard-Jones potential is implemented using

$$U_{\mathrm{LJ}}(r_{ij}) = 4\varepsilon \left[\left(\frac{\sigma}{r_{ij}} \right)^{12} - \left(\frac{\sigma}{r_{ij}} \right)^{6} \right], \tag{1.4.1}$$

where r_{ij} is the distance between neighbouring pairs of particles i and j and ε and σ have the same interpretations as in (1.2.40).

1.4.3.1.2 Square well potential

Taylor et al. [2009a] and Růžička et al. [2012] have modelled particle interactions in a single freely-jointed polymer chain using the square well potential,

$$U_{\mathrm{SW}}(r_{ij}) = \begin{cases} \infty & 0 < r_{ij} < \sigma, \\ -\varepsilon & 1 \leq r_{ij} \leq \lambda\sigma, \\ 0 & r_{ij} > \lambda\sigma, \end{cases} \tag{1.4.2}$$

where r_{ij} is the distance between neighbouring particles i and j, σ and ε are the Lennard-Jones distance and energy parameters, and λ is a measure of the width of the square well (see Figure 1.17). Simulations of the freely-jointed square well polymer chain have been investigated by Taylor and collaborators [Taylor et al., 2009a], [Taylor et al., 2010], [Taylor et al., 2013] and [Taylor and Aung, 2013], who have used several different values of λ in the range $\lambda \in [1.02, 1.5]$. We choose $\lambda = 1.05$, which makes non-bonded interactions sufficiently short-ranged that, if a particle has many neighbours that are close enough to be in the attractive square well, the neighbours are likely to be in crystal order.

Taylor et al. [2009a] have computed a temperature-interaction range (T-λ) phase diagram using Wang-Landau sampling [Wang and Landau, 2001]. By computing the phase diagram, the authors found that, for short-range interactions ($\lambda \leq 1.05$) the chain freezes directly from an expanded coil state to a crystallite. Furthermore, [Taylor et al., 2009b] use the "density of states" function generated from Wang-Landau sampling to compute free energy barriers for chains of length $N = 128$ and $N = 256$ as a function of the total square-well energy. The total square-well potential is obtained by summing over all pair-potentials (1.4.2).

Leitold and Dellago [2014] adapt (1.4.2) by using a smoothed square well potential to conduct molecular dynamics simulations, which require a continuous interaction potential. As previously discussed in §1.3.1, we are using a discrete order parameter, so cannot conduct molecular dynamics. Thus, we do not require the smoothed square well potential and (1.4.2) is sufficient.

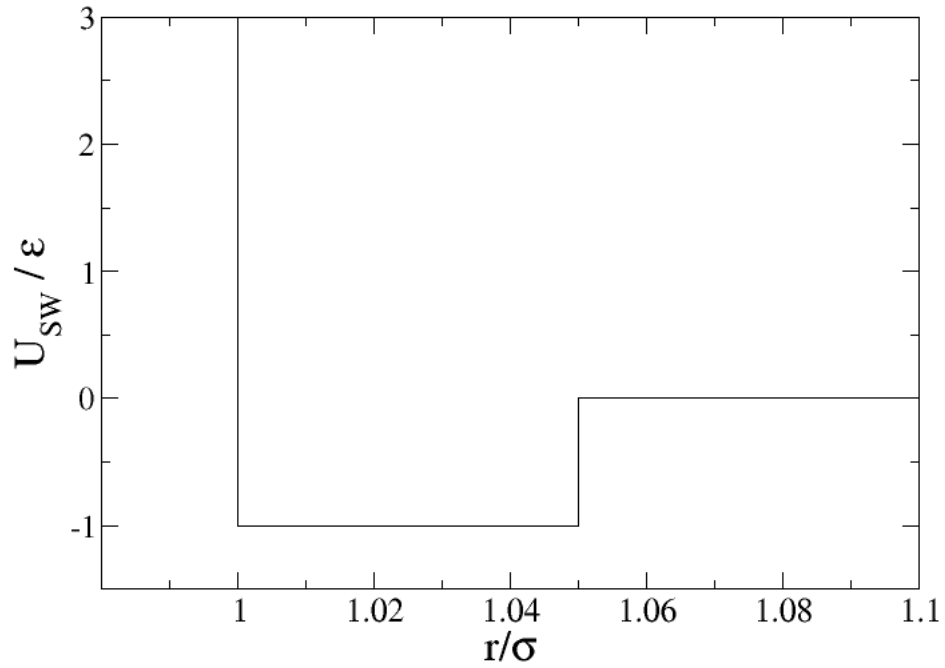


Figure 1.17: The square well potential (1.4.2). Particles repel like hard spheres for $r < \sigma$ and are attractive for $\sigma < r < \lambda\sigma$. Here, $\lambda = 1.05$.

1.4.3.2 Bonded interactions

We define a set of bonded interactions to model the interaction of connected particles in a polymer chain. The bonded interactions have three components, which are described below.

1.4.3.2.1 Bond length

Some authors, such as Taylor et al. [2010], model polymer chains with fixed bond length by exclusively proposing moves that preserve bond length. Others, such as Leitold and Dellago [2014] impose a harmonic spring potential between adjacent beads,

$$U_{\text{bond}}(l) = k_l(l - l_0)^2, \tag{1.4.3}$$

where l is the length of the bond, k_l is the bond stretching constant and l_0 is the unstretched length of the bond. In our simulations, we fix the bond length to $l = \sigma$, matching the interaction length, and so do not apply a bond potential.

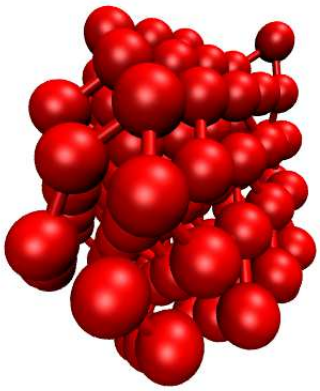


Figure 1.18: Hexagonal close-packed crystal structure formed by the freely-rotating chain. The bond lengths are fixed, but θ and φ are arbitrary.

1.4.3.2.2 Bond and torsion angles

A simple model of a polymer is the **freely-rotating chain**, for which there is no potential imposed on bond angles θ and torsion angles φ. This model allows the formation of tight hexagonal crystal structures in single chain simulations, such as the structure illustrated in Figure 1.18 [Taylor et al., 2009b].

To model stiff chains, we impose a bond angle potential $U_{\text{angle}}(\theta)$ and a torsion angle potential $U_{\text{torsion}}(\varphi)$, where

$$U_{\text{angle}}(\theta) = k_\theta(\theta - \theta_0)^2, \tag{1.4.4a}$$

$$U_{\text{torsion}}(\varphi) = \frac{1}{2}[k_1(1-\cos\varphi) + k_2(1-\cos 2\varphi) + k_3(1-\cos 3\varphi)], \tag{1.4.4b}$$

where k_θ is the angle-bending constant, θ_0 is the preferred complement of the bond angle and the k_i are torsion constants. All constants in (1.4.4) define the stiffness of the chain, and are given appropriate reference values. The torsion potential (1.4.4b) is illustrated in Figure 1.20. Use of the stiff chain is appropriate for modelling nucleation of polymer melts, where adjacent chains stretch out and align to form crystal nuclei [Yi and Rutledge, 2011]. Adaptation of this model for a single chain is difficult, since polymer crystallisation requires the formation of tight folds in the chain to align parallel straight segments, which is energetically unfavourable, given (1.4.4b). A typical crystal structure for a stiff

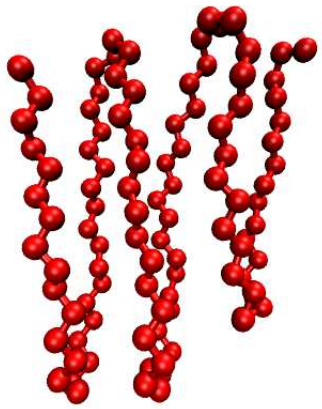


Figure 1.19: Typical crystal structure for a stiff chain, where $U_{\text{angle}}(\theta)$ and $U_{\text{torsion}}(\varphi)$ are included in the potential calculation. The structure involves straight segments between tight folds.

chain is illustrated in Figure 1.19.

In summary, the total potential energy of a polymer chain is given by

$$\begin{aligned} U_{\text{total}} \;=\; & \sum_i \sum_{j>i} U_{non-bonded}(r_{ij}) \\ & + A_1 \sum_i U_{\text{bond}}(l_i) + A_2 \sum_i U_{\text{angle}}(\theta_i) \\ & + A_3 \sum_i U_{\text{torsion}}(\varphi_i), \end{aligned} \tag{1.4.5}$$

where the A_k are parameters that are set to 0 or 1 depending on the polymer model being used. For example, $A_1 = 0$ for fixed bond length and $A_2 = A_3 = 0$ for the freely-rotating chain. A chain containing N particles will have $N-1$ bonds, which corresponds with

- $N-1$ bond energies, $U_{\text{bond}}(l_i)$,
- $N-2$ bond angle energies, $U_{\text{angle}}(\theta_i)$,
- $N-3$ torsion angle energies, $U_{\text{torsion}}(\varphi_i)$.

1.4.3.3 Gauche and trans states

Figure 1.20 shows a plot of (1.4.4b), with reference values for the k_i taken from Yi and Rutledge [2009], who have extracted these values from experimental data

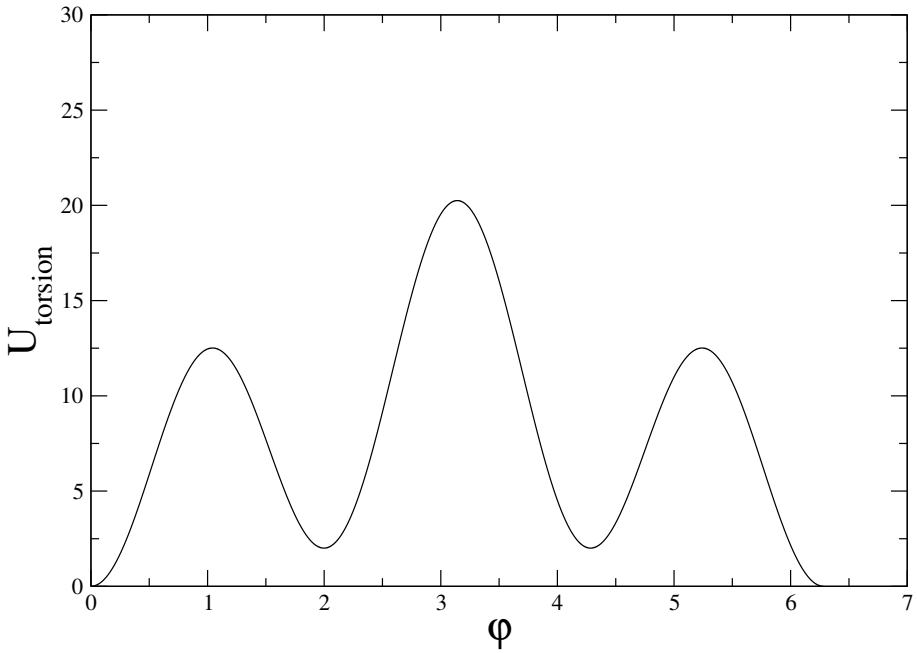


Figure 1.20: A plot of $U_{\text{torsion}}(\varphi)$ for $\varphi \in [0, 2\pi]$, with $k_1 = 6.69$ kJ/mol, $k_2 = -3.63$ kJ/mol, $k_3 = 13.56$ kJ/mol [Yi and Rutledge, 2009].

on short alkanes. These values have been used here to produce a qualitative plot of the bond torsion potential that is physically realistic.

The features that we wish to highlight in Figure 1.20 are the three distinct local minima (the minimum at $\varphi = 2\pi$ is equivalent to the minimum at $\varphi = 0$). These minima correspond to the conformational states of the physical polymer, taking into account all of the individual atoms before applying the united atoms model. A change in the torsion angle φ corresponds to a rotation of the polymer chain about a carbon-carbon bond, which results in a change of conformation. The minima refer to conformation states where the distance between the hydrogen atoms or methyl groups is maximised [Strobl, 2007]. The conformational state that has the lowest potential energy at $\varphi = 0$ is the *trans* conformation. The other two local minima at $\varphi = 2\pi/3$ and $\varphi = 4\pi/3$ are called the *gauche+* and *gauche-* respectively. In terms of our united atoms model, the trans state corresponds to when all three carbon-carbon bonds lie in one plane, while the gauche states are non-planar.

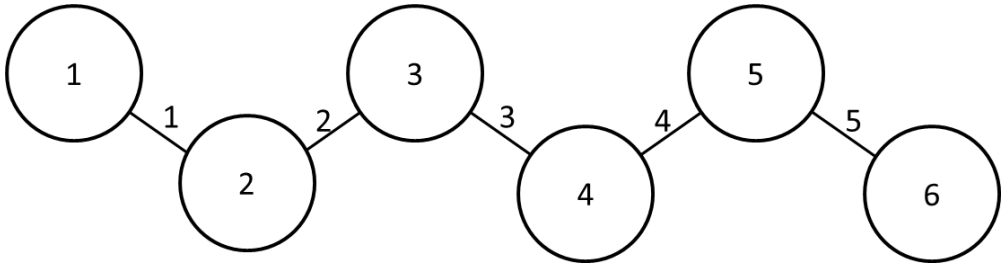


Figure 1.21: Labelling convention for particles and bonds. Particles i and $i+1$ are connected by bond i.

b

1.4.4 Monte Carlo moves for polymers

More care must be taken when proposing Monte Carlo moves for polymer chains than free particles. We must account for the connectivity of the bonded particles and also for the strong bonded potentials that reduce the likelihood of many moves being accepted in the Metropolis algorithm. In addition, since bonds either have fixed length or are very rigid (§1.4.3.2.1), moves that do not preserve bond length are unlikely to be accepted. When simulating long chains, the chain can collapse into an amorphous globule [Taylor et al., 2009a], where the chain-ends are trapped in the core and so are unable to move, causing the simulation to be "trapped". To overcome this problem, we propose a set of move-types, as suggested by Taylor et al. [2009b] and Leitold and Dellago [2014], that allow the chain to escape the collapsed globule.

Our labelling convention for particles and bonds in a polymer chain is provided in Figure 1.21. Using this labelling convention for bonds and particles i, we compute bond lengths l_i, bond angles θ_i and torsion angles φ_i using the formulae described in §1.4.2.

1.4.4.1 Bond pivot moves

To perform a bond pivot move, a bond i, with torsion angle φ_i, is selected at random. Bonds 1 and $N-1$ are discounted from the random draw, as rotating either of these bonds does not change the local configuration of the chain and so does not advance the simulation. A random rotation angle, $\delta\varphi$, is selected uniformly from the interval $[-\delta\varphi_i^{\max}, \delta\varphi_i^{\max}]$. Bond pivot moves preserve the bond

angles, θ_i, which makes pivot moves effective for advancing simulations of stiff chains, since U_{angle} is unchanged by the move. The value of $\delta\varphi_i^{\max}$ is adjusted for each bond i such that the acceptance ratio of bond rotation moves of bond i are 50% on average. We must adjust each of the $\delta\varphi_i^{\max}$ separately, since large rotations of the bonds near the ends of the chain are more likely to be accepted than large rotations bonds near the centre, since there are typically fewer particles to collide with at the ends. In the centre, rotation moves are likely to be rejected due to large changes in the non-bonded potential U_{LJ}, whereas, for stiff chains, the acceptance rotation moves near the ends of the chain will be dominated by changes in the torsion potential U_{torsion}. Dynamically adjusting the $\delta\varphi_i^{\max}$ allows us to propose moves that are likely to be accepted half the time and so optimise the convergence of the simulation.

For stiff chains, we periodically propose rotations of a random bond i by $\delta\varphi_i = \pm 2\pi/3$. These moves are rarely accepted, particularly near the centre of the chain due to particle repulsion, but these moves allow bonds to switch from trans to gauche states (§1.4.3.3). Near the ends, where particles are less likely to overlap, these moves are likely to be accepted, allowing configurations to transition between the local minima of U_{torsion}. Facilitating these switches enables the chain to form folds that can lead to the crystalline structure of Figure 1.19. The simulation can explore the vicinity of each local minimum via the standard angular move $\delta\varphi_i \in [-\delta\varphi_i^{\max}, \delta\varphi_i^{\max}]$. Without these moves, bonds will rarely switch between trans and gauche states, particularly when simulating long, stiff chains, due to particle overlapping being very likely.

1.4.4.2 Reptation moves

Typically, we need to simulate long chains in order to observe crystallisation. If we simulate long chains, it is highly likely that the ends of the chain become trapped in states that correspond with local minima of E_{total}, but are amorphous. Angular moves of the chain ends are generally favourable because they advance the simulation without drastically changing the local configuration of the chain. If the chain ends become trapped, any folds that have formed in the chain persist, meaning that our sampling is heavily biased to a particular minimum in the potential, leading to a loss of ergodicity. Since we wish to gain accurate sampling of the energy landscape of the chain, we need to provide

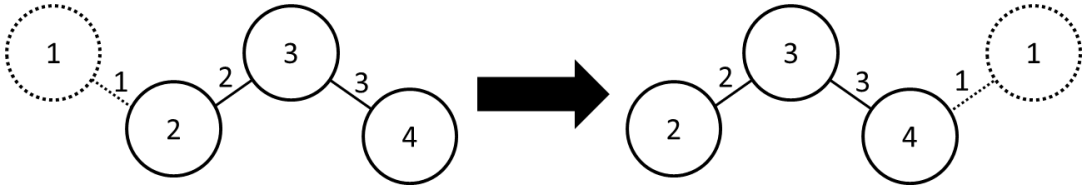


Figure 1.22: Illustration of the forward reptation move in the case $N = 4$. Particle 1 and bond 1 are detached, then reattached to particle N, such that bonds N and 1 lie in the same plane.

a sequence of moves that release the chain-ends from regions of high density, allowing the chain to form new folds and continue resolving the energy landscape. To achieve this, we propose *reptation* moves, which translate the chain along its own length. A long sequence of reptation moves eventually translates folds to one of the ends of the chain, where the angular moves are more likely to be accepted and remove the fold.

To perform a *forward* reptation move, we detach particle 1 and bond 1 from particle 2 in Figure 1.21, then reattach bond 1 to particle N such that the bond angle θ between bonds $N-1$ and 1 is preserved and bonds $N-1$ and 1 are in the same plane (See Figure 1.22). For stiff chains, we rotate bond N by a uniform random angle $\delta\varphi_{\text{rept}} \in [-\pi, \pi)$ to assign a new position to particle 1 while preserving the bond angle between bond $N-1$ and 1. For freely-rotating chains, we randomly place particle 1 on a sphere with radius l_0, centred at the position of particle N. The *reverse* reptation move is proposed by removing bond $N-1$ and particle N, then reattaching particle N to particle 1, placing particle 1 using the appropriate method for stiff or flexible chains as described above. Having completed the reptation move, we relabel all particles and bonds appropriately to recover the labelling convention in Figure 1.21.

1.4.4.3 Moves exclusive to the freely-rotating chain

The following moves are only appropriate when simulating the freely-rotating chain, since they significantly change the bond angles, θ_i, and torsion angles, φ_i. Though these moves can be performed for stiff chains, they are very likely to be rejected due to large, unfavourable changes to U_{angle} and U_{torsion}.

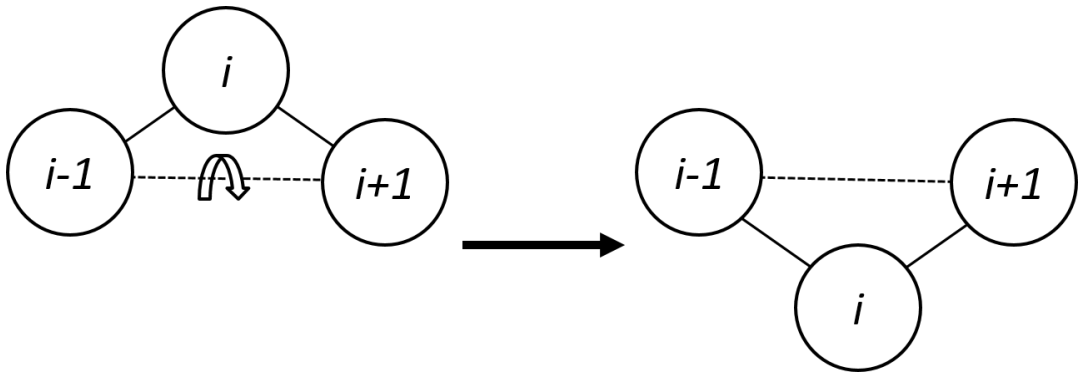


Figure 1.23: Illustration of the crank-shaft move. Particle i is rotated by $\delta\phi_{\text{crank}}$ about the vector connecting particles $i-1$ and $i+1$.

1.4.4.3.1 Crank-shaft moves

We perform a crank-shaft move by choosing a particle at random, discounting particles 1 and N. The randomly selected particle, i, is rotated by $\delta\phi_{\text{crank}}$ about the vector between particles $i-1$ and $i+1$ (see Figure 1.23). The rotation angle, $\delta\phi_{\text{crank}}$, is selected randomly from the interval $[-\delta\phi^{\text{max}}_{\text{crank}}(i), \delta\phi^{\text{max}}_{\text{crank}}(i)]$, where the $\delta\phi^{\text{max}}_{\text{crank}}(i)$ are adjusted so that the acceptance ratio of crank-shaft moves of particle i is 50% on average. Crank-shaft moves are useful for making small changes to the local conformation of the chain by displacing a single particle while maintaining all bond lengths.

1.4.4.3.2 End rotation moves

We randomly select an end particle, i ($i \in \{1, N\}$). Particle i is then rotated about a random axis by a random angle, $\delta\phi_{\text{end}}$. The random angle, $\delta\phi_{\text{end}}$ is chosen randomly from the interval $[-\delta\phi^{\text{max}}_{\text{end}}(i), \delta\phi^{\text{max}}_{\text{end}}(i)]$, where the $\delta\phi^{\text{max}}_{\text{end}}(i)$ are adjusted so that the acceptance ratio of end rotation moves is 50% on average. End rotation moves are the equivalent of crank-shaft moves, but for the end particles. As for crank-shaft moves, end rotations are useful for making small, local changes to the chain configuration.

1.4.4.3.3 End bridging moves

Here, we summarise the detailed description of end bridging moves by Leitold and Dellago [2014]. First, we randomly select and end particle, as in the first

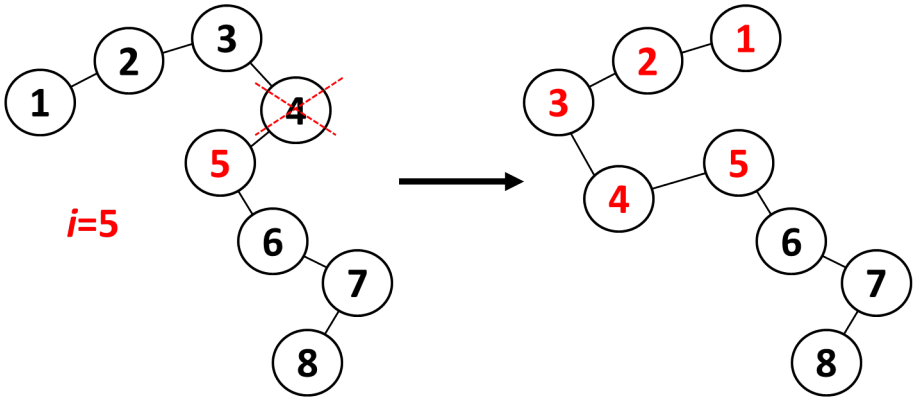


Figure 1.24: Illustration of the end bridge move. Particle 5 has been selected as the bridging partner for end, 1. Particle 5 is re-bridged to end 1 via removal and reinsertion of particle 4. Finally, particles 1 to 3 are relabelled accordingly.

step of an end rotation move. Next, we identify all internal particles ($j \in [3, N-2]$) that are within $2l_0$ of the chosen end, where l_0 is the bond length, and one of these is chosen at random and denoted i. The *bridging partner*, particle i, is then reconnected to the end via removal and reinsertion of the next connected neighbour in the direction of the chosen end, namely particle $i-1$ or $i+1$. This process is illustrated in Figure 1.24.

The effect of the end bridging move is to relabel the particles to change the index of the chain end, while displacing a single particle. The move is particularly useful for exploring highly crystal states, since they allow small modifications of the crystalline core [Leitold and Dellago, 2014]. Furthermore, sequence of successful end bridge moves will improve the probability of accepting reptation moves for highly crystal states where the ends may have become trapped, since end bridging moves alter the indices of the end particles.

To ensure that the end bridging moves obey detailed balance, we accept the moves with probability

$$P_{\text{ac}}(m \to n) = \min\left[1, \frac{b_m R_n}{b_n R_m} \exp(-\beta \Delta U_{\text{total}})\right], \tag{1.4.6}$$

where b_m (b_n) denotes the number of possible bridging partners present in state m (n), R_m is the distance of particle i to the previously selected chain end, R_n is the distance of particle i to the new chain end after the move, and ΔU_{total} is the change in the potential energy (1.4.5) as a result of the move.

1.4.5 Order parameter

To simulate the crystallisation of polymer chains, we clearly need a measure of the crystal order of the particles in the system. Several different order parameters have been suggested in the literature. The appropriate choice of order parameter depends on the type of polymer chain being simulated and its associated force field.

1.4.5.1 Stiff chains

Waheed et al. [2002] suggest two order parameters, one based on density and the other on orientation. To define the density order parameter, the authors use a 3D Voronoi tessellation of space based on atom positions. Voronoi tessellation for a set of points for a discrete set of points in 3D involves dividing up the space into polyhedra centered at each point such that the faces of neighboring polyhedra are equidistant from their corresponding central points. They then follow the algorithm of Gerstein et al. [1995] to compute the volume of the polyhedron, surrounding each atom then invert this to define a density. These densities are then binned in the z-direction, where z is in the direction of the polymer's principle axis, to produce a density profile. They then go on to define $\rho(z)$, a normalised density in the region $(z - \sigma/2)$ to $(z + \sigma/2)$, where σ is the bin width, by convoluting the distribution of densities with a box function, giving the formula

$$\rho(z) = \int_{-\sigma/2}^{\sigma/2} \sum_{i=1}^{n} \frac{V_0}{V_i} \delta(z - z_i - u)\, \mathrm{d}u, \tag{1.4.7}$$

where z_i is the z-coordinate of particle i, V_i is the Voronoi volume associated with particle i, V_0 is the Voronoi volume of a particle in the crystal in its configuration at absolute zero, the value of which can be obtained from classical lattice dynamics situations [Lacks and Rutledge, 1994]. Note that when we set $\sigma = L$, where L is the side-length of the simulation box, we recover the bulk number density for the whole system.

Though we make use of the density parameter, $\rho(z)$, (1.4.7), we shall focus on the orientational order of the particles, as in §1.2.3.3. The question of whether to use a local or global orientational parameter arises. Liu and Muthukumar

[1998], Fujiwara and Sato [1999] comment that a global order parameter is appropriate for simulating the nucleation of short polymer chains and in the later stages of nucleation when entanglement is not an important factor. In general, global order parameters are useful in molecular dynamics simulations, since they are sensitive to when the system has completely nucleated and so can be used as a marker for the time taken to cross the top of the nucleation barrier. The disadvantage is that we do not ascertain much information about the details of the barrier.

We made the argument in §1.2.3.3 that a local order parameter is useful in our simulations in order to track each stage of the nucleation process. Yi and Rutledge [2011] claim that a local orientational order parameter is useful for simulating long, flexible chains. They implement an alignment parameter originally proposed by Lavine et al. [2003] and developed by Yu et al. [2008]. The local order parameter of the ith bead is defined by

$$p_2(i) = \left\langle \frac{3\cos^2\theta_{ij} - 1}{2} \right\rangle_j , \tag{1.4.8}$$

where θ_{ij} is the angle between the vector from the $(i-1)$th bead to the $(i+1)$th bead and the vector from the $(j-1)$th bead to the $(j+1)$th bead. The average in (1.4.8) is taken over all j beads that are within a cutoff radius $r_{ij} < r_{p_2}$, where r_{ij} is the distance between beads i and j. A bead i is considered crystalline if $p_2(i)$ is above a threshold $p_{2\text{th}}$. Further, two crystal beads i and j are considered to be part of the same cluster if the distance between them is less than r_{th} for an appropriate distance threshold r_{th}. We construct the global order parameter

$$P_2 = \left\langle \frac{3\cos^2\theta_{ij} - 1}{2} \right\rangle_{i\neq j} , \tag{1.4.9}$$

where the average is taken over all pairs of beads (i, j), with $j \neq i$, in the system regardless of the distance between i and j.

An alternative measure of local order for multiple chains has been suggested by Yi and Rutledge [2009]. Two chain molecules belong to the same nucleus if they are neighbours and have the same orientation [Esselink et al., 1994]. The chains have the same orientation if the angle between their main axes is less than or equal to 10° and are neighbours if their centres of mass are less than or equal to 1.5σ. One final measure of global order used by Yi and Rutledge [2009] for stiff

chains is the fraction of torsions in the system that are in the trans state. This is appropriate, since almost all the bonds in the ordered structure are in the trans state [Fujiwara and Sato, 1999].

1.4.5.2 Freely-rotating chains

The local and global order parameters (1.4.8) and (1.4.9) are appropriate for determining crystal states of stiff polymer chains, which have crystal structures involving straight segments and tight folds, as illustrated in Figure 1.19. For freely-rotating chains, which can access hexagonal close-packed crystal structures, such as that illustrated in Figure 1.18, we can update the Steinhardt-Nelson (SN) order parameter described for free particles in §1.2.3.3. Leitold and Dellago [2014] suggest the following adaptation of the SN order parameter for a single polymer chain.

First, compute the connection coefficients $d_6(i,j)$ using (1.2.49), where particles i and j are considered to be near neighbours if the distance between them is less than r_c, where r_c is similar to σ, typically 1.05σ. Two neighbouring particles i and j are *connected* if $d_6(i,j) \geq 0.5$. The number of (bonded and non-bonded) near neighbours and the number of connected neighbours of particle i are denoted $N_n(i)$ and $N_c(i)$ respectively. Particle i is defined as *crystalline* if $N_n(i) \geq 5$ **and** $N_c(i) \geq N_n - 1$. This definition allows us to identify particles near the surface of a crystal structure as crystalline, while also preventing us from incorrectly defining particles in the core of a dense, unordered region as crystalline. A particle is defined as *coil-like* if $N_n(i) \leq 4$ regardless of the value of $N_c(i)$, and the particle is *intermediate* if it is neither crystalline nor coil-like.

1.4.5.2.1 Freely-rotating square-well chains

Chains with square-well non-bonded interactions (§1.4.3.1.2) have a discrete energy spectrum $E = -n\varepsilon$, where ε is the depth of the square-well in (1.4.2) and n is the number of square-well overlaps in the chain configuration. Taylor et al. [2009b] propose an $-E$ as an order parameter for freely-rotating chains, as we can expect the number of square-well overlaps to correlated with the number of crystalline particles for high crystal states if the square-well potential is sufficiently short-ranged. This amounts to prescribing a value of λ in (1.4.2)

that is small. Typically values suggested by Taylor et al. [2009b] are $\lambda \leq 1.05\sigma$. We explore this idea in more depth in Chapter 4.

1.4.6 Flow-induced crystallisation

Nucleation of polymers is more difficult to model than that of free particles, due to their highly entangled nature. When a polymer melt crystallises, it transforms from an amorphous liquid state into a semi-crystalline state, which consists of regions of highly-ordered crystalline regions where chains are parallel to one another (crystallites) interspersed with amorphous regions [Graham, 2011]. During industrial polymer processing, the melt is subjected to flow, which significantly encourages nucleation events. This is known as flow-induced crystallisation (FIC). The qualitative effect of the flow is that it stretches the polymer chains in the melt and imposes a common direction for the principle axes of all chains, lowering the entropic penalty of a nucleation event.

The various arrangements and proportions of crystallised and amorphous regions is collectively known as the polymer's morphology. The morphology has a strong effect on the semi-crystalline polymer's physical properties, such as strength, permeability and transparency, so effective modelling of the crystallisation process and the effect of flow is of great interest to the plastics industry.

Several methods for modelling the crystallisation of polymers were reviewed by Graham [2011], who found that an effective model for a polymer melt under fast flow is the Graham-Likhtman and Milner-McLeish (GLaMM) model [Graham et al., 2003], which has been extensively verified against experimental results. Figure 1.25 illustrates the formation of a nucleus in a polymer melt. Our aim is to investigate the effect of stretching a polymer chain in our simulation of polymer nucleation, which is an approximation of the introduction of flow.

1.5 Thesis outline

As stated at the beginning of this chapter, we implement several different modelling and simulation techniques to investigate the effect of flow on the nucleation of polymer chains. In order to do this, we must carefully consider the appropriate conditions for the simulation and how to correctly track the crystal

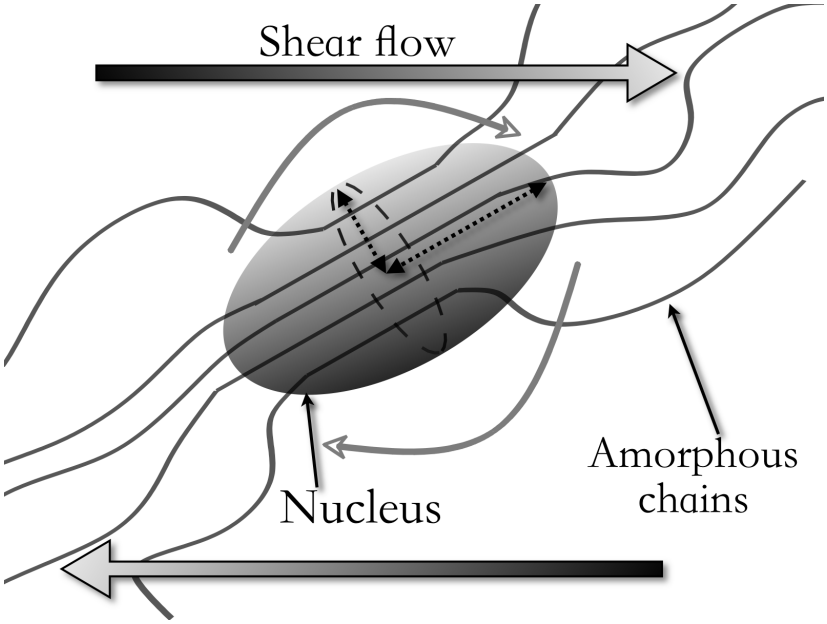


Figure 1.25: Diagram of a nucleus in a semi-crystallised polymer melt with flow, as used in the Graham-Olmsted simulations [Graham and Olmsted, 2009].

state of the system as the simulation evolves.

In Chapter 2, we simulate the nucleation of free particles, in particular using a detailed order parameter to detect the early formation and growth of ordered crystal states. The development of this model allowed us to improve our understanding of the modelling process, with particular regard to sampling techniques employed to correctly monitor the transitions into very rare crystal states located near the maximum of the free energy barrier.

Results from the free particle model are presented in Chapter 3 to illustrate our novel simulation techniques. These results are useful to benchmark our polymer simulation results and to be certain that our method has been implemented correctly to a simplified model.

Chapter 4 concerns the development of a model for a single chain forming nuclei along its own length. This simple model allows us to adapt the model from Chapter 2 to include appropriate moves for connected particles in an infinite simulation box. We implement the parallel tempering algorithm to this model to encourage good exploration of the single chain's state-space. We use our simulation to compute free energy barriers for the square-well chain studied by

Taylor et al. [2009a] and Leitold and Dellago [2014], with a focus on resolving the top of the barrier. We also provide some comparative analysis of different choices of order parameter for polymer crystal states.

In Chapter 5, we stretch the single chain studied in Chapter 4 to approximate flow-induced crystallisation. We observe the effect of stretch on the nucleation barrier, using results from the quiescent case for comparison.

Chapter 6 includes a discussion of the key results and conclusions of the thesis, followed by some ideas for future work in the study of polymer nucleation.

Appendices include further background material and derivations of key formulae to support the main body of the thesis and a user guide for our code, which is available online.

In summary, the work described above investigates a range of simulation techniques for simulating the crystallisation process of polymers. We seek to understand the effect of stretch on the liquid-solid transition and crystal morphology of the resulting nucleated material.

CHAPTER 2

Implementation of simulating the nucleation barriers of free Lennard-Jones particles

In this chapter, we develop an algorithm for simulating the nucleation of Lennard-Jones particles and computing the associated free energy barrier as a function of the crystal state of the system. We use the Metropolis Monte Carlo Method introduced in §1.3.2. We develop several simulation techniques to implement our simulation. We simulate using a biased acceptance rule (see §1.3.3), which leads to a a biased distribution, in which we use sampling windows to resolve specified sections of the state-space, then periodically collate the results. We also implement an adapted order parameter that detects *sub-crystal* states, which are used to achieve good sampling of the transition from the zeroth to first crystal states.

Designing an effective algorithm for simulating the nucleation of free particles is essential for subsequent simulations of polymer nucleation. These simulations involve resolving very steep barriers for large systems of particles, so it is highly advantageous to use a technique designed to accurately and efficiently sample rare crystal states. We achieve good sampling of systems with steep free energy barriers by feeding back the current estimate of the barrier into the biasing function. We can also use the current estimate of the free energy barrier as an initial estimate of the perfect biasing function for another system with similar macroscopic properties, then periodically feed back estimates of the es-

timated barrier to refine the sampling. Our method also takes advantage of the availability of multiple cpu cores to implement several parallel copies of the simulation in different sampling windows so that good sampling of the available crystal states can be achieved in a reasonable time. We have produced the method described above involving the use of sampling windows, with an automated feedback biasing algorithm by combining ideas from several sources, namely Allen and Tildesley [1989], Auer and Frenkel [2004], Filion et al. [2011].

2.1 Simulation details

Particles interact according to the Lennard-Jones pair potential (1.2.40). The simulation is conducted in a finite cube with periodic boundary conditions, as described in §1.3.2.7. We conduct the simulation either in the **Canonical Ensemble (NVT)**, where the number of particles N, the volume of the simulation cube V and the temperature T are constant, or the **Isobaric-Isothermal Ensemble (NpT)**, where N, the pressure p and T are constant. Interactions are computed explicitly for $r_{ij} < 0.8L$, where r_{ij} is the distance between particles i and j, and $L = V^{1/3}$ is the side length of the simulation cube. Particles interact via the Lennard-Jones pair-potential. The total potential of a configuration in state k, $\mathbf{r}^k$ from explicit interactions is obtained by summing over all pair-potentials of particles i and j with $r_{ij}^k < 0.8L$,

$$U_{\text{ex}}(\mathbf{r}^k) = 4\varepsilon \sum_{i=1}^{N} \sum_{j>i} \left[\left(\frac{\sigma}{r_{ij}^k} \right)^{12} - \left(\frac{\sigma}{r_{ij}^k} \right)^{6} \right], \tag{2.1.1}$$

where ε and σ are the energy and length scales for the Lennard-Jones potential (see Figure 1.3).

We compute the contribution to the virial of a configuration from explicit pair interactions by summing over all pair virials (1.3.23). Since $U(r_{ij})$ is given by the Lennard-Jones potential, the pair virial is given by

$$w(r_{ij}) = r \frac{\mathrm{d}U(r_{ij})}{\mathrm{d}r_{ij}} = -48\varepsilon \left[\left(\frac{\sigma}{r_{ij}} \right)^{12} - \frac{1}{2} \left(\frac{\sigma}{r_{ij}} \right)^{6} \right]. \tag{2.1.2}$$

The contribution to the total virial for a configuration in state k from explicit

interactions is computed using (1.3.22),

$$\mathcal{W}_{\text{ex}}(\mathbf{r}^k) = -\frac{1}{3}\sum_{i=1}^{N}\sum_{j>i} w(r_{ij}^k) = 16\varepsilon\sum_{i=1}^{N}\sum_{j>i}\left[\left(\frac{\sigma}{r_{ij}^k}\right)^{12} - \frac{1}{2}\left(\frac{\sigma}{r_{ij}^k}\right)^{6}\right]. \tag{2.1.3}$$

A mean field correction (§1.3.2.6) is used to approximate the contribution to the potential and virial from interactions not explicitly calculated. Substituting the Lennard-Jones potential into (1.3.24) and (1.3.26), we obtain

$$U_{\text{LRC}} = 2\pi N\rho\int_{r_c}^{\infty} r^2 U(r)\,\mathrm{d}r = \frac{8\pi N\rho\varepsilon}{9}\left[\left(\frac{\sigma}{r_c}\right)^9 - 3\left(\frac{\sigma}{r_c}\right)^3\right], \tag{2.1.4a}$$

$$\mathcal{W}_{\text{LRC}} = U_{\text{LRC}} + \frac{2}{3}\pi N\rho r_c^3 U(r_c) = \frac{16\pi N\rho\varepsilon}{9}\left[2\left(\frac{\sigma}{r_c}\right)^9 - 3\left(\frac{\sigma}{r_c}\right)^3\right], \tag{2.1.4b}$$

where ρ is the bulk number density N/V, and r_c is the cuttoff radius, $0.8L$. Combining the explicit and mean field contributions to the potential and virial, we obtain, respectively, the total potential and virial for a configuration in state k,

$$U_{\text{tot}}(\mathbf{r}^k) = U_{\text{ex}}(\mathbf{r}^k) + U_{\text{LRC}}, \tag{2.1.5a}$$

$$\mathcal{W}_{\text{tot}}(\mathbf{r}^k) = \mathcal{W}_{\text{ex}}(\mathbf{r}^k) + \mathcal{W}_{\text{LRC}}. \tag{2.1.5b}$$

The average pressure is computed using (1.3.19), where

$$p = \rho k_{\text{B}} T + \frac{\langle \mathcal{W}_{\text{tot}}\rangle}{V}, \tag{2.1.6}$$

where k_{B} is Boltzmann's constant, and $\mathcal{W}_{\text{tot}}$ is averaged over all Monte Carlo cycles. Particles are initialised in a regular cubic lattice, followed by an equilibration period of 25,000-50,000 Monte Carlo steps before commencing the main simulation.

2.1.1 Canonical ensemble

We reduce temperature and energy using the Lennard-Jones formation [Vrabec and Hasse, 2002]. We use the Lennard-Jones parameters, σ and ε, as our length and energy scales respectively. Thus, we obtain the following nondimensional parameters, where a variable with a '*' is a nondimensional quantity:

$$T = \frac{\varepsilon}{k_{\text{B}}}T^*,\quad U = \varepsilon U^*,\quad L = \sigma L^*,\quad p = \frac{\varepsilon}{\sigma^3}p^*,\quad \mathcal{W} = \varepsilon\mathcal{W}^*,\quad \rho = \frac{1}{\sigma^3}P^*,\quad r_{ij}\sigma r_{ij}^*. \tag{2.1.7}$$

See the previous section for the definitions of each symbol in (2.1.7).

At the start of the simulation, we compute the total potential and virial of the system in the initial configuration. Applying the scales (2.1.7) to (2.1.1) and (2.1.3), we obtain, on dropping the stars, the dimensionless contributions to the potential and virial respectively,

$$U_{\text{ex}}(\mathbf{r}^k) = 4\sum_{i=1}^{N}\sum_{j>i}\left[\left(\frac{1}{r_{ij}^k}\right)^{12} - \left(\frac{1}{r_{ij}^k}\right)^{6}\right], \tag{2.1.8a}$$

$$\mathcal{W}_{\text{ex}}(\mathbf{r}^k) = 16\sum_{i=1}^{N}\sum_{j>i}\left[\left(\frac{1}{r_{ij}^k}\right)^{12} - \frac{1}{2}\left(\frac{1}{r_{ij}^k}\right)^{6}\right]. \tag{2.1.8b}$$

Similarly, the dimensionless mean field corrections for the potential and virial are, respectively

$$U_{\text{LRC}} = \frac{8\pi N\rho}{9}\left[\left(\frac{1}{r_c}\right)^{9} - 3\left(\frac{1}{r_c}\right)^{3}\right], \tag{2.1.9a}$$

$$\mathcal{W}_{\text{LRC}} = \frac{16\pi N\rho}{9}\left[2\left(\frac{1}{r_c}\right)^{9} - 3\left(\frac{1}{r_c}\right)^{3}\right]. \tag{2.1.9b}$$

The total dimensionless potential and virial is computed using (2.1.5), with the dimensionless quantities (2.1.8) and (2.1.9). The dimensionless pressure is

$$p = \rho T + \frac{\langle \mathcal{W}_{\text{tot}}\rangle}{V}, \tag{2.1.10}$$

where $V = L^3$ is the dimensionless volume of the simulation cube.

In order to carry out an MC step in the Metropolis algorithm, we must compute the acceptance probability of a proposed move from state m to n, which is given by (1.3.13),

$$P_{\text{ac}}(m \to n) = \min[1, \exp(-\beta\Delta U_{mn})], \tag{2.1.11}$$

where $\beta = (k_{\text{B}}T)^{-1}$ is the inverse temperature, and ΔU_{mn} is computed using (1.3.14). Substituting the Lennard-Jones potential into (1.3.14) and using the scales (2.1.7), we obtain, on dropping the stars,

$$\beta\Delta U_{mn} = 4\beta\left\{\sum_{j=1}^{N}\left[\left(\frac{1}{r_{ij}^n}\right)^{12} - \left(\frac{1}{r_{ij}^n}\right)^{6}\right] - \sum_{j=1}^{N}\left[\left(\frac{1}{r_{ij}^m}\right)^{12} - \left(\frac{1}{r_{ij}^m}\right)^{6}\right]\right\}, \tag{2.1.12}$$

where $\beta = 1/T$ is the dimensionless inverse temperature. Particle moves are attempted using the criteria outlined in §1.3.2.4 (see Figure 1.9).

2.1.2 Isothermal-isobaric ensemble

In the NpT ensemble, we use the same scalings as NVT (2.1.7) and accept moves with probability $\min(1, \beta\Delta H_{mn})$, where

$$\beta\Delta H_{nm} = \beta\Delta U_{nm} + \beta p(V_n - V_m) - N\log(V_n/V_m), \tag{2.1.13}$$

with ΔU_{nm} having the definition given by (2.1.12) (see §1.3.2.9 for a derivation of this). We attempt particle moves using the same criteria as in the NVT ensemble. In addition, we attempt volume change moves in order to maintain the imposed pressure. Particle moves and volume change moves are attempted with ratio 9 : 1 respectively. Volume moves are attempted much less often than particle moves, since they are computationally more expensive.

2.2 Order parameter

We define the crystal order of a configuration in our simulation using the Steinhardt-Nelson order parameter. For a configuration, $\mathbf{r}$, the number of crystalline particles is given by

$$n_C = \sum_{i=1}^{N} H(\xi(i) - \xi_c). \tag{2.2.1}$$

Here, H is the Heaviside step function, $d_c = 0.7$ is the dot product threshold, $\xi(i)$ is the number of near-neighbours of particle i that are arranged in crystal order, and ξ_c is the minimum number of crystal neighbours required for particle i to be considered crystalline. See §1.2.3.3 for a detailed description of how $\xi(i)$ is computed.

2.2.1 Subdividing the zero-state

We have found that using (2.2.1) with $\xi_c = 8$ to define crystal particles results in the identification of well-ordered crystal states for relatively small n_C. This is advantageous, since we bias in favour of any observed crystal states, increasing the likelihood of generating a sequence of configurations that grows the crystal nucleus. However, setting $\xi_c = 8$ results in very low rates of attempted moves from the zero-state ($n_C = 0$) to the one-state ($n_C = 1$). This is because

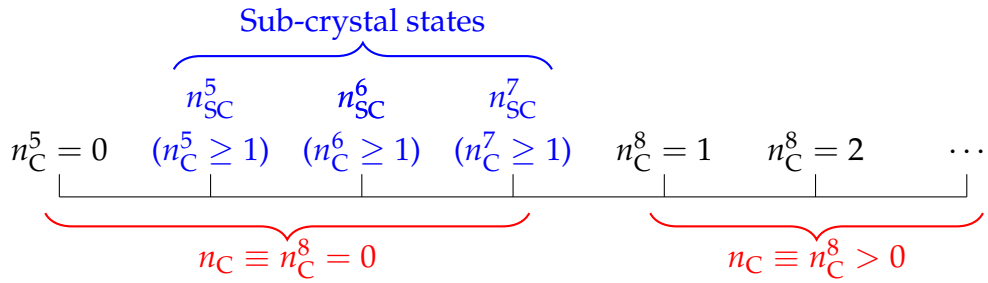


Figure 2.1: Schematic of the labelling and ordering of the sub-crystal states, N_{SC}^5, N_{SC}^6, N_{SC}^7 in relation to the genuine crystal states, n_C.

the conditions for one particle to be considered crystal-like are very strict. As a result, simulations can become "locked" in the zero state, giving uneven sampling of the available states. We require the particle to have 8 near-neighbours $N_b(i) \geq 8$ and for each of the neighbours to be aligned with i and each other such that $\xi(i) \geq 8$, which is incredibly unlikely. To overcome this difficulty, we subdivide the zero-state by generalising (2.2.1) to

$$n_C^q = \sum_{i=1}^{N} H(\xi(i) - q), \tag{2.2.2}$$

where q is the number of crystal neighbours a particle requires to be considered crystal-like, i.e. the number of particles i with $\xi(i) \geq q$. Using (2.2.2), we can define *sub-crystal* states, which represent pathways to our definition of the first genuine crystal state (one particle i with $\xi(i) \geq 8$. We find that it is sufficient to define a sub-crystal state as the existence of at least one particle i with $\xi(i) = q$, with no further subdivision necessary. This allows us to identify configurations where crystal structures have partially formed, then use biasing to preferentially sample these states (see §2.3). Using only the original parameter, n_C^8 in the notation of (2.2.2), these states would have been ignored resulting in a loss of valuable information. We use the shorthand n_{SC}^q to denote the sub-crystal state where the configuration has $n_C^q \geq 1$. See Figure 2.1 for an illustration of this notation and the ordering of states with the addition of sub-crystal states. To compute the free energy barrier in the same terms as the original problem, we sum the occupancies of the zero-state and sub-crystal states and treat this as one state and use (1.3.33) to compute the free energy barrier.

We typically only need to define sub-crystal states with $5 \leq q \leq 7$, though this can be tuned for each simulated system by monitoring the proportion of all upward move proposals from the zero state. To gain reasonable statistics of

the zero-to-one transition, an acceptable attempted move ratio is 1%. We have found that no further subdivision is necessary once the system has reached the first genuine crystal state, as the attempted move ratio between subsequent states is already acceptable ($\geq 1\%$).

For ease of notation, we shall henceforth refer to crystal states using n_C, as defined by (2.2.1), and sub-crystal states using n_{SC}^q, where $n_{\mathrm{C}}^q \geq 1$ in (2.2.2) (see Figure 2.1).

2.3 Biasing

We introduced the process of biasing in a Metropolis Monte Carlo simulation in §1.3.3 as a tool for obtaining good sampling of rare states. We sample in the biased distribution, which is defined by the probability density function (C.1.3). We must choose a suitable weight function $W(n_{\mathrm{C}})$ to achieve the best possible sampling of states. We write W as a function of n_{C} rather than a function of $\mathbf{r}$, the coordinates of all particles in the configuration, since n_{C} is computed using $\mathbf{r}$ (§1.2.3.3). We sample in the biased distribution by modifying (2.1.11) so that moves from state m to n, with order parameters n_{C}^m and n_{C}^n respectively, are accepted with probability

$$P_{\mathrm{ac}}^{\mathrm{w}}(m \to n) = \min\left[1, \frac{W(n_{\mathrm{C}}^m)}{W(n_{\mathrm{C}}^n)} \exp(-\beta \Delta U_{mn})\right]. \tag{2.3.1}$$

2.3.1 Feedback algorithm for biasing states

The most effective way of sampling the state-space of a system is to achieve uniform sampling of all available states in the biased distribution. If the system spends an equal amount of time in each state, we gain good statistics for computing the true transition probabilities between states and the corresponding free energy. To achieve this uniform sampling, the appropriate weight function $W(n_C)$ is related to an estimate of the true free energy barrier given by (1.3.33),

$$\Delta F(n_C) = \text{constant} - \log P_{\mathrm{o}}(n_C), \tag{2.3.2}$$

where $P_{\mathrm{o}}(n_C) = \langle P(n_C) \rangle_{\mathrm{o}}$ is the average probability of the system being in state n_C in the unbiased distribution. The constant in (2.3.2) is chosen such that

$\Delta F(0) = 0$, so that (2.3.2) can be written as

$$\Delta F(n_C) = \log\left[\frac{P_o(0)}{P_o(n_C)}\right]. \tag{2.3.3}$$

We nondimensionalise ΔF using k_B, where T is the **dimensional** temperature. Hence, $k_B T$ has units of energy and

$$\Delta F = k_B T \Delta F^*, \tag{2.3.4}$$

where ΔF^* is the dimensionless free energy. Henceforth, we drop the '*' and refer to the dimensionless free energy as ΔF.

In a biased simulation, we log the occupancies of crystal states n_C to estimate $P_w(n_C) = \langle P(n_C)\rangle_w$, the probability of the system visiting state n_C in the biased distribution. We then use (1.3.37) to obtain $P_o(n_C)$, the corresponding probability of visiting state n_C in the unbiased distribution, such that

$$P_o(n_C) = \frac{\langle P(n_C)/W(n_C)\rangle_w}{\langle 1/W\rangle_w}, \tag{2.3.5}$$

where $\langle\cdot\rangle_w$ indicates an average taken over the biased distribution. Having obtained $P_o(n_C)$, we use (2.3.3) to estimate the free energy of the system as a function of the crystal states n_C. A suitable weight function that enables good sampling of all states is defined using the estimate for $\Delta F(n_C)$,

$$W(n_C) = \exp\left[U_{\text{bias}}(n_C)\right], \tag{2.3.6}$$

where

$$U_{\text{bias}} = \Delta F(n_C). \tag{2.3.7}$$

Writing $W(n_C)$ in the form (2.3.6) allows us to simplify (2.3.1),

$$P_{\text{ac}}^w(m \to n) = \min[1, \exp(\Delta U_{\text{bias}}^{mn} - \beta \Delta U_{mn})], \tag{2.3.8}$$

where $\Delta U_{\text{bias}}^{mn} = U_{\text{bias}}^n - U_{\text{bias}}^m$ is the change in U_{bias} in moving from state m to n. Clearly, we do not have an estimate for P_o at the start of the simulation, so we execute the following algorithm.

1. *Equilibration*: we initialise the system with the particles in a regular cubic lattice and perform a sequence of 5×10^4 Monte Carlo moves without collecting any statistics to mix the particles. We use $W(n_C) = 1$ as the initial biasing function, meaning that the biased and unbiased distributions are equivalent.

2. We begin collecting statistics for a prescribed interval (typically at least 10^6 Monte Carlo steps), with $W(n_C) = 1$ to obtain an initial guess for $\Delta F(n_C)$ using (2.3.3).

3. Update the biasing function, $W(n_C)$, using (1.3.37) for states that have occupancies that are sufficiently large that we can rely on the statistics collected from them. We simulate systems for which the free energy is at least initially an increasing function of n_C, so we expect to evaluate $\Delta F(n_C)$ for $n_C \leq n_C^{\max}$. For the remaining states, the biasing function is updated using $\Delta F(n_C^{\max})$. This is summarised by

$$U_{\text{bias}}(n_C) = \begin{cases} \Delta F(n_C), & n_C \leq n_C^{\max}, \\ \Delta F\left(n_C^{\max}\right), & n_C > n_C^{\max}. \end{cases} \tag{2.3.9}$$

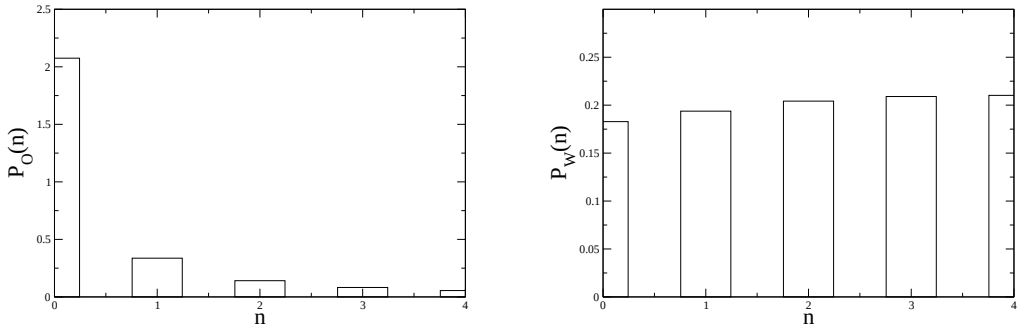
4. Reset the counters for the state occupancies and collect statistics in the biased distribution with the updated weight function for the same prescribed interval as in step 2.

5. Repeat steps 3 and 4 until the distribution of occupancies of available states is approximately uniform (within 30%). If the biased distribution is approximately uniform, this is an indication that our current estimate of $\Delta F(n_C)(= W(n_C))$ is a good estimate for the true free energy.

Successive iterations of the feedback step (3) results in more attempted moves to the states that were previously very rare. Consequently, $n_C^{\max}$ increases with each iteration, eventually allowing us to bias all available states and gain good statistics of the entire state-space. On completion of the feedback algorithm in step 5, we fix the biasing function with the latest approximation of $\Delta F(n_C)$ and execute a long simulation to gain a reliable estimate of the free energy barrier for the system. An implementation of the feedback technique is illustrated in Figure 2.2.

2.3.2 Sampling windows

Implementing the feedback algorithm of §2.3.1 results in a sampling scheme where each state is visited with equal probability in the limiting distribution.



(a) Original distribution

(b) Biased distribution after completion of the feedback algorithm

Figure 2.2: The feedback algorithm applied to an exemplar simulation in the NVT ensemble, with $N = 20$, $\rho = 0.3$, $T = 0.7$, where we monitor the first 5 states. After several iterations of the feedback algorithm, we have achieved approximately uniform sampling of these states.

Effectively, this reduces the simulation to a diffusion problem, which is preferable when dealing with small systems ($N \approx 20$), but causes problems when dealing with larger, more physically realistic systems. This is particularly a problem when using the feedback algorithm to estimate the free energy for all states, since the system will spend the majority of the simulation time diffusing between the states for which the free energy has already been resolved, with attempted moves into the desired higher crystal states being rare.

To solve this problem, we employ a modified version of umbrella sampling [Torrie and Valleau, 1974], running the simulation in a sampling window which targets a small number of states, $N_{\min} \leq n_C \leq N_{\max}$. Since we use the simulation to collect statistics on the relative occupancy of states to compute the free energy barrier, we sample different parts of the state-space using various windows independently, then construct the complete barrier retrospectively. The sampling window is defined such that we run a simulation only collecting statistics when $n_C \in [N_{\min}(w), N_{\max}(w)]$, where $N_{\min}(w)$ and $N_{\max}(w)$ are respectively the lower and upper limits of the window and w is the window number. To implement the sampling windows, we make further use of biasing. We now define the weight function in window w to be $W(n_C, w) =$

$\exp(U_{\text{bias}}(n_C, w))$, with

$$U_{\text{bias}}(n_C, w) = \begin{cases} -\kappa(w)(n_C - N_{\min}(w))^2 + \Delta F(n_C), & n_C < N_{\min}(w), \\ \Delta F(n_C), & N_{\min}(w) \leq n_C \leq N_{\max}(w), \\ -50, & n_C > N_{\max}(w), \end{cases} \tag{2.3.10}$$

where $\kappa(w)$ is a parameter than can be tuned to control the number of visits to states with $n_C < N_{\min}(w)$.

The $N_{\min}(w)$ and $N_{\max}(w)$ are chosen such that adjacent windows contain at least one common state. We define an overlap, ϵ, where

$$\epsilon = N_{\max}(w) - N_{\min}(w+1) + 1. \tag{2.3.11}$$

We wish to allow some visits to outside the window, particularly to the zero-state in order to maintain the ergodicity of the system. If we regularly allow the system to escape the window and return to the zero state, any crystals that have formed will melt. The biasing will encourage rapid growth of a new cluster to return the system to the sampling window and continue collecting statistics. Over the course of a long simulation, the most likely crystal cluster will be sampled most often in each sampling window. This is desirable, since we do not want one or more of the sampling windows get stuck in a potential well caused by sampling an unlikely crystal structure, causing the window in question to decorrelate from results collected from the rest of the windows. For the upper limit, we do not have this issue, so we set U_{bias} to a large constant so that any proposed moves to states $n_C > N_{\max}(w)$ are rejected.

In each window, we evaluate the free energy using (2.3.2). Recall that ΔF is a measure of the *change* in free energy of moving between states, so we are free to choose the constant such that $\Delta F(N_{\min}) = 0$ in each window, such that

$$\Delta F(n_C, w) = \log\left[\frac{P_o(N_{\min}(w))}{P_o(n_c)}\right]. \tag{2.3.12}$$

Details of how we construct the barrier using the estimates from independent windows, (2.3.12), are provided in §2.5.

2.4 Additional equilibration period

In §2.3, we mentioned that the system initially undergoes an equilibration period, whereby several Monte Carlo steps are carried out without collecting any statistics, which has the effect of "mixing" the particles. With the introduction of the new order parameter described in §2.2.1, we found it useful to conduct a further equilibration period when sampling in the window for which $N_{\text{min}} = 0$. This equilibration period is used to execute the feedback algorithm for all available crystal and sub-crystal states in the window. After this equilibration has taken place, the $n_C = 0$ to $n_C = 1$ transition has been sufficiently resolved so that good statistics can be collected for the first few crystal states immediately. Although this equilibration is not strictly necessary, it is useful for quickly gaining interesting data, particularly for systems where the free energy barrier is initially very steep, i.e. $\Delta F'(n_C) \gg 1$ for $n_C = 0, 1, 2, \ldots$.

Since the system is initialised in the zero-state, with all particles arranged on a regular cubic lattice (§2.3), for windows, w, with $N_{\text{min}}(w) \gg 0$, we found it necessary to extend the equilibration period so that the system has entered the sampling window before statistics are collected. The length of the equilibration period can be extended dynamically by monitoring the number of visits to state $n_C = N_{\text{min}}(w)$ and beginning the main simulation when the occupancy exceeds a specified threshold, which we have taken to be 10^3.

2.5 Pseudo-parallelisation of the simulation

Since we have divided the work required to construct a complete free energy barrier into several windows in §2.3.2, it appears that more computation time must be allowed to execute the simulation. If this is the case, the time taken would be comparable with not implementing the windows and just waiting for the original simulation to resolve the barrier. However, we note that simulations in each sampling window are independent of each other and use computers with several cpu cores. We initialise all sampling windows separately and run them in parallel on one cluster, regularly outputting information regarding the occupancy of states. We then combine the data to construct the complete free energy barrier.

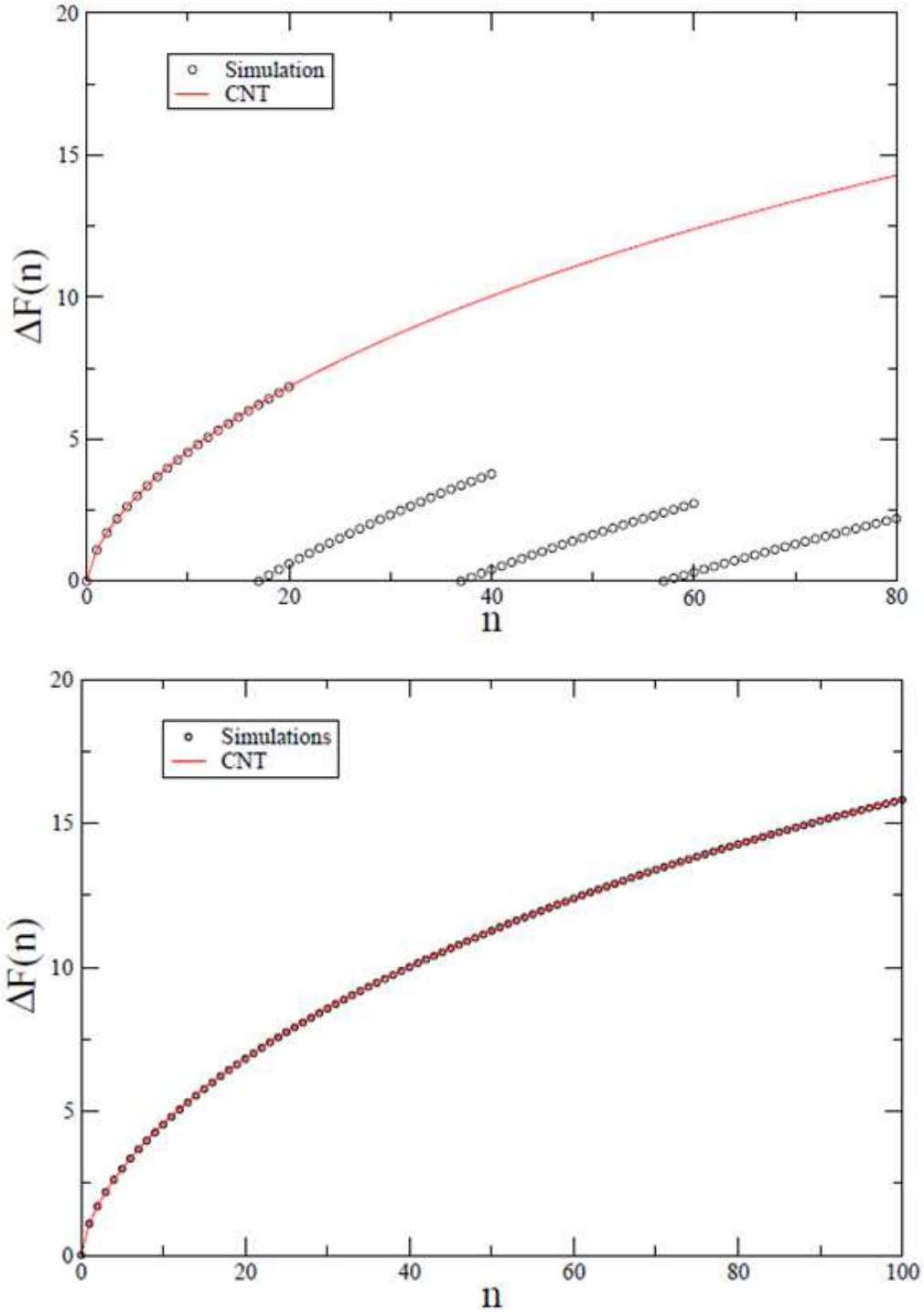


Figure 2.3: We compute the free energy barrier in four independent simulations (top), then use the results to construct the complete barrier (bottom).

This process is illustrated in Figure 2.3, where we have implemented the parallel sampling windows to a simple simulation that replicates classical nucleation theory (§1.2.1). The continuous curve has been computed using (1.2.2), with $A = 0.1$, $B = 1.2$,

$$\beta\Delta F(n) = -0.1n + 1.2n^{2/3}, \tag{2.5.1}$$

where $\beta = 1/T$ is the inverse temperature and n is the cluster size. The simulation is carried out in one dimension, initialised in state $n = 0$. We randomly propose upwards and downward moves in n, with moves from state n_1 to n_2 being accepted with probability

$$P_{\text{ac}}(n_1 \to n_2) = \min\{1, \exp[-\beta(\Delta F(n_2) - \Delta F(n_2))]\}. \tag{2.5.2}$$

Boundary conditions are imposed by rejecting downward moves from state $n = 0$ and upward moves from state $n = 80$. We sample using 4 independent sampling windows, with biasing function (2.3.10), using the feedback algorithm in §2.3.1 to update the biasing function with the latest estimate of the free energy. The window limits $N_{\min}(w)$ and $N_{\max}(w)$ are chosen such that the 4 windows span $n \in [0, 80]$ with overlap $\epsilon = 5$ (2.3.11). This is essential, since to construct the complete barrier from the partial barriers resolved in each window, we need at least one state which is in two adjacent windows.

The first window in Figure 2.3, $n \in [0, 20]$, corresponds to the complete barrier, so we move in the positive n direction from here to construct the complete barrier. We shift the estimate of $\Delta F(n)$ in each of the remaining windows by a constant $k(w)$, such that the absolute difference in the estimates of $\Delta F(n)$ for the ϵ overlapping states of adjacent windows w and $w-1$, $\delta F_\epsilon(w)$, is minimised, where

$$\delta F_\epsilon(w) = \sum_{i=1}^{\epsilon} |\Delta F_{w-1}(i) - (\Delta F_w(i) + k(w))|, \tag{2.5.3}$$

and $\delta F_w(i)$ is the estimate of ΔF for state i from window w. Minimising (2.5.3) with respect to $k(w)$, we obtain

$$k(w) = \frac{1}{\epsilon}\sum_{i=1}^{\epsilon}(\Delta F_{w-1}(i) - \Delta F_w(i)). \tag{2.5.4}$$

Applying (2.5.4) to each of the adjacent windows yields the complete free energy barrier. Note that $k(1) = 0$, since the first window corresponds with the original free energy barrier. For each new window, moving in the positive n

direction, we accumulate the $k(w)$, such that the complete free energy curve constructed by collating the windows can be expressed as

$$\tilde{\Delta}F(n) = \Delta F_w(n) + \sum_{i=1}^{w} k(w), \tag{2.5.5}$$

where we have used $\tilde{\Delta}$ to emphasise that (2.5.5) has been constructed from sampling windows.

2.6 Discussion

Implementing the Metropolis Monte Carlo algorithm to the simplified free particle model revealed several algorithmic and computational difficulties. It is important to devise modifications to the methods in the literature to make them more suitable for the simulation of polymer nucleation, which involves large systems with steep energy barriers.

The use of biasing with a feedback mechanism to obtain uniform sampling of states is similar to umbrella sampling, which is a common technique seen in the literature. The feedback approach is advantageous compared to umbrella sampling, since one requires no knowledge of the rarest states in the system before executing the simulation. We also achieve even sampling of states within the window, reducing the risk of poor sampling at the edges of the window, allowing for more reliable matching of adjacent windows (§2.5). Furthermore, if we wish to simulate a system with similar macroscopic properties to a system for which we have already resolved the free energy barrier, we can simply use the resolved barrier as an initial guess of U_{bias} for the new system.

We have made use of the fact that this problem is *embarrassingly parallelisable*, since each window is an independent copy of the simulation which just samples different subsets of the state-space. This allows us to make use of multiple cores and run simulations in parallel with no genuine parallel programming implementation required. A modification to this algorithm would be to run the simulation using an explicit parallel implementation, such as OpenMP, which would manage the barrier collocation process at specified intervals. This is not necessary for this simulation, since the simulations in each window do not need to share information to resolve their respective free energies. However, we

shall explore this avenue in future chapters when shared information between independent parallel simulations is required to implement parallel tempering (§1.3.4).

In the next chapter, we present some example results using the simulation techniques discussed in this chapter.

CHAPTER 3

Results from simulating the nucleation barriers of free Lennard-Jones particles

3.1 Overview

Here, we provide some results obtained using the free particle nucleation model described in Chapter 2. The purpose of these results is to confirm that the simulation techniques introduced in Chapter 2 are appropriate for improving current approaches to simulating the nucleation process.

3.2 Simulation details

In order to diagnose the simulation, we record the relative occupancy of states, $P(n_C)$, over long runs of the simulation, where states n_C are defined using the order parameter (2.2.1), defined in §1.2.3.3. For convenience, we write the order parameter again,

$$n_C = \sum_{i=1}^{N} H(\xi(i) - \xi_c), \tag{3.2.1}$$

where H is the heaviside step function, $\xi(i)$ is the number of crystalline neighbours of particle i, and ξ_c is the number of crystalline neighbours required for a particle to be crystal-like, which we are free the choose.

Throughout this chapter, we use the Steinhardt-Nelson order parameter (3.2.1)

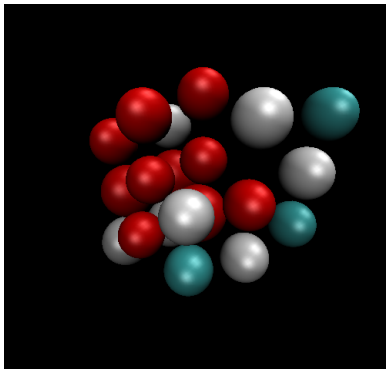


Figure 3.1: Snapshot of a simulation. Red particles are *crystal-like*, meaning they have at least five neighbours that are oriented in a crystal-structure. White particles are *dense*, since they have at least five neighbours, but they are not oriented in a crystal structure. The remaining blue particles do not satisfy any of the above criteria.

to define the crystal state of a configuration. In Figure 3.1 we show a snapshot of a simulation, where the particles have been colour coded according to their crystal state defined by the order parameter, with $\xi_c = 5$.

We also record the free energy of the system, $\Delta F(n_C)$ which can be computed from the relative occupancy of states using (2.3.2). Using these results, we identify the barrier to crystallisation of a system of particles from the maximum in the free energy. The value of n_C which corresponds to the maximum of the free energy barrier corresponds with the critical nucleus size, the minimum nucleus size required for the given system to spontaneously crystallise. The height of the barrier is the energy of formation of the critical nucleus, ΔF^*, which can be used to indicate the nucleation rate τ, since

$$\tau \propto \exp(-\Delta F^*/k_B T), \tag{3.2.2}$$

where k_B is Boltzmann's constant and T is the temperature.

3.3 Biasing

Since we make frequent use of sampling in a biased distribution to obtain our results for steep barriers, we first verify that our implementation is correct. To do this, we apply several different biasing functions to the same system and check that we compute the same free energy system in each case. In particular,

we confirm that our feedback algorithm defined in §2.3.1 allows approximately uniform sampling of all available states and subsequently computes the same free energy barrier as the unbiased simulation.

Since we are verifying the biasing algorithm here, we use a simplified system, where we simulate in the NVT ensemble (see §1.2.2.9), with $N = 20$ particles, number density $\rho = 0.3$ and reduced temperature $T = 0.7$. We use the order parameter (3.2.1) with $\xi_c = 5$. We use this somewhat permissive value of ξ_c to identify crystal states so that the simulation provides a rapidly convergent estimate of the free energy of the system. We use the sampling windows introduced in §2.3.2 to restrict the sampling to window w, such that $n_C \in [N_{\min}(w), N_{\max}(w)]$. To achieve this, we use the biasing function

$$W(n_C) = \exp[U_{\text{bias}}(n_C, w)], \tag{3.3.1}$$

with

$$U_{\text{bias}}(n_C, w) = \begin{cases} -\kappa(w)(n_C - N_{\min}(w))^2 + f(n_C), & n_C < N_{\min}(w), \\ f(n_C), & N_{\min}(w) \leq n_C \leq N_{\max}(w), \\ -50, & n_C > N_{\max}(w), \end{cases} \tag{3.3.2}$$

where $\kappa(w)$ is a parameter that can be tuned to control the number of visits to states $n_C < N_{\min}(w)$, and $f(n_C)$ is the biasing function inside the window, which we are free to choose. We set an appropriate value of $\kappa(w)$ for each window w such that the simulation can occasionally escape the window and melt and reform crystal states of a given size. This guarantees that the simulation with sampling windows is ergodic (all states are accessible through some sequence of moves) and decorrelates from particular crystal states within the time-frame of a typical simulation. The simulation is initialised with particles arranged in a regular cubic lattice. We then run an equilibration period, which ends after at least 50,000 Monte Carlo steps and when there have been a sufficient number of visits to state $N_{\min}$.

We sample states $n_C \in [5, 10]$ to avoid the issues in sampling the zero-to-one transition, which is dealt with in the next section. The weight function inside the window is the simple linear function

$$f(n_C) = \alpha(n_C - 5), \tag{3.3.3}$$

where α is a parameter which is used to control the strength of the biasing function. We expect that, as we increase α, the sampling of states with larger n_C will be sampled more often than those with n_C close to 5. The results for the free energy estimates from each of the simulations for various values of α are presented in Figure 3.2. All free energies were computed over long runs (5×10^7 Monte Carlo steps) of the simulation.

Also shown in 3.2 is a simulation for which we have implemented the feedback algorithm in §2.3.1, specifying $f(n_C)$ using the current estimate of the free energy, ΔF, which was updated every 10^7 steps. Initially, the simulation is unbiased inside the window ($f(n_C = 0)$. The result shown is after 5 iterations of the feedback algorithm.

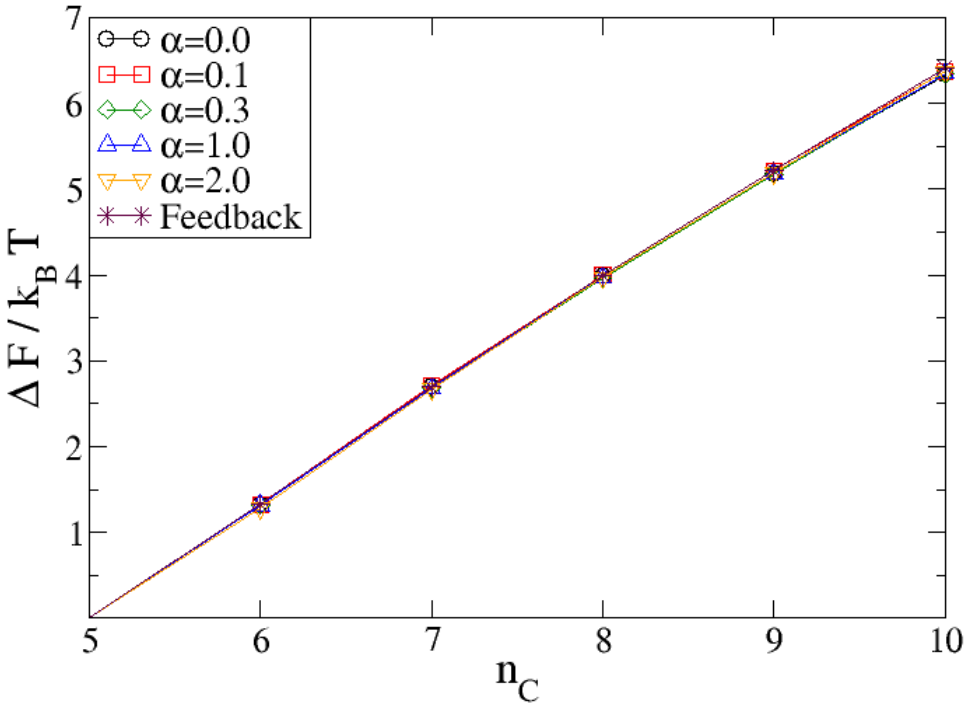


Figure 3.2: Comparison of the resulting free energy curves from nine independent simulations of the same system ($N = 20$, $\rho = 0.3$, $T = 0.7$) with different biasing potentials applied to each.

We readily confirm from this figure that all of the six independent simulations with different biasing functions are in close agreement in their free energy cal-

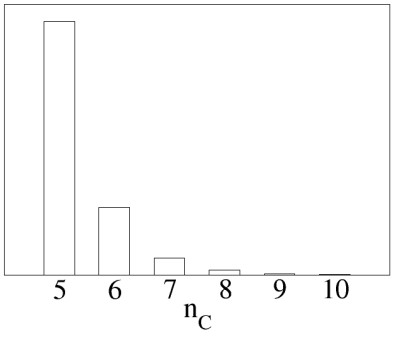
culation. Note that this is a relatively steep nucleation barrier, where the free energy increases by approximately $1\,k_BT$ when n_C increases by one. This means that a very long simulation is required to resolve this barrier reliably, since there will be very seldom visits to the rare states. The corresponding histograms from implementing the biasing function (3.3.3) with various values of α are presented in Figure 3.3.

We observe in Figures 3.3(a), 3.3(b) and 3.3(c) that small values of α ($\alpha < 1$) have little effect on the unbiased distribution. Furthermore, although we have managed to resolve the nucleation barrier when $\alpha = 0$ in Figure 3.2, visits to state $n_C = 10$ are much rarer than visits to state $n_C = 5$, making this system an ideal candidate for the implementation of a biased simulation. There is a significant change in the distribution when $\alpha = 1$ (Figure 3.3(d)), where we achieve better sampling of states with larger n_C. If α is too large, as is the case when $\alpha = 2$ in Figure 3.3(e), we *over-bias* the simulation and do not sufficiently sample states with small n_C. Using these results, one might suggest a trial and error technique to optimise the value of α to achieve good sampling across all states within the sampling window.

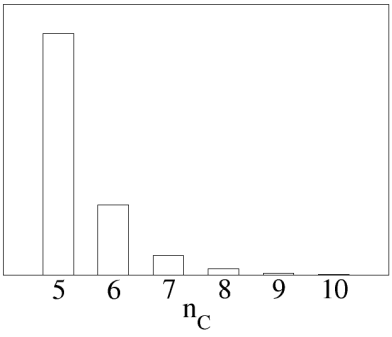
However, Figure 3.3(f) shows that using the Feedback algorithm allows us to achieve almost precise uniform sampling across all available states within the sampling window. The added advantage of the feedback algorithm is that it is completely automated, with no prior knowledge of the nucleation barrier being necessary. For systems with particularly tough, steep nucleation barriers, we can repeatedly apply the feedback algorithm to refine the biasing function. In Figure 3.4 we show the effect on the biased distribution of 3 iterations of the feedback algorithm on the same simulation, for an extended the window of $n_C \in [5, 15]$. Having optimised $W(n_C)$, we use it to fix the biasing function and perform a long run of the biased simulation to obtain an accurate evaluation of the free energy barrier for the system.

3.3.1 Sub-crystal states

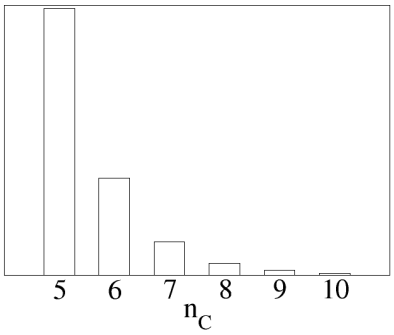
Since the biasing algorithm only increases the likelihood of *accepting* desirable moves, it is ineffective when the number of move *proposals* is insufficient. In §2.2.1, we defined sub-crystal states for simulations in which we use the order



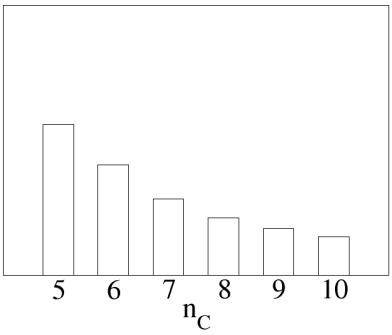
(a) Unbiased ($\alpha = 0$)



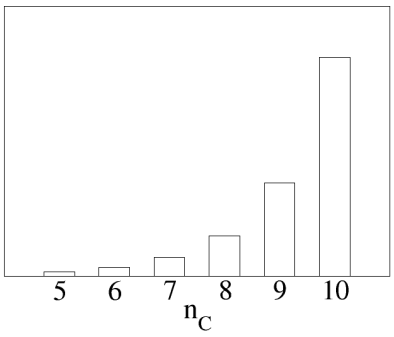
(b) $\alpha = 0.1$



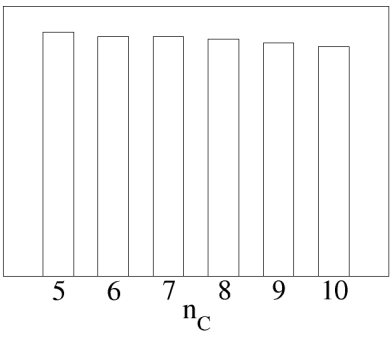
(c) $\alpha = 0.3$



(d) $\alpha = 1.0$

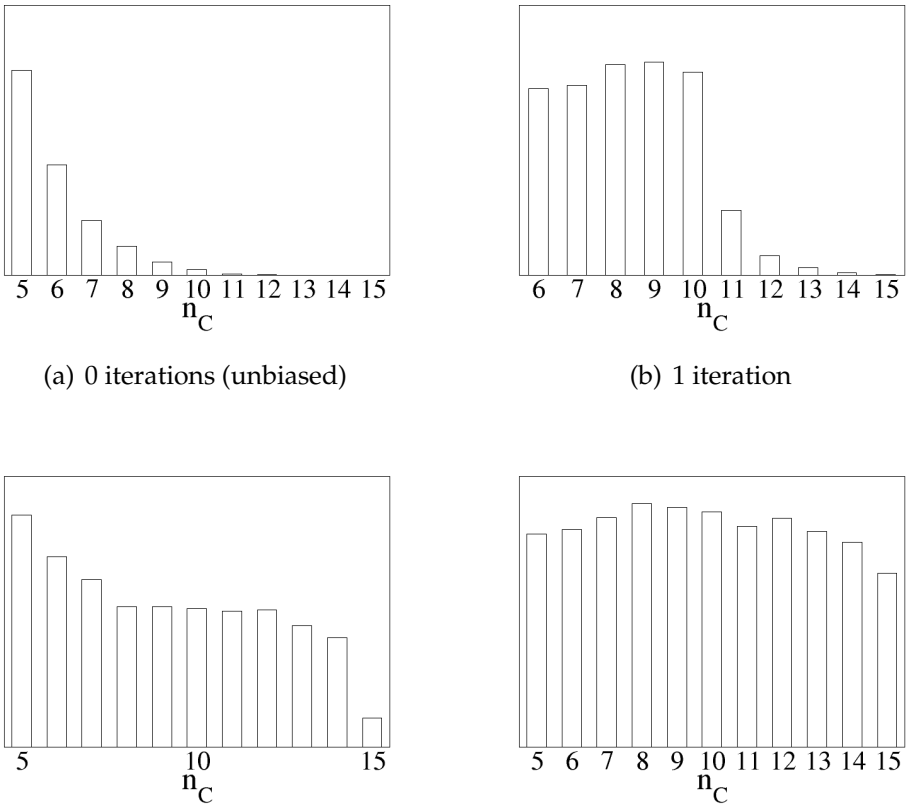


(e) $\alpha = 2.0$



(f) Feedback

Figure 3.3: Histograms of the occupancies of states $n_C \in [5, 10]$ in six independent runs of a simulation in the NVT ensemble ($N = 20$, $\rho = 0.3$. $T = 0.7$), each of which is biased using the biasing function (3.3.3), with α specified as labelled.



(a) 0 iterations (unbiased)

(b) 1 iteration

(c) 2 iterations

(d) 3 iterations

Figure 3.4: Histograms illustrating the feedback algorithm for states $n_C \in [5, 15]$, where $N = 20$, $\rho = 0.3$, $T = 0.7$. After an initial unbiased run, the estimate for the free energy is repeated fed back to the biasing function.

Table 3.1: Summary of the available sub-crystal states in each simulation.

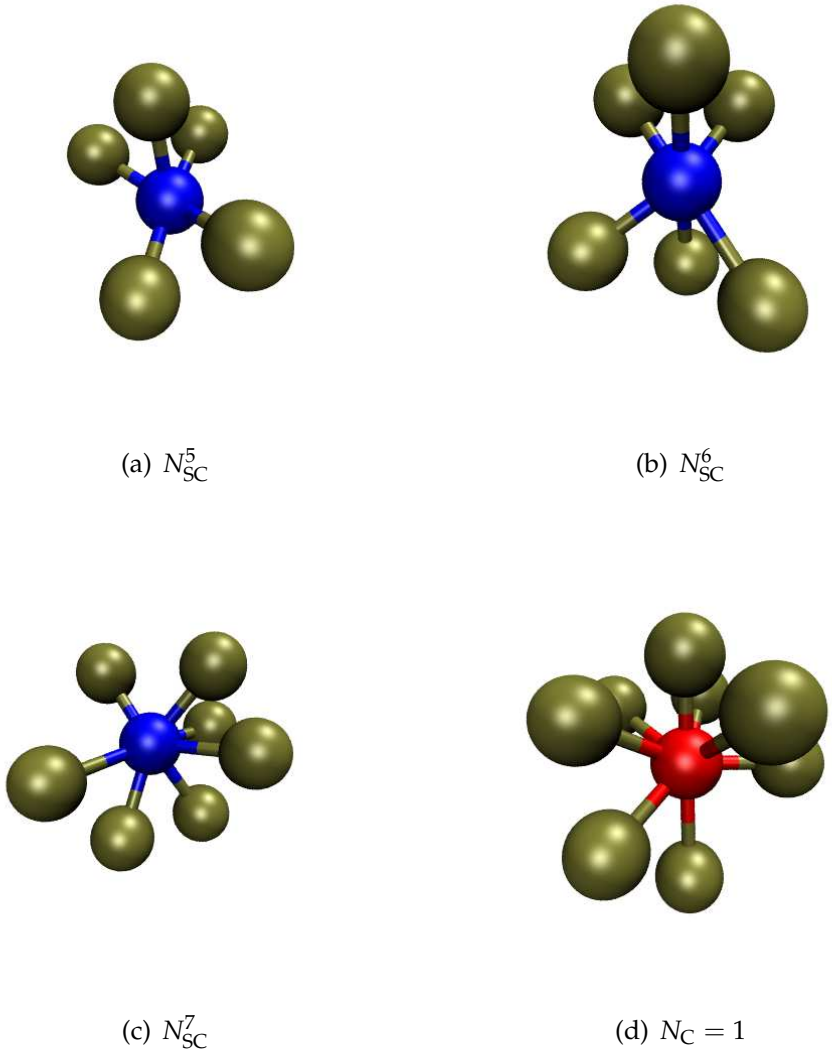
# of sub-crystal states	Available sub-crystal states
0	*None*
1	n_{SC}^5
2	n_{SC}^5, n_{SC}^6
3	n_{SC}^5, n_{SC}^6, n_{SC}^7

parameter (3.2.1) with $\xi_c = 8$. Sub-crystal states have $n_{\mathrm{C}} = 0$, but are close to $n_{\mathrm{C}} = 1$ because there are some particles in the configuration with some crystal neighbours. By biasing into the sub-crystal states, we hope to encourage more frequent attempted transitions from the zero ($n_{\mathrm{C}} = 0$) to first ($n_{\mathrm{C}} = 1$) crystal states in systems with a particularly steep free energy barrier. Effectively, we sub-divide the zero-to-one transition into smaller steps. We determine whether we need sub-crystal states by computing

$$\phi(A, B) = \frac{\#\,\text{proposed moves from } n_{\mathrm{C}} = A \text{ to } n_{\mathrm{C}} = B}{\text{occupancy of state } n_{\mathrm{C}} = A}, \tag{3.3.4}$$

the ratio of the number of proposals from state $n_{\mathrm{C}} = A$ to $n_{\mathrm{C}} = B$, which corresponds with the probability of proposing moves from crystal state A to B. If $\phi(0, 1) < 0.01$, i.e. fewer than 1% of moves proposed from the zeroth to the first crystal state, we are required to sample at least one sub-crystal state to accurately resolve the zero-to-one transition. We then monitor the proportion of move proposals between all available states to diagnose whether the introduction of the new sub-crystal states has improved the sampling of the zero-to-one transition.

In order to demonstrate the effectiveness of the sub-crystal states, we run four simulations of a system, where we use 0, 1, 2 and 3 sub-crystal states. Our sub-crystal states are defined such that a configuration in state n_{SC}^k contains at least one particle i with $\xi(i) \geq k$, where $\xi(i)$ is defined in (1.2.50). To avoid ambiguity, the available sub-crystal states in each simulation are summarised in Table 3.1. See Figure 3.5 for illustrations of typical examples of the listed sub-crystal states and a particle in the first crystal state with $\xi_c = 8$. We refer to crystal states with integers n_{C}, where a configuration is in state n_C if it contains n_C particles i with $\xi(i) \geq 8$.



(a) N_{SC}^{5} (b) N_{SC}^{6}

(c) N_{SC}^{7} (d) $N_{\mathrm{C}} = 1$

Figure 3.5: Illustrations of particles in sub-crystal and crystal states as labelled. We have drawn bonds between the central particle and its near neighbours in each case to emphasise their orientation. Blue particles are in the indicated sub-crystal states and the red particle is in the first crystal state, since it has eight crystal neighbours.

We use the feedback algorithm to obtain approximately uniform sampling of all available states in the biased distribution, then compute the free energy barrier. We have chosen an NVT simulation in which there are $N = 70$ particles, with reduced temperature $T = 0.5$ and number density $\rho = 0.8$. We choose this system as it has an initially steep free energy barrier, so the proposal of moves from zero-to-one are rare. However, the barrier is not so steep that it is impossible to obtain reasonable sampling of the available crystal states without using sub-crystal states or a biasing function. Hence, we can run an initial unbiased simulation to compute an estimate of the free energy barrier for comparison with our results from a biased simulation.

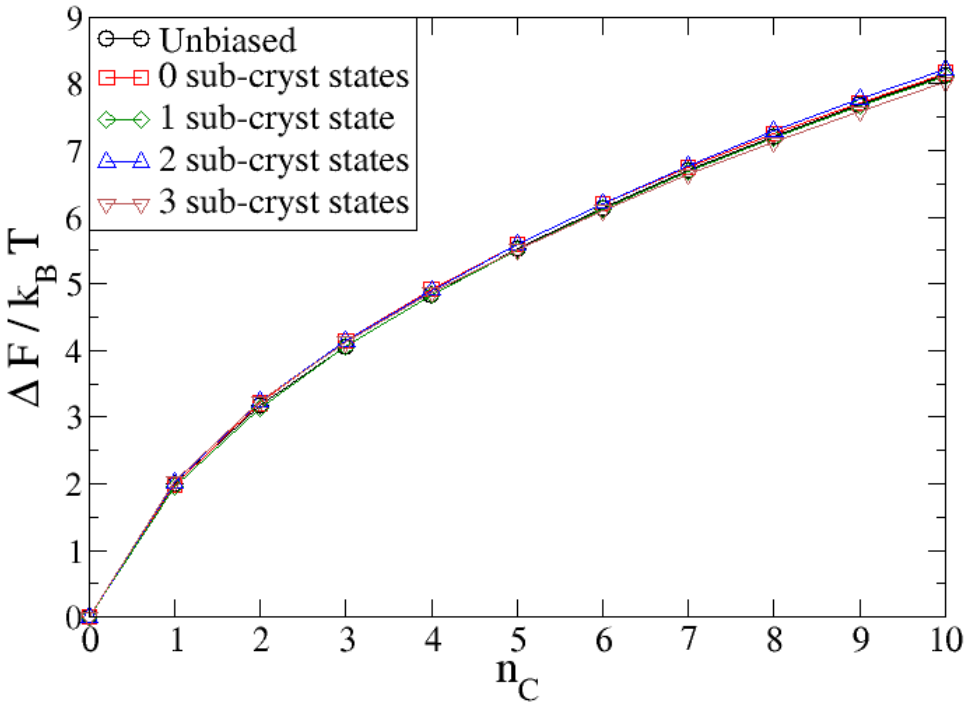


Figure 3.6: Comparison of the free energy curves computed for the same system ($N = 70$, $\rho = 0.8$, $T = 0.5$) where the indicated number of sub-crystal states are sampled and biased using the feedback algorithm.

The resulting free energy barriers from each simulation are plotted in Figure 3.6 as a function of the crystal order parameter n_C. We combine the unbiased

Table 3.2: Summary of the move proposals between the first few available states in each simulation, where $\phi(A, B)$ is the probability of proposing a move from state A to state B.

# of sub-crystal states	Move proposal probabilities
0	$\phi(0,1) = 0.0058$
1	$\phi(0, n_{SC}^7) = 0.0151, \phi(n_{SC}^7, 1) = 0.0178$
2	$\phi(0, n_{SC}^6) = 0.0253, \phi(n_{SC}^7, 1) = 0.0179$
3	$\phi(0, n_{SC}^5) = 0.0397, \phi(n_{SC}^7, 1) = 0.0176$

occupancies of the sub-crystal states to compute the free energy curve so that the curves can be compared in identical terms. Furthermore, the detailed information obtained from sampling the sub-crystal states is only required to achieve satisfactory sampling of the zero-to-one transition and is not required to produce frequent transitions between higher crystal states and thus obtain the overall free energy curve. Clearly, the estimated free energy barriers from each simulation are in close agreement, regardless of the number of sub-crystal states available.

To expose the difference between simulations, we refer to the probability of certain move proposals in Table 3.2. In particular, we focus on moves from the zeroth crystal state to the first available (sub-)crystal state in each simulation and moves to the first crystal state from the previous available state. We see that $\phi(0,1)$ in the simulation with zero sub-crystal states is incredibly small ($\approx 0.5\%$), so the proposal of moves from the zeroth to first crystal state are very unlikely, as predicted. However, moves from sub-crystal state n_{SC}^7 are almost four times more likely, so it is advantageous to bias in favour of this sub-crystal state to help the simulation escape the zero-state. Although introducing one sub-crystal state is sufficient in this case, we observe that we can further improve the simulation by adding more sub-crystal states, since the probability of proposing a move to the first sub-crystal state increases as we increase the number of sub-crystal states available, which enables swifter resolution of barriers such as that in Figure 3.2 and can also be used to resolve even steeper barriers.

3.4 Sampling windows

We have shown in the previous sections that the feedback algorithm is effective for obtaining strong sampling of available states by repeatedly tuning the estimate of the true free energy of a system. This process is sufficient for systems with relatively few states of interest, but is inefficient for systems with many available states. When there are a large number of states, the simulation diffuses between the states for which we have fed back the free energy estimate and so very rarely attempt moves to the higher states for which we have limited information. When applying the feedback algorithm for such systems, the convergence of the free energy is slow and can even be rendered completely ineffective, with the majority of sampling taking place over states that are already well resolved rather than rare states of interest. To overcome this problem, we partition the state-space into sampling windows. We conduct independent simulations of a given system within each window and apply the feedback algorithm so that states within each window are sampled with equal priority. The windows are sufficiently narrow that the diffusion problem described above is avoided and the free energy is well resolved for every window. A small number of states are shared between neighbouring windows to enable the construction of the complete barrier by matching the curves in the overlapping region, e.g. w_1: $n_C \in [0, 10]$, w_2: $n_C \in [8, 20]$, w_3: $n_C \in [18, 30]$, etc. (see Figure 2.3). As in previous simulations with sampling windows, each simulation is initialised with particles on a regular cubic lattice. We then conduct an equilibration period, biasing the simulation into the desired sampling window, w. We begin collecting statistics once the number of visits to state $N_{\min}(w)$ passes a threshold.

There is a concern that simulations in sampling windows that concentrate on high crystal states are not ergodic, since the high crystal states are strongly biased and are very unlikely to melt and reform, meaning that we only explore a particular type of crystal structure, rather than obtaining a genuine representation of the energy landscape. In §2.3.2, we discussed the implementation of a quadratic function to bias states where $n_C < N_{\min}$, where $N_{\min}$ is the lower bound of the window (see (2.3.10)). To verify that a simulation with sampling windows is not affected by these ergodicity issues, we run two independent

simulations of the same system, where the state-space has been partitioned into three windows in the first simulation and ten windows in the second. For these simulations, $N = 50$, $\rho = 0.3$ and $T = 0.7$, which yields a very steep barrier, which are typically very hard to simulate.

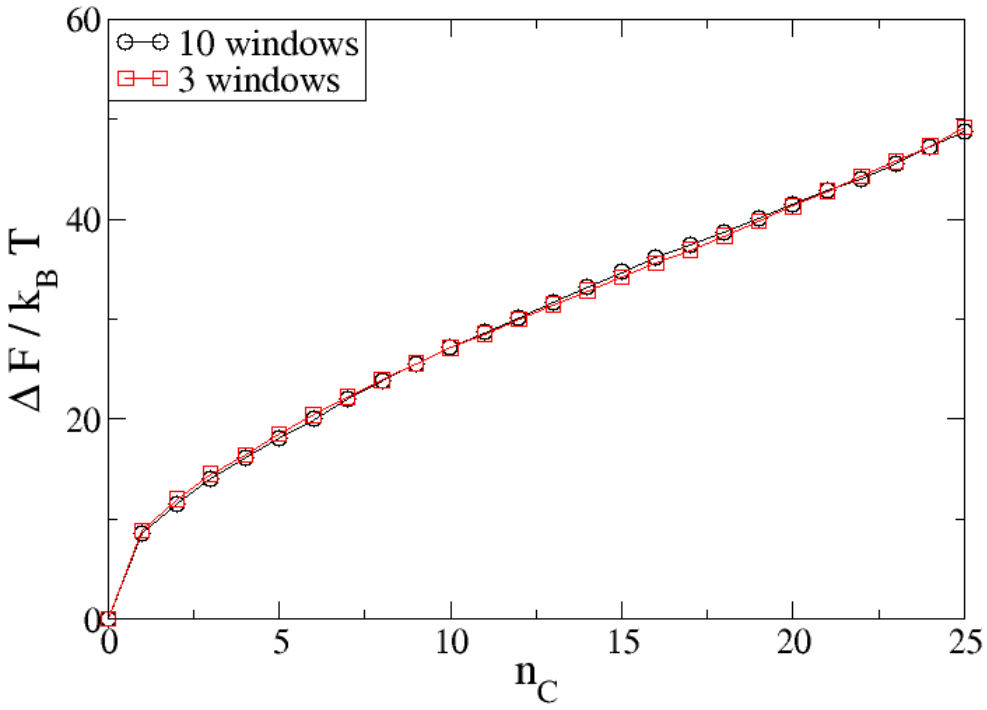


Figure 3.7: Comparison of two simulations of the same system ($N = 50$, $\rho = 0.3$, $T = 0.7$) with three and ten sampling windows.

The results from these two simulations are presented in Figure 3.7. There is close agreement for states $n_C \leq 25$. For the higher crystal states, there is less close agreement, but the data for these states is noisy due to the rareness of the proposal of moves to these states, which renders the biasing ineffective. Since this system has a low number density and high temperature, it is unsurprising that there is no maximum and there is no critical nucleus size that can be reached for spontaneous nucleation to ensue. Nevertheless, this result displays the effectiveness of the sub-crystal states, which were used in the first sampling window of each simulation to obtain good statistics of the zero-to-one transition, and the sampling windows, which increase the effectiveness of the biasing

to accurately sample the remaining states.

3.5 Discussion

In this chapter, we have used a simple free particle model to test and verify the necessary simulation techniques that will be used to simulate the nucleation of single polymer chains in the remainder of the thesis. The advantage of simulating free particles, particularly in the NVT ensemble, is that we only need to define particle displacement Monte Carlo moves to advance the simulation and are not concerned with the chemical connectivity of particles. We have also developed our implementation of the Steinhardt-Nelson order parameter, where we have introduced ξ_c to control the permissiveness of the order parameter, n_C, that can be adjusted to describe the crystal order of the system. Thus, we implemented sub-crystal states, which can be used for systems where we wish to use a strict crystal order parameter, but prevent the simulation from being "trapped" in the zero state due to a lack of move proposals to the first crystal state. This is useful when one desires the growth of strict, well-ordered crystals, which have a steep free energy barrier.

We have verified that our new simulation techniques have a positive effect on simulations, in that they allow us to sample difficult systems with steep free energy barriers accurately and efficiently without changing the results obtained in the original expensive simulation. We have found that our feedback algorithm for sampling windows is an effective and automated alternative to umbrella sampling, which is implemented by several authors, such as Filion et al. [2011]. Like umbrella sampling, the feedback method allows simulations in each window to be run simultaneously on different cpu cores, since they are independent and do not need to share any information until the final stage of constructing the complete barrier. An added advantage is that each of the states are equally represented in the biased distribution, yielding good resolution of the overall free energy. Furthermore, we are not required to have any initial guess of the free energy barrier to produced overlapping "umbrellas" for umbrella sampling, since our feedback technique is automated.

3.5.1 Comparison with theoretical results in the literature

Throughout this chapter, all of the simulations were performed in the NVT ensemble. We chose to do so, since we plan to use the techniquest developed in this chapter to simulate an isolated polymer chain, which is simulated in the NVT ensemble. As a result, we have not developed our techniques for an NpT simulation, although this is possible.

In the literature, authors who conduct simulations with free particles typically do not use the Lennard-Jones potential. Instead, they use hard spheres [Auer and Frenkel, 2005] or the Weeks-Chandler-Anderson potential, which is a truncated Lennard-Jones potential [Filion et al., 2011]. Although some authors simulate with Lennard-Jones particles, ten Wolde et al. [1996b] for example, they typically simulate at NpT, since the nucleation barrier is highly sensitive to pressure.

Due to the reasons above, we cannot obtain a quantitative comparison between our results for free particles with those in the literature. However, for a qualitative comparison, the barriers calculated in the literature have heights in the range 20-40$k_{\mathbf{B}}T$. Our results in Figures 3.2, 3.6 and 3.7 predict barriers of at least this height. Furthermore, we have shown that our results are internally consistent and that our simulations quantitatively reproduce literature results for the pressure-density behaviour of Lennard-Jones fluids in the gas, liquid and supercritical region [NIST, 2007].

3.5.2 Summary

The work in this chapter has provided us with a framework for performing a Metropolis Monte Carlo simulation for a thermodynamic system, with the addition of the new techniques described above. This is a strong basis on which to build our polymer model in the next chapter.

CHAPTER 4

Simulating the nucleation barrier of a single, quiescent, freely-rotating square-well polymer chain

4.1 Overview

In this chapter, we investigate the nucleation barrier of a single polymer chain, which is composed of a chain of particles, which are connected by chemical bonds. The particles interact via the square-well potential (1.4.2). Since we model the freely-rotating chain, we do not impose any bonded interactions, such as those described in §1.4.3.2, and instead just impose a fixed bond length, l. We simulate the nucleation process using the Metropolis Monte Carlo method (§1.3.2). We implement our simulation of a polymer chain by adapting the free particle model of Chapters 2 and 3. To represent the interactions along a polymer chain, we use the united atoms model described in §1.4.2, where each particle in the chain could represent a group of several atoms. To correctly model the connectivity of the particles in the chain, we propose the set of moves described in §1.4.4, since displacing individual particles, as in the free particle model, will not preserve bond length, l.

4.2 Simulation details

We restrict attention to the quiescent case, where there are no external forces on the chain and it can only interact with itself. We conduct our simulation in the canonical ensemble, where the number of particles, N, and the temperature, T, are fixed. The simulation box is very large, so we have no boundary conditions and the box is effectively infinite. We consider a single isolated chain, so do not require boundary conditions.

4.2.1 Particle interaction

The only potential in the freely-rotating chain model is the non-bonded square-well potential,

$$U_{\mathrm{SW}}(r_{ij}) = \begin{cases} \infty & 0 < r_{ij} < \sigma, \\ -\varepsilon & 1 \leq r_{ij} \leq \lambda\sigma, \\ 0 & r_{ij} > \lambda\sigma, \end{cases} \tag{4.2.1}$$

where r_{ij} is the distance between **non-neighbouring** particles i and j, ε is the depth of the square-well, σ is the interaction length (see Figure 1.3), and λ is the width of the square well. See Figure 1.17 for a graphical representation of (4.2.1). The bonds have fixed length $l = \sigma$. Neighbouring bonded particles do **not** interact. We nondimensionalise (4.2.1) using the following scales, where a variable with a '*' is a nondimensional quantity:

$$T = \frac{\varepsilon}{k_{\mathrm{B}}} T^*, \quad U = \varepsilon U^*, \quad r_{ij} = \sigma r_{ij}^*, \tag{4.2.2}$$

where k_{B} is Boltzmann's constant. Applying the scales (4.2.2) to (4.2.1), we obtain, on dropping the stars, the dimensionless square-well potential,

$$U_{\mathrm{SW}}(r_{ij}) = \begin{cases} \infty & 0 < r_{ij} < 1, \\ -1 & 1 \leq r_{ij} \leq \lambda, \\ 0 & r_{ij} > \lambda. \end{cases} \tag{4.2.3}$$

Using the square well potential (4.2.3), the total potential energy of a configuration is

$$E(\mathbf{r}) = \sum_{i=1}^{N} \sum_{j>i} U_{\mathrm{SW}}(r_{ij}), \tag{4.2.4}$$

where $\mathbf{r}$ is the vector containing the configurational coordinates of the N particles in the chain. Note that (4.2.4) can be written as

$$E(\mathbf{r}) = -n(\mathbf{r}), \tag{4.2.5}$$

where $n(\mathbf{r}) \in \mathbb{Z}$ is the number of square-well interactions in configuration $\mathbf{r}$.

4.2.2 Monte Carlo moves

Throughout the simulation, we track the position of each particle in the chain, $\mathbf{r}$. The single chain is initialised in a state that we are free to choose. The initial coordinates are generated using crystal configurations captured from previous simulations. At each Monte Carlo step, we propose crank-shaft, bond pivot, reptation, end rotation and end bridging moves in the ratio 6:1:1:1:1 respectively. Details of how the moves are implemented are provided in §1.4.4. Crank-shaft moves are proposed the most frequently, since the move proposes a small displacement of a single particle, which is likely to be accepted and provide small adjustments to configurations in dense states that can form a crystal nucleus. The remaining move types are proposed in order to change the configuration in a more drastic way, by displacing several particles at once in the case of bond pivots, or changing the index of the end particle in the case of end bridging.

Moves from state m to n are accepted with probability

$$P_{\text{ac}}(m \to n) = \min[1, \Gamma \exp(-\beta \Delta E_{mn})], \tag{4.2.6}$$

where $\beta = 1/T$ is the inverse temperature, and $\Delta E_{mn} = E(\mathbf{r}^n) - E(\mathbf{r}^m)$ is the change in the total square-well potential as a result of moving from state m to n. We use the parameter Γ to modify P_{ac} for end-bridging moves, such that

$$\Gamma = \begin{cases} \dfrac{b_m R_n}{b_n R_m}, & m \to n \text{ is an end bridge move,} \\ 1, & \text{otherwise,} \end{cases} \tag{4.2.7}$$

where b_m and b_n are the number of possible bridging neighbours in state m and n respectively, and R_m and R_n are the distances of the chosen bridging particle to the old and new end particles respectively. See §1.4.4.3.3 for details of the geometry and acceptance probability of end bridging moves.

4.2.2.1 The importance of reptation moves

Reptation moves are essential to fully explore the energy landscape of a system. A sequence of N reptation moves in the same direction is sufficient to fully forget a given configuration. Thus, it is advantageous to ensure that there is a reasonably high acceptance ratio of reptation moves ($\geq 10\%$) to be confident that the simulation explores the state-space effectively. If the simulation explores the state-space effectively by continually forgetting and reforming crystal configurations, we say it is *well-mixed*. An additional diagnostic tool to monitor the mixing of our simulation is the *reptation number*, n_{rept}, where $n_{\text{rept}} = 0$ at the start of the simulation, then

$$n_{\text{rept}} = \begin{cases} n_{\text{rept}} + 1 & \text{if a forward reptation move is accepted,} \\ n_{\text{rept}} - 1 & \text{if a reverse reptation move is accepted,} \end{cases} \tag{4.2.8}$$

and remains unchanged otherwise. We use the reptation number to define *end crossings*. To define an end crossing, we initialise an integer variable, $k^* = 0$. Whenever $n_{\text{rept}} = kN$, $k \in \mathbb{Z} \setminus \{k^*\}$, then a sequence of N forward or backward reptation moves have been accepted and an end crossing has occurred. An end crossing indicates that the chain has renewed its configuration and "forgotten" any trapped or tightly bound configurations that would have persisted in the absence of reptation moves. We update the value of k^* such that $k^* = k$ and continue the simulation. We log the number of these *end crossings* to be sure that the simulation is not just exploring a particular crystal configuration and is constantly gaining new information and mixing effectively. By "mixing effectively", we mean that the chain is able to decorrelate from its current configuration in a number of Monte Carlo steps that is much smaller than the total simulation length.

4.2.3 Order parameter

We have a choice of two discrete order parameters to identify crystal states of the freely-jointed square-well chain. The first, as used by Taylor et al. [2009a], is

$$n_E(\mathbf{r}) = -E(\mathbf{r}), \tag{4.2.9}$$

where E is the total number of square-well interactions in configuration $\mathbf{r}$ and is computed using (4.2.5). The energy order parameter (4.2.9) is based on the principle that if the square-well potential is sufficiently short-ranged, i.e. λ is sufficiently small, the only way for a particle i to have several neighbours j with $r_{ij} \leq \lambda$ is for the neighbours to be in crystal order. We follow the work of Taylor et al. [2009a], who claim that $\lambda = 1.05$ is sufficiently small for n_E to be a suitable order parameter.

The alternative choice for the order parameter is the adaptation of the Steinhardt-Nelson order parameter proposed by Leitold and Dellago [2014], which is described in detail in §1.4.5.2. Essentially, this order parameter computes $n_C(\mathbf{r})$, the **number of crystalline particles** in the configuration $\mathbf{r}$.

We use n_E to define the crystal order of configurations in order to compare our results to those of Taylor et al. [2009a]. The calculation of n_E is also computationally cheaper than that of n_C, as we are required to evaluate spherical harmonics to determine n_C. Furthermore, we need to compute $E(\mathbf{r})$ to determine the acceptance probability of each Monte Carlo move in the Metropolis algorithm, so no further expensive operations are required to find n_E. We compare the order parameters n_E and n_C in §4.4 to determine whether the cheap energy parameter, n_E, is a reasonable order parameter to use instead of the detailed orientational order parameter, n_C.

4.2.4 Free energy estimation

We compute an estimate for the free energy, ΔF, as a function of the order parameter, n_E, using the average occupancy of states sampled in the simulation, $\langle P(n_E)\rangle$, and (1.3.33), where

$$\Delta F(n_E) = \text{constant} - \log\langle P(n_E)\rangle. \tag{4.2.10}$$

The constant in (4.2.10) is arbitrary, since we are only interested in *changes* in the free energy when considering the nucleation barrier. Hence, we are free to choose the constant to vertically translate the free energy curve appropriately when comparing results from multiple simulations. We nondimensionalise ΔF using $k_B T$, where T is the **dimensional** temperature. Hence, $k_B T$ has units of energy and

$$\Delta F = k_B T \Delta F^*, \tag{4.2.11}$$

where ΔF^* is the dimensionless free energy. Henceforth, we drop the '*' and refer to the dimensionless free energy as ΔF.

4.2.5 Biasing algorithm

We use the biasing technique outlined in §1.3.3, where we sample in a biased distribution by modifying (4.2.6), such that moves from state m to n, with order parameters n_E^m and n_E^n respectively, are accepted with probability

$$P_{\text{ac}}^{w}(m \to n) = \min\left[1, \frac{W(n_E^n)}{W(n_E^m)}\Gamma \exp(-\beta \Delta E_{mn})\right]. \tag{4.2.12}$$

The biasing function, $W(n_E)$, is a function of the order parameter, n_E. To define $W(n_E)$, we use the same approach as that in the free particle model of Chapter 2, in that we use an estimate of the free energy, $\Delta F(n_E)$. We can either use an estimate of the free energy for sets of parameters used in the literature [Taylor et al., 2010], or implement the feedback algorithm described in §2.3.1, where we periodically update $W(n_E)$ using the latest estimate of the estimate of $\Delta F(n_E)$ from the simulation (4.2.10). We compute the average occupancy of states in the biased distribution, $\langle P(n_E)\rangle_{\text{w}}$, then use (1.3.37) to compute the average occupancy of each state in the original distribution. Thus, we compute

$$\Delta F(n_E) = \text{constant} - \log\langle P(n_E)\rangle_{\text{o}}, \tag{4.2.13}$$

where $\langle P(n_E)\rangle_{\text{o}}$ is the average probability of the system being in state n_E in the unbiased distribution. The average occupancy of states in the unbiased distribution is given by

$$\langle P(n_E)\rangle_{\text{o}} = \frac{\langle P(n_E)/W(n_E)\rangle_{\text{w}}}{\langle 1/W\rangle_{\text{w}}}, \tag{4.2.14}$$

where $\langle\cdot\rangle_{\text{w}}$ indicates an average taken over the biased distribution. Finally, we define the biasing function,

$$W(n_E) = \exp[U_{\text{bias}}(n_E)], \tag{4.2.15}$$

where

$$U_{\text{bias}}(n_E) = \Delta F(n_E). \tag{4.2.16}$$

We therefore rewrite (4.2.12) as

$$P_{\text{ac}}^{w}(m \to n) = \min[1, \Gamma \exp(\Delta U_{\text{bias}}^{mn} - \beta \Delta E_{mn})], \tag{4.2.17}$$

where $\Delta U^{mn}_{\text{bias}} = U_{\text{bias}}(n^n_E) - U_{\text{bias}}(n^m_E)$ is the change in U_{bias} as a result in moving from state m to n.

We also use the biasing function to define sampling windows to restrict the state-space of interest in a simulation. We sample in a window such that $n_E \in [N_{\text{min}}(w), N_{\text{max}}(w)]$, where w is the window number. We follow the implementation in §2.3.2 by defining the biasing function $W(n_E) = \exp(U_{\text{bias}}(n_C, w))$, with

$$U_{\text{bias}}(n_E, w) = \begin{cases} -\kappa_{\text{L}}(w)(n_E - N_{\text{min}}(w))^2 + \Delta F(n_E), & n_E < N_{\text{min}}(w), \\ \Delta F(n_E), & N_{\text{min}}(w) \leq n_E \leq N_{\text{max}}(w), \\ -\kappa_{\text{R}}(w)(n_E - N_{\text{max}}(w))^2 + \Delta F(n_E), & n_E > N_{\text{max}}(w), \end{cases} \tag{4.2.18}$$

where $\kappa_{\text{L}}(w)$ and $\kappa_{\text{R}}(w)$ are parameters that control the number of visits to states $n_E < N_{\text{min}}(w)$ and $n_E > N_{\text{max}}(w)$ respectively. Since small changes in configuration can lead to large changes in the energy order parameter, n_E, we found it necessary to control visits to states $n_E > N_{\text{max}}$ using a quadratic arm rather than the step function used in the free particle model of the previous chapters. Using the step function introduces window edge-effects, where states close to N_{max} are sampled too often. Using the quadratic function with a reasonably strong bias ($\kappa_{\text{R}} \approx 0.01$) considerably reduces this effect.

4.2.6 Ergodicity and mixing

When simulating the nucleation of free particles in the previous two chapters, we found that the sampling of states was both *ergodic* and *well-mixed*. However, when simulating a polymer chain, the ergodicity of the simulation is necessary but insufficient for good mixing. The simulation is well-mixed if the simulation explores the entire state-space, making regular transitions between states, without the simulation becoming "trapped" in any particular state.

The free particle model is well mixed because, by proposing random displacements of free particles, we can eventually find a sequence of moves that melt and reform a new version of a given high crystal state. In the case of polymer chains, the connectivity of the particles means that the acceptance of moves is reduced due to the increased likelihood of particle overlap. Consequently, the simulation is more likely to become trapped in a dense or crystal state, par-

ticularly when sampling at a temperature close to the melting temperature of the chain. For example, if the ends of the chain are trapped in the centre of a crystal or dense cluster, then reptation moves are unlikely to be accepted, and so the particular crystal or dense configuration persists. This gives the illusion of achieving good sampling of the trapped state, but in fact correlations in the simulation persist, meaning that we only obtain statistics in the region of the trapped state. We compute the following measures of good mixing to be sure that configurations decorrelate, allowing representative sampling of the whole state-space.

4.2.6.1 Criteria for determining good mixing in the simulation

We use the following diagnostic tools to measure the mixing of our simulation.

4.2.6.1.1 The acceptance ratio of all move-types

Clearly, if all move-types have very low acceptances, then the simulation has become trapped. However, the simulation may still be trapped if most move types still have high acceptance ratios, since the adaptive move sizes (see §1.4.4) may have become so small that the moves do not appreciably change the configuration. The most important move-type to monitor is reptation, since, as discussed in §4.2.2.1, the simulation can find new configurations by accepting sequences of reptation moves.

4.2.6.1.2 The frequency of end crossings

End crossings are defined in §4.2.2.1 as the number of times the reptation number n_{rept} (4.2.8) has been equal to a **different** multiple of N. We monitor the end crossings to ensure that the reptation is having the desired effect of facilitating the formation of new random configurations. An example of a false positive for good mixing is observing a high acceptance ratio of reptation moves that confine the reptation number to a small interval. This is a symptom of the simulation being trapped, but would be undetected by considering the reptation move acceptance ratio alone.

4.2.6.1.3 Upward move proposals and acceptances

We define an upward move proposal as a move that increases the order parameter. We log the ratio of upward move proposals to state occupancies and their corresponding acceptance ratio. This information is useful in addition to the acceptance ratio of move-types, since high proportions of upward move proposals is an indication that the moves are having a positive effect on the ability of the simulation to explore the state-space. The acceptance of the moves can be controlled using a biasing function (§4.2.5), given that the moves are proposed sufficiently often.

For the polymer model in this chapter, it is not possible to meet measures 4.2.6.1.1, 4.2.6.1.2 and 4.2.6.1.3 at sufficiently low temperatures to crystallise, especially in high energy regions. However, higher temperatures readily meet all three. This is because

(i) $\beta = 1/T$ is smaller for higher temperatures, so reptation moves are more likely to be accepted, given the acceptance probability (4.2.17).

(ii) Low crystal states are visited much more often, reducing the influence of excluded volume and providing more available space for the chain to reptate.

Hence, we use parallel tempering to improve the mixing of the simulation. Parallel tempering allows us to simulate at the low temperature of interest, where crystallisation of the chain occurs, while simultaneously simulating at higher temperatures, where we achieve good mixing. We periodically swap configurations between the parallel simulations.

4.2.7 Parallel tempering

We implement the parallel tempering technique described in §1.3.4 into our simulations to allow the simulation to explore the state-space effectively. We follow the following algorithm

1. At the start of the simulation, we choose a range of temperatures in which to simulate, $T_1, T_2, \ldots, T_M$, where M is the total number of tempering levels. One of the T_i is the temperature of interest, $\overline{T}$.

2. Initialise M copies of the simulation with identical initial conditions and run them in parallel using OpenMP for several thousand Monte Carlo steps.

3. Periodically pause the simulation copies and propose n_{swaps} swaps of configurations between adjacent temperatures. Swaps are only proposed between adjacent simulations with temperatures T_i and $T_i + 1$, where $i \in [1, M-1]$ is chosen at random. Swaps between adjacent windows, with inverse temperatures $\beta_i = 1/T_i$ and $\beta_{i+1} = 1/T_{i+1}$, are accepted with probability

$$P_{\text{ac}}(\beta_i \rightleftarrows \beta_{i+1}) = \min[1, \exp(-\Delta\beta\Delta E)], \tag{4.2.19}$$

where $\Delta\beta = \beta_{i+1} - \beta_i$ and $\Delta E = E(\mathbf{r}_{i+1}) - E(\mathbf{r}_i)$ are the changes in the inverse temperature and square-well potential as a result of swapping configurations $\mathbf{r}_i$ and $\mathbf{r}_{i+1}$.

4. Restart the M copies of the simulation with temperatures T_i, using the updated configurations for each temperature as the initial configuration.

5. Repeat steps 3 and 4 until the desired number of Monte Carlo steps have been executed.

4.2.7.1 Biasing implementation

We modify our implementation of the biasing algorithm (§4.2.5) in order to correctly simulate in a biased distribution with parallel tempering. We use a fixed biasing function, $W(n_E)$, for the whole simulation, so we must choose a $W(n_E)$ that enables good sampling of states for the simulation at the temperature of interest, $\overline{T}$. To do this, we define a new biasing function, $\overline{W}(n_E)$, which is rescaled using the temperature of interest, such that

$$\overline{W}(n_E) = \exp(\overline{U}_{\text{bias}}(n_E)), \tag{4.2.20}$$

where

$$\overline{U}_{\text{bias}}(n_E) = \overline{T}U_{\text{bias}}(n_E). \tag{4.2.21}$$

We then rewrite (4.2.17), so that moves from state m to n of a simulation running at temperature T_i are accepted with probability

$$P^{w}_{\text{ac}}(m \to n)_i = \min[1, \Gamma \exp(\beta_i(\Delta\overline{U}^{mn}_{\text{bias}} - \Delta E_{mn}))], \tag{4.2.22}$$

where $\beta_i = 1/T_i$ is the inverse temperature. Note that when $T_i = \overline{T}$, the acceptance probability (4.2.22) is equivalent to the acceptance probability without parallel tempering (4.2.17). This has the effect of achieving good, uniform sampling of states for the temperature of interest, while ensuring that the parallel simulations at different temperatures still obey detailed balance.

For configuration swaps to obey detailed balance, we must also include the biasing function in the swap acceptance probability (4.2.19), such that swaps between simulations with temperatures T_i and T_{i+1} are accepted with probability

$$P_{\text{ac}}^{w}(\beta_i \rightleftarrows \beta_{i+1}) = \min[1, \exp(\Delta\beta(\Delta\overline{U}_{\text{bias}} - \Delta E)], \tag{4.2.23}$$

where $\Delta\overline{U}_{\text{bias}}$ is the change in $\overline{U}_{\text{bias}}$ as a result of swapping configurations $\mathbf{r}_i$ and $\mathbf{r}_{i+1}$.

We verified our algorithm by using known analytic results for the distribution of the end-to-end vector for short chains. We found that our results corresponded with the analytic results and were independent of the *number* and *distribution* of temperatures. Furthermore, the results were independent of the choice of biasing function, W.

4.2.7.2 Choice of temperature range

In §1.3.4.2, we discussed the different approaches in the literature to choosing a suitable temperature range when implementing parallel tempering. We must choose a range of temperatures such that the hottest temperature has enough energy to explore the state space without getting trapped, while ensuring that the temperatures are close enough that a sufficient number of attempted swaps between adjacent temperatures are accepted.

As an initial approximation for the temperature range, we use a geometric approximation to define the T_i, such that $T_{i+1}/T_i = c$, where $c > 1$ is the common ratio. We typically choose $c \leq 1.05$ so that the temperatures are sufficiently close so that swaps involving the temperature of interest $\overline{T}$ are likely to be accepted. We then perform some initial short runs of a few thousand Monte Carlo steps to gauge how well-mixed the simulation is.

4.2.7.3 Additional criteria for determining good mixing of the simulation

We use the three criteria described in §4.2.6.1, along with the following additional criteria for parallel tempering.

4.2.7.3.1 Acceptance ratio of reptation moves for the hottest chain

The chain simulated at the hottest temperature must have a good acceptance ratio of reptation moves compared to the coldest chain. In addition, the hottest chain must have completed a reasonable number of end crossings to show that the reptation moves are having the effect of exploring the states. This guarantees that the hottest chain is "forgetting" configurations and forming new configurations, which can be propagated to colder temperatures via swaps.

4.2.7.3.2 The acceptance ratios of swaps between adjacent windows

The swap acceptance ratios must be sufficiently high ($\geq$ 10%) to ensure that the new information being gathered by the hottest chain has a chance of being propagated to the coldest chain via configuration swaps.

4.2.7.3.3 Energy traces of each simulation replica at each swap interval

We monitor the traces of energies as a result of configuration swaps to confirm that the simulation at the temperature of interest is frequently visiting all available states of interest as a result of receiving information from the parallel processes. We can use the traces to check whether a pair of simulations have become trapped, only exchanging the same pair of configurations with each other rather than swapping new configurations through all M temperatures. Monitoring the swap acceptance ratios alone could result in a similar false positive indication of good mixing to monitoring the acceptance ratio of reptation moves without considering end crossings.

4.3 Computing the free energy barrier

4.3.1 Summary of literature results

We compare our results for the computation of the free energy barrier to those of Taylor et al. [2009b]. In a previous paper, Taylor et al. [2009a] use Wang-Landau sampling [Wang and Landau, 2001] to compute the (reduced temperature, T, against square-well interaction length, λ) phase diagram for a single, freely-jointed square-well chain of length $N = 128$. In the phase diagram, the authors define three categories of conformational state of the single chain: expanded coil, collapsed globule and frozen crystallite. The collapsed globule is a dense, but amorphous state. According to the phase diagram, for a sufficiently short interaction length, λ, and low temperature, T, the chain collapses directly from the expanded coil state to the frozen crystallite. Taylor et al. [2009b] choose values $\lambda = 1.05$ and $T = 0.446$ to further investigate this phenomenon.

The free energy barrier for this system computed by Taylor et al. [2009b] is plotted in Figure 4.1. The authors compute the density of states using a Wang-Landau simulation, then compute the free energy by conducting a multicanonical simulation [Berg and Neuhaus, 1992], using their previous Wang-Landau results [Taylor et al., 2009b] as weight factors in the move acceptance criteria. For a long chain, such as the $N = 128$ case, the authors conduct simulations in overlapping energy windows, $n_E \in [0, E_{\text{join}} - 50]$ and $n_E \in [E_{\text{join}} - 50, E_{\text{max}}]$, where $E_{\text{join}} \approx 250$ and $E_{\text{max}} \in [438, 448]$ to compute the density of states. The authors combine the estimates for the density of states from each sampling window by matching the derivatives of the curves in the overlapping region. Ultimately, this produces the overall barrier in Figure 4.1.

4.3.2 Comparison with a literature result computing the free energy barrier for a single chain

We conduct a simulation using our model described in the previous sections. Initially, we do not use sampling windows, effectively letting $N_{\text{min}}(w) = 0$, $N_{\text{max}}(w) = 450$ for all w in (4.2.18). We fix the number of particles in the single chain to $N = 128$, the square-well interaction length to $\lambda = 1.05$, and the tem-

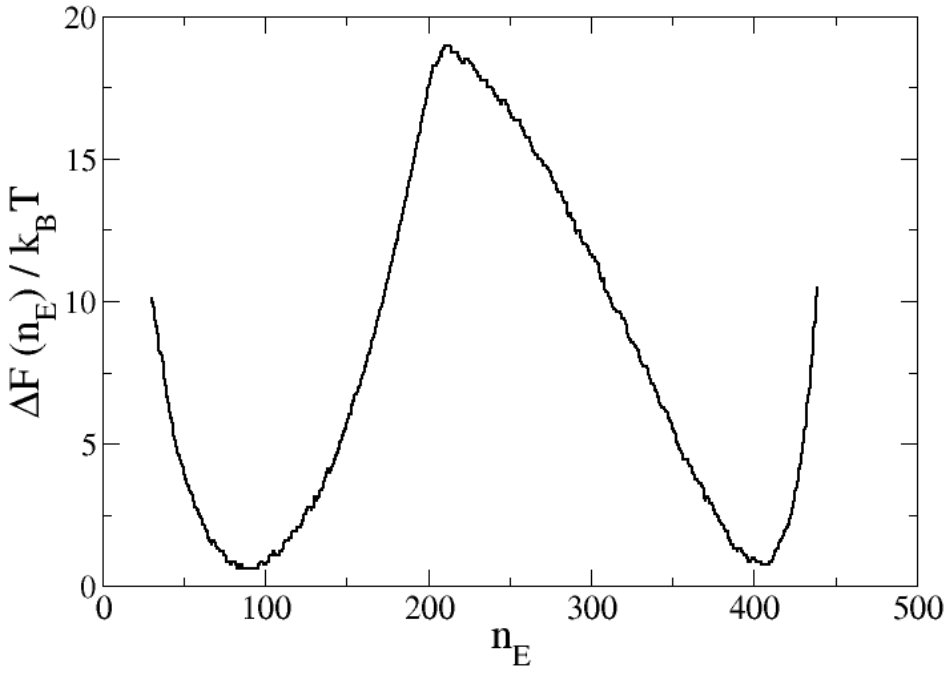


Figure 4.1: Free energy barrier for the $N = 128$ square-well chain, where $\lambda = 1.05$ and $T = 0.446$ computed by Taylor et al. [2009b] as a function of the energy order parameter, n_E. Taylor et al. [2009b] nondimensionalise ΔF with ε using our scales in (4.2.2), we multiply the literature result by $1/T(= 1/0.446)$ to match with our scaling of ΔF (4.2.11). The left-hand minimum corresponds with the melted state and the right-hand with the crystal state, with the maximum between them corresponding with the nucleation barrier.

Table 4.1: Temperature distribution for the simulation comparing our computation of the free energy to that of Taylor et al. [2009b] in Figure 4.1. The temperature of interest is T_0. Temperatures T_i, $i > 1$ are defined using a geometric progression, such that $T_{i+1}/T_i = 1.05$

$T_1\ (= \overline{T})$	0.446
T_2	0.46830
T_3	0.49172
T_4	0.51630
T_5	0.54212
T_6	0.56922
T_7	0.59768
T_8	0.62757
T_9	0.65895
T_{10}	0.69189
T_{11}	0.72649
T_{12}	0.76281

perature of interest, $\overline{T} = 0.446$ to match the system simulated by Taylor et al. [2009b].

We implement parallel tempering using twelve tempering levels. The twelve simulations are run in parallel on multiple cpu cores using OpenMP. The temperature distribution is provided in Table 4.1. This level of separation between the temperatures allows for reasonable acceptance ratios of configuration swaps between adjacent temperatures.

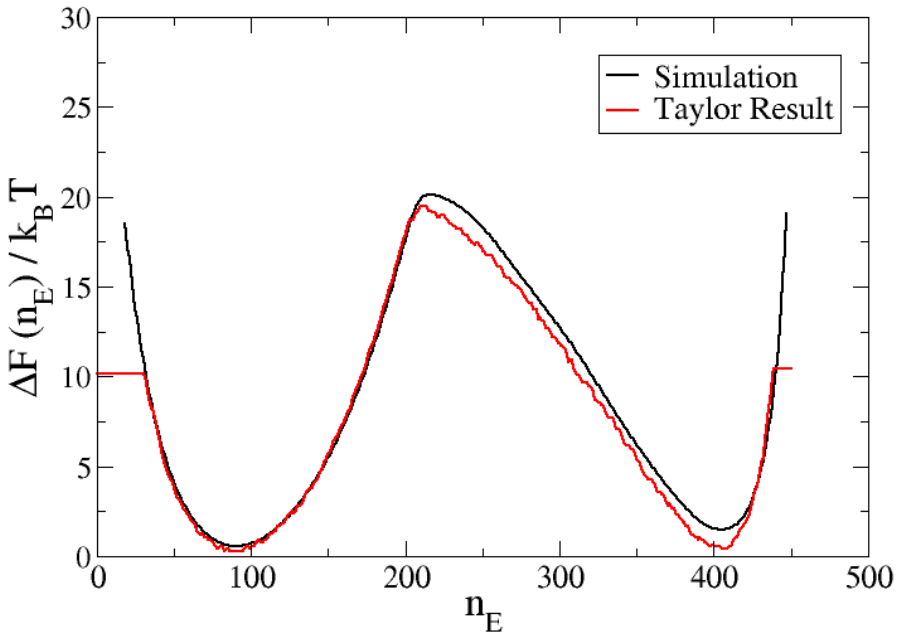
Using a configuration from a previous simulation, each chain is initialised in state $n_E = 100$, which is in the vicinity of the local minimum predicted by Taylor in Figure 4.1 that corresponds with the coil state. We use the literature result for the free energy barrier in Figure 4.1 to define U_{bias} in the biasing algorithm (4.2.18). After an initial equilibration of 5×10^4 Monte Carlo steps, we begin collecting statistics of the biased occupancy, $\langle P(n_E)\rangle_{\text{w}}$, from which we compute the free energy using (4.2.13). Every 5×10^5 Monte Carlo steps, we pause all twelve parallel simulations and attempt fifty configuration swaps between adjacent temperatures T_i and T_{i+1}, where $i = 1, 2, \ldots, 11$ is chosen randomly. We then repeat this process, frequently entering "swap cycles" to propagate infor-

mation from different temperatures through each parallel simulation.

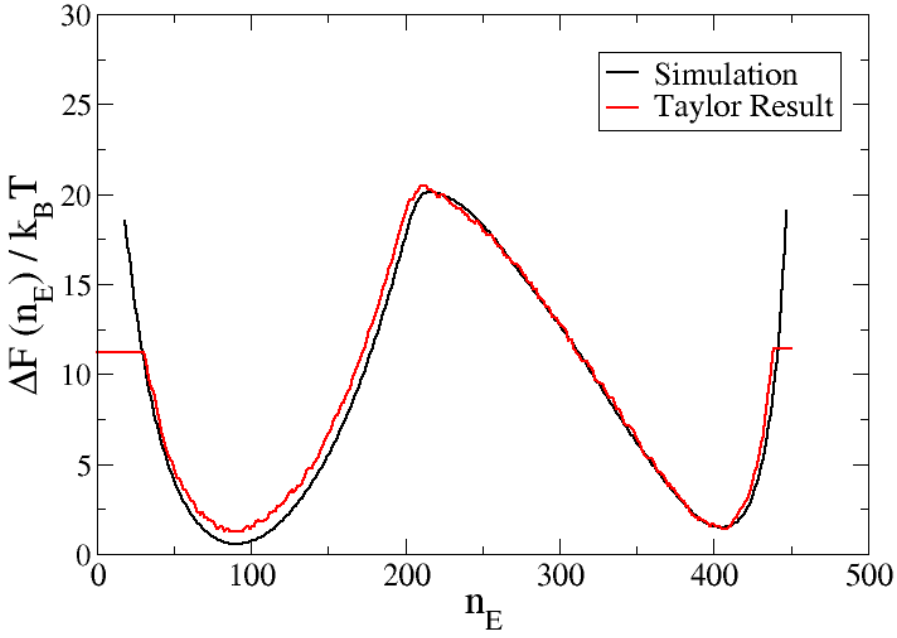
The resulting free energy barrier for the temperature of interest is plotted in Figure 4.2. By comparing Figures 4.2(a) and 4.2(b) we see that we have close agreement with the literature result for the two local minima in the barrier. The fact that we need to vertically shift our result to observe the agreement suggests that we have good resolution of the two minima, but not of the barrier maximum. The corresponding biased occupancy of states for the temperature of interest are plotted in Figure 4.3. This plot shows that the biasing algorithm is working effectively in the regions either side of the barrier maximum, since we achieve approximately uniform sampling there. However, there is a clear change in the behaviour of the simulation in the region $n_E = 200$, which corresponds with the point at which Taylor et al. [2009b] join their two sampling windows. This is some evidence of a lack of mixing of the simulation in the state-space close to the barrier maximum. Although the barrier appears to be well resolved in our simulation, it is likely that the simulation has rarely traversed from one side of the barrier to the other and has simply spend a large proportion of the simulation time exploring each of the minima.

To confirm this diagnosis, we conduct two more independent simulations, using the literature results for the free energy to bias the states, but with a modified temperature distribution. The new temperature distribution (given in Table 4.2) includes four temperatures *below* the temperature of interest, which allows us to pass information from states likely to be in high crystal states into the simulation of interest. In effect, we are creating the possibility of swaps that bypass the barrier peak. Also, the temperatures are much closer together near the temperature of interest, $\overline{T}$. Chains around $\overline{T}$ tend to get trapped either to the left or the right of the barrier. This reduces the chance of swaps being accepted around $\overline{T}$. The closer temperatures improve the swap acceptance probability because $\Delta\beta$ is small in (4.2.23). This also allows swaps with large ΔE to be accepted, i.e. swaps that bypass the barrier. In one simulation, each of the chains are initialised in a high crystal state ($n_E = 400$) and a low crystal state ($n_E = 100$) in the other. The resulting free energy barriers for the temperature of interest are plotted in Figure 4.4.

As in the previous simulation, neither of the computed free energy barriers accurately replicated the literature results. Furthermore, they are not consistent



(a) Taylor result shifted to match our result for $n_E \approx 100$.



(b) Taylor results shifted to match our result for $n_E \approx 400$.

Figure 4.2: Comparison of our estimate of the nucleation barrier to that of Taylor et al. [2009b] for $T = 0.446$.

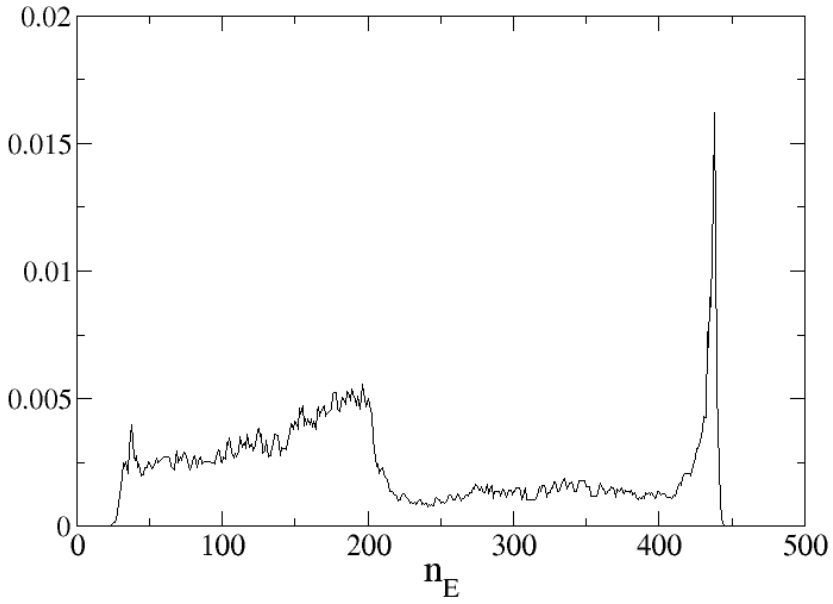


Figure 4.3: Histogram of state occupancies for the simulation to compute the free energy barrier in Figure 4.2.

Table 4.2: Modified temperature distribution. Starting with the distribution in Table 4.1, we have added some temperatures lower than $\overline{T}$ and reduced the difference between adjacent temperatures about $\overline{T}$ to increase the acceptance ratio of configuration swaps in and out of the simulation with $T = \overline{T}$

T_1	0.36000
T_2	0.38000
T_3	0.41000
T_4	0.43600
T_5 $(= \overline{T})$	0.44600
T_6	0.44900
T_7	0.45200
T_8	0.45600
T_9	0.46300
T_{10}	0.46830
T_{11}	0.47500
T_{12}	0.49172

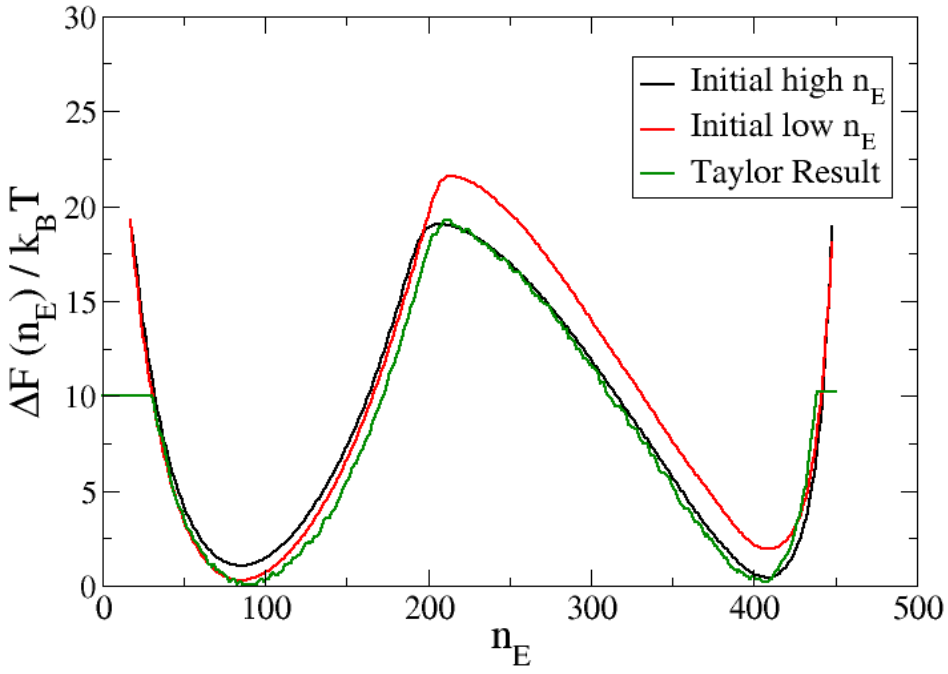


Figure 4.4: Free energy barriers computed from two independent simulation of a square-well chain of $N = 128$ particles, with interaction length $\lambda = 1.05$ and temperature of interest $T = 0.446$. We use the result of Taylor et al. [2009b] as a biasing function, where the system is initialised in a state with either high or low n_E.

with each other either. We take note of the interesting feature that the minimum at $n_E \approx 100$ computed by our simulation initialised at low n_E is in close agreement with the literature results. The same is true for the minimum at $n_E \approx 400$, in that the simulation initialised at high n_E is in close agreement with the literature result near the minimum. The corresponding state occupancies for the two simulations at the temperature of interest are plotted in Figure 4.5. As for the previous simulation, there is a clear feature in both distributions at $n_E = 200$, suggesting an issue with the biasing of states in this region.

We log the value of n_E for each configuration after each attempted configuration swap, whether it is accepted or not. We use this data to plot the energy trace as a result of the swaps. The swap energy traces for these simulations is plotted in Figure 4.6. In Figure 4.6(a), we see that there is reasonably good mixing throughout the simulation that was started in a high crystal state, though there are some intervals where the simulation is "trapped" on one side of the barrier for several swap cycles. This can be seen in the intervals where states $n_E \in [200, 300]$ are rarely visited. In the case of the simulation initialised in the low crystal state, we see in Figure 4.6(b) that it takes approximately half the simulation time (2.5 days in real time) for the simulation to cross the barrier once, after which the trace plot is similar to that in Figure 4.6(a).

Observing distributions of state occupancies and traces in Figures 4.5 and 4.6, we see that we do not have frequent exploration visits to states in the vicinity of the barrier maximum. In particular, the simulation frequently approaches $n_E = 200$, but it unable to pass through; a feature seen on both sides of $n_E \approx 200$. Instead, the simulation occasionally tunnels from one side of the barrier through configuration swaps. This feature is most prevalent in the simulation initialised in the low crystal state. In this simulation, although the simulation spends half of the total runtime in the low crystal region, there is a higher overall occupancy in the high crystal region in Figure 4.5. We can see in the trace plot (Figure 4.6(b)) that, once the simulation crosses the barrier the first time, it is "trapped" in the high crystal region, and only makes occasional excursions back to the low crystal region.

The swap statistics are acceptable for this simulation (all swap acceptance ratios for adjacent temperatures are more than 10%). Also, the hot simulations have high reptation move acceptance ratios and frequent end crossings, which,

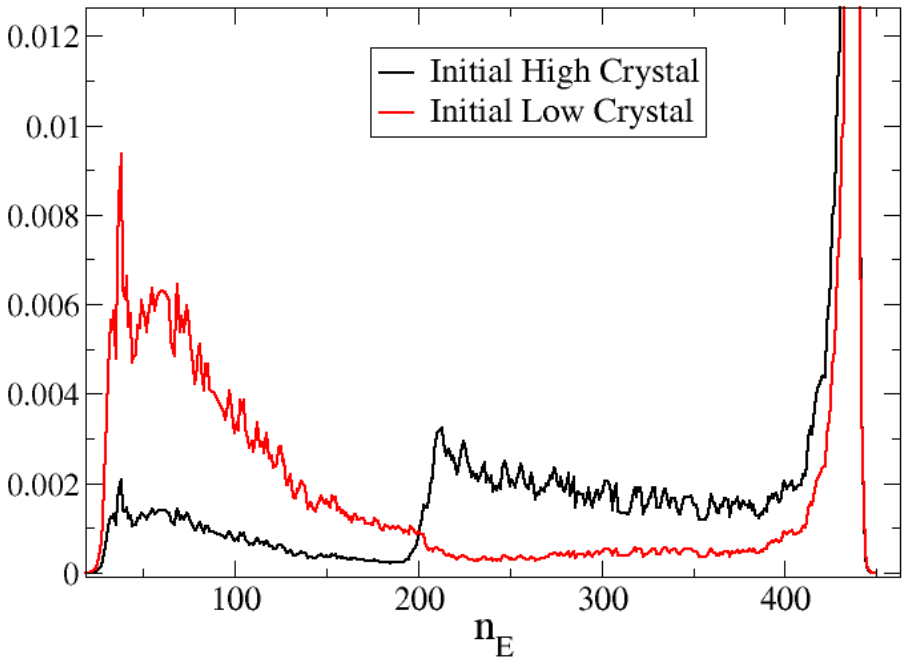
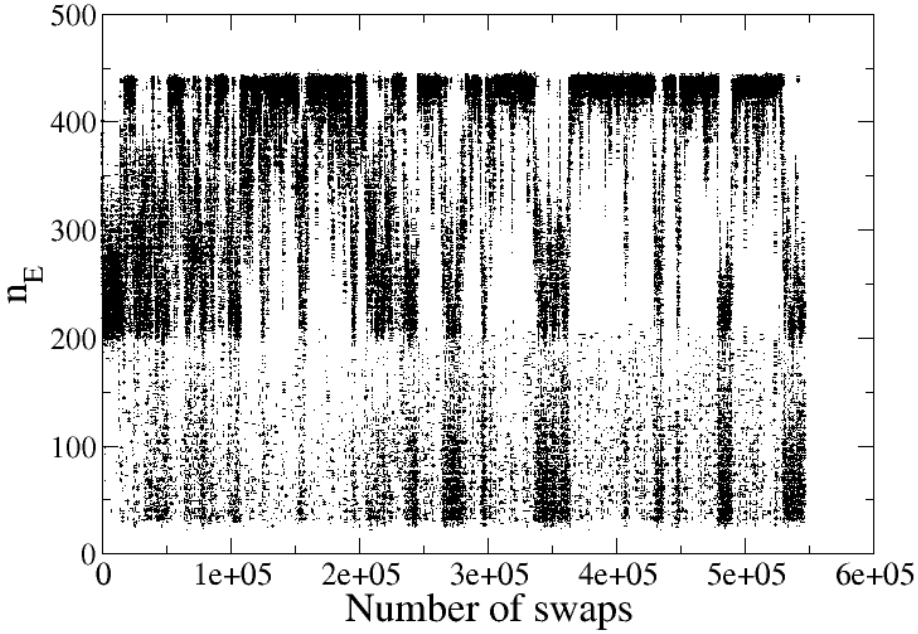
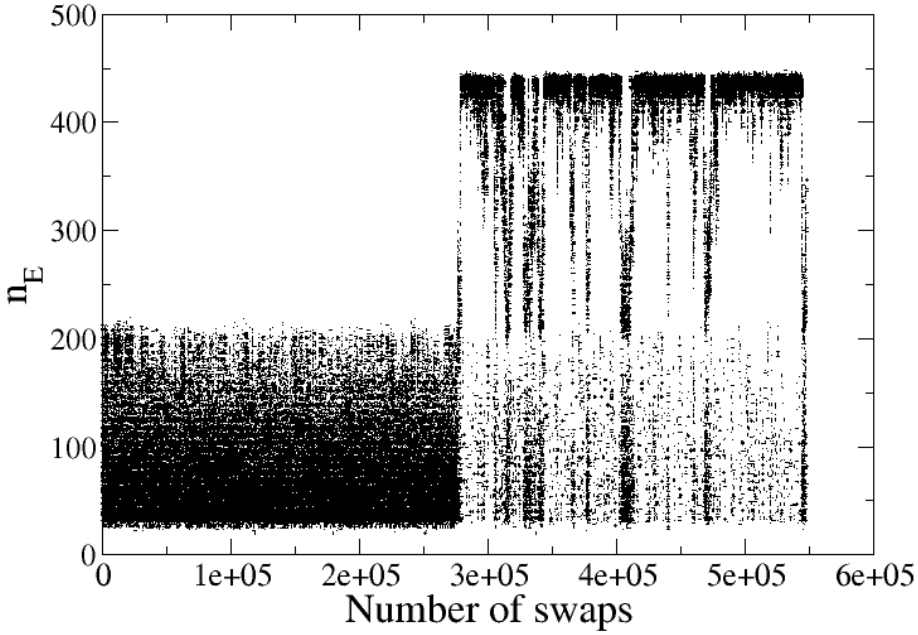


Figure 4.5: Histograms of the state occupancies for the two simulations computing the free energy barriers in Figure 4.4.

combined with the reasonable swap acceptance ratio, propagates new configurational information to the simulation at the temperature of interest. Since this is the case, the temperature distribution in Table 4.2 passes our tests described in §4.2.6.1 and §4.2.7.3 for good mixing. Hence, we deduce that the discrepancy in the free energy calculations in Figure 4.4 is caused by an inappropriate biasing function. States near the barrier maximum are *under-biased* by using the literature values for the free energy barrier, suggesting that the true barrier for the system is higher than that computed by Taylor et al. [2009b]. It is stated by Taylor et al. [2009a] that there is some distortion in their results in the extreme upper edge of their "low energy" window, which corresponds with the extreme lower edge of our high energy window due to difference of a minus sign in our implementation of the energy scale. This suggests that the discrepancy in the results for the overall free energy barrier may be due to the joining of the results from the separate sampling windows by Taylor et al. [2009a].



(a) Initial high crystal



(b) Initial low crystal

Figure 4.6: Trace plots of configuration swaps for the simulations computing the free energy barriers in Figure 4.4.

4.3.2.1 Resolving the top of the barrier

Since one of the aims of this thesis is to simulate the nucleation barrier of a polymer chain, we focus attention on resolving the maximum of the nucleation barrier for this system, which was not a focus of Taylor et al. [2009b]. In particular, we wish to obtain reproducible results for the location, height and shape of the maximum. To this aim, we repeat the simulation of the previous section, where we simulate the same square-well chain with $N = 128$, $\lambda = 1.05$, and $\overline{T} = 0.446$, but with a revised biasing strategy, based on our algorithm in Chapter 3. Our new strategy is to resolve the low crystal region and barrier peak using a sampling window. We implement parallel tempering, using twelve temperatures, with the distribution in Table 4.2.

For this simulation, we restrict the sampling to a window using the weight function $W(n_E) = \exp[U_{\text{bias}}(n_E)]$, where $U_{\text{bias}}(n_E)$ is given by (4.2.18). We only sample in one window, so $w = 1$. We set $N_{\text{min}}(1) = 0$ and $\kappa_{\text{R}}(1) = 0.005$. For N_{max}, we use 278, 280, 285, 290, 300. Each of these values defines a window that includes the barrier maximum, but increasing N_{max} allows us to sample states further into the crystal region. We are cautious about how large we allow N_{max} to be, since we do not want the simulation to become "trapped" in the crystal region, as in the previous section.

We bias the simulation for states $n_E \in [0, N_{\text{max}}]$ by using the best estimate of the free energy. We obtain the best estimate of the free energy using our feedback algorithm (§2.3.1). We initially use the literature result for the free energy in Figure 4.1 as a biasing function, then complete several iterations of the feedback algorithm, performing very long simulation runs ($O(10^8)$ Monte Carlo steps) between each update to provide enough simulation time to obtain a reasonable resolution of the barrier for states where the biasing is too weak. The resulting free energy estimate used to bias the simulation is plotted in Figure 4.7. Note that we have flattened the biasing function for states $n_E < 85$, since we already have a good resolution of the barrier for low crystal states and only need to resolve the barrier minimum, which is used to compute the barrier height. We only have good resolution of the barrier for states $n_E < 250$, so we apply a linear projection of the curve for states $n_E > 250$.

The free energy curves computed from these simulations are plotted in Figure

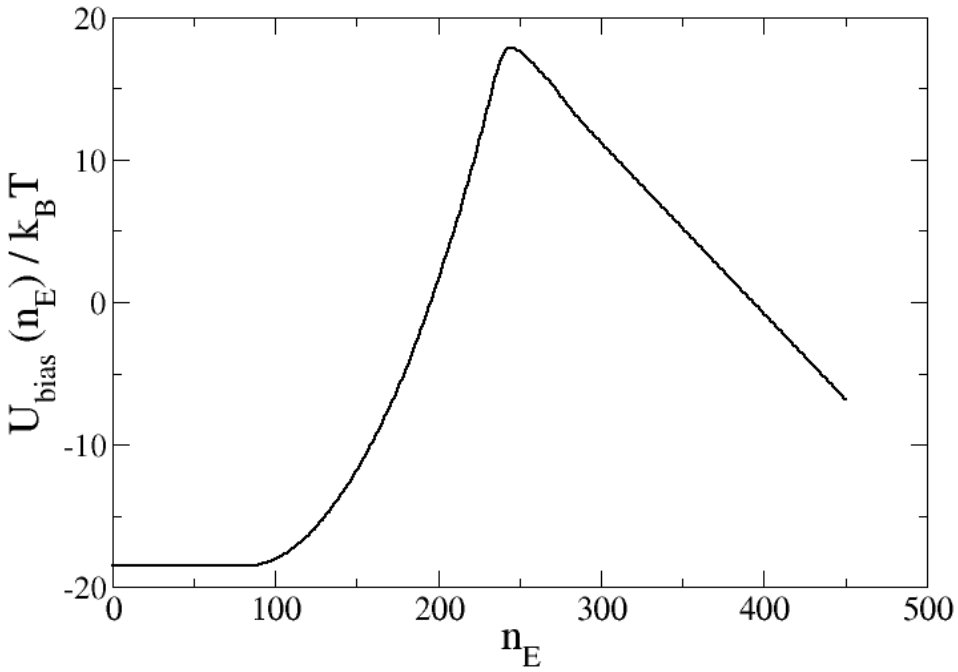


Figure 4.7: The best estimate of the true free energy barrier of the square-well chain with $N = 128$, $\lambda = 1.05$, $T = 0.446$. We compute this function by performing several iterations of the feedback algorithm to obtain successively better estimates of the free energy. We flatten the function for states $n_E < 85$, for which we have no interest in resolving the free energy. A linear extension is added for states $n_E > 250$, where the free energy is currently unresolved. We use this function to bias our simulation inside the sampling window ($n_E \in [N_{\min}, N_{\max}]$).

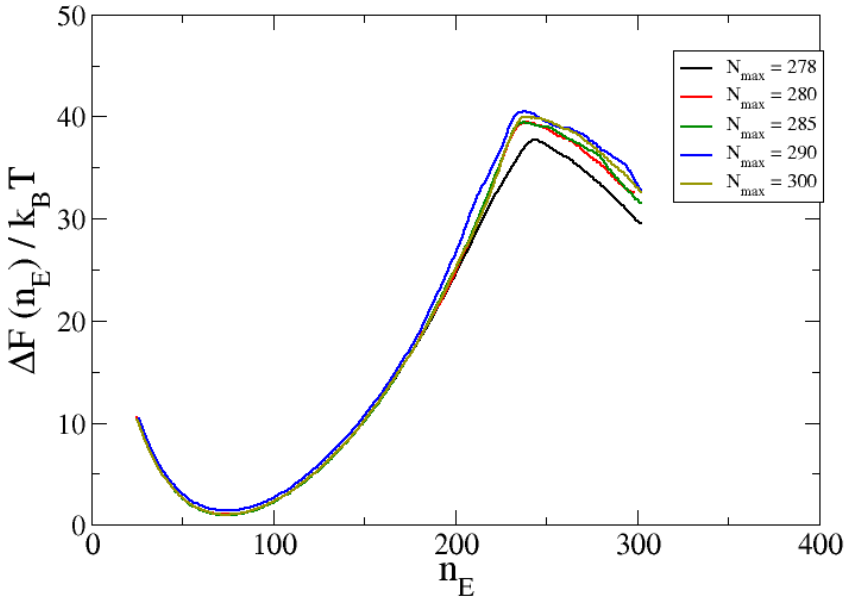


Figure 4.8: Free energy barriers computed for the square-well chain with $N = 128$, $\lambda = 1.05$, $T = 0.446$ and biasing function plotted in Figure 4.7 for various values of $N_{\max}$.

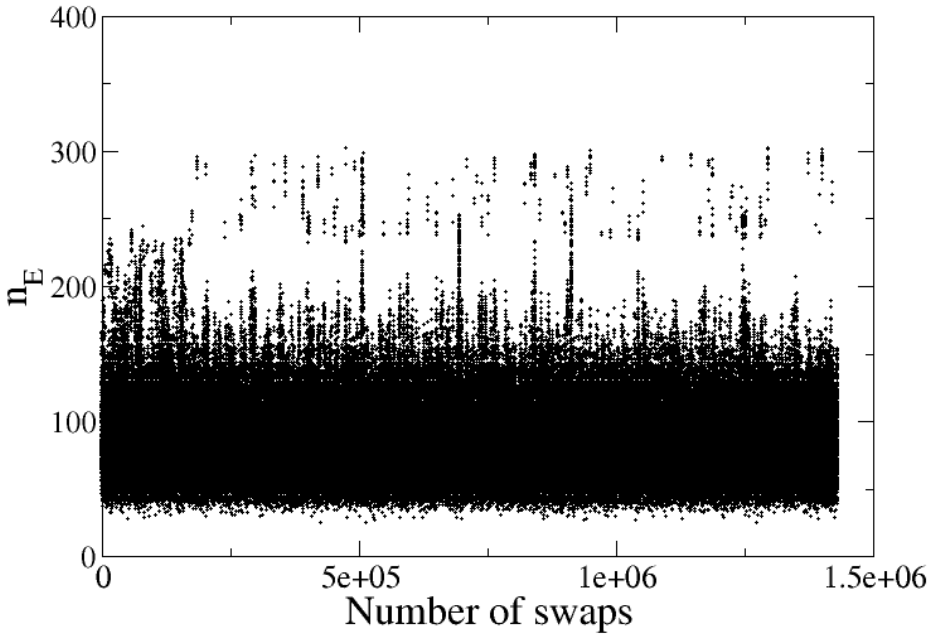


Figure 4.9: Trace plots of configuration swaps for the simulations computing the free energy barriers in Figure 4.8, with $N_{\max} = 285$.

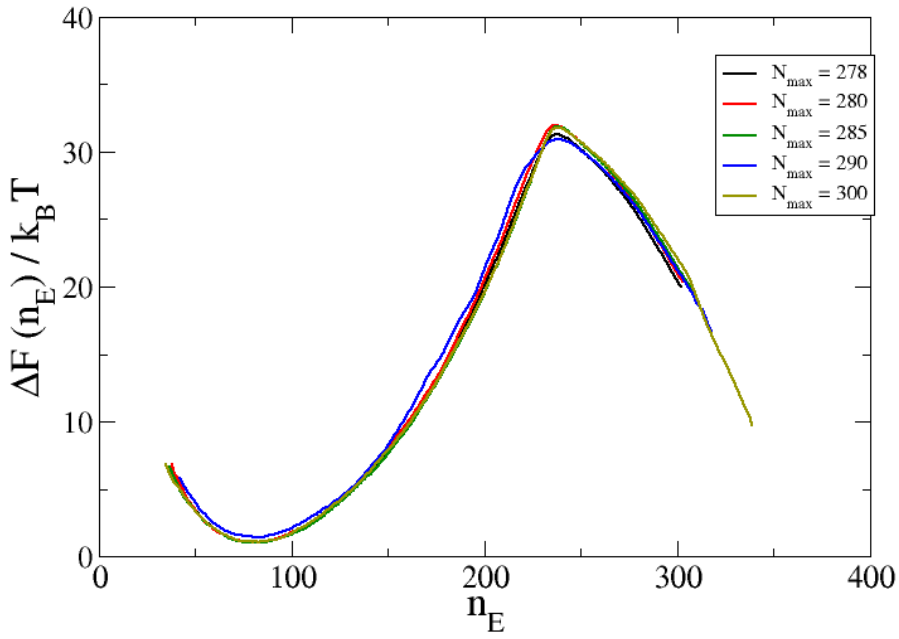


Figure 4.10: Free energy barriers computed for the square-well chain with $N = 128$, $\lambda = 1.05$, $T = 0.436$, with various values of $N_{\max}$.

4.8. We see that there is reasonable consistency between the independent runs. Each run predicts a barrier that is approximately $20k_BT$ higher than the literature estimate. The location of the maximum, which corresponds with the critical nucleus is also approximately 30ε larger than the literature estimate. Finally, the shape of the maximum is similar, though all barriers are strongly peaked. The corresponding trace plot for configuration swaps in the simulation with $N_{\max} = 285$ is plotted in Figure 4.9, which is representative of the swap traces for each simulation. The trace shows that, although there are regular swaps across the barrier, the sampling is dominated by visits to the low crystal states, hence the close agreement in that region compared to the barrier maximum.

We can obtain useful information from the simulations running at temperatures $T_i \neq \overline{T}$, even though they are not systems at the temperature of interest. The free energy curves for $T_4 = 0.436$, which is the first temperature below $\overline{T}$, are plotted in Figure 4.10. At this temperature, we obtain a similar barrier, though the height is approximately $5k_BT$ lower than that for $T = 0.446$. There is much closer agreement at the barrier maximum between the independent simulations

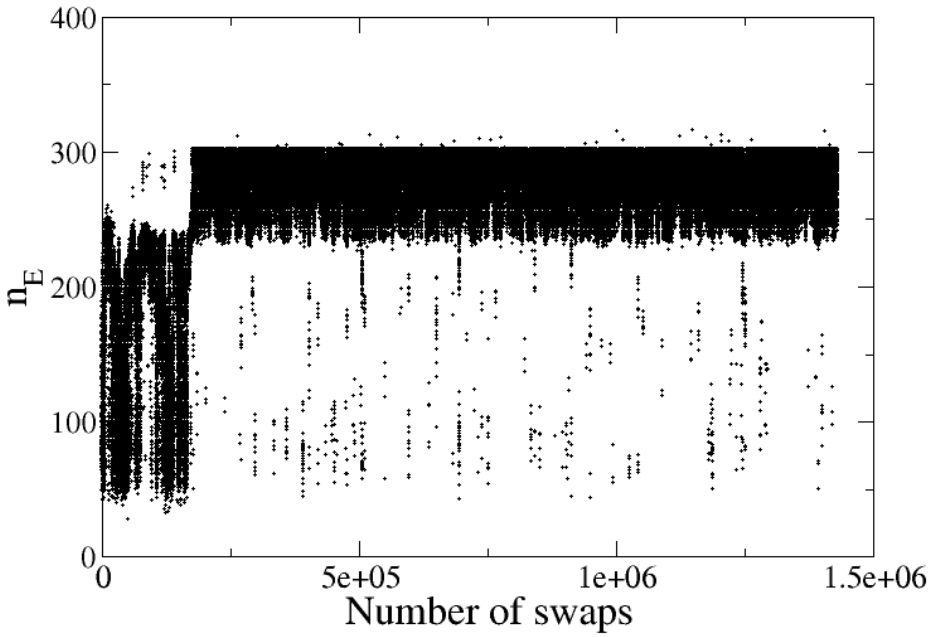


Figure 4.11: Trace plots of configuration swaps for the simulations computing the free energy barriers in Figure 4.10, with $N_{\max} = 285$.

in this case. This is further corroborated by the corresponding trace plot in Figure 4.11. Although at first glance, the simulation appears to be trapped, there are regular excursions to very low crystal states ($n_E \approx 50$), but further there is frequent exploration to states on either side of the barrier. Hence, we can be confident that the mixing is improved and the simulation explores the states in the vicinity of the barrier maximum. This is further supported by the reproducibility between simulations with different $N_{\max}$.

4.3.2.2 Strong reproducibility between independent simulations

We perform one final iteration of the feedback algorithm to modify the biasing function. Performing two long simulations, we obtain the free energy barriers plotted in Figure 4.12. Here, we see strong reproducibility between the two independent simulations, which is significant evidence that the simulations are mixing well across all states for which we have resolved the free energy in Figure 4.12.

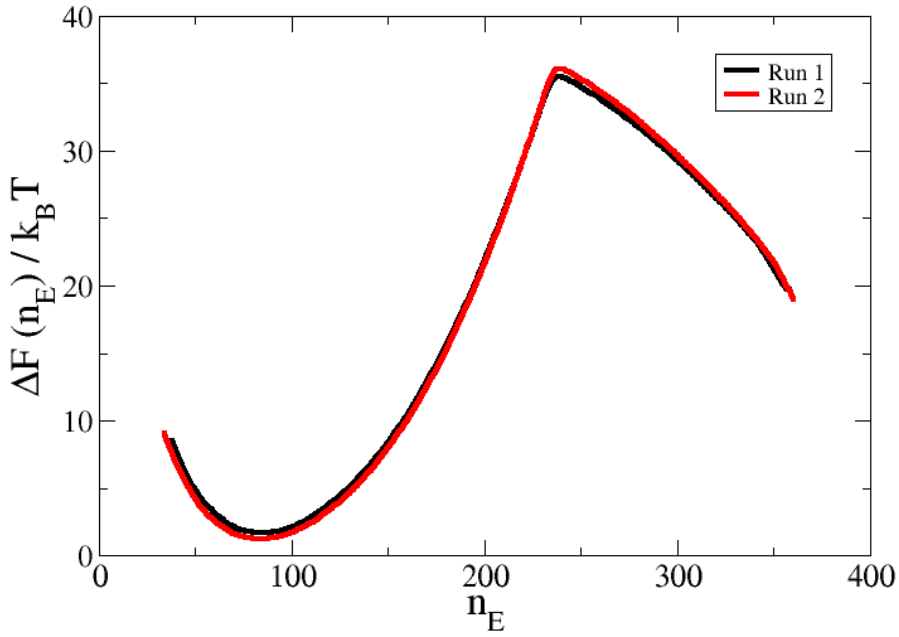


Figure 4.12: Free energy barriers computed from two long independent runs of the simulation of the square-well chain with $N = 128$, $\lambda = 1.05$, $T = 0.436$, having completed a final iteration of the feedback algorithm to update the biasing function.

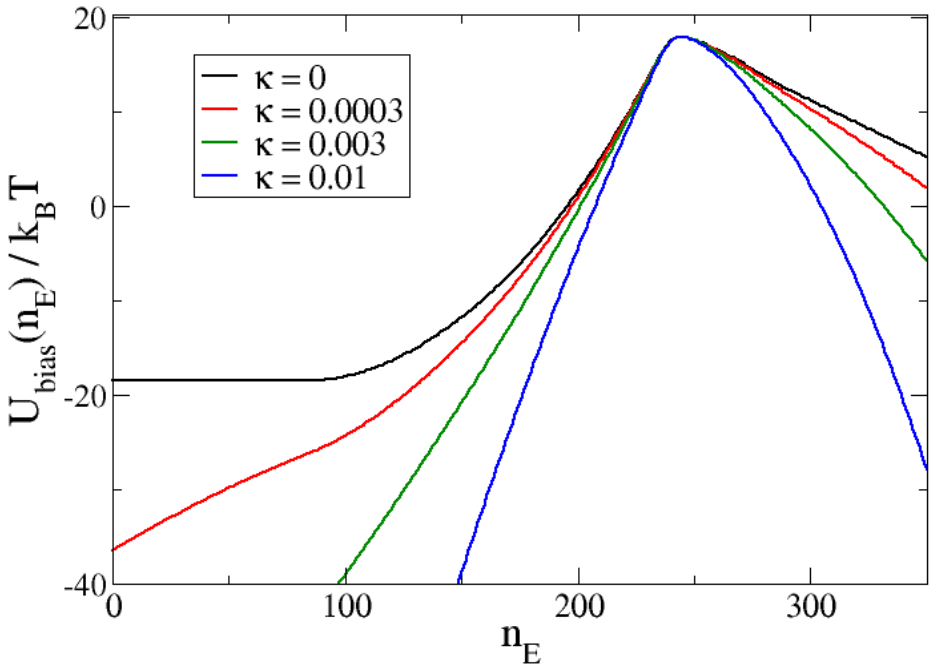


Figure 4.13: Weight files for the simulations where we have modified the biasing function in Figure 4.7 using (4.3.1).

4.3.2.3 Modified umbrella sampling

Figure 4.12 shows that we have obtained a reproducible estimate for the height of the barrier. To further verify that we have a reproducible result for the location and shape of the barrier peak, we bias the simulation using a hybrid feedback and umbrella sampling approach to restrict sampling to the vicinity of the barrier maximum. To do this, we subtract a quadratic function in n_E centred at the estimated barrier maximum in Figure 4.12, such that the new biasing function, $U^*_{\text{bias}}(n_E)$, is given by

$$U^*_{\text{bias}}(n_E) = U_{\text{bias}}(n_E) - \kappa(n_E - 245)^2, \tag{4.3.1}$$

where the parameter κ controls the strength of the quadratic. As for umbrella sampling, the large the value of κ, the narrower the sampling window centred at $n_E = 245$. We perform three simulation for $\kappa = 0.0003, 0.003, 0.01$. The resulting biasing functions are plotted in Figure 4.13.

The free energy barriers computed for each of the simulations are plotted in Fig-

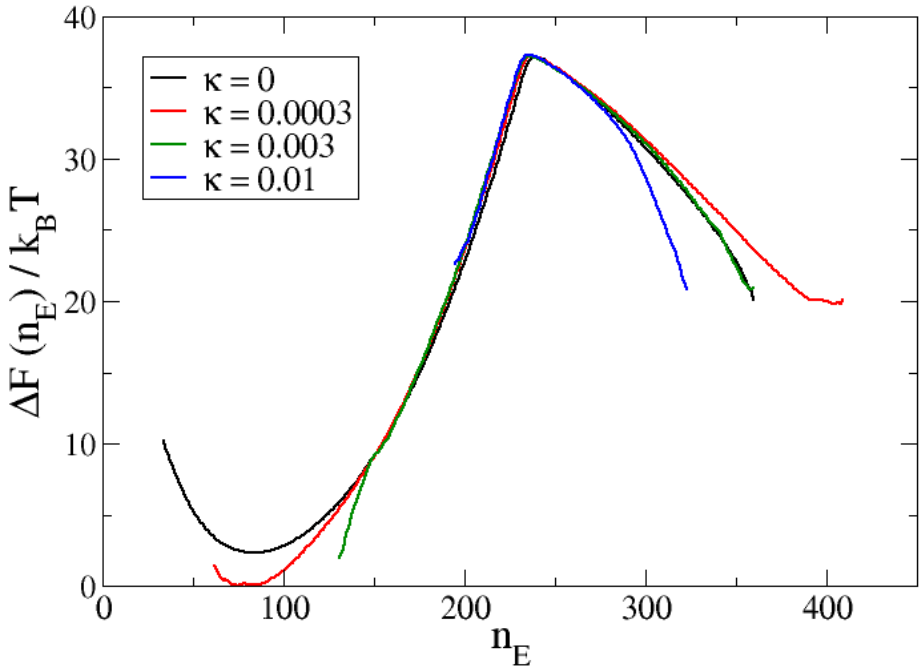
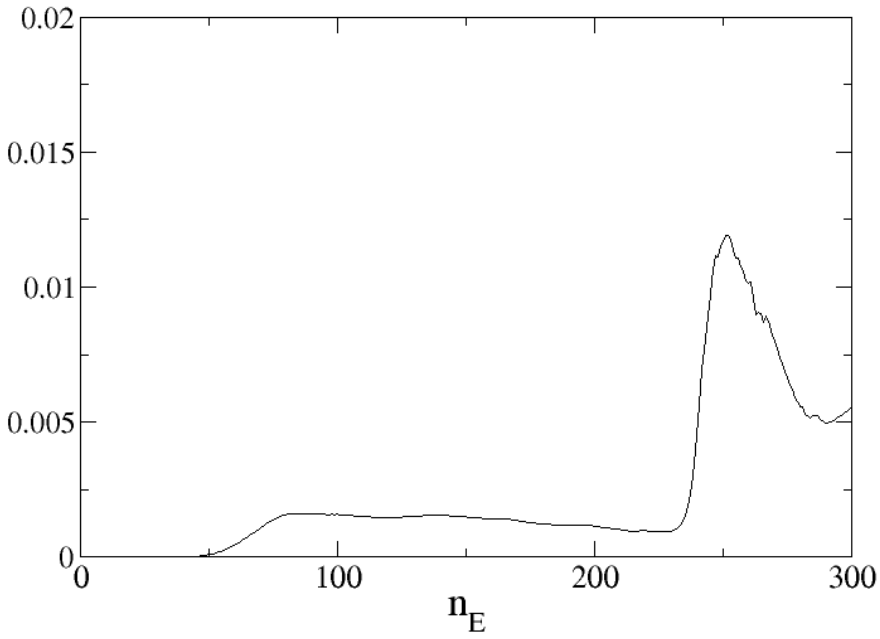


Figure 4.14: Free energy barriers for the square-well chain with $N = 128$, $\lambda = 1.05$, $T = 0.446$, using the biasing functions in Figure 4.13.

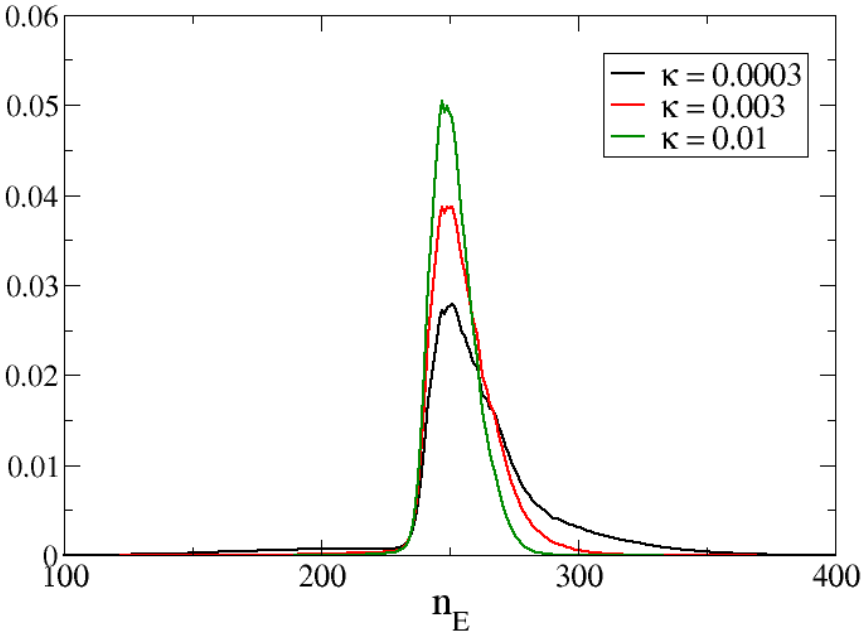
ure 4.14, where we have also plotted the original best estimate without umbrella sampling for comparison. This plot shows that we have strong reproducibility between independent simulations for the barrier location and shape. The corresponding state occupancy plots in Figure 4.15 show that even the weak umbrella sampling has a significant effect on the biased occupancy of states, meaning we do not collect enough meaningful statistics near for states $n_E \approx 100$. However, the umbrella sampling has had the desired effect in that the majority of the sampling takes places at the barrier maximum, so we have a reliable estimate of the free energy for $n_E \approx 245$.

Finally, the configuration swaps trace plots for the four simulations plotted in Figure 4.16 show that the simulation have good mixing in all cases. Furthermore, as we increase κ, we see that the amplitude of the oscillations of each of the trace plots decreases, meaning that the sampling windows get increasingly narrower without trapping the simulation in any particular state.

For completeness, we also include the free energy plots and configuration swaps

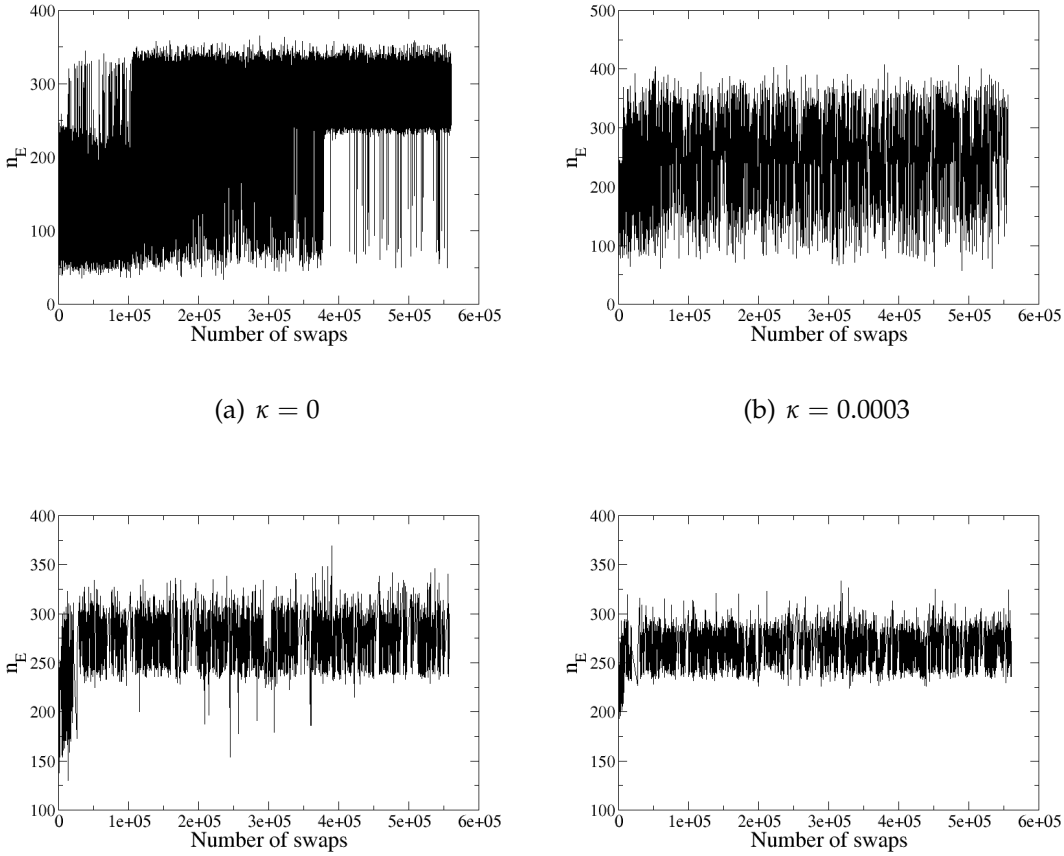


(a) $\kappa = 0$



(b) $\kappa > 0$

Figure 4.15: Histograms of state occupancies for the simulations to compute the free energy barriers in Figure 4.14.



(a) $\kappa = 0$

(b) $\kappa = 0.0003$

(c) $\kappa = 0.003$

(d) $\kappa = 0.01$

Figure 4.16: Trace plots of configuration swaps for the simulations computing the free energy barriers in Figure 4.14.

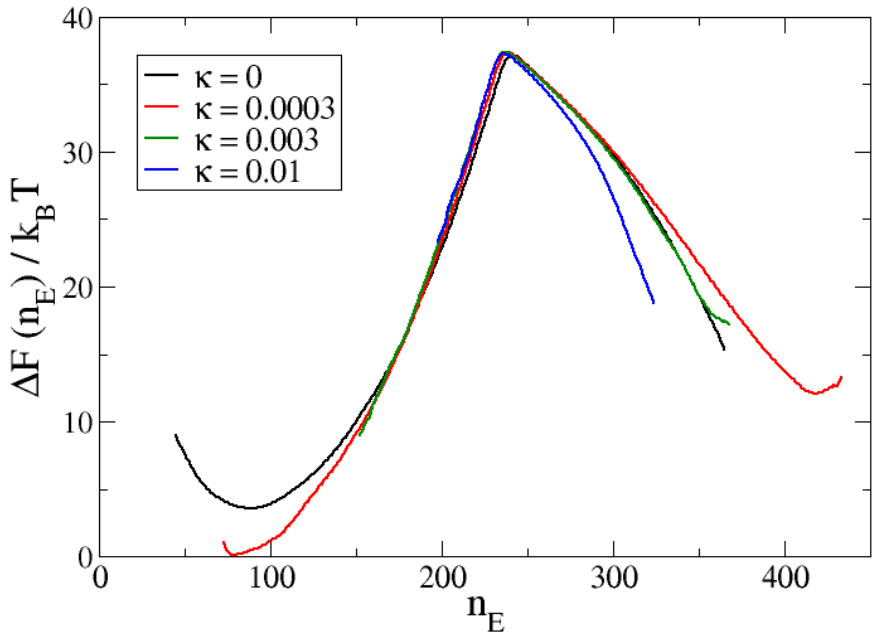


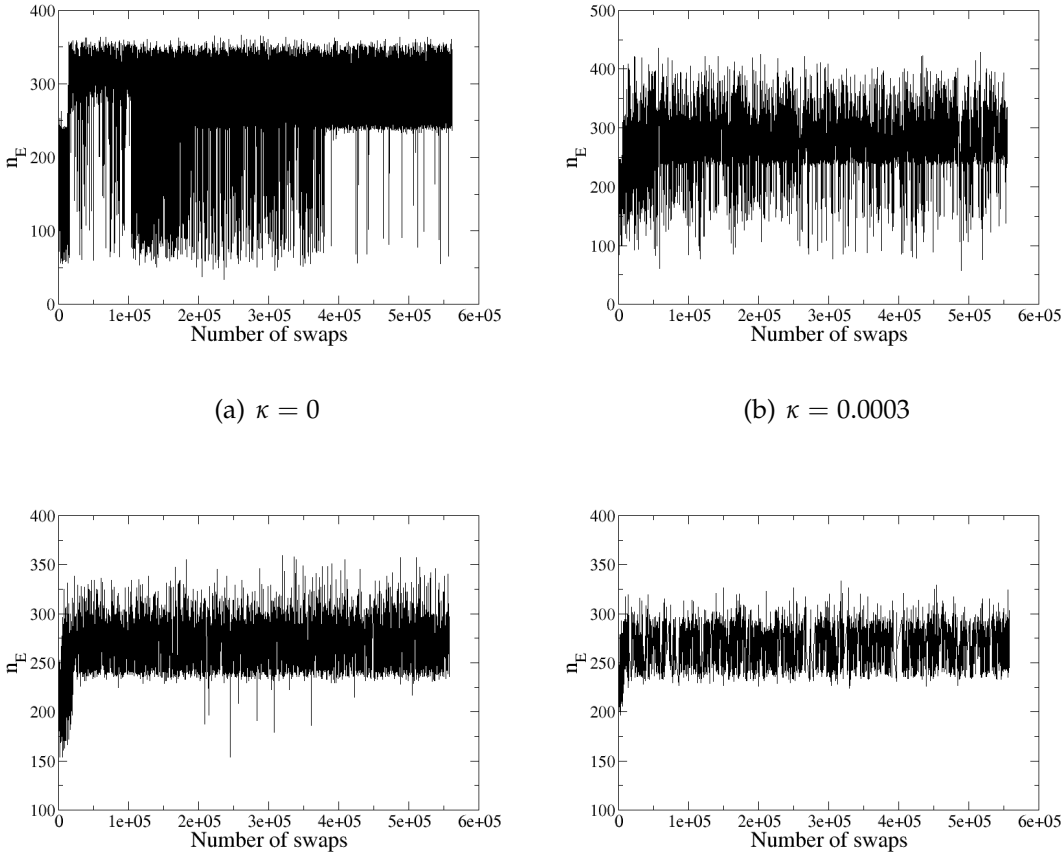
Figure 4.17: Free energy barriers for the square-well chain with $N = 128$, $\lambda = 1.05$, $T = 0.436$, using the biasing functions in Figure 4.13.

trace plots for $T = 0.436$ in Figures 4.17 and Figure 4.18. These plots show that we have similarly good agreement between each of the simulations, all of which have strong evidence of good mixing.

These results show that simulating using biasing with a feedback mechanism with parallel tempering is an effective algorithm for computing the free energy of a single chain with a steep energy barrier. The parallel tempering is used to mitigate the high likelihood of the simulation getting "trapped" in particular configurations with high n_E by swapping configurations with higher temperatures that have a good chance of exploring the coil states with low n_E.

4.3.3 Summary

To simulate the nucleation barrier of a single chain, there are inherent difficulties related to mixing that arise due to the connectivity of particles, which leads to self-entanglement and excluded volume. We tackle these issues using parallel tempering with a non-standard choice of temperature range, where



(a) $\kappa = 0$

(b) $\kappa = 0.0003$

(c) $\kappa = 0.003$

(d) $\kappa = 0.01$

Figure 4.18: Trace plots of configuration swaps for the simulations computing the free energy barriers in Figure 4.17.

we simulate systems at temperatures *lower* than the temperature of interest, $\overline{T}$, and increase the density of temperatures close to $\overline{T}$. We prescribe strict conditions that are monitored throughout the simulation to ensure good mixing of the simulation. These include the monitoring of the acceptance ratios of proposed configuration swaps between adjacent tempering levels, evolution of the energy traces after attempted configuration swaps, and ensuring effective decorrelation of configurations through the reptation number. Combining these techniques, we use the feedback algorithm to determine the optimum biasing, leading to a well-mixed and reproducible simulation that yields a nucleation barrier with a height $20k_{\mathrm{B}}T$ higher than the literature value [Taylor et al., 2009b]. We also develop a hybrid biasing strategy, where we combine our feedback algorithm with umbrella sampling. We use this biasing to verify that the shape and position of the barrier peak is highly reproducible under different biasing functions.

4.4 Order parameter comparison

In §4.2.3, we discussed the use of two different order parameters. We have produced results in this chapter using n_E as an order parameter to make comparisons between results in the literature and predictions from our model. However, more detailed order parameters are available, such as the Steinhardt-Nelson n_{C} to detect crystal geometries of particles with many neighbours. We provide results showing the correlation between the two choices of order parameter, n_E and n_{C}.

For a given crystal configuration, we could compare n_E and n_{C} in a global sense by reporting the integer value of each order parameter to determine whether they both detect a similar level of crystal order of the whole configuration. We gain a greater insight into the discrepancy of the two order parameters if, for a given configuration, we compare the classification of near neighbours of individual particles, i. A near-neighbour of i is a particle j with $r_{ij} < 1.05\sigma$, where r_{ij} is the distance between particles i and j. In addition, we define the following quantities:

$N_{\mathrm{n}}(i)$: The number of near-neighbours of particle i.

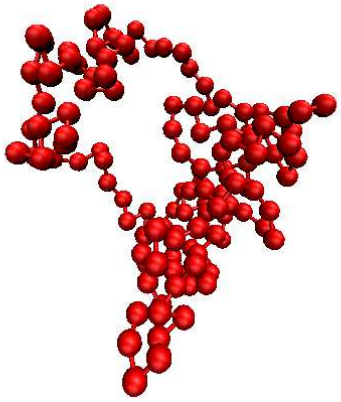
$N_C(i)$: The number of crystal neighbours, as determined by the Steinhardt-Nelson order parameter.

For each particle i in a configuration, we compute $N_n - N_C$, which is a measure of the inaccuracy of using n_E as an order parameter instead of n_C.

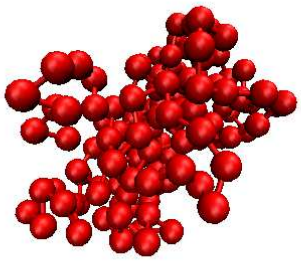
We take three snapshots of a chain of length $N = 128$ particles, whose nucleation is simulated at reduced temperature $T = 0.446$. The three configurations are chosen such that $n_E = 100, 200, 434$, since each of these configurations represent a distinct part of the free-energy landscape. The configuration with $n_E = 100$ corresponds with a melted coil state. The configuration with $n_E = 200$ represents a state near the top of the nucleation barrier, which could be an intermediate state between the melted and crystallised states. Finally, the configuration with $n_E = 434$ corresponds with a well-ordered crystal state. The three chosen configurations are illustrated in Figure 4.19.

In Figure 4.20, we inspect the near neighbours of each particle, i, in a configuration, comparing $N_n(i)$ with with $N_n(i) - N_C(i)$. Effectively, this is a comparison of the number of crystal and **non**-crystal neighbours of particle i. The number of non-crystal neighbours is a measure of the discrepency between the two order parameters n_E and n_C. The smaller the discrepancy, the more correlated the two order parameters are at a local level, so the number of near-neighbours, N_n is a good indicator for the number of crystal neighbours, N_c.

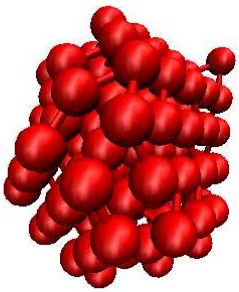
Figure 4.20(c) shows that the two order parameters are strongly correlated for high n_E, since there are very few particles for which there is significant disagreement between N_n and N_c. In particular there are many particles with $N_n = N_c = 12$. Since the square-well interactions are short-ranged ($\lambda = 1.05$), the maximum number of near-neighbours a spherical particle, i, can have is 12, with 6 neighbours arranged in a planar hexagon centred at i, then 3 in planes above and below the hexagon arranged such that their distance from i is minimised. The particles can only be in such an arrangement if they are in face-centred cubic or hexagonal close-packed crystal order (see §1.2.3). For high n_E, we expect many particles to be in crystal order, since this is the only way to achieve enough square-well interaction that contribute to the sum in (4.2.4). Hence, we can conclude that the parameter n_E is an effective, computationally cheap, alternative to the Steinhardt-Nelson parameter n_C.



(a) $n_E = 100$

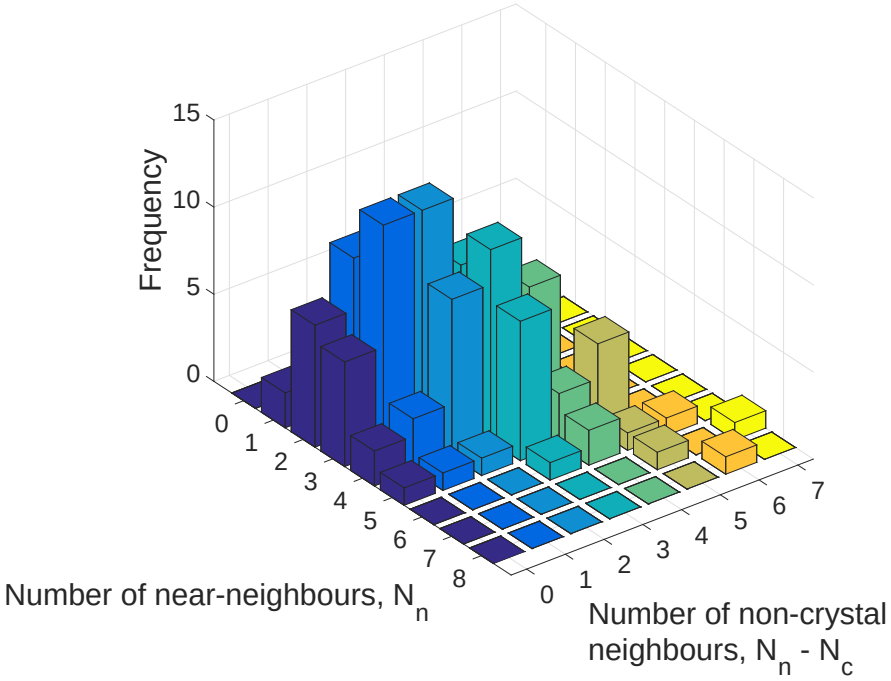


(b) $n_E = 200$

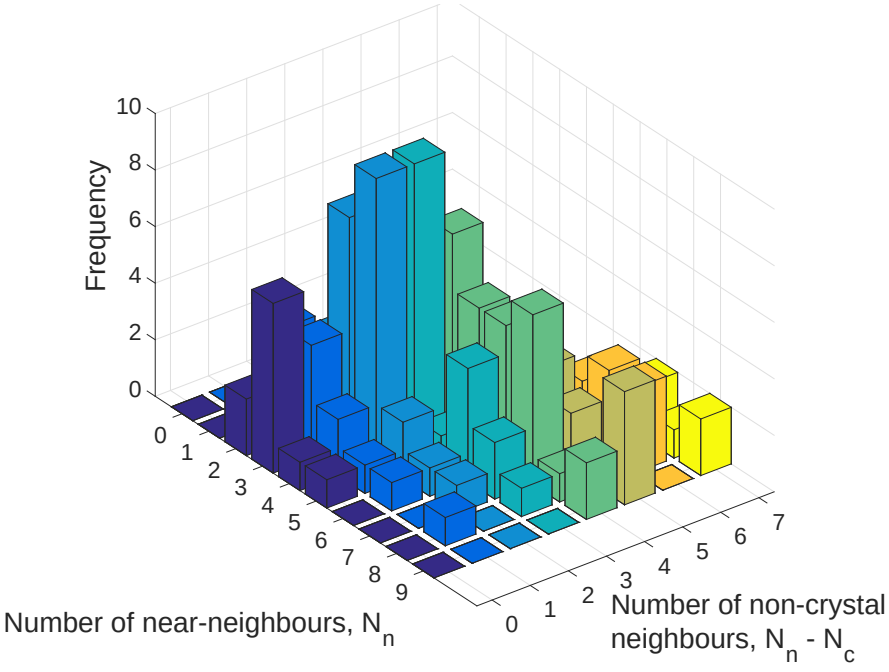


(c) $n_E = 434$

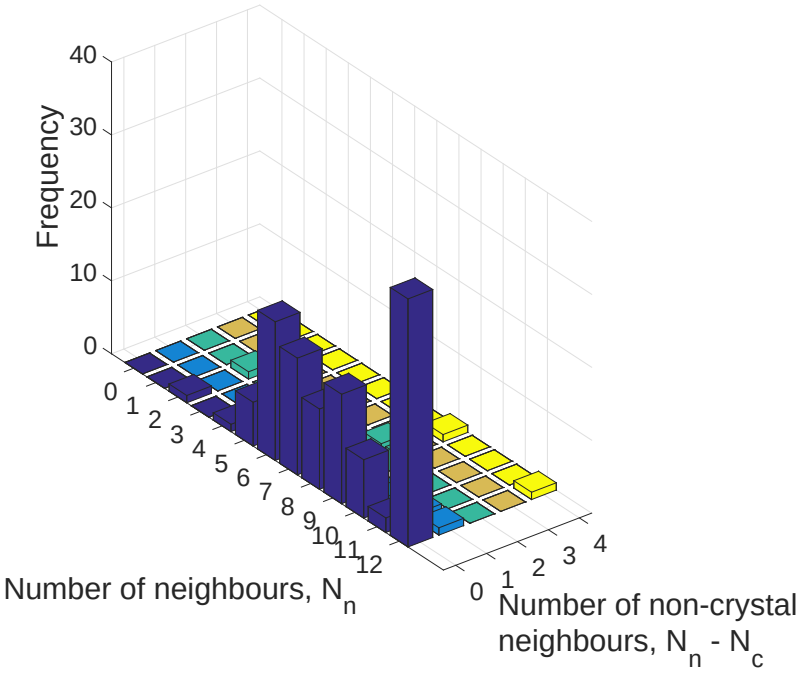
Figure 4.19: 3D representations of configurations of a single polymer chain containing $N = 128$ particles, with energy parameter n_E as labelled.



(a) $n_E = 100$



(b) $n_E = 200$



(c) $n_E = 434$

Figure 4.20: Three-dimensional histograms to compare the order parameters n_E and n_C for states $n_E = 100, 200, 434$. For each particle, we compute N_n and N_c, the number of neighbours and crystal neighbours respectively and plot points $(x, y) = (N_n, N_n - N_c)$ for each particle, with the frequency of each (x, y) on the z-axis.

For smaller values of n_E, we see in Figures 4.20(a) and 4.20(b) that the correlation of N_n and N_c is considerably weaker. In the case of $n_E = 100$, most particles have 1 to 4 neighbours, and typically at least one of them is not a crystal neighbour. This is to be expected, since $n_E = 100$ corresponds with the location of the first minimum of the free energy barrier for this system, which we have already identified as a melted state. Therefore, the contribution to the total energy is the overall effect of the particles being connected, since it is likely that many will be close enough to contribute to the total potential, but few particles will have many neighbours and so be in crystal order. We may still have confidence in using n_E as an alternative to n_C, since we observe that the first minimum of the free energy curve corresponds with sum of the interactions of the melted state.

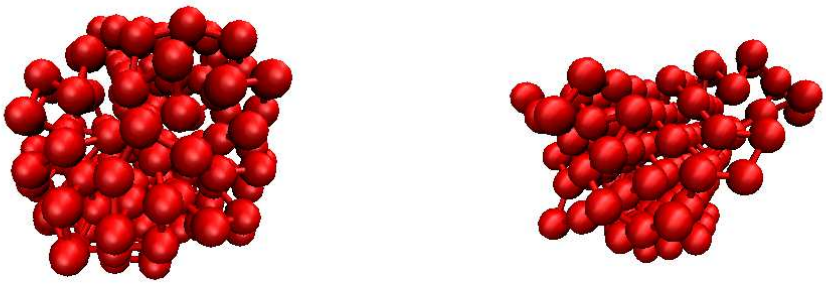
However, the significantly weak correlation of N_n and N_c in Figure 4.20(b) for $n_E = 200$ may pose problems for correctly resolving the maximum in the nucle-

ation barrier. Typically, most particles have very few crystal neighbours, even when they have as many as 9 near-neighbours. Thus, there are many contributions to n_E from particles that are not crystalline, resulting in the seemingly high order parameter n_E. However, if we were to count the total number of crystal neighbour interactions in the configuration, the order parameter would be much lower. Physically, the difference between the two order parameters is whether we count neighbours that are simply near to each other (n_E) or neighbours that are near **and** arranged in crystal order (n_C).

4.4.1 Detailed crystal order parameter

In the previous section, we identified a problem with using n_E as an order parameter for identifying intermediate states near the top of the nucleation barrier as a computationally cheap alternative to n_C, which takes into account the geometric coherence of each particle's near-neighbours. In our previous study of the Steinhardt-Nelson order parameter, we have gone a step further in defining the order parameter. Having defined the crystal neighbours, $N_c(i)$ of each particle i, we define *crystalline particles* as particles that have $N_c(i) \geq q$, where q is some threshold for the minimum number of crystal neighbours required. We then define the order parameter as the number of crystal particles in the configuration (see (2.2.2)). Leitold and Dellago [2014] use an adapted version of this order parameter that accounts for particles on the surface of a crystal nucleus having fewer crystal neighbours than those in the core. We describe this parameter in detail in §1.4.5.2, and refer to its integer output as n_C, the number of crystalline particles in the system.

We choose two configurations to study, whose energy order parameters are $n_E = 357, 358$. Since their interaction energies are similar, the order parameter n_E, computed directly from the total number of interactions, finds the configurations virtually indistinguishable. However, one can see from the images of the two configurations in Figure 4.21 that the configuration with $n_E = 357$ is an amorphous dense globule, while the other configuration with $n_E = 358$ is a crystal nucleus. We also see a significant difference between the two configurations when we consider the correlation of the particle near-neighbours and crystal neighbours in Figure 4.22. The amorphous state ($n_E = 357$) has a similar



(a) $n_E = 357$ (b) $n_E = 358$

Figure 4.21: 3D representations of configurations of a single polymer chain containing $N = 128$ particles, with $n_E = 357$ and $n_E = 358$. Configuration (a) is an amorphous globule, while (b) is a crystal nucleus, though the order parameter n_E identifies little difference between the two.

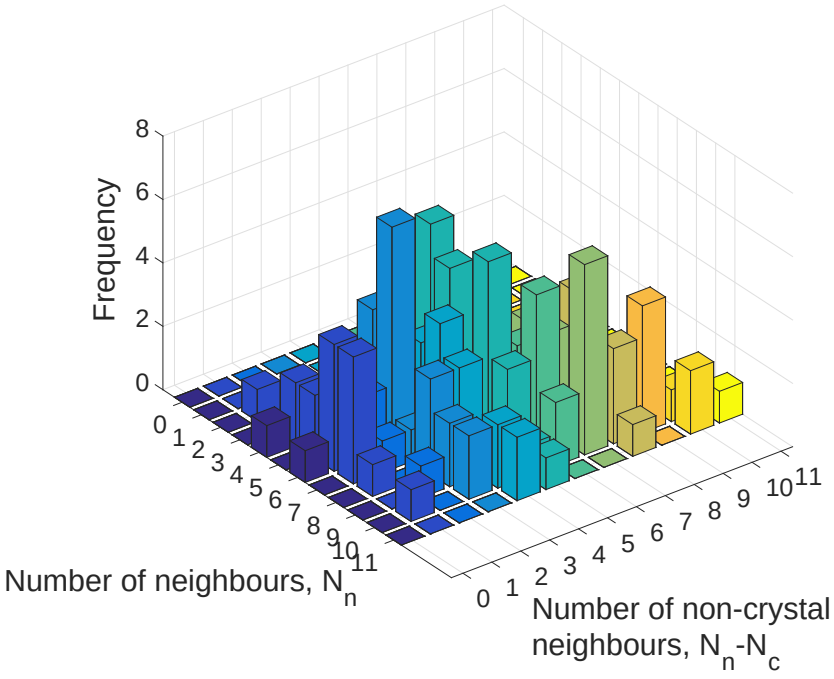
correlation plot to the decorrelated states in Figures 4.20(a) and 4.20(b), while the crystal state ($n_E = 358$) is strongly correlated, as was the case for $n_E = 434$ in Figure 4.20(c).

This suggests that we ought to use a more detailed order parameter to distinguish between intermediate states, such as n_C. We compute n_C for the configurations studied in Figures 4.21 and 4.22, and find that for these particular configurations,

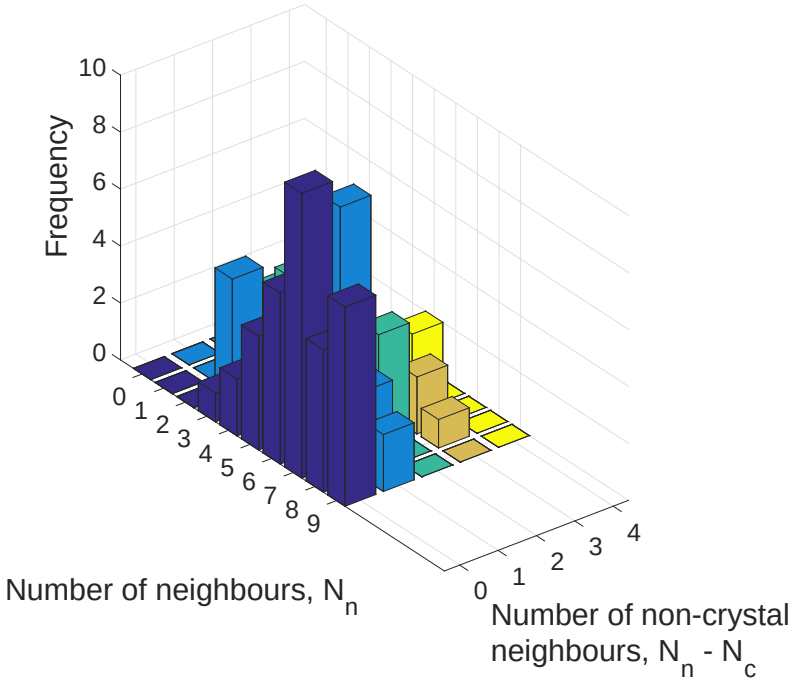
$$n_E = 357 \quad \Rightarrow \quad n_C = 13, \tag{4.4.1a}$$

$$n_E = 358 \quad \Rightarrow \quad n_C = 92. \tag{4.4.1b}$$

The Leitold and Dellago [2014] (LD) crystal order parameter reveals the significant difference between the crystal structures of the two configurations. In Figure 4.23, we illustrate the identification of crystal, coil and intermediate particles according to LD's algorithm. Coil particles contribute very little to the crystal nucleus structurally or energetically. Meanwhile, intermediate particles have enough near-neighbours to be considered dense and so contribute significantly to the square-well potential, but are not in crystal order. The contribution to the total square-well potential of the $n_E = 357$ configuration from the inter-

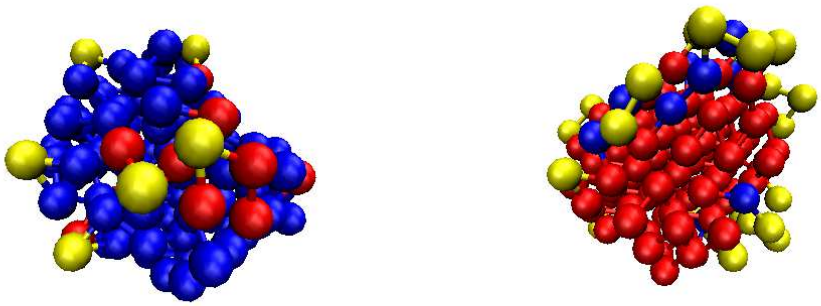


(a) $n_E = 357$



(b) $n_E = 358$

Figure 4.22: Three-dimensional histograms to compare configurations with almost identical energy order parameters, but significantly different crystal conformations.



(a) $n_E = 357$, $n_C = 13$ (b) $n_E = 358$, $n_C = 92$

Figure 4.23: 3D representations of configurations of a single polymer chain containing $N = 128$ particles, with $n_E = 357$ and $n_E = 358$. The particles have been colour coded according to the convention of Leitold and Dellago [2014]. Red particles are crystalline, blue particles are intermediate, and yellow particles are coil-like.

mediate particles is responsible for the high, but inappropriate order parameter n_E.

The discrepancy between the order parameters observed for the specific configurations in Figure 4.23 may be isolated cases and not typical configurations for $n_E = 357, 358$. We investigate whether this is the case by computing the distribution of the crystal order parameter, n_C for each value of n_E over a long run of the simulation of the $N = 128$ chain. Histograms for five values of n_E that span the state-space are plotted in Figure 4.24. We see for the extreme values of the energy order parameter in the coil and crystal regions ($n_E = 100$ and $n_E = 440$ respectively) the distributions of n_C are narrow and we can clearly distinguish one from the other. However, for values of n_E in the region of the barrier ($n_E \in [200, 350)$, the distributions are much wider, with many points overlapping. This means that the correlation between n_E and n_C is much weaker in this region. This is particularly prevalent in the case of $n_E = 200$, where there are cases of this state corresponding with $n_C \in [3, 62]$.

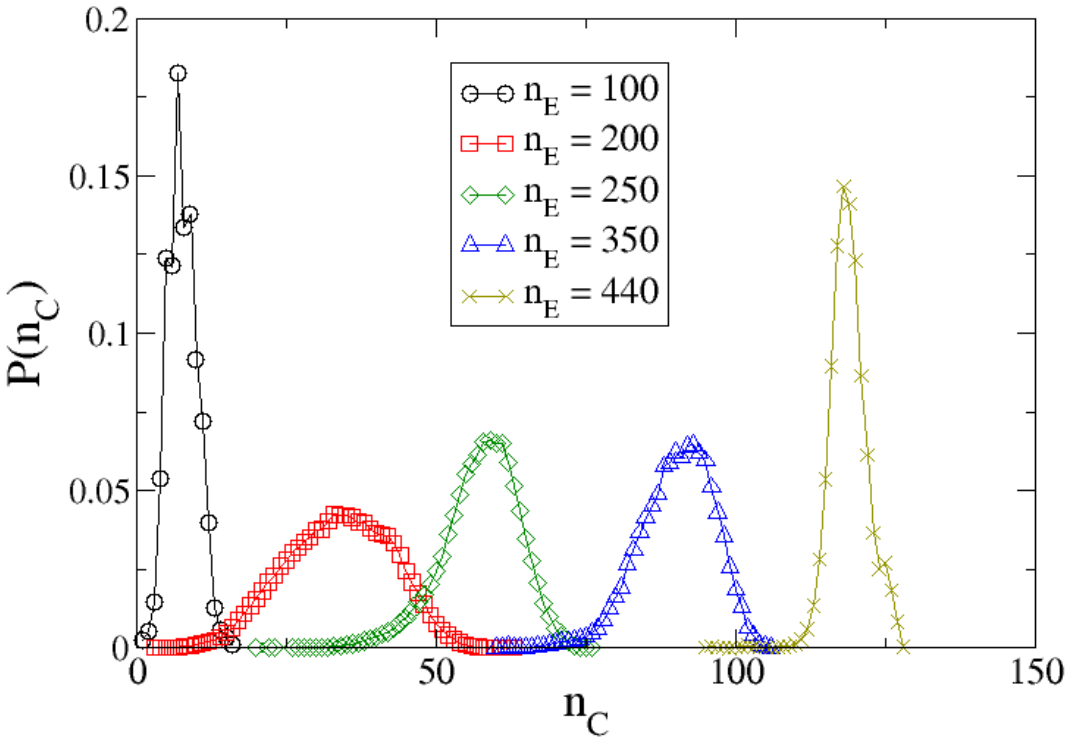


Figure 4.24: Histograms of the crystal order parameter, n_C for five fixed values of the energy order parameter, n_E.

4.4.2 Computing the occupancy of n_C using the occupancy of n_E

In Figure 4.24, we found that, for a given energy, $E_n = -n_E$, there is a distribution of configurations that have energy E. When $n_E \approx 200$, the configurations in the distribution have vastly different crystal configurations, as measured by the Steinhardt-Nelson order parameter n_C. Here, we show that the energy barrier for order parameter n_C can be extracted from our Wang-Landau approach. We define

$$P(n_C|E_n), \tag{4.4.2}$$

the probability of the system being in state n_C, given that the configuration has energy E_n (i.e. it is in energy state $-n_E$). We can obtain an estimate of the values of (4.4.2) from our simulation by regularly outputting configurations with the corresponding energy value E_n. An independent process can run in parallel to the main simulation, which reads in the configurations and computes the expensive Steinhardt-Nelson order parameter for each configuration. Proceeding in this way, we can obtain data for the distribution of n_C for each E_n without slowing down the main simulation. An important note to make is that (4.4.2) is independent of temperature, since we are working at fixed E_n. In a parallel tempered simulation, this means that we can use configurations from all temperatures to compute the distribution of n_C given E_n.

We can then compute $P(n_C, T_i)$, the probability of the system being in crystal state n_C at any temperature T_i using the Theorem of Total Probability, whence

$$P(n_C, T_i) = \sum_{n=1}^{N} P(E_n, T_i) P(n_C|E_n), \tag{4.4.3}$$

where N is the total number of available energy states, E_n, and $P(E_n, T_i)$ is the probability of the system being in energy state E_n at temperature T_i, which can be obtained directly from the simulation.

Hence, the discrete energy is an ideal order parameter, since it is cheap to compute, especially with the implementation of shortcuts in the computation of energy changes due to Monte Carlo moves to remove the N^2 scale factor with respect to system size. Furthermore, we are already required to compute E in the simulation to obtain the acceptance criterion for each Monte Carlo move, so

there is no additional computation required. The above results show that we can still use results obtained using n_E as an order parameter if we are interested in the detailed crystallisation of the system.

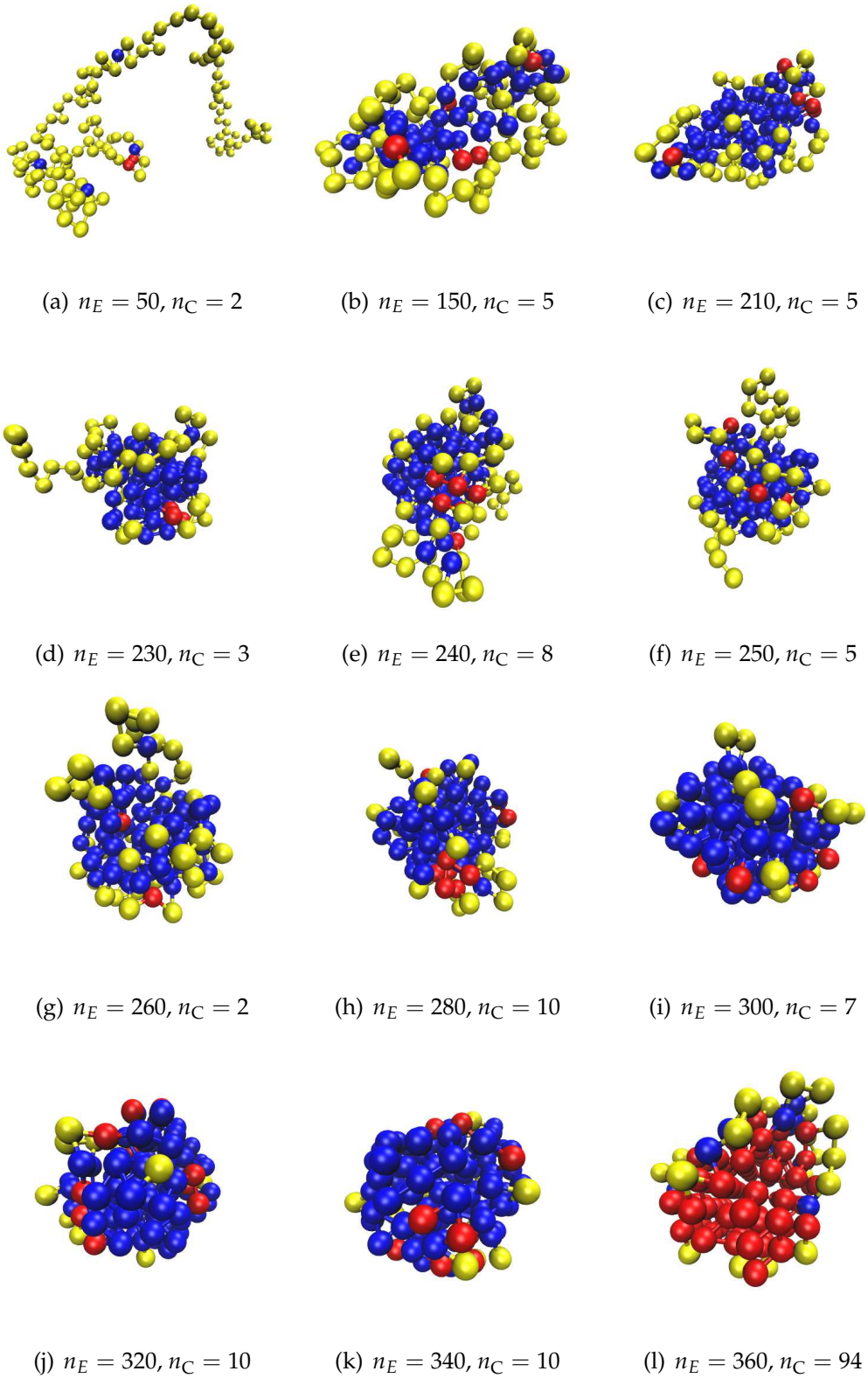
4.4.3 Summary

We compute two choices of order parameter for a single chain. One choice of parameter (n_E) is cheap to compute and just counts the number of close neighbours, while the second (n_C) is computationally expensive, but accounts for the geometric coherence of close neighbours. We found close correlation between the order parameters for the highly crystal state and reasonable correlation for the amorphous coil state. However, for intermediate states, we readily found anomalous configurations. To quantify the difference between order parameters, we plotted histograms of n_C for fixed values of n_E. This confirmed a close correlation for high and low n_E and that the average n_C increases with n_E. However, for intermediate n_E, the number of neighbours is not strongly correlated with the number of crystalline particles. Some additional snapshots of configurations and their corresponding values of n_E and n_C are provided in Figure 4.25.

4.5 Discussion

4.5.1 Free energy barrier calculation

We use our new simulation techniques to simulate the nucleation barrier investigated by Taylor et al. [2009b]. The two approaches are similar in that both use the same polymer model, with short-ranged square-well interactions, and the same set of moves proposed at each Monte Carlo step. However, our model uses a feedback mechanism to bias the acceptance probability of moves to achieve good sampling of available states of interest. Another crucial feature of our simulation is parallel tempering, which allows us to perform a simulation with good mixing, allowing frequent transitions between available states, enabling the simulation to explore the state space. In addition, parallel tempering provides us with the diagnostic tool to monitor the mixing of the simu-



(a) $n_E = 50, n_C = 2$ (b) $n_E = 150, n_C = 5$ (c) $n_E = 210, n_C = 5$

(d) $n_E = 230, n_C = 3$ (e) $n_E = 240, n_C = 8$ (f) $n_E = 250, n_C = 5$

(g) $n_E = 260, n_C = 2$ (h) $n_E = 280, n_C = 10$ (i) $n_E = 300, n_C = 7$

(j) $n_E = 320, n_C = 10$ (k) $n_E = 340, n_C = 10$ (l) $n_E = 360, n_C = 94$

Figure 4.25: Some additional representations of various crystal configurations, with their corresponding values of n_E and n_C as labelled. The colour coding is the same as that in Figure 4.23

lation, where we use energy trace plots from data generated when attempting configuration swaps between adjacent temperatures.

We found that achieving good exploration of the state space is particularly important when resolving the barrier maximum, since the resolution of the free energy in this reason is highly sensitive to the quality of sampling each side of the barrier. This is apparent when comparing our barrier with that of Taylor et al. [2009b], who estimate a shallower barrier with a critical nucleus corresponding with a smaller n_E. We found that the discrepancy emerges from the difficulty of estimating the free energy in a robust way near to the barrier maximum. Although the distribution of states sampled in this region can appear reasonable, the swap traces reveal that the simulation can become trapped in states on either side of the barrier, rarely proposing moves that explore the maximum.

4.5.2 Order parameter comparison

Our analysis comparing the two choices of order parameter suggest some weaknesses in using n_E as a reaction coordinate. We have significant evidence, particularly from Figure 4.24, that n_E is an effective order parameter, in that a configuration with high or low values of n_E are likely to have correspondingly high or low values of n_C respectively. However, for states near the maximum in the free energy barrier that is plotted as a function of n_E, there are much broader distributions for the corresponding values of n_C. Since $n_E = 200$ is a state near the top of the free energy barrier, according to our simulations, we expect to sample configurations that, given the right sequence of Monte Carlo steps, will find a configuration that will spontaneously crystallise. Without the presence of dynamics in our simulations, we do not know whether a pathway to a configuration with a large number of crystal interactions exists from the configuration with $n_E = 200$, pictured in Figure 4.19(b).

Therefore, it may be the case that the energy parameter n_E is useful as a cheap order parameter, since it can clearly distinguish between the melted and crystal phases, but is not useful as a reaction coordinate to inform us on the progress of the crystallisation, since it struggles to identify intermediate states. However, based on the analysis conducted in this chapter, we cannot conclude that n_C *is*

an effective reaction coordinate either. Given the distributions in the 3D histograms of §4.4, we see that if we were to repeat the experiment, and compute histograms of n_E for various fixed values of n_C, we would obtain a similar distribution plot to Figure 4.24. However, this analysis motivates the further study of various order parameters to measure the crystallisation of a single chain.

CHAPTER 5

Nucleation of a single polymer chain under stretch

5.1 Overview

In this chapter, we simulate stretch-induced nucleation by adapting the single, quiescent polymer chain model described in Chapter 4. To approximate the effect of flow, we isotropically stretch the single chain, then compare the resulting free energy barrier with the quiescent case. Stretching reduces the chain entropy, as it penalises coiled configurations. Thus, stretching lowers the entropic penalty for crystalisation, increasing the thermodynamic driving force for crystalisation and hence lowers the nucleation barrier. Clearly, if we stretch the chain too strongly, we inhibit nucleation, since folds are highly unfavourable in the energy landscape associated with the strongly stretched polymer. However, we expect sufficiently weak stretching to have a noticeable effect on the nucleation barrier, given previous study on flow-induced crystallisation for both experiments [Keller and Kolnaar, 1997] and simulation [Graham and Olmsted, 2009, Anwar et al., 2014].

5.2 Simulation details

We add stretching to our simulation by introducing a field conjugate to the square end-to-end vector. We use this field to control the degree of stretching. This approach has been previously used as a tool to model strain [Olmsted and

Milner, 1994] and flow-induced phase transitions [Graham and Olmsted, 2010].

We begin with the implementation of the Metropolis Monte Carlo method in Chapter 4, and provide a brief summary of the key components here. We simulate in the canonical ensemble, fixing the number of particles, N, the simulation cube volume, V, and temperature, T. Particles interact via the square-well potential (4.2.1). We use the scales (4.2.2) to nondimensionalise the simulation, such that

$$T = \frac{\varepsilon}{k_B} T^*, \quad U = \varepsilon U^*, \quad r_{ij} = \sigma r_{ij}^*, \quad \Delta F = k_B T \Delta F^*, \tag{5.2.1}$$

where a variable with a '*' is a nondimensional quantity. Here, ε and σ are parameters in the square-well potential (4.2.1), k_B is Boltzmann's constant, U is the potential energy, r_{ij} is the distance between particles i and j, and ΔF is the free energy. We compute the total dimensionless potential energy of a configuration by summing over the square well interactions (4.2.3),

$$E(\mathbf{r}) = \sum_{i=1}^{N} \sum_{j>i} U_{SW}(r_{ij}), \tag{5.2.2}$$

where $\mathbf{r}$ is the vector containing the configurational coordinates of the N particles in the chain. Since the square-well potential returns either 0 or -1, we can represent the total potential as

$$E(\mathbf{r}) = -n(\mathbf{r}), \tag{5.2.3}$$

where $n(\mathbf{r}) \in \mathbb{Z}$ is the number of square-well interactions in $\mathbf{r}$.

We carry out the Metropolis method by initialising the chain using crystal configurations captured from previous simulations, then proposing crank-shaft, bond-pivot, reptation, end rotation and end bridging moves in the ratio 6:1:1:1:1 respectively. Details of the implementation of each move-type are provided in §1.4.4. States are defined using the energy order parameter, $n_E(\mathbf{r})$ (§4.2.3), where

$$n_E(\mathbf{r}) = -E(\mathbf{r}). \tag{5.2.4}$$

Monte Carlo moves from state m to n are accepted with probability

$$P_{ac}(m \to n) = \min[1, \Gamma \exp(-\beta \Delta E_{mn})], \tag{5.2.5}$$

where $\beta = 1/T$ is the inverse temperature, and $\Delta E_{mn} = E(\mathbf{r}^n) - E(\mathbf{r}^m)$ is the change in the total square-well potential as a results of moving from state m to

n. We use the parameter Γ to modify P_{ac} for end-bridging moves, such that

$$\Gamma = \begin{cases} \dfrac{b_m R_n}{b_n R_m}, & m \to n \text{ is an end bridge move,} \\ 1, & \text{otherwise,} \end{cases} \tag{5.2.6}$$

where b_m and b_n are the number of possible bridging neighbours in state m and n respectively, and R_m and R_n are the distances of the chosen bridging particle to the old and new end particles respectively. See §1.4.4.3.3 for details of the geometry and acceptance probability of end bridging moves.

We estimate the free energy of the system, ΔF, as a function of n_E using the average occupancy of states sampled in the simulation, $\langle P(n_E) \rangle$, and (1.3.33), where

$$\Delta F(n_E) = \text{constant} - \log \langle P(n_E) \rangle. \tag{5.2.7}$$

The constant in (5.2.7) is arbitrary, since we are interested in relative *changes* in the free energy. We choose a value for the constant to vertically translate a free energy curve computed in a simulation so that we can compare the nucleation barrier with other results on the same scale.

5.2.1 Biasing

We bias our simulation such that the resulting limiting distribution restricts sampling to a window, w, such that $n_E \in [N_{\min}(w), N_{\max}(w)]$. Inside the window, we bias using our best estimate of the free energy, computed using (5.2.7), so that each state is equally likely to be visited. The appropriate weight function to achieve this biased distribution is

$$W(n_E) = \exp[U_{\text{bias}}(n_E)], \tag{5.2.8}$$

where

$$U_{\text{bias}}(n_E, w) = \begin{cases} -\kappa_{\text{L}}(w)(n_E - N_{\min}(w))^2 + \Delta F(n_E), & n_E < N_{\min}(w), \\ \Delta F(n_E), & N_{\min}(w) \leq n_E \leq N_{\max}(w), \\ -\kappa_{\text{R}}(w)(n_E - N_{\max}(w))^2 + \Delta F(n_E), & n_E > N_{\max}(w). \end{cases} \tag{5.2.9}$$

Here, $\kappa_{\text{L}}(w)$ and $\kappa_{\text{R}}(w)$ are parameters that control the number of visits to states $n_E < N_{\min}(w)$ and $n_E > N_{\max}(w)$ respectively. The derivation of (5.2.9) is given

in §4.2.5. We refine our estimate of ΔF to define the biasing function inside the window using the feedback algorithm of §2.3.1. Monte Carlo moves from state m to n in the biased simulation are accepted with probability

$$P_{\text{ac}}^{w}(m \to n) = \min[1, \Gamma \exp(\Delta U_{\text{bias}}^{mn} - \beta \Delta E_{mn})], \tag{5.2.10}$$

where $\Delta U_{\text{bias}}^{mn} = U_{\text{bias}}(n_E^n, w) - U_{\text{bias}}(n_E^m, w)$ is the change in the biasing function (5.2.9) in moving from state m to n.

5.2.2 Parallel tempering

Simulating a single chain with stretching involves the same issues with mixing that were discussed in Chapter 4, so we implement parallel tempering using the algorithm in §4.2.7. In a parallel tempered simulation we simulate several copies of the single polymer chain simultaneously at different temperatures T_i, one of which is the temperature of interest, $\overline{T}$. We rescale the biasing function (5.2.8) using $\overline{T}$, such that

$$\overline{W}(n_E) = \exp[\overline{U}_{\text{bias}}(n_E)], \tag{5.2.11}$$

where

$$\overline{U}_{\text{bias}}(n_E) = \overline{T} U_{\text{bias}}(n_E). \tag{5.2.12}$$

Moves from state m to n of a simulation conducted at temperature T_i are accepted with probability

$$P_{\text{ac}}^{w}(m \to n)_i = \min[1, \Gamma \exp(\beta_i(\Delta \overline{U}_{\text{bias}}^{mn} - \Delta E_{mn}))], \tag{5.2.13}$$

where $\beta_i = 1/T_i$ is the inverse temperature. We periodically propose configuration swaps between systems with adjacent temperatures T_i, T_{i+1}. Swaps of configurations $\mathbf{r}_i$ and $\mathbf{r}_{i+1}$ from systems with inverse temperatures β_i and β_{i+1} are accepted with probability

$$P_{\text{ac}}^{w}(\beta_i \rightleftarrows \beta_{i+1}) = \min[1, \exp(\Delta\beta(\Delta \overline{U}_{\text{bias}} - \Delta E)], \tag{5.2.14}$$

where $\Delta\beta = \beta_{i+1} - \beta_i$, and $\Delta \overline{U}_{\text{bias}}$ and ΔE are respectively the changes in $\overline{U}_{\text{bias}}$ and the total square-well potential as a result of swapping configurations $\mathbf{r}_i$ and $\mathbf{r}_{i+1}$.

5.2.3 Implementation of stretching

We apply an isotropic stretching to the single chain by adding a term to the total potential that is proportional to the square of the length of the end-to-end vector of the chain,

$$\mathcal{R}^2 = |\mathbf{r}_N - \mathbf{r}_1|^2, \tag{5.2.15}$$

where $\mathbf{r}_1$ and $\mathbf{r}_N$ are the position vectors of the first and last beads of the chain respectively. We define the stretching potential,

$$E_S(\mathcal{R}) = \sigma\mathcal{R}^2, \tag{5.2.16}$$

where σ is the stretching coefficient. We modify (5.2.13) such that moves from state m to n in a simulation at temperature T_i are accepted with probability

$$P_{\text{ac}}^{w}(m \to n)_i = \min[1, \Gamma \exp(\beta_i(\Delta\overline{U}_{\text{bias}}^{mn} - \Delta E_{mn} - \Delta E_S^{mn}))], \tag{5.2.17}$$

where ΔE_S^{mn} is the change in E_S in moving from state m to n. We also modify (5.2.14) so that configuration swaps between adjacent temperatures are accepted with probability

$$P_{\text{ac}}^{w}(\beta_i \rightleftarrows \beta_{i+1}) = \min[1, \exp(\Delta\beta(\Delta\overline{U}_{\text{bias}} - \Delta E - \Delta E_S))], \tag{5.2.18}$$

where ΔE_S is the change in E_S as a result of swapping the configurations.

5.2.3.1 Computing the stretch ratio

We translate our stretching coefficient, σ, into a stretch ratio, which is physically meaningful. We do so by conducting an unbiased simulation of the single chain with a given value of σ. We compute the average square length of the end-to-end vector, $\langle\mathcal{R}^2(\sigma)\rangle$, over a long run of the simulation ($\approx 10^7$ Monte Carlo steps). The stretch ratio, Λ, is defined as

$$\Lambda(\sigma) = \sqrt{\frac{\langle\mathcal{R}^2(\sigma)\rangle}{\langle\mathcal{R}^2(0)\rangle}}, \tag{5.2.19}$$

where $\langle\mathcal{R}^2(0)\rangle$ is the average square length of the end-to-end vector of the unstretched chain ($\sigma = 0$).

5.2.3.2 Stretching the chain-centre

Stretching the chain by using a potential that is proportional to the end-to-end vector results in a significant physical change in conformation, even for modest values of σ (≈ 0.01). Significant stretching potentially inhibits the crystallisation of the chain. Hence, we propose a regime in which we stretch a short section at the centre of the chain by defining a stretching potential,

$$E_S(\mathbf{r}, c) = \sigma |\mathbf{r}_{N/2+c} - \mathbf{r}_{N/2-c}|^2, \tag{5.2.20}$$

where c is a constant that we use to control the length of the stretched segment at the centre of the chain. Applying the potential (5.2.20) allows the chain to nucleate about the stretched central segment. This mimics flow-induced stretching of polymers, in which chains stretch most strongly at the centre [Doi and Edwards, 1986].

5.3 Results

We simulate the nucleation barrier of the same system as that studied in §4.3.2, where we set the number of particles in the single chain to $N = 128$, the interaction length $\lambda = 1.05$ and the temperature of interest $\overline{T} = 0.446$. We bias the simulation using a sampling window with biasing function $W(n_E) = \exp[U_{\text{bias}}(n_E)]$, where $U_{\text{bias}}(n_E)$ is given by (5.2.9). We follow the method of §4.3.2.1, where we use a single sampling window ($w = 1$) with $N_{\text{min}}(1) = 0$, $N_{\text{max}}(1) = 250$, with $\kappa_R = 0.02$. We bias states within the window using the biasing function in Figure 4.7. Stretching the chain will change the free energy barrier, so this will not allow us to achieve uniform sampling of states within the window, but it is a good first approximation for the true barrier if the stretching is sufficiently weak. To improve the sampling, we can use the feedback algorithm, updating the biasing function using estimates of the free energy from long simulation runs.

We implement parallel tempering with twelve temperatures, using the temperature distribution in Table 4.2. Each simulation is initialised in state $n_E = 100$, where the initial configuration is generated using a snapshot from a previous simulation. We equilibrate the simulation for 5×10^4 Monte Carlo steps, then

Table 5.1: Corresponding stretch ratios, Λ, corresponding to the stretching coefficients, σ, computed using (5.2.19).

Stretching coefficient, σ	Stretch ratio, Λ
0.000	1.00
0.001	1.11
0.002	1.28
0.003	1.62
0.005	4.61

begin collecting statistics to compute the free energy from the unbiased occupancy of states using (5.2.7), where we use (1.3.37) to obtain the occupancy in the unbiased distribution. Swap cycles occur every 5×10^5 Monte Carlo steps, in which each parallel simulation is paused. We then attempt fifty configuration swaps between adjacent temperatures T_i and T_{i+1}, where $i = 1, 2, \ldots, 11$ is chosen randomly. This process is repeated for several thousand swap cycles to obtain an accurate estimate of the free energy barrier.

5.3.1 End-to-end stretching

Our first experiment is to apply end-to-end stretching using (5.2.16). Since we are applying a stretching potential that is proportional to the square length of the end-to-end vector, the simulation is sensitive to small changes in the stretching coefficient, σ. We translate the stretching coefficient values to the stretch ratio, Λ, using (5.2.19), where we obtain the averages $\langle \mathcal{R}(\sigma) \rangle$ and $\langle \mathcal{R}(0) \rangle$ over long, unbiased runs of the simulation with all parameters set to the values stated above. The stretch ratios corresponding to the values of σ used in our simulations are summarised in Table 5.1.

The free energy estimates from each of the simulations with the stretch ratios listed in Table 5.1 are plotted in Figure 5.1. The free energy barrier for $\Lambda = 0.001$ shows that very weak end-to-end stretching has a significant effect on the barrier. The height is reduced by approximately $8k_{\mathrm{B}}T$ and the location of the maximum is shifted to $n_E \approx 210$. Energetically, it is easier for the simulation to cross the barrier, so weak stretching appears to aid the nucleation of the chain. Increasing the level of stretching, makes the nucleation increasingly more dif-

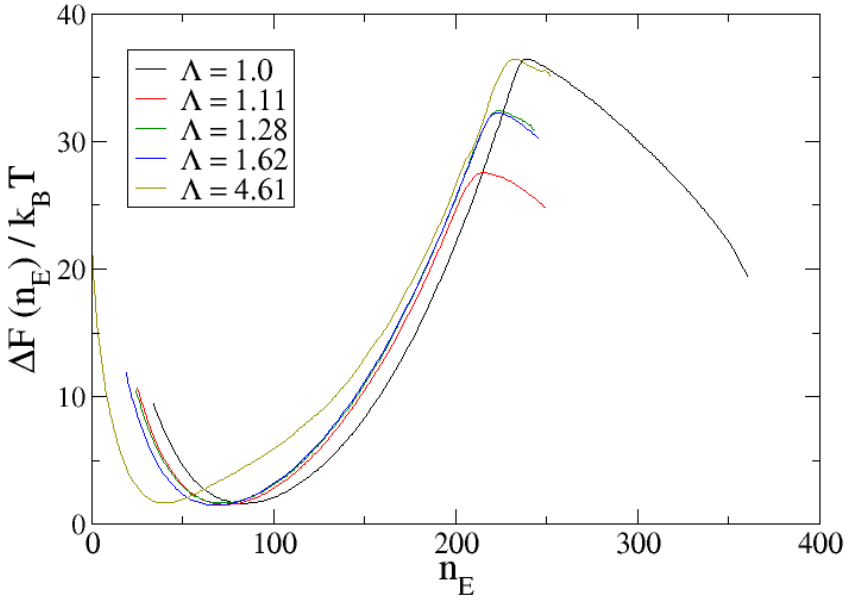
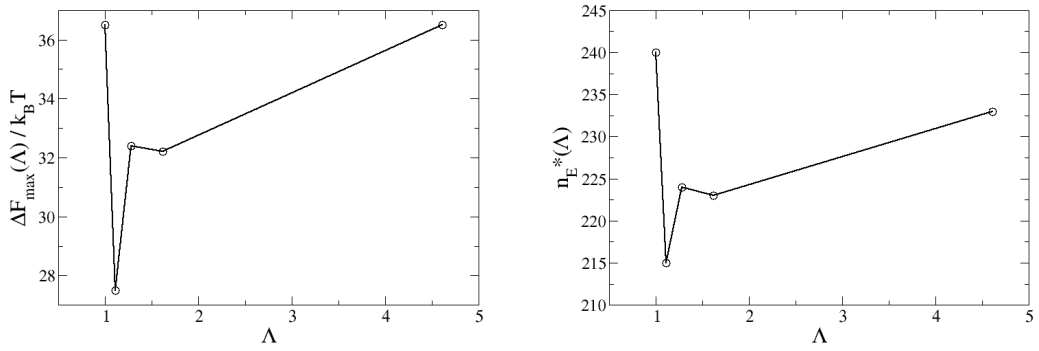


Figure 5.1: Free energy barriers for the square-well chain with $N = 128$, $\lambda = 1.05$, $T = 0.446$, for various different stretch ratios, Λ. The curve with $\Lambda = 1.0$ corresponds with the result for the quiescent chain from Chapter 4.

ficult, where the curve for $\Lambda = 4.61$ has a maximum that is similar to the unstretched chain. Another feature is that increasing the level of stretching shifts the location of the minimum to smaller values of n_E. We expect this to be the case, since stretching the chain will encourage the unfolded coil states to have fewer square-well interactions on average, resulting in the shift. In Figure 5.2, we have plotted the heights, $\Delta F_{\max}$, and locations, n_E^*, of the five free energy barriers in Figure 5.1. Figure 5.2(a) show the significant effect of weak end-to-end stretching on the barrier height. However, as the strength of stretching increases, $\Delta F_{\max}$increases back to that of the unstretched height. We see a similar behaviour in the location of n_E, where weak stretching causes n_E^* to decrease, but increasing Λ causes n_E^* to increase .

The configuration swaps trace plots for the end-to-end stretching simulations are plotted in Figure 5.3. These show that the simulation mixes well for weak stretching, but the frequency of traversals from one side of the barrier to the other is reduced for the simulation $\Lambda = 4.61$. The lack of mixing for strong



(a) $\Delta F_{\max}(\Lambda)$. (b) $n_E^*(\Lambda)$.

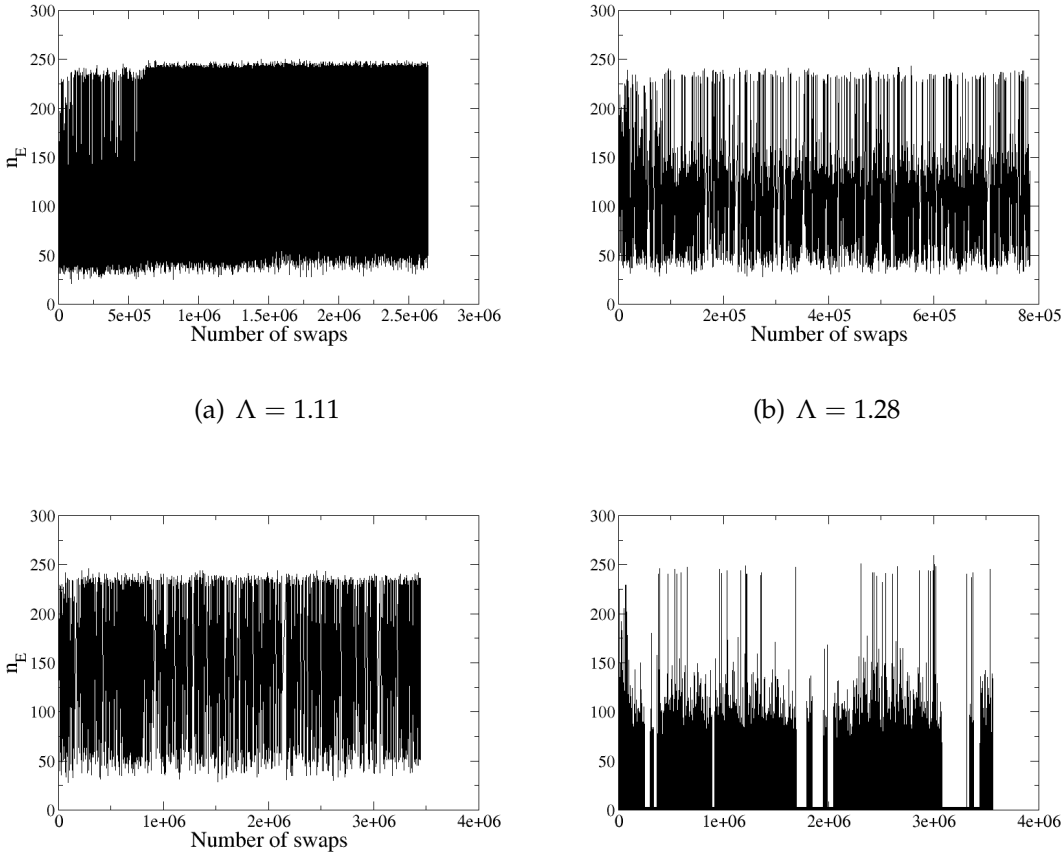
Figure 5.2: Plots of the barrier height, $\Delta F_{\max}$, and location, n_E^* as functions of stretch.

stretching is a symptom of nucleation being inhibited in this case. Since we bias the simulation using the free energy barrier for $\Lambda = 1.0$, we over-bias states in the vicinity of the barrier peak for $\Lambda = 1.11$. This, combined with $N_{\max} = 250$ enables us to obtain good sampling of states in the vicinity of the barrier peak, yielding a good resolution of the free energy maximum.

5.3.2 Chain-centre stretching

In the previous section, we found that strong end-to-end stretching has a significant destabilising effect on the nucleation barrier, even for very weak stretching. Hence, we simulate using chain-centre stretching (see §5.2.3.2), which has a much weaker effect on the end-to-end vector. We expect that stretching the chain-centre has the effect of creating a straight segment of chain that provides a core for the growth of a crystal nucleus. Since we are stretching on a restricted interval, we can apply stronger stretching without completely inhibiting nucleation, as is the case for end-to-end stretching. Stretching the chain centre also mimics flow-induced stretching, so has some value for approximating models of polymers under flow.

We implement chain-centre stretching by adding the stretching potential $E_S(\mathbf{r}, c)$ defined in (5.2.20) to the total potential used for accepting moves and configuration swaps (see §5.2.3). We choose $c = 2$, since we expect the ideal crystallite for a chain containing $N = 128$ particles to approximate a $5 \times 5 \times 5$ cubic lat-



(a) $\Lambda = 1.11$

(b) $\Lambda = 1.28$

(c) $\Lambda = 1.62$

(d) $\Lambda = 4.61$

Figure 5.3: Trace plots of configuration swaps for the simulations computing the free energy barriers in Figure 5.1.

tice. Hence, we expect that imposing a stretching potential that encourages the formation of a straight segment of length five will aid the generation of such a crystallite.

We simulate using the same stretching coefficients, σ, as in the previous section with end-to-end stretching, namely $\sigma = 0.001, 0.002, 0.003, 0.005$. We also run a simulation with $\sigma = 0.1$ to test the hypothesis that we can stretching the chain-centre more strongly than the end-to-end vector. Computing the effective stretch ratios using (5.2.19) with the stated values of σ shows that the simulations are indistinguishable in terms of Λ, since all stretch ratios computed were not significantly different to one. Hence, we distinguish between the simulations using the stretching coefficient, σ.

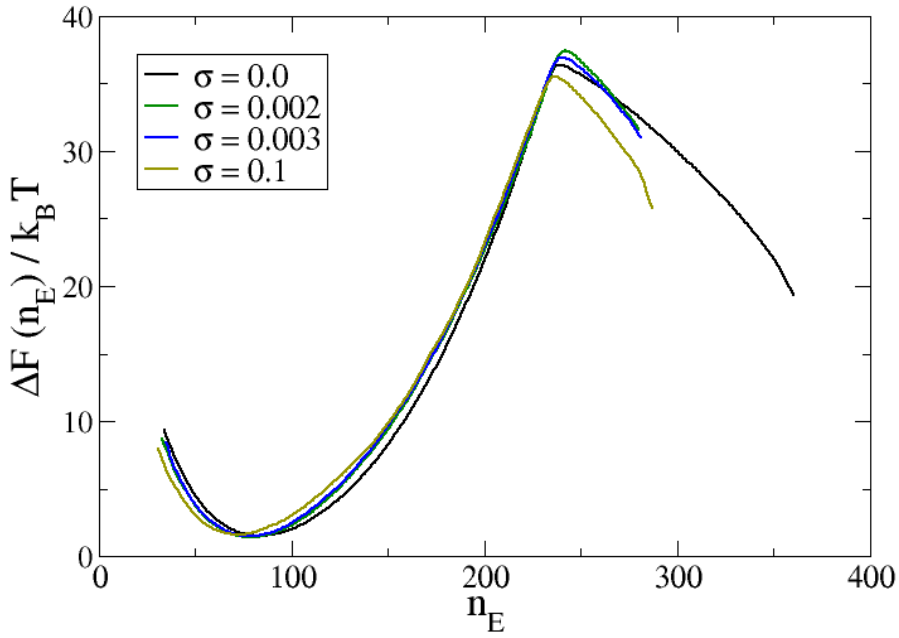


Figure 5.4: Free energy barriers for the square-well chain with $N = 128$, $\lambda = 1.05$, $T = 0.446$, for various different stretching coefficients, σ. The curve with $\sigma = 0.0$ corresponds with the result for the quiescent chain from Chapter 4.

The free energy curves computed in each of the simulations is plotted in Figure 5.4. We see that there is little effect on the quiescent free energy barrier for weak stretching of the chain-centre. We expect such results, since a weak stretching of a short section of the chain far from the the ends will have little influence on the

chain conformation. In the case where we apply strong stretching of the same section, however, we see a similar feature to that in Figure 5.1 for weak end-to-end stretching. The stretching has the effect of lowering the barrier height and shifting the location of the barrier minimum in the negative n_E direction. The minimum is also shifted in the negative n_E direction, so the chain-centre stretching has some effect on the average total square-well interactions of the coil state, but an insignificant effect on the average end-to-end vector for these values of σ. We see a similar picture here, in that very small changes in Λ can alter the barrier, potentially leading to dramatic changes in the nucleation rate.

5.4 Discussion

Applying a stretching potential to a single polymer chain simulation approximates the effect of simulating a chain under flow to observe flow-induced crystallisation. Our preliminary results show that there is evidence that stretching the chain reduces the energy cost of crossing the nucleation barrier as a function of n_E, and thus aids the nucleation process. Although end-to-end stretching is the simplest method for stretching the chain, its implementation allowed us to observe that small changes in the stretching coefficient, σ, result in significant changes to both the stretch ratio, Λ, and the nucleation barrier. However, strong end-to-end stretching inhibits the nucleation. Chain-centre stretching, though less effective in changing the global conformation of the chain, has the advantage that we can apply a much stronger stretching coefficient to lower the height of the nucleation barrier with less risk of eliminating the nucleation process entirely.

We note that for a very modest level of stretching ($\approx$ 10%), we see a significant reduction in the nucleation barrier, which we would expect to drastically increase the nucleation rate. This explains an observation by Graham and Olmsted [2009] concerning polymer nucleation measurements by Coccorullo et al. [2008] on a flowing polydisperse melt. They noted that, even though only a very small fraction of the melt's chains were strongly stretched in the flow experiments, there was a dramatic increase in nucleation rate. Our result suggests that chain lengths that were only weakly stretched (constituting a much larger fraction of the overall molecular weight distribution than chains that

were strongly stretched) would contribute significantly to the increased nucleation rate.

CHAPTER 6

Conclusions and Future Work

6.1 Summary

The aim of this thesis is to develop molecular simulation algorithms to resolve nucleation barriers for free particles and single polymer chains. In Chapter 2, we developed a Metropolis Monte Carlo simulation, which uses an automated feedback algorithm to bias the sampling of rare crystal states. Each iteration of the feedback algorithm updates the biasing function using the most recent estimate of the free energy curve. Using the feedback technique allows us to resolve steep nucleation barriers without either a priori knowledge of the true barrier or the need to provide an initial guess. We further modify our biasing function to simulate in restricted sampling windows, each of which can be run simultaneously. This allows us to obtain a good resolution of the barrier in each window, then retrospectively construct the overall free energy barrier. We use a free particle model in Chapter 3 to test our implementation of the feedback algorithm and sampling windows developed in Chapter 2.

In Chapter 4, we use the new simulation techniques developed in Chapters 2 and 3 to simulate the nucleation of a single, freely-jointed square-well polymer chain. There are two inherent difficulties with resolving the nucleation barrier for this system: sampling rare configurations and mixing well to overcome locally trapped configurations. We bias the simulation using the feedback algorithm to overcome the issue of the rare sampling of high crystal states of the single chain. In this model, mixing issues arise due to the connectivity of particles, which leads to self-entanglement and excluded volume. We overcome

these issues using parallel tempering with a non-standard choice of temperature range, where we simulate systems at temperatures *lower* than the temperature of interest, $\overline{T}$, and increase the density of temperatures close to $\overline{T}$ (see Table 4.2). We stipulate careful criteria to ensure good mixing of the simulation. These include the monitoring of the acceptance ratios of proposed configuration swaps between adjacent tempering levels, evolution of the energy traces after attempted configuration swaps, and ensuring effective de-correlation of configurations through the reptation number. Combining these techniques, we use the feedback algorithm to determine the optimum biasing, leading to a well-mixed and reproducible simulation that yields a nucleation barrier with a height $20k_BT$ higher than the literature value [Taylor et al., 2009b]. We also develop a hybrid biasing strategy, where we combine our feedback algorithm with umbrella sampling. We use this biasing to verify that the shape and position of the barrier peak is highly reproducible under different biasing functions.

We also investigate the correlation of two choices of order parameter for the single chain model in Chapter 4. One choice of parameter (n_E) is cheap to compute and just counts the number of close neighbours, while the second (n_C) is computationally expensive, but accounts for the geometric coherence of close neighbours. We found close correlation between the order parameters for the highly crystal state and reasonable correlation for the amorphous coil state. However, for intermediate states, we readily found anomalous configurations. To quantify the difference between order parameters, we plotted histograms of n_C for fixed values of n_E. This confirmed a close correlation for high and low n_E and that the average n_C increases with n_E. However, for intermediate n_E, the number of neighbours is not strongly correlated with the number of crystalline particles.

Finally, we investigate the effect of stretching a single polymer chain on its nucleation barrier in Chapter 5. We implement stretching by adding a term to the potential function used to control the acceptance probability of Monte Carlo moves in our polymer simulation of Chapter 4. We bias the simulation using the reproducible barrier for the quiescent chain. We find that very weak stretching of the end-to-end vector of the single chain has a significant effect on the nucleation barrier, reducing the height and changing the location of the barrier peak. Conformations of the single chain are highly sensitive to end-to-end

stretching, meaning that nucleation is inhibited by relatively modest stretching proportional to the end-to-end vector. Hence, we apply stretching that is restricted to a short segment at the chain-centre. This allows us to apply much stronger stretching to the chain, which has little effect on the stretch ratio, but has a noticeable effect on the nucleation barrier.

6.2 Overarching conclusions

Simulating the nucleation barrier of a single polymer chain has two inherent difficulties. The first is the rareness of visits to states around the barrier peak. We overcome this issue using biasing with our feedback technique. The key advantage of our feedback technique is that we do not require any initial guess of the nucleation barrier is required. The feedback technique is an automated process, in that we can initialise a simulation with no biasing, then execute successive iterations of the feedback algorithm to refine our estimate of the free energy barrier. We then use this estimate to fix the biasing function and perform a long run of the simulation to obtain an accurate resolution of the true barrier.

The feedback algorithm is sufficient for the free particle model, since free particles do not suffer from the configurational issues of self-entanglement and excluded volume that arise with connected particles. Around the crystallisation temperature, these configurational restrictions lead to poor mixing of simulation, which compounds the issues with the rarity of visits to the states of interest. We overcome the issue of poor mixing with parallel tempering. In particular, we choose a non-standard distribution of temperatures with which to conduct parallel tempering. This choice of temperatures propagates information from both the high crystal region sampled by temperatures lower than the temperature of interest and the amorphous coil region from hotter temperatures. Hence, we enable frequent large jumps in the order parameter both forwards and backwards across the barrier. These jumps, in combination with the hybrid feedback and umbrella sampling biasing strategy, allow us to obtain a highly reproducible resolution of the barrier peak.

Our stretching results reveal that the nucleation barrier for the single chain is highly sensitive to the onset of stretching. We observe significant changes to the height and location of the barrier peak for both end-to-end and chain-

centre stretching. The chain-centre stretching are advantageous over end-to-end stretching because we can apply stronger stretching to the chain-centre with less risk of inhibiting the nucleation process. Furthermore, chain-centre stretching mimics the effective stretch of the chain under flow, meaning we can use chain-centre stretching to simulate flow-induced stretching [Doi and Edwards, 1986]. The results from our stretching simulations suggest an explanation for a phenomenon observed by Graham and Olmsted [2009] in experiments conducted by Coccorullo et al. [2008]. They noted that, even though only a very small fraction of the chains in the melt were strongly stretched in the flow experiments, there was a dramatic increase in nucleation rate. Our result suggests that chain lengths that were only weakly stretched (constituting a much larger fraction of the overall molecular weight distribution than chains that were strongly stretched) would contribute significantly to the increased nucleation rate.

6.3 Future work

The work of §4.3.2.1 can be extended by increasing the value of $N_{\max}$, extending the upper limit of the sampling window. This will allow us to resolve the second minimum in the free energy barrier. Although this result is not needed to investigate nucleation, there is still value in obtaining good resolution of the entire barrier, allowing further comparison with the results of Taylor et al. [2009b].

The analysis comparing the order parameters n_E and n_C in §4.4. We suggest that computing the free energy barrier with the Steinhardt-Nelson order parameter, n_C, would be an interesting comparison to the results for the barrier as a function of n_E. We could simulate using the biasing with the function of n_E from Chapter 4, but also log the occupancies of states defined with n_C. Using the occupancies of n_C we can plot the free energy barrier as a function of n_C. Figure 4.24 showing the wide distribution of n_C for fixed n_E for states in the vicinity of the barrier peak suggests we may have issues with the two order parameters being poorly correlated. In particular, intermediate values of n_E are necessary but not sufficient to see intermediate values of n_C. These intermediate values are of great interest because they define the barrier peak. However,

our hybrid feedback and umbrella sampling biasing strategy would be useful here to reduce the effect of the wide distributions, since this biasing forces the simulation to occupy the problematic states. Hence, over a long enough simulation, the correlation of n_E and the peaks of the n_C distributions for fixed n_E in Figure 4.24 will emerge.

There are opportunities to use our model to investigate the effect of changing the temperature and chain length on the nucleation barrier for the single chain, since these are always experimental variables. The chain length is of interest, since the chain length influences the local density of the chain, analogous to pressure and volume for free particles. We can use our model to test the expectation that the nucleation barrier is independent of chain length [Yi et al., 2013]. The investigation of temperature is also of interest, since there is evidence that stretching has a more dramatic effect at high temperatures [Graham, 2014]. One can also investigate the effect of persistence length on nucleation by introducing potentials on the bending and torsional angles of the chain. Since we have already implemented the united atoms model, we can alter the parameters governing the relevant potentials in §1.4.3.2.2 to "switch on" the bond and torsion interactions and control the stiffness (and effectively the persistence length) of the chain. This leads to elongated nuclei, so we postulate that stretching has a more drastic effect on stiff chains.

For the stretching model, we could investigate the change in the free energy as a function of the stretch ratio, Λ. A key weakness of current coarse grained models of flow-induced crystallisation is their empirical treatment of the interplay between chain stretching and the nucleus structure [Graham and Olmsted, 2009]. This work simulates directly from molecular principles. Using our model, we can locate systematic patterns in the effect of Λ on the change in the free energy, which can be used to inform models of flow-induced crystallisation. In particular, we could predict the effect of temperature on the sensitivity to stretching. Experiments [Pantani et al., 2010] and coarse grained simulations [Graham and Olmsted, 2009] suggest that the higher the temperature, the more drastic effect stretching has on nucleation. This important effect is quite poorly understood in the current literature.

We can readily use our model to investigate the interplay of persistence length and the sensitivity of the barrier to stretch by introducing potentials to the

bonding and torsional angles in the stretching model. Since increasing the chain stiffness increases the likelihood of configurations with straight segments, crystal states are forced to consist of straight segments of particles separated by tight folds (Figure 1.19). Since tight folds are energetically expensive in a model under the influence of a torsion potential (Figure 1.20), these states are very rare in the quiescent case. Stiff chains are likely to give elongated nuclei, as are stretched chains as previously mentioned. This similarity in nucleus structure suggests that stiff chain nucleation should be more sensitive to stretching than for fully flexible chains, since aligned straight segments of the chain will be favourable.

Our model can be used to examine in detail the geometry of the critical nucleus. Using our hybrid feedback and umbrella sampling biasing technique, we can obtain good sampling of states in the vicinity of the barrier peak, i.e. critical nuclei. In addition, parallel tempering guarantees the propagation of information from low crystal states, melting and reforming the critical nuclei and de-correlating any trapped conformation within the timescale of a simulation. Using several snapshots of configurations sampled in the critical region, we can investigate geometric features of critical nuclei, such as the shape and aspect ratio. By adding stretching and stiffness to the chain using the appropriate potentials, we can investigate the effect of flow and chain stiffness on the geometric features of the critical nucleus.

Finally, we can compare our results in Chapters 4 and 5 with experimental results for flow-induced crystallisation and unfolding of biomolecules. Such experiments provided the motivation for our study of the nucleation of a single chain, so it is important to verify that they are similar to those obtained in the laboratory. Phenomena found in experiments can also be used to motivate further theoretical study using simulations to see if the results can be reproduced by our model.

Our code has been made publicly available on GitHub with a basic user guide (see Appendix D) for use by the research community to explore the areas described above.

APPENDIX A

The Legendre transform

The Legendre transform is a useful and versatile tool in thermodynamics. Its primary use in this work is to determine the appropriate quantities to compute, given the desired ensemble for a simulation. We outline the derivation of the Legendre transform here for convenience.

A.1 Derivation

Suppose we have a function $f(x, y)$, then

$$\mathrm{d}f = \left(\frac{\partial f}{\partial x}\right)_y \mathrm{d}x + \left(\frac{\partial f}{\partial y}\right)_x \mathrm{d}y, \tag{A.1.1}$$

where $(\partial/\partial x)_y$ denotes partial differentiation with respect to x keeping y fixed. We define

$$u = \left(\frac{\partial f}{\partial x}\right)_y, \quad v = \left(\frac{\partial f}{\partial y}\right)_x, \tag{A.1.2}$$

then

$$\mathrm{d}f = u\,\mathrm{d}x + v\,\mathrm{d}y. \tag{A.1.3}$$

The quantities (u, x) and (v, y) are conjugate pairs of variables. Further, we can write

$$\mathrm{d}(vy) = y\,\mathrm{d}v + v\,\mathrm{d}y. \tag{A.1.4}$$

Subtracting (A.1.4) from (A.1.3), we define

$$g(x, v) = f(x, y(v)) - vy(v), \tag{A.1.5}$$

so that

$$\mathrm{d}g = \mathrm{d}(f - vy) = u\,\mathrm{d}x - y\,\mathrm{d}v. \tag{A.1.6}$$

We define $g(x, v)$ as the Legendre transform of $f(x, y)$. We have exchanged the independent variable y in f for its conjugate variable v in g. Similarly, we could find other Legendre transforms of $f(x, y)$, namely $h(u, y)$ and $k(u, v)$. In general, a function of n variables has 2^n Legendre transforms.

A.2 Application to Thermodynamics

In the case of thermodynamics, the functions are called potentials and we have three conjugate pairs of variables: (S, T), $(V, -p)$, (N, μ). Therefore, we have 8 possible potentials, though we restrict attention to the potentials listed in Table 1.1.

For example, when transforming from the internal energy $U(S, V, N)$ to the Helmholtz free energy $F(T, V, N)$, we are exchanging the conjugate variables S and T. Using the expression in Table 1.1 for U, we identify that

$$\mathrm{d}U = T\,\mathrm{d}S - p\,\mathrm{d}V + \mu\,\mathrm{d}N. \tag{A.2.1}$$

Following the procedure above for the Legendre transform, we define the Helmholtz free energy $F = U - TS$ such that

$$\begin{aligned} \mathrm{d}F &= \mathrm{d}U - T\,\mathrm{d}S - S\,\mathrm{d}T, && \text{(A.2.2a)} \\ &= -S\,\mathrm{d}T - p\,\mathrm{d}V + \mu\,\mathrm{d}N, && \text{(A.2.2b)} \end{aligned}$$

so we have a new thermodynamic potential that has T dependence instead of S.

APPENDIX B

Principles of statistical mechanics

Here, we outline the foundations of statistical mechanics, which lead to Boltzmann's formula for entropy and ultimately the Boltzmann distribution of states. We begin with some notes on probability, which are useful for understanding the underlying principles of statistical mechanics.

B.1 Probability

Given the nature of a thermodynamic system outlined in the previous section, the most sensible way to model the set of states that the system can be in is with a probability distribution. In this section, we derive precisely which distribution is appropriate. To begin, we consider two types of probability.

Classical probability is the common, intuitive interpretation of probability. For a certain trial, one lists all possible outcomes in a sample space, then assigns each outcome i the probability p_i. If there are multiple ways f_i of obtaining outcome i, then

$$p_i = \frac{f_i}{\sum_i f_i}, \tag{B.1.1}$$

such as when two fair, cubic dice are rolled. In this case, there are 3 ways to roll a 4, $\{(1,3),\ (2,2),\ (3,1)\}$, so $p_4 = 1/12$.

In contrast, *statistical probability* is where p_i is approximated based on the results of a set of trials. If one conducts N trials and the number of occurances of event i is n_i, then if n_i/N tends to a constant as N gets large, we define the statistical

probability as

$$p_i = \lim_{N\to\infty} \frac{n_i}{N}. \tag{B.1.2}$$

Clearly, one cannot conduct infinitely many trials, but fluctuations in p_i decrease as $N^{-1/2}$, so we simply need to make N sufficiently large to obtain an accurate estimate of p_i. In the context of thermodynamics, the outcomes are states that the system can be in. We define precisely what we mean by a state in the next section.

To close this section on probability, we discuss the way in which we count the total number of states available in order to compute these probabilities. Consider N identical systems and let n_i of them be in state i. Then there are

$$\binom{N}{n_i} = \frac{N!}{n_i!(N-n_i)!} \tag{B.1.3}$$

ways of arranging the systems in such a way. Overall, we wish to have n_1 systems in state 1, n_2 in state 2 and so on, subject to $\sum_i n_i = N$. Since each of these events are independent we obtain, for k available states,

$$\begin{aligned} W &= \binom{N}{n_1}\binom{N-n_1}{n_2}\binom{N-n_1-n_2}{n_3}\cdots\binom{N-n_1-\ldots-n_{k-1}}{n_k} \\ &= \frac{N!}{n_1!(N-n_1)!}\times\frac{(N-n_1)!}{n_2!(N-n_1-n_2)!}\times\ldots\frac{N-n_1-\ldots-n_{k-1}}{n_k!} \\ &= \frac{N!}{n_1!\times n_2!\times\ldots} \end{aligned} \tag{B.1.4}$$

for the total number of ways that the N identical systems can be arranged into this distribution of states.

B.2 Motivation

The key principles of statistical mechanics, as laid out by [Bowley and Sanchez, 1999] are as follows:

1. The state of a system is specified as completely as possible. This means that one can measure the position, momentum and energy of each particle at one instant.

2. In practice, we cannot make a complete set of simultaneous measurements to specify the quantum state, so we cannot obtain the statistical

probability of a certain state. Thus, we are forced to use the classical probability.

3. To assign the classical probability of a quantum state, we imagine an *ensemble* of systems. An ensemble is a collection of systems, all of which have been prepared identically, i.e. they have the same number of particles, energy, volume etc. The quantum state of each system in the ensemble can be different, since there are many quantum states that correspond to the way every system in the ensemble was prepared. Thus, the probability that a system is in a particular quantum state is given by the fraction of systems in the ensemble in this state.

4. Suppose there are W possible quantum states that satisfy the set of constraints on the system when it is prepared. We assume that all quantum states are equally likely a priori. Thus, the probability of the system being in any state is $1/W$. We can then calculate the average of any quantity, such as the average kinetic energy of a particle. Averages obtained in this fashion are assumed to be identical to the time average of a single system.

Finally, we have Boltzmann's hypothesis, which states that the entropy of a system is related to the probability of its being in a quantum state. Now, $p_i = 1/W$ if there are W available states, so Boltzmann's hypothesis can be expressed as

$$S = \phi(W), \tag{B.2.1}$$

where ϕ is some unknown function of W.

Consider two non-interacting systems, A and B, whose entropies are

$$S_A = \phi(W_A), \tag{B.2.2a}$$
$$S_B = \phi(W_B). \tag{B.2.2b}$$

Considering the two systems as a single system, AB,

$$S_{AB} = \phi(W_{AB}) = S_A + S_B, \tag{B.2.3}$$

since the total entropy is the sum of the entropies of the two systems. Since the systems are independent, $W_{AB} = W_A W_B$, so

$$S_{AB} = \phi(W_{AB}) = \phi(W_A W_B) = \phi(W_A) + \phi(W_B). \tag{B.2.4}$$

The only solution to this equation for $\phi(W_{AB})$ is

$$S = k_B \ln W, \tag{B.2.5}$$

where k_B is Boltzmann's constant. This is Boltzmann's formula for entropy, which relates entropy to the number of accessible states of a system via a universal constant, $k_B \approx 1.38 \times 10^{-23}\,\mathrm{J\,K^{-1}}$.

APPENDIX C

Recovering averages over the unbiased distribution

When we conduct a Metropolis Monte Carlo system, we collect statistics and compute averages over the Boltzmann distribution, which has the probability density function,

$$P_{\mathrm{o}}(\mathbf{r}) = \frac{1}{Z_{\mathrm{o}}} \exp(-\beta U(\mathbf{r})), \tag{C.1.1}$$

where $\beta = 1/(k_{\mathrm{B}}T)$ is the inverse temperature, U is the potential energy, $\mathbf{r}$ is the configurational coordinates of the particles, and Z_{o} is the partition function for the unbiased distribution,

$$\int \exp(-\beta U(\mathbf{r}))\, \mathrm{d}\mathbf{r}. \tag{C.1.2}$$

In §1.3.3, we introduce the idea of sampling in a biased distribution, which has the probability density function,

$$P_{\mathrm{o}}(\mathbf{r}) = \frac{1}{Z_{\mathrm{w}}} W(\mathbf{r}) \exp(-\beta U(\mathbf{r})), \tag{C.1.3}$$

where W is the weight function, and Z_{w} is the partition function for the biased distribution,

$$Z_{\mathrm{w}} = \int W(\mathbf{r}) \exp(-\beta U(\mathbf{r}))\, \mathrm{d}\mathbf{r} \tag{C.1.4}$$

In our biased simulations, we compute average of a quantity, X, over the biased distribution (C.1.3), $\langle X \rangle_{\mathrm{w}}$. We wish to use $\langle X \rangle_{\mathrm{w}}$ to compute $\langle X \rangle_{\mathrm{o}}$, the average

of X over the original distribution (C.1.1). Manipulating $\langle X \rangle_{\mathrm{o}}$,

$$\begin{aligned}
\langle X \rangle_{\mathrm{o}} &= \frac{1}{Z_{\mathrm{o}}} \int X(\mathbf{r}) \exp(-\beta U(\mathrm{r}))\, \mathrm{d}\mathbf{r}, \\
&= \frac{1}{Z_{\mathrm{o}}} \int \frac{X(\mathbf{r})}{W(\mathbf{r})} W(\mathbf{r}) \exp(-\beta U(\mathrm{r}))\, \mathrm{d}\mathbf{r}, \\
&= \frac{Z_{\mathrm{w}}}{Z_{\mathrm{o}}} \int \frac{X(\mathbf{r})}{W(\mathbf{r})} \frac{W(\mathbf{r}) \exp(-\beta U(\mathrm{r}))\, \mathrm{d}\mathbf{r}}{Z_{\mathrm{w}}}, \\
&= \frac{Z_{\mathrm{w}}}{Z_{\mathrm{o}}} \left\langle \frac{X}{W} \right\rangle_{\mathrm{w}}. \qquad \text{(C.1.5)}
\end{aligned}$$

Consider the ratio $Z_{\mathrm{o}}/Z_{\mathrm{w}}$

$$\begin{aligned}
\frac{Z_{\mathrm{o}}}{Z_{\mathrm{w}}} &= \frac{1}{Z_{\mathrm{w}}} \int \exp(-\beta U(\mathbf{r}))\, \mathrm{d}\mathbf{r}, \\
&= \frac{1}{Z_{\mathrm{w}}} \int \frac{1}{W(\mathbf{r})} W(\mathbf{r}) \exp(-\beta U(\mathrm{r}))\, \mathrm{d}\mathbf{r}, \\
&= \left\langle \frac{1}{W} \right\rangle_{\mathrm{w}}. \qquad \text{(C.1.6)}
\end{aligned}$$

Substituting (C.1.6) into (C.1.5), we finally obtain (1.3.37),

$$\langle X \rangle_{\mathrm{o}} = \frac{\langle X/W \rangle_{\mathrm{w}}}{\langle 1/W \rangle_{\mathrm{w}}}. \qquad \text{(C.1.7)}$$

APPENDIX D

Code user guide

The code used in Chapters 4 and 5 can be found at `https://github.com/twicks2011/spmmc`. We provide the user guide in this appendix.

D.1 Author contact details

If you have any enquiries or issues regarding use of this code, please contact me via *tom.wicks1988@gmail.com*.

D.2 Installation and compilation

This code is entirely written in C and can be compiled using either *icpc* or *gcc*, with OpenMP enabled. Follow the steps below to install the code on your system.

1. Download the zip file containing all files and extract them to your desired working directory.
2. Compile the code, using *make*. By default, the code compiles with *icpc*. To compile with *gcc*, comment and uncomment the appropriate lines in the makefile, following the instructions in the file itself.
3. On successful compilation, the executable *singleChainOMP* is generated.

D.3 Running the code

There are two components to running the code successfully:

1. Enter the desired inputs into an input file. An exemplar input file can be found in the root directory (inputFileEXAMPLE.dat). This file contains some typical values for the parameters to get you started.
2. Create the output directories for storing the data generated by the code.

Details are provided in the following subsections for how to perform these tasks.

D.3.1 Setting up the input file

The input file contains all of the information needed by the code to perform a simulation. The following table provides a description of each input variable to be set by the user, along with its type.

Variable	Type	Description
`N`	int	Number of particles in the chain.
`LAMBDA`	double	Square-well interaction range.
`TEMP_INTEREST`	int	The indicial value of the temperature of interest (where the counter starts from 0).
`NUM_TEMPS`	int	The number of temperatures to be simulated. This is also the number of parallel threads to be initiated.
`TEMPS_FILE`	string	File containing the temperature profile.
`fileInput`	int	Toggles whether to use an input file for initial configurations or to generate a planar zigzag.
`CONFIGS_FILE`	string	File containing the initial configurations of each chain.
`MSIZE_DIR`	string	Directory containing files with initial move sizes for each type of move.
`PHI_MAX`	double	The initial maximum size of a bond-torsion move.

Variable	Type	Description
ALPHA	double	The relative influence of the Square-Well potential on the move acceptance probability.
BETA GAMMA	double double	The relative influence of the bond-torsion and bond-angle energies respectively on the move acceptance probability. These are not fully tested.
sigma	double	Stretching coefficient.
stretchStart stretchFinish	int int	Stretching occurs between the particles with indices stretchStart and stretchFinish.
Nmin Nmax	int int	Energy states between Nmin and Nmax will be sampled.
kappaLHS kappaRHS	double double	These variables define the steepness of the quadratic arm on the left and right-hand side of the sampling window.
UPDATE	int	Toggles the automated feedback mechanism for the biasing function.
tolerance	double	The tolerance (0.0-1.0) for the automated checks for uniform sampling in the biased distribution.
PRINT_INTERVAL	double	Number of moves between updates of the output files.
PHI_UPDATE_INTERVAL	double	Number of moves between updates of PHI_MAX.
UPDATE_INTERVAL	double	Number of SWAP_INTERVALs between updates of the biasing function.
SWAP_INTERVAL	double	Number of moves between swap cycles.
NUM_SWAPS	int	Number of configuration swaps in each swap cycle.
NUM_SWAP_INTERVALS	double	Number of swap intervals to complete over the whole simulation.

Variable	Type	Description
RESET_INTERVAL	double	Number of moves between resetting the counters and initiating a new sampling block.
SMALL_ANGLE_MOVES REPTATION_MOVES CRANK_MOVES END_ROT_MOVES END_BRIDGE_MOVES	double double double double double	The probability of proposing each move-type. These should sum to 1.
OUTPUT_DIR	string	The directory where all the output files are to be sent.
weightFile	string	The file containing the biasing function.
VMDdir	string	The directory to send files for generating 3D images of configurations using Visual Molecular Dynamics.
initCoords	string	File containing initial coordinates to be copied to all configurations if `fileInput` is set to 0.

Examples of the input files referred to in the table are provided in the *EXAMPLES* folder.

D.3.2 Creating the output directories

Use the executable *createDirs.sh* to create the output directories for storing the output data from the code. You will be asked to enter a name for your desired directory. The following subdirectories are generated:

- outputFiles/`MyOutputDirectory/`
  - outputFiles/`MyOutputDirectory`/occupancies/
  - outputFiles/`MyOutputDirectory`/restartInfo/
- VMD/`MyOutputDirectory/`

Note: generate the output directory *EXAMPLE* **to use the exemplar input file** *inputFileEXAMPLE.dat*.

D.3.2.1 Deleting output data

To delete an output directory, use the executable *deleteDirs.sh*. WARNING: this will remove the directory and all of its contents, so use with caution!

D.4 Output files

The following files are outputted by the code into the directory outputFiles/`MyOutputDirectory/`:

- *crossings_chainX.dat*: These files contain information regarding the number of "zero crossings" chain X has experienced. This is an indication of the number of times the chain has completed N (forward or backward) reptation moves and so has replenished its configuration.
- *EtraceX.dat*: At each swap interval, these files update with the current temperature of chain X, where each integer refers to the index in the temperature profile.These temperature traces are used to determine how well-mixed the simulation is.
- *FE_chainX.dat*: The latest estimate of the free energy of chain X as a function of energy state.
- *occW_chainX.dat*: The latest occupancy of each energy state by chain X.
- *upwardMoves_chainX.dat*: The latest information regarding number of attempted and accepted upward moves from each energy state visited by chain X.
- *occupancies/block_K.dat*: These file contain the occupancies of each chain in sampling block K. The nth entry in line m of the file refers to chain n's occupancy of state m.
- *restartInfo/*: The files in this subdirectory are used to restart a simulation where the current one finishes. This is particularly useful to avoid the need to repeat a long equilibration process.
  - *configs.dat*: This file contains the latest configurations of all chains.

  - *crankMax.dat*, *endMax.dat*, *phiMax.dat*: These files contain the latest optimal move sizes for crank-shaft, end-rotation and torsional moves respectively.

If the Visual Molecular Dynamics output is enabled, the following files are outputted to the directory VMD/`MyOutputDirectory/`:

- *coordinatesX_Y.pdb*: A file that can be interpreted by VMD containing information for the Yth realisation of a configuration in energy state X.
- *bondsX_Y.psf*: A file containing the corresponding bond information.
- *StateX_Y.vmd*: An executable that can be run in the command line of VMD to generate the visualisation corresponding with the coordinates and bonds files with given values of X and Y.

References

M.P. Allen and D.J Tildesley. *Computer Simulation of Liquids*. Clarendon Press, Oxford, 1989.

M. Anwar, J. T. Berryman, and T. Schilling. Crystal nucleation mechanism in melts of short polymer chains under quiescent conditions and under shear flow. *J. Chem. Phys.*, 141(12):124910, 2014.

S. Auer and D. Frenkel. Prediction of absolute crystal-nucleation rate in hard-sphere colloids. *Nature*, 409:1020, 2001.

S. Auer and D. Frenkel. Numerical prediction of absolute crystallization rates in hard-sphere colloids. *J. Chem. Phys.*, 120:3015–3029, 2004.

S. Auer and D. Frenkel. Numerical simulation of crystal nucleation in colloids. *Adv. Polym. Sci.*, 173:149–207, 2005.

R. Becker and W. Döring. Kinetic treatment of nucleation in supersaturated vapors (transl.). *Ann. Phys.*, 24:719, 1935.

B. A. Berg and T. Neuhaus. Multicanonical ensemble: a new approachto simulate first-order phase transitions. *Phys Rev. Lett.*, 68:9–12, 1992.

R. Bowley and M. Sanchez. *Introductory Statistical Mechanics*. Clarendon Press, Oxford, 2nd edition, 1999.

J. N. Cape, J. L. Finney, and L. V. Woodcock. An analysis of crystallization by homogeneous nucleation in a 4000-atom soft-sphere model. *J. Chem. Phys.*, 75:2366, 1981.

P.M. Chaikin and T.C. Lubensky. *Principles of Condensed Matter Physics*. Cambridge University Press, Cambridge, 1st edition, 1995.

I. Coccorullo, R. Pantani, and G. Titomanlio. Spherulitic nucleation and growth rates in an iPP under continuous shear flow. *Macromolecules*, 41(23):9214–9223, 2008.

M Doi and S F Edwards. *The Theory of Polymer Dynamics*. Oxford University Press, Oxford, 1986.

D. J. Earl and M. W. Deem. Parallel tempering: Theory, applications, and new perspectives. *Phys. Chem. Chem. Phys.*, 7:3910–3916, 2005.

K. Esselink, P.A.J. Hilbers, and B.W.H. van Beest. Molecular dynamics study of nucleation and melting of n-alkanes. *J. Chem. Phys.*, 101:9033, 1994.

L. Filion, M. Hermes, R. Ni, and M. Dijkstra. Crystal nucleation of hard spheres using molecular dynamics, umbrella sampling, and forward flux sampling: A comparison of simulation techniques. *J. Chem. Phys.*, 133:244115, 2010.

L. Filion, R. Ni, D. Frenkel, and M. Dijkstra. Simulation of nucleation in almost hard-sphere colloids: The discrepancy between experiment and simulation persists. *J. Chem. Phys.*, 134:134901/1–134901/7, 2011.

D. Frenkel and B. Smit. *Understanding Molecular Simulation: From Algorithms to Applications*. Academic Press, San Diego, 1st edition, 1996.

S. Fujiwara and T. Sato. Molecular dynamics simulation of structure formation of short chain molecules. *J. Chem. Phys.*, 110:9757, 1999.

M. Gerstein, J. Tsai, and M. Levitt. The volume of atoms on the protein surface: calculated from simulation, using Voronoi polyhedra. *J. Mol. Biol.*, 249:955–966, 1995.

J. W. Gibbs. *The Scientific Papers of J. Willard Gibbs: Thermodynamics*. Longmans, Green and Company, London, 1906.

R. S. Graham. Modelling flow-induced crystallisation in polymers. *Chem. Commun.*, 50:3531–3545, 2014.

R. S. Graham and P. D. Olmsted. Kinetic Monte Carlo simulations of flow-induced nucleation in polymer melts. *Faraday Discuss*, 144:71–92, 2010.

R.S. Graham. Molecular modelling of flow-induced crystallisation in polymers. *J. Eng. Math.*, 71:237–251, 2011.

R.S. Graham and P.D. Olmsted. Coarse-grained simulations of flow-induced nucleation in semi-crystalline polymers. *Phys Rev. Lett.*, 103:115702, 2009.

R.S. Graham, A.E. Likhtman, and T.C.B. McLeish. Microscopic theory of linear, entangled polymer chains under rapid deformation including chain stretch and convective constraint release. *J. Rheol.*, 47:1171–1200, 2003.

W. Humphrey, A. Dalke, and K. Schulten. VMD - visual molecular dynamics. *Molec. Graphics*, 14:33–38, 1996.

M. Kawakami, K. Byrne, B. Khatri, T. C. B. Mcleish, S. E. Radford, and D. A. Smith. Viscoelastic properties of single polysaccharide molecules determined by analysis of thermally driven oscillations of an atatom force microscope cantilever. *Langmuir*, 20:9299–9303, 2004.

T. Kawasaki and H. Tanaka. Formation of a crystal nucleus from a liquid. *Proc. Natl. Acad. Sci.*, 107:14036, 2010.

A Keller and H.W. Kolnaar. Flow induced orientation and structure formation in processing of polymers. In HEH Meijer, editor, *Processing of Polymers*, volume 18, page 189. Wiley-VCH, Weinheim, 1997.

D. A. Kofke. On the acceptance probability of replica-exchange Monte Carlo trials. *J. Chem. Phys.*, 117:6911, 2002.

A. Kone and D. A. Kofke. Selection of temperature intervals for parallel-tempering simulations. *J. Chem. Phys.*, 122:206101, 2005.

D.J. Lacks and G.C. Rutledge. Simulation of the temperature dependence of mechanical properties of polyethylene. *J. Phys. Chem.*, 98:1222–1231, 1994.

M.S. Lavine, N. Waheed, and G.C. Rutledge. Molecular dynamics simulation of orientation and crystallization of polyethylene during uniaxial extensional. *Polymer*, 44:1771–1779, 2003.

C. Leitold and C. Dellago. Folding mechanism of a polymer chain with short-range attractions. *J. Chem. Phys.*, 141:134901, 2014.

C. Liu and M. Muthukumar. Langevin dynamics simulations of early-stagepolymer nucleation and crystallization. *J. Chem. Phys.*, 109:2536, 1998.

M. J. Mandell, J. P. McTague, and A. Rahman. Crystal nucleation in a three-dimensional lennard-jones system: A molecular dynamics study. *J. Chem. Phys.*, 64:3699, 1976.

V. I. Manousiothakis and M. W. Deem. Strict detailed balance is unnecessary in Monte Carlo simulation. *J. Chem. Phys.*, 110:2753, 1999.

R. McGraw and A. Laaksonen. Scaling properties of the critical nucleus in classical and molecular-based theories of vapor-liquid nucleation. *Phys. Rev. Lett.*, 76:2754, 1996.

R. McGraw and A. Laaksonen. Interfacial curvature free energy, the Kelvin relation, and vapor-liquid nucleation rate. *J. Chem. Phys.*, 106:5284, 1997.

N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *J. Chem. Phys.*, 21: 1087–1092, 1953.

R. D. Mountain and A. C. Brown. Molecular dynamics investigation of homogeneous nucleation for inverse power potential liquids and for a modified Lennard-Jones liquid. *J. Chem. Phys.*, 80:2730, 1984.

NDT Education Resource Centre. Primary metallic crystalline structures, 2001a. [Online]. Available: http://www.ndt-ed.org/ EducationResources/ CommunityCollege/ Materials/Structure/ metallic_structures.htm.

NDT Education Resource Centre. Similarities and difference between the fcc and hcp structure, 2001b. [Online]. Available: http://www.ndt-ed.org/ EducationResources/ CommunityCollege/ Materials/ Structure/ fcc_hcp.htm.

R. Ni. *Entropy-Driven Phase Transitions in Colloidal Systems*. PhD thesis, Utrecht University, 2012.

National Institute of Standards and Technology, NIST. EOS-TMMC: Equations of state (pressure as a function of density), 2007. URL `http://cstl.nist.gov/srs/LJ_PURE/eostmmc.htm`.

P. D. Olmsted and S. T. Milner. Strain-induced nematic phase-separation in polymer melts and gels. *Macromolecules*, 27:6648–6660, 1994.

D. W. Oxtoby and R. Evans. Nonclassical nucleation theory fot the gas-liquid transition. *J. Chem. Phys.*, 89:7521, 1988.

R. Pantani, I. Coccorullo, V. Volpe, and G. Titomanlio. Shear-induced nucleation and growth in isotactic polypropylene. *Macromolecules*, 43:9030–9038, 2010.

W. Paul, D.Y. Yoon, and G.D. Smith. An optimized united atom model for simulations of polymethylene melts. *J. Chem. Phys.*, 103:1702, 1995.

G. Racah. Theory of complex spectra. ii. *Phys Rev.*, 62:438–462, 1942.

S. Rastogi, D. R. Lippits, G. W. M. Peters, R. Graf, Y. Yao, and H. W. Spiess. Heterogeneity in polymer melts from melting of polymer crystals. *Nat. Mater.*, 4: 635–641, 2005.

N. Rathore, M. Chopra, and J.J. de Pablo. Optimal allocation of replicas in parallel tempering simulations. *J. Chem. Phys.*, 122:024111, 2005.

H. Reiss, W. K. Kegel, and J. L. Katz. Resolution of the problems of replacement free energy, 1/s, and internal consistency in nucleation theory by consideration of the length scale for mixing entropy. *Phys Rev. Lett.*, 78:4506, 1997.

M. Rotenberg, R. Bivins, N. Metropolis, and J. K. Wooten, Jr. *The 3-j and 6-j symbols*. Cambridge Technology Press, Massachusetts Institute of Technology, 1959.

P.E. Rouse. A theory of the linear viscoelastic properties of dilute solutions of coiling polymers. *J. Chem. Phys.*, 21:1272–1280, 1953.

Š. Růžička, D. Quigley, and M. P. Allen. Folding kinetics of a polymer. *Phys. Chem. Chem. Phys.*, 14:6044–6053, 2012.

K. Y. Sanbonmatsu and A. E. García. Structure of met-enkephalin in explicit aqueous solution using replica-exchange molecular dynamics. *Proteins*, 46: 225–234, 2002.

A. Schug, T. Herges, and W. Wenzel. All-atom folding of the three-helix HIV accessory protein with an adaptive parallel tempering method. *Proteins*, 57: 792–798, 2004.

P.J. Steinhardt, D.R. Nelson, and M. Ronchetti. Bond-orientational order in liquids and glasses. *Phys. Rev. B*, 28:784–805, 1983.

G. Strobl. *The Physics of Polymers: Concepts for Understanding Their Structures and Behaviour*. Springer-Verlag, Berlin, 3rd edition, 2007.

Y. Sugita and Y. Okamoto. Replica-exchange molecular dynamics method for protein folding. *Chem. Phys. Lett.*, 314:141–151, 1999.

R. H. Swendsen and J Wang. Replica Monte Carlo simulation of spin-glasses. *Phys Rev. Lett.*, 57:2607–2609, 1986.

M. Tanemura, Y. Hiwatari, H. Matsuda, T. Ogawa, N. Ogita, and A. Ueda. Geometrical analysis of crystallization of the soft-core model. *Prog. Theor. Phys.*, 58:1079, 1977.

M. P. Taylor and P. P. Aung. Partition function zeros and phase transitions fora square-well polymer chain. *Phys Rev. E*, 88:012604, 2013.

M. P. Taylor, W. Paul, and K. Binder. Phase transitions of a single polymer chain: A Wang-Landau simulation study. *J. Chem. Phys.*, 131:114907, 2009a.

M. P. Taylor, W. Paul, and K. Binder. All-or-none protein-like folding transition of a flexible homopolymer chain. *Phys. Rev. E*, 79:050801(R), 2009b.

M. P. Taylor, W. Paul, and K. Binder. Two-state protein-like folding of a homopolymer chain. *Physics Procedia*, 4:151–160, 2010.

M. P. Taylor, W. Paul, and K. Binder. Applications of the Wang-Landau algorithm to phase transitions of a single polymer chain. *Polymer Science Ser. 3*, 55:23–38, 2013.

P.R. ten Wolde. *Numerical Study of Pathways for Homogeneous Nucleation*. PhD thesis, FOM Institute for Atomic and Molecular Physics, 1998.

P.R. ten Wolde and D. Frenkel. Computer simulation study of gas-liquid nucleation in a Lennard-Jones system. *J. Chem. Phys.*, 109:9901–9918, 1998.

P.R. ten Wolde, M.J. Ruiz-Montero, and D. Frenkel. Numerical evidence for bcc ordering at the surface of a critical fcc nucleus. *Phys Rev. Lett.*, 75:2714–2717, 1995.

P.R. ten Wolde, M.J. Ruiz-Montero, and D. Frenkel. Numerical calculation of the rate of crystal nucleation in a Lennard-Jones system at moderate undercooling. *J. Chem. Phys.*, 104:9932–9947, 1996a.

P.R. ten Wolde, M.J. Ruiz-Montero, and D. Frenkel. Simulation of homogeneous crystal nucleation close to coexistence. *Faraday Discuss.*, 104:93–110, 1996b.

G.M. Torrie and J.P. Valleau. Monte Carlo free energy estimates using non-Boltzmann sampling: application to the sub-critical Lennard-Jones fluid. *Chem. Phys. Lett.*, 28:578, 1974.

D. Turnbull and J.C. Fisher. Rate of nucleation in condensed systems. *J. Chem. Phys.*, 17:71–73, 1949.

J.S. van Duijneveldt and D. Frenkel. Computer simulation study of free energy barriers in crystal nucleation. *J. Chem. Phys.*, 96:4655–4668, 1992.

M. Volmer and A. Weber. Nuclei formation in supersaturated states (transl.). *Z. Phys. Chem.*, 119:227–301, 1926.

J. Vrabec and H. Hasse. Grand equilibrium: vapour-liquid equilibria by a new molecular simulation method. *Molec. Phys.*, 100:3375–3383, 2002.

N. Waheed, M.S. Lavine, and G.C. Rutledge. Molecular simulation of crystal growth in n-eicosane. *J. Chem. Phys.*, 116:2301, 2002.

N. Waheed, M.J. Ko, and G.C. Rutledge. Molecular simulation of crystal growth in long alkanes. *Polymer*, 45:8689–8702, 2005.

F. Wang and D. P. Landau. Efficient, multiple-range random walk algorithm to calculate the density of states. *Phys Rev. Lett.*, 86:2050–2053, 2001.

C. H. M. Weber, A. Chiche, G. Krausch, S. Rosenfeldt, M. Ballauff, L. Harnau, I. Göttker-Schnetmann, Q. Tong, and S. Mecking. Single lamella nanoparticles of polyethylene. *Nano Lett.*, 7:2024–2029, 2007.

P. Yi and G.C. Rutledge. Molecular simulation of crystal nucleation in n-octane melts. *J. Chem. Phys.*, 131:134902, 2009.

P. Yi and G.C. Rutledge. Molecular simulation of bundle-like crystal nucleation from n-eicosane melts. *J. Chem. Phys.*, 135:024903, 2011.

P. Yi, C. R. Locker, and G. C. Rutledge. Molecular dynamics simulation of homogeneous crystal nucleation in polyethylene. *Macromolecules*, 46:4723–4733, 2013.

X. Yu, B. Kong, and X Yang. Molecular dynamics study on the crystallization of a cluster of polymer chains depending on the initial entanglement structure. *Macromolecules*, 41:6733–6740, 2008.