THE UNIVERSITY *of* EDINBURGH

Computer Simulations of Polymers and Gels

Dean Wood

Doctor of Philosophy
University of Edinburgh
2013

Declaration

I declare that this thesis was composed by myself and that the work contained therein is my own, except where explicitly stated otherwise in the text.

(*Dean Wood*)

Abstract

Computer simulations have become a vital tool in modern science. The ability to reliably move beyond the capabilities of experiment has allowed great insights into the nature of matter. To enable the study of a wide range of systems and properties a plethora of simulation techniques have been developed and refined, allowing many aspects of complex systems to be demystified. I have used a range of these to study a variety of systems, utilising the latest technology in high performance computing (HPC) and novel, nanoscale models.

Monte Carlo (MC) simulation is a commonly used method to study the properties of system using statistical mechanics and I have made use of it in published work [1] to study the properties of ferrogels in homogeneous magnetic fields using a simple microscopic model . The main phenomena of interest concern the anisotropy and enhancement of the elastic moduli that result from applying uniform magnetic fields before and after the magnetic grains are locked in to the polymer-gel matrix by cross-linking reactions. The positional organization of the magnetic grains is influenced by the application of a magnetic field during gel formation, leading to a pronounced anisotropy in the mechanical response of the ferrogel to an applied magnetic field. In particular, the elastic moduli can be enhanced to different degrees depending on the mutual orientation of the fields during and after ferrogel formation. Previously, no microscopic models have been produced to shed light on this effect and the main purpose of the work presented here is to illuminate the microscopic behaviour. The model represents ferrogels by ensembles of dipolar spheres dispersed in elastic matrices. Experimental trends are shown to be reflected accurately in the simulations of the microscopic model while shedding light on the microscopic mechanism causing these effects. These mechanisms are shown to be related to the behaviour of the dipoles during the production of the gels and caused by the chaining of dipoles in magnetic fields. Finally, simple relationships between the elastic moduli and the magnetization are proposed. If supplemented by the magnetization curve, these relationships yield the dependencies of the elastic moduli on the applied magnetic field, which are often measured directly in experiments.

While MC simulations are useful for statistical studies, it can be difficult to use them to gather information about the dynamics of a system. In this case, Molecular Dynamics (MD) is more widely used. MD generally utilises the classical equations of motion to simulate the evolution of a system. For large systems, which are often of interest, and multi-species polymers, the required computer power still poses a challenge and requires the use of HPC techniques. The most recent development in HPC is the use of Graphical Processing Units (GPU) for the fast solution of data parallel problems. In further published work [2], I have used a bespoke MD code utilising GPU acceleration in order to simulate large systems of block copolymers(BC) in solvent over long timescales. I have studied thin films of BC solutions drying on a flat, smooth surface which requires long timescales due to the 'slow' nature of the process. BC's display

interesting self-organisation behaviour in bulk solution and near surfaces and have a wide range of potential applications from semi-conductors to self-constructing fabrics. Previous studies have shown some unusual behaviour of PI-PEO diblock co-polymers adsorbing to a freshly cleaved mica surface. These AFM studies showed polymers increasing in height over time and proposed the change of affinity of mica to water and the loss of water layers on the surface as a driver for this change. The MD simulation aimed to illuminate the process involved in this phenomena. The process of evaporation of water layers from a surface was successfully simulated and gave a good indication that the process of solvent evaporation from the surface and the ingress of solvent beneath the adsorbed polymer caused the increase in height seen in experiment.

Contents

Chapter 1

Introduction

The area of soft condensed matter has become one of immense importance. With applications utilising soft matter ranging from liquid crystals [3, 4] to elastomers [5, 6], the study of soft matter has had a direct impact on modern life. Although much of science in the media and the public domain give the impression that all of science works on esoteric subjects such as particle physics or cosmology, the fact is that a large percentage of scientists could be said to work in the area of soft matter. In part, this large body of scientific effort reflects the wide range of materials and subjects which can be claimed to be soft condensed matter. This area of research crosses disciplinary boundaries and is of interest to everyone from biologists and chemists through to engineers and physicists with applications in technology, biology[7] and even bacteriology[8].

Despite this wide study and impressive range of uses, there remains many challenges in understanding and controlling the complex behaviour of soft systems. Inherent in the challenges soft matter presents is one of scale. Unlike hard condensed matter, where atoms are largely fixed to their crystalline lattice points, soft matter can exhibit emergent phenomena where the interactions on the molecular scale can lead to behaviours on much larger length scales. Examples of this emergent phenomena can be seen in all areas of soft matter including colloids[9] and the recently included area of biological systems [10].

In order to deal with this wide range of length scale, there has been an increase in the number of attempts to bridge a number of length scales [11, 12, 13] be that at the smallest scale [14] or at the almost human scale [15]. This has met with differing degrees of success as the technical challenges posed by attempting to cross length scales are significant. Nevertheless, for certain problems a coupling of continuum properties, such as elasticity, with particle interactions, such as interactions between dipolar particles, can sometimes yield useful results and is a growing area of interest.

In this section we will briefly explore two general areas of soft matter related to the work presented in later chapters. Firstly, we will discuss the general background for polymer science. This will be a brief overview as this has become a vast field due to the incredible growth in the use of polymers in modern life. Secondly, we will discuss dipolar particles with a view to their relationship to the relatively new material of ferrogels. Lastly, in this section we will touch on the place of computer simulation in modern science.

1.1 Polymers

Polymers are defined in the Oxford English Dictionary as "a compound with a molecular structure in which a (usually large) number of similar polyatomic units are bonded together". This covers an enormous range of compounds today with almost everything in a modern house containing a polymeric material of one kind or another, with a good number being synthetic. This ubiquity highlights the impressive rise of polymers, from the first synthetic polymer in 1909 [16] to the hundreds in every home today.

The word 'polymer' was first coined in 1833 by Jöns Jacob Berzelius although the original meaning differed slightly from the modern definition and the work being carried out would differ significantly from the current idea of polymer science. The term stuck, and today is taken to mean everything from synthetically produced macromolecules to naturally occurring biomolecules. Initially, work on polymers consisted of manipulating naturally occurring materials such as rubber. As stated before, by 1909 a synthetic polymer had been created and from there the science has not looked back.

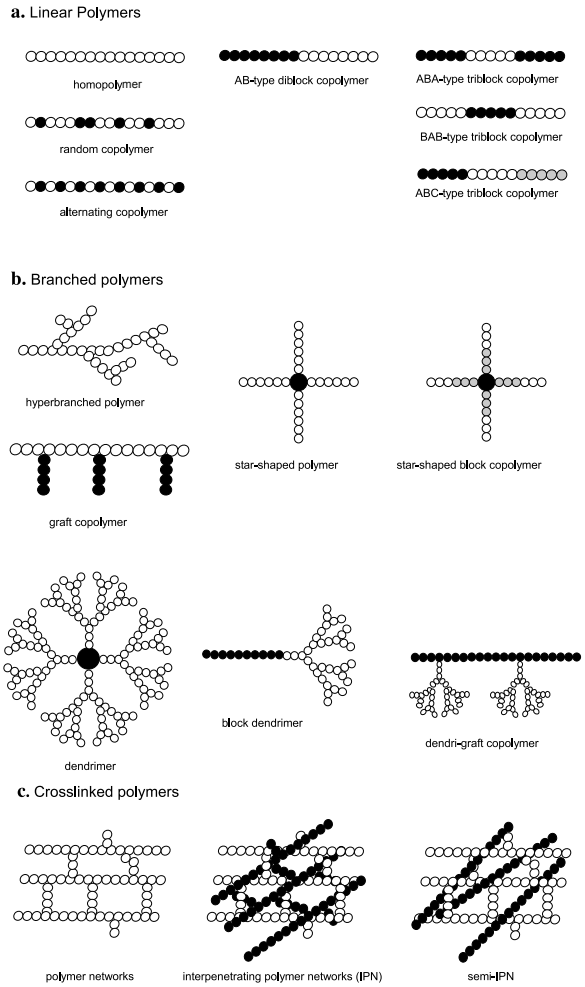


Figure 1.1: A by no means exhaustive summary of possible conformations for synthesised polymers. Taken from [17].

Modern polymer science behaves very differently. Experimental techniques have evolved to the point where single homopolymers (polymers consisting of one type of monomer) are just the

beginning of the range of materials available. With modern techniques it is possible to construct polymers with multiple 'blocks' of different types and control their lengths quite precisely [18, 19]. This has led to a vast array of different types of polymers able to be produced with a plethora of different responses to a variety of environments. Fig 1.1 shows a small selection representing the diversity of polymer types that can be constructed. Considering simply the AB-type diblock copolymer, a wide range of behaviours can be extracted from simply using polymers with different responses to a solvent [20, 21]. Such polymers with different responses have many applications and can be utilised in many ways. Adsorbing them onto surfaces under certain solvent conditions can lead to usable films for applications such as nanolithography [22].

Experimentally, visualising objects as small as polymers remains a difficult issue. For adsorption studies use can be made of microscopy techniques such as AFM [23, 24, 25]. These studies can lead to detailed pictures of the structure of the polymers on the surface and can even visualise single polymers. However it cannot visualise the evolution of the polymers over time. This limit to single snapshots constrains what can be inferred from such studies. Further to this, AFM studies can only be carried out on polymers on surfaces. For questions of what happens in solution, options are further limited. Light or neutron scattering studies can yield information for the distribution of polymers through a solvent but suffers from the single image problem of AFM. Extra information can be gleaned from alternative properties such as surface tension and rheological studies but these are not direct measures and can only give limited insight into the microscopic behaviour.

It is here that computer simulation can be effectively utilised. With the increasing power of computers it has become possible to simulate systems of sufficient size for many problems to yield microscopic insight. While this has become much simpler in recent years, computational power is not at a point where realistic systems can be simulated in a naive brute force manner. Instead, there are a number of compromises which must be made in order to carry out useful simulations. More will be said on this in chapter 3 whilst more will be said on polymers in chapter 5.

1.2 Magnetic dipolar particles and ferrogels

Ferrogels are a magnetic field sensitive polymer gel which were first made in the mid-90's [26, 27]. They consist of a polymer gel embedded with magnetic dipolar particles whose movement is fixed by the polymer matrix. To understand the implications of this, it is first necessary to consider the behaviour of magnetic dipolar particles.

Magnetic dipolar particles are particles which respond to an external magnetic field. They contain a dipole moment which is a vector pointing between the north and south poles on the particle and align with an applied external field [28].

The pure dipole-dipole interaction was heavily studied via simulations in the mid-90s [29, 30, 31]. These seminal studies looked at systems of dipoles at cryogenic temperatures to remove any thermodynamic perturbations of the system and only consider pure dipolar interactions. Weis and Levesque [Levesque1994, Weis1993] looked at large systems of dipoles at a low density in 3D. In these studies it was found that with no other effects involved, dipoles would choose to form chains with the dipoles lining up nose to tail.

The fact that dipoles formed chains was surprising, as the minimum energy conformation for a system of more than 5 dipoles would be a ring configuration [31]. To understand the reason

for this lack of ring structures in 3D we must consider the thermodynamics of the system. By forming rings, the dipoles would reduce the entropy of the system. This means there is a delicate balance between the minimisation of energy and entropy effects which stop large chains of dipoles forming into rings.

In order to relate all of the above behaviour to the real world, it is necessary to add one small complication. All of the above has been considering simple spherical particles of one size. Systems such as this are known as monodisperse. In reality, systems may be more complicated, containing particles of different size and shape and are polydisperse. This can affect things in two ways: by changing the effective dipole-dipole interaction and making the particle multi-domain.

Dipolar particles are quite sensitive to a change in shape. Any stretching of the particles in one direction can change the interaction energies such that a nose to tail configuration is equally likely to an anti-parallel side by side structure [32, 33] allowing clusters to develop. Any system which contains such stretched particles will include clusters as well as chains and rings.

Above a certain size, particles can no longer be thought of as containing a single dipole. Larger particles can contain independent domains, each containing a discrete dipole moment of its own. The size at which this happens depends strongly on the material and can range from 40nm to 200nm. The transition from single domain to multi-domain is accompanied by a decrease of the particle coercivity. The coercivity quantifies the stability of the magnetic properties of a material or object. This is typically measured using a B-H analyser and relates to the intensity of an applied magnetic field, B, required to reduce the magnetization, M, of a magnetically saturated material to zero. The region of single domain particle size is bounded below by the regime of superparamagnetic particles. These particles are so small that the energy barrier for fluctuations of the dipole moment is low enough to be effected by thermal fluctuations. This means that, over time, the net dipole moment is seen to be 0. This inherently has a dependence on temperature that will be looked at below. Typically, a superparamagnetic particle exhibits no coercivity, and the single domain particles have the highest coercivity which means that they are the most stable magnetically. For magnetite particles single domain limit is around 80 nm [34, 35, 36], below this size and above around 25 nm they exhibit single domain characteristics and their coercivity is maximised. Multi-domain particles are more complex and considerably less studied due to their unsuitability for most applications. It is worth clarifying that this discussion does consider the bulk material regime but rather the behaviour of a single dipolar particle in a dipolar fluid or colloidal suspension.

As stated previously, with regards to superparamanetic particles and the lower bound for the size of a single domain particle, there is a role to be played by temperature. From the definition of a superparamagnetic particle, it is necessary for the temperature to be high enough to supply the energy to overcome the energy barrier holding a dipole in an orientation. Below a certain temperature, known as the blocking temperature, there will not be sufficient thermal energy to cause the dipolar fluctuations to exhibit superparamagnetic behaviour. For the purposes of the discussion above, it is assumed that the temperature is room temperature and the blocking temperature is well below this.

Returning to ferrogels, it is now clear how the embedding of dipolar particles could induce a deformation of a gel. Restricting the movement of dipolar particles under a magnetic field would induce a stress on the gel body and cause deformation or changes in the rheological properties of the gel. The behaviour of ferrogels is explored much further in chapter 4 where

work on the effect of the microscopic structure is presented.

1.3 Computer Simulations

All of these areas, which are currently of significant interest, pose significant problems for experimentalists. The range of tools available to view and measure such systems at the scales of interest can only reveal limited data and make it difficult to understand the processes at hand. It is this difficulty that computer simulations can fill. The ability to actually see microscopic processes unfold can yield insights which are unavailable through traditional experimental approaches. In this way, computation has become an indispensable tool in modern science.

With the rise of the computer in all areas of life, there has been a parallel rise in the number of ways to utilise them in simulations. In fact, the underlying ideas which underpin most, if not all, of the modern algorithms for simulations were developed in the 1940s and 1950s [37, 38] and have changed little since. In the case of many body simulations, these approaches can be summarised into two basic types: Molecular Dynamics (MD) and Monte Carlo (MC).

Briefly, MD is a way of calculating the equilibrium and kinetic properties of a many body system. It generally integrates Newtons equations of motion with a pairwise potential governing the inter-particle interactions. This can be extended to handle long range interactions, such as Coulombic potentials, in a relatively straightforward way. The classical approximation inherent in the use of Newtons equations of motions, turns out to be a good one for a large number of systems.

MC takes a different approach. This uses 'importance-sampling' to generate probability distributions according to physical laws. It is very difficult, although not impossible, to examine transport properties with this approach, but it can yield useful information on equilibrium states. Further to this it can be easily biased to probe rarely explored regions of phase space which would be near impossible to study with an MD simulation. Further detail of both MC and MD will be given in Chapter 3.

In many ways, a computer simulation is analogous to an experiment. Frequently, experiments are a simplification of a real world system with the environmental conditions controlled in order to be able to take measurements and analyse the system. This means that care must be taken in experiment to ensure initial conditions and restricted interactions do not bias the results. This care over conditions is even more important when considering computer experiments. Simulation brings with it the ability to configure systems in ways which are completely unrealistic and can introduce pathological errors before the simulation has even begun. This freedom to configure things at will is also important for the power of computer experiments. It is, in part, this ability which allows simulations to explore systems difficult to access with experiment and to easily guide systems to test ideas difficult to reach with real experiments. As long as sufficient care is taken when choosing an approach computers provide an indispensable tool in studying complex systems.

The rest of this document is organised in the following way. The next chapter outlines some of the technology and computer science necessary to carry out modern computer simulations. There is a focus on the cutting edge technology of GPUs as during the course of the work presented here, a bespoke code was developed making use of this technology.

Following this, there is a chapter on Computer Simulation Methods, outlining more detail on MC and MD. The next 2 chapters outline published work on ferrogels and adsorption of

polymers.

Chapter 2

Parallel Computing

2.1 Introduction

Computational solution of equations and simulation have become indispensable to modern science. Ever since the use of the abacus in ancient Babylon, man has been making use of machines designed to aid in scientific progress. Today, the use of computers is ubiquitous. Computational modelling touches every corner of science, from the numerical solution of abstracted equations in mathematics, to the complexities of climate modelling so often discussed in the media. This extension of the use of computers to problems which are complex, multi-variable and above all require huge data sets has seen the need for ever faster processors, larger hard disks and more memory.

To fulfil these requirements, large supercomputing centres have developed, often on a national scale such as the High-End Computing Terascale Resource (HECToR) in the UK. HECToR is a specialised supercomputer, designed and built by Cray [39]. It consists of a large number of 'blades', large boards which contain a number of processors and random access memory, which are connected together by a fast backbone to allow data to be passed between them. All of these blades, the interconnect between them and the drivers to run it all are bespoke and are vastly different to anything seen on a desktop.

An alternative approach to this large bespoke approach is taken by the Barcelona Supercomputing Centre (BSC) with their Mare Nostrum cluster. In this case, off-the-shelf rack mounted servers have been supplied by IBM and connected together using a similar approach to that taken in server farms for internet use. This simple approach has allowed for surprisingly impressive results, with Mare Nostrum being among the top 5 of the 500 fastest supercomputers list on launch [40].

The aim of large national supercomputing centres such as these, is to allow scientific access to computing resources without needing the expertise to construct and administer a cluster and to avoid the sometimes significant financial outlay necessary to purchase the hardware for a cluster. In practice, the use of these large supercomputing centres can be frustrating due to the delays of waiting in queues for free resources, regular maintenance downtime and the difficulty of knowing, in some cases, precisely what the cost of your simulation will be. If the simulations required were small enough it could be cost effective to maintain a small cluster with a simple fast ethernet backbone. In 2009, £10000 was sufficient to buy a 44 core cluster, consisting of 42 compute cores with 2 cores used as a head node, and a fast ethernet backbone. There was

(a) (b)

Figure 2.1: (a) The former church holding Mare Nostrum and (b) the row of racks inside the church showing the rack mounted servers and inter-connect. Even the simple concept of Mare Nostrum requires the complex, high speed backbone (the orange mass of cables, shown in the centre of (b)) to be fully exploited.

a reasonable cost in time to set up and administer the cluster but this was gained back in the minimal delays waiting in queues and the number of development simulations that were run with no usable data. It would be desirable to be able to access something approaching the undeniable power of these clusters with a fraction of the cost and preferably on a desktop with little need for extra training and administration. Until recently, there were no alternatives which could be utilised by anyone but a specialist in computer science, but in 2007, Nvidia launched the first version of their snappily titled Compute Unified Device Architecture (CUDA).

This first version of CUDA consisted of two application programming Interfaces (API) which allowed the hardware designed for graphics representations, the graphical processing unit (GPU), to be used for calculations. This was not the first time that GPUs had been used for general purpose calculation. Indeed, the phrase 'General Purpose computing on Graphical Processing Units', often reduced to GPGPU, was coined in a 2002 paper by Mark Harris [41] who went on to become chief technologist for Nvidia. However, in 2002, the use of this hardware required specialist knowledge and remained the playground of computer scientists. It was the release of a high-level API which reduced the technical knowledge necessary to utilise GPUs that began the move to mainstream scientific computing. CUDA was not the only attempt to make the power of GPUs accessible. An open source project known as OpenCL was also developed but has not gained the level of acceptance that CUDA has. This is largely due to the slower pace of development of OpenCL causing it to lag behind CUDA.

Since those early days, GPUs have made great leaps forward with new versions of APIs realeased regularly and many mainstream computational codes already ported to exploit the capabilities of GPGPU. The GPU has even made the leap to the national supercomputing centres with the fastest supercomputer on the list in November 2012 being accelerated by NVIDIA GPUs [42]. For the short term at least, GPUs are here to stay.

The information held in this chapter played a large part in the work presented in this thesis. The simulation sizes and length of simulation runs formed a significant technological challenge which would have been very difficult to solve with a traditional approach to computing. Given the long run times required and the development time necessary for developing simulations of new systems, it was clear the flexibility and power supplied by GPU's was the most suitable solution. This meant a large amount of time during which the work presented in this thesis was carried out, was spent learning about and adapting algorithms for the use on massively parallel

systems. It is that information which is presented in this chapter.

The rest of this chapter will give a brief overview of the computer science needed to understand parallel computing on both the CPU and the GPU. The next section will look at fundamental ideas in parallel computing and explore the types of parallelism and important considerations when developing a parallel algorithm. Following this there are two short sections covering the specifics of CPU and GPU computing and the advantages and limitations of each. The fundamental principles of computer architecture and some coding practices are adapted from references [43], [44] and [45].

2.2 Parallelism in computing

In order to understand the changes which must be made to an algorithm when using parallel computing it is first necessary to understand something about the computer science. The internal architecture of a computer or a cluster of computers has a large effect on the way algorithms can be optimised. A program which may run extremely quickly on a serial computer will not necessarily be suitable for a parallel approach. There are certain aspects of programming which are universal regardless of the architecture, but in general, in order to make the largest gains on any particular system, its architecture should be taken into account.

Any program can be thought of as consisting of an instruction set and a data set. The instruction set is simply the arithmetic instructions that make up your program, such as the numerical integration in the case of molecular dynamics (MD). Likewise the data set consists of all the properties of all of the particles in a many-body system. Breaking our program into two in this very straightforward way, allows us to characterise our system according to their approach to parallelism. This method of classifying parallel architectures was first proposed by Flynn in 1972 [46].

The first and most obvious type of program is a serial program. In terms of instruction and data sets we can label this as Single Instruction, Single Data (SISD). To visualise this simple kind of computing, the first computers can be brought to mind. Early computers, as well as filling a whole room, were programmed via a series of punch cards which were inserted in a linear fashion. Each card carried a single instruction to be carried out on a single piece of data. This kind of inherently serial approach leaves little room for optimisation and in the event of an error it is generally necessary to rerun the whole program to regenerate the solution. This approach to computing is largely obsolete with the advent of multi-core CPUs in desktops and most commercial programs utilise parallelism of some sort. Serial programmes do have their uses however. The simplicity of writing a serial code means that for small problems or as initial steps in writing a new code, serial programming is a useful tool. In fact, before moving on to make a code parallel it is recommended that a serial version be optimised as this will yield returns in speed later on and problems can be found much more simply.

The next approach which can be mooted in these terms is Multiple Instruction, Single Data (MISD) architectures. This isn't an entirely intuitive form of computing and hasn't really been seen as an independent architecture. Rather, it is seen as a model for something called *pipelining* which is a method used by processors to schedule work more efficiently. As such, it is entirely handled by modern compilers and has no real effect on coding or algorithm efficiency and is mentioned here purely for completeness. Moving on from this, we come to Multiple Instruction, Multiple Data (MIMD). For modern computer users, familiar with multi-core processors and,

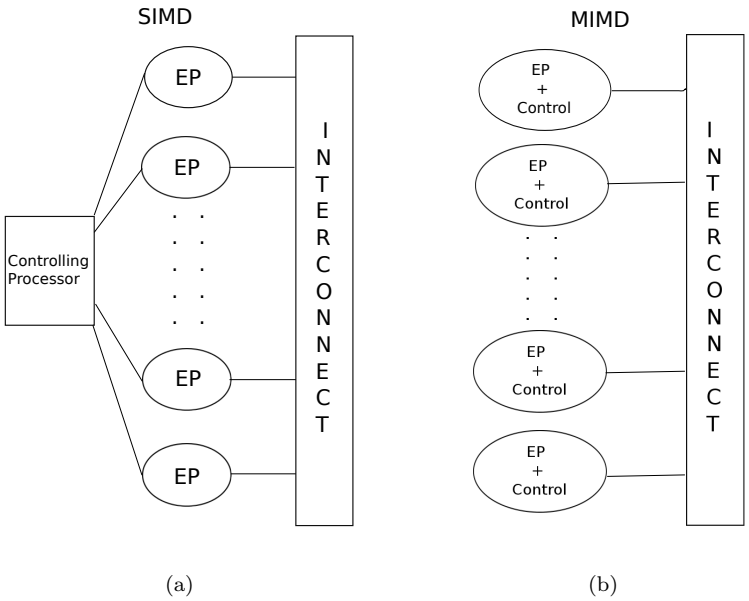


Figure 2.2: Schematic architectures for SIMD (a) and MIMD (b). The controlling processor (labelled 'Control' in (b)) co-ordinates the instruction to be carried out which is then passed to the execution processors (labelled 'EP' in the figures). In a SIMD architecture, all EPs are passed the same instruction to operate on different data.

if HPC users, multiple computer clusters, this may be the most familiar architecture. In this case, each processor has access to a copy of the programme and copies of some, if not all of the data. The simplest example of this type of system would be a modern multi-core computer. In this case, each 'core' of the system is a fully functional CPU, with its own scheduler and caches. Despite each core having access to the same data store in the case of single machines, from a programming point of view, this model would still generally generate multiple copies of the data set to be rationalised at the end of the routine or program. This may seem wasteful and to require an overly large data store, however, it is the simplest way to avoid clashes in data access between the processors. If there were only one copy of a variable, the use of so-called 'atomic' operations would become necessary, enforcing serial access to a variable and reducing the parallelism of the code. This would make any code using atomic operations essentially serial at this point.

The HPC extension of this is the compute cluster. Each node in this case is a fully functional computer in its own right and is connected to others through costly interconnects. Generally, there will be a 'master' processor coordinating the work to some level, but this is simply an identical processor to the compute cores and this decision on which is the coordinating processor is carried out at the software level. MIMD systems in general require the most memory of all the methods as each processor requires a copy of the program and data. This duplication of the data means this model requires large amounts of memory per computation core as entire copies of the system must be held in different memory locations. There is also a risk of inducing race conditions with different values for the same variable being held in different places. Race conditions will be explored further later on in this chapter. This kind of architecture is reminiscent of a multi-node cluster made up of independent CPUs linked together by an ethernet backbone.

Finally, we come to Single Instruction, Multiple Data (SIMD) parallelism. This kind of parallelism places constraints on the problem being addressed and requires what, up until recently, could be considered an unusual architecture. SIMD architectures generally consist of a single controlling processor connected to a large number of smaller, simpler execution processors. This may seem like a trivial difference from the MIMD architecture outlined above, but the simplicity of the execution processors means there is a significant difference to these systems, highlighted by fig 2.2. These systems generally require less memory as there is only one copy of the programme being executed and one copy of the data. However, the requirement of all processors running the same instruction is a strong constraint and can limit the suitability of some algorithms for this architecture. This will be touched on again briefly in section 2.4 with more detail on the effect on algorithms in chapter 3.

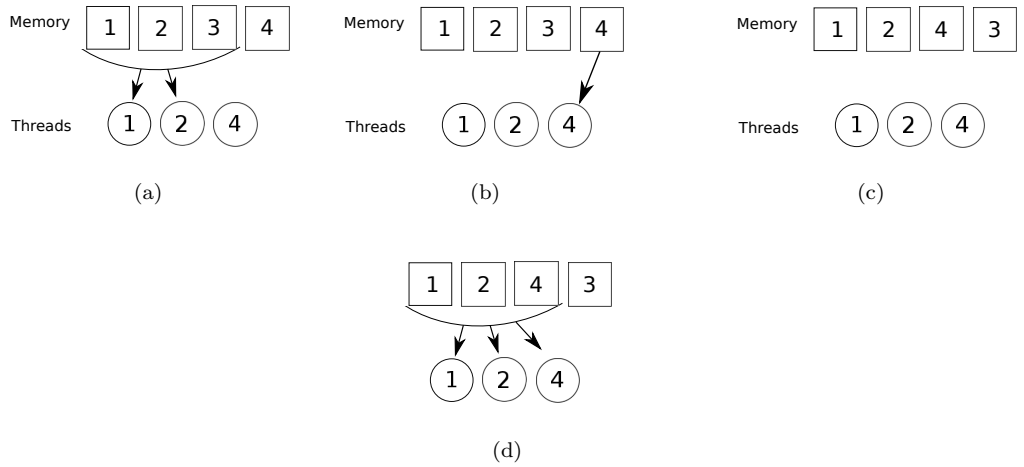


Figure 2.3: Schematic showing coalesced and uncoalesced memory access. In (a) the memory addresses to be loaded are 1,2 and 4. The first memory read takes addresses 1,2 and 3 meaning a second memory read is necessary for address 4 (b). In (c) the memory has been restructured, allowing addresses 1,2 and 4 to be accessed in one load (d).

The final general computer science idea we will explore before moving on to more real-world considerations will be memory access. As early as 1978, it was suggested that the architecture of computers was inherently flawed and limited [47]. The design of separate memory and processor systems incurs an inherent delay in the passing of data between them. If the time taken to retrieve the data for the processor to work on is too high, the processor can be forced to wait for a number of clock cycles before it can continue with its task. This problem with memory latency was given the name *the von Neumann bottleneck*.

Increases in processor speeds have the potential to compound the problem with a real possibility of memory buses being unable to keep up. The use of caching and small memory stores being available on the processor are aimed at reducing this problem but it still remains an issue as memory this close to the processor is so small that even the smallest of programs must resort to some kind of global memory. Surprisingly large gains can be made to even a well-designed algorithm if the some attention is paid to optimising the data structures.

An obvious way to reduce the impact of this memory latency is to minimise the number of memory loads. One way to do this is to ensure that in the algorithm design, once data is loaded all the operations it is needed for are carried out at once. In practice, this may be difficult, if

not impossible, to implement.

A different, and more realistic, approach would be to look at the data structures used and arrange these in a way that minimises the number of data loads required. In order to do this, it is necessary to understand something about the way that memory is accessed and read into the processor. When reading values for variables from memory, the values are not read individually. Memory is read in a parallel way with any read during a clock cycle reading in a section of physical memory consisting of a number of memory addresses. The size of this read is heavily dependent on the hardware involved. If the data which are required are spread across physical memory locations that span more than can be accessed in a single read, two reads will be needed to collect the data and the memory access is known as *strided*. Where the data required is stored in concurrent memory addresses which can be read at the same time, the memory is *coalesced*. Figure 2.3 illustrates this principle. The square boxes represent the memory addresses with the circles showing the data required by individual thread. In fig 2.3(a), the memory is strided with three physical addresses read in one cycle. This leaves one memory address unread requiring a second read to gather all the data. Figure 2.3(c) shows the physical memory coalesced to allow all the required data to be recovered in a single read. This may seem like a small gain, but with modern simulations consisting of thousands, if not hundreds of thousands of variables, the number of extra accesses necessary for strided memory can have a significant impact.

There is one more common problem which can arise in parallel programmes which stems from memory management. When many routines are running at the same time, it is possible that data can be corrupted by overwriting from several routines at the same time. The result can then become dependent on the order in which these routines finish in leading to false results and a program which is not reliable. The easiest way to understand the problem is in the flow chart shown in fig 2.4. In this figure we have a simple sum split over two nodes. Each node must add 1 to the initial value of the variable which is set to 0, meaning the correct result will be 2. The first step is each node making a copy of the variable to its local memory. Next, each

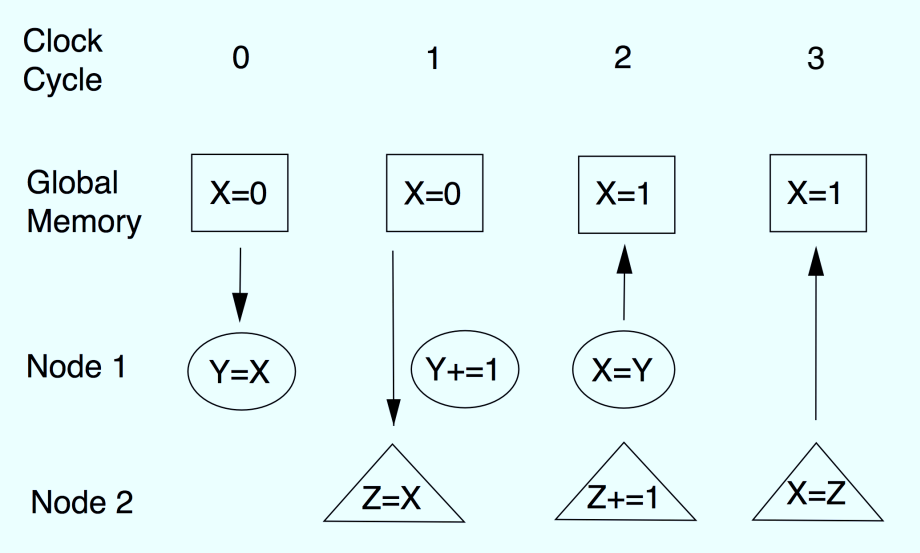


Figure 2.4: Example of a simple race condition. With no barriers or synchronisation the returned node 1 value is overwritten by the value calculated by node 2 leading to the wrong answer of 1.

node carries out the arithmetic procedure. Finally, each node copies the variable back to the initial memory location. However, because these nodes accessed the variables without locking it or synchronising their access in some way, the value copied back is 1. This is a trivial example where the algorithm is clearly incorrect but it is surprisingly easy for errors such as this to be found in a complex algorithm. When the outcome depends on the order of execution and there is no synchronisation, this is known as a *race condition*. To avoid errors like this, methods of communications between nodes have a variety of barriers, which hold execution until all nodes have reached a certain point, and locks (known as atomic operations) on variable access. This kind of error is not just limited to MIMD and distributed architectures but could be an issue in any kind of parallel architecture as the use of caches and different memory spaces are common.

From here, we will move on to consider the uses of the individual architectures inherent in CPU and then GPU computing.

2.3 CPU computing

Since the beginning of the modern microprocessor it has been the CPU which has dominated the computational landscape, forming the core of every home computer since the Commodore 64.

The technology involved has moved a long way since the initial microprocessors were developed, moving at least as fast, if not faster, than the well known Moore's law[1]. Until recently, the speed implications of Moore's law were adhered to by a simple increasing of transistors in a single processor with little change to architecture. As physical limitations, both in manufacturing and physical laws, have been reached, the increase required by Moore's law has led to vast changes in chip design with multi-core processors becoming the norm, which in turn has led to the concept of parallel computing reaching every home computer produced today. Despite this, most desktop applications do not take advantage of this increase in computer power and it is research which is leading the way in exploiting this extra capacity [48].

Even with the introduction of multi-processors, the research approach to parallel computing on CPUs has actually changed very little from a hardware viewpoint. Despite the ability to run multiple processes on one machine with multiple core processors, when limited to run on one physical machine the options are still limited. To be able to run large systems for reasonable lengths of time using a CPU based approach, it is necessary to chain a number of individual machines together and pass data between them. This immediately highlights a limitation of this approach. The speed and efficiency of the backbone which connects the different nodes of a cluster is of great importance if much data is required to be passed backwards and forwards. The problems this engenders can be viewed as an extension of the von Neumann bottleneck mentioned above. Shuffling data backwards and forwards between nodes can lead to nodes waiting for data, wasting clock cycles in a similar way a single CPU can wait for data from global memory. A great deal of money and research is concentrated towards developing faster, more efficient methods and technology to increase the bandwidth between nodes. Large modern clusters invest significant time and cost in setting up advanced, complex systems to increase bandwidth and reduce overhead for node communication. An example of this is Myrinet [49]. This consists of a low overhead optical fibre switching system, with a bandwidth as high as

[1]The number of transistors on an integrated circuit doubles approximately every two years. Sometimes misquoted as 'the speed of a processor'. Large chip manufacturers have been known to use this law in their planning of processor development.

tens of Gigabits per second, for connecting nodes to the cluster and a more efficient network protocol to reduce network messages. Myrinet even goes so far as to include modules that can be incorporated into any code that allows processes to access it directly without the need for initiating a call to the operating system. This direct access to the data transfer system can reduce the overhead for transfering data.

(a) (b)

Figure 2.5: The small cluster bought in 2010 for the use of the Camp group in Edinburgh. The cluster consists of 11 quad-core compute nodes, connected by a fast ethernet router to a head node housing 2TB of storage. and Even a small cluster such as this requires a significantly complex backbone in order to be interconnected sufficiently as shown in (b).

Moving away from the high-end, large national clusters, the rapid increases in processor power and networking technology are making it more feasible for small clusters to be built up by individual groups. While buying, setting up and maintaining a cluster brings with it a cost in terms of time and training for administration, this is more than gained back in terms of productivity with small or non-existent waiting times for execution and no costs for development runs which yield no data. With the high and sometimes uncertain costs of large, centrally run clusters, this smaller scale, less expensive cluster is often sufficient for all but the largest of simulations.

The interconnect in this small cluster is also significantly slower and less sophisticated than the Myrinet systems described above. The interconnect pictured in fig 2.5(b) is a simple, fast ethernet router with a bandwidth of 1000 Mbps connected by standard ethernet cables. This is 10 times slower than the Myrinet system outlined above and would definitely have an impact on the scaling efficiency across the cluster. Tests run on a system of 24066 atoms interacting via Lennard-Jones and coulombic potentials with the software LAMMPS [50] run on the Camp group cluster, demonstrated the impact of the backbone on the quick deviation away from linear scaling (fig 2.6) although it is likely this could improved slightly with more care on domain decomposition and with a larger system. Even though linear scaling remains unattainable with a low-end backbone, some usable increases in speed were still seen. When considered along with the reduced waiting time and free development simulations even this admittedly modest acceleration still makes the approach of building a low end cluster a useful and viable approach.

While all of the computer hardware and architecture explored above is important to understand when using and running a cluster it would be useless without some way to use it. Standard programming languages such as C, Fortran and C++ do not have ways to manage

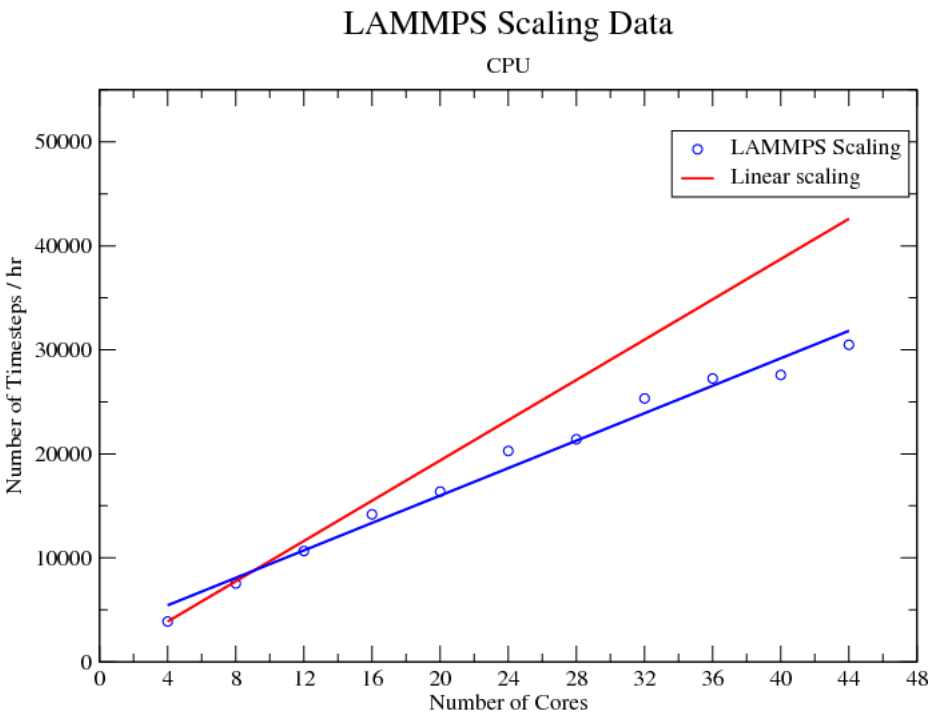


Figure 2.6: Time for simulations of 24066 atoms with Lennard-Jones and coulombic potentials using LAMMPS utilising an increasing number of cores on the Camp group cluster. Although a long way from linear scaling, largely due to the limitations of the interconnect, some speedup was supplied and the lack of waiting time and free development simulations more than make up for the slower speeds.

distributed systems inherently built into them. To make use of the power of both multi-core processors and multi-node clusters, language extensions have been developed which allow programs to be distributed amongst processors. The less complex if more limited of these methods is OpenMP.

OpenMP consists of a set of directives which can be added around sections of a programme, such as loops, which are inherently parallel, to share that loop across a number of processors. It is designed for use in shared memory architectures where all processors access the same global memory bank. This is exactly the architecture of today's multi-core desktop machines and with its simplicity of use, OpenMP is ideally suited to exploiting the full capability of a modern desktop. This strength also holds its limitation. Being purely designed for shared memory machines, it is not suitable for use on multi-node clusters where the memory spaces are separate and maintain independent address spaces. It can be used in multi-node clusters but only in cases where the user can be sure the processors will all be on the same node or as a subsidiary process to the more flexible message passing interface.

MPI is a standard which defines how communications will be carried out for the execution of parallel applications and libraries. The first version of the standard was completed in 1995 [51] with the first version of the now widespread library for using it, known as MPICH, appearing shortly afterwards [52]. With such a broad scope, MPI has become the dominant method for developing parallel programmes. Anything developed making use of MPI is immediately portable and scaleable to any other architecture which contain MPI libraries. It is this portability which is the strength of MPI.

As would be expected, something which contains such a large amount of portability and

generality is not as simple to use as OpenMP. Instead of being directive driven, MPI requires a number of library function calls to set up and run the multiple copies of sub-routines and variables that are necessary. This overhead to the code adds a certain amount of complexity to developing, and care must be taken with MPI to ensure that code executes in the correct order and that there are no errors such as race conditions. This added complexity means that MPI is not an automatic choice when developing parallel code, and if it is known that an application will only require the limited resources of a single node, it would be prudent to make use of the much simpler OpenMP.

While all of the advances in the use of hardware described above have been a great step forward in terms of computing capability, there is an associated increase in cost which came along with this technology. As compute capability and speed has increased, so too has power consumption. As can be seen in Table 2.1, for a given processor architecture the power consumption has increased. The process was retarded slightly with the fundamental change of architecture to multi-core processors, but the latest generation of processors are again increasing in power consumption.

Model	Release Date	Clock Speed	Thermal Design Power
Pentium III 1333	1999	1.33GHz	33.9W
Pentium 4 HT 672	2000	3.8GHz	115W
Pentium D 960	2005	3.6GHz	130W
Core 2 Duo E8600	2006	3.33GHz	65W
Core i3-560	2010	3.33GHz	73W
Core i5-680	2009	3.6Ghz	73W
Core i7-2700K	2011	3.5GHz	95W

Table 2.1: Table showing power usage of Intel processors over time. The general trend in increasing was broken by the introduction of the first multi-core CPU and then continued increasing. Information taken from [53]

While this increase in power consumption is not a huge issue for a desktop computer, when scaling up to a cluster with potentially thousands of processors, running costs can become an issue. With this in mind, there has been a recent movement towards lower power approaches with the phrases such as 'FLOPs per Watt', which means the number of floating point operations carried out per Watt of power consumed, appearing in more publications and promotional material of manufacturers and clusters. The idea of curbing the seemingly endless appetite for more power of compute clusters has even gone as far as to bring into being an alternative 'Top 500' list where the rankings are based on the energy efficiency of the machine [54, 55]. This has, in part, helped with the introduction of GPUs as a major tool in HPC and eased the path for their acceptance by the mainstream of computational science.

2.4 GPU computing

Although the conventional CPU has come to dominate HPC, it was not always the case. From the 70s until the early 90s, the large supercomputer suppliers focussed largely on vector processors. With the rise of use of home PCs and the development of the conventional microprocessor that came along with it, the vector processor was superseded in terms of price-to-performance ratios. Since then, despite attempts to find an alternative approach for HPC, the CPU has seen off all contenders due to the fast pace of development and their ease of use with respect

to programming.

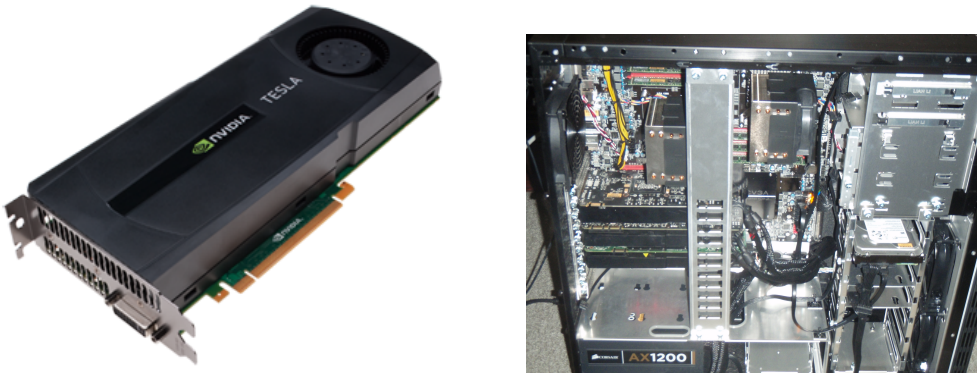
The lessons from this dominance and the attempts at alternative architectures and approaches to find a different hardware and programming model are clear. Firstly, in order to encourage major manufacturers to fund the development efforts necessary to make processors or accelerators useful in HPC there must be an alternative mainstream use for the hardware. The huge developments in CPU design are largely driven by the home PC market which provides a large source of income for manufacturers with HPC a distant secondary motivation.

Secondly, they must use a fairly familiar programming paradigm and not require vast specialist knowledge to use. The importance of this is demonstrated by Field Programmable Gate Arrays (FPGA) which despite being invented in the mid-80's and having been shown to accelerate certain computational approaches and simulations [56, 57, 58] has not been adopted by the research community. This reluctance in the face of proven gains seems strange but is largely due to the specialist knowledge that is necessary to make use of FPGA and the lack of available open source libraries[59].

GPUs are the latest technology to find a use in HPC. On their emergence as a possible tool for HPC in 2002, it was still necessary to have significant specialist knowledge in order to use them but the release in 2007 of a high level API by Nvidia changed things considerably. The initial version of this API, but there has been rapid development and the effort required to develop code to utilise GPU is now equivalent to that necessary to write MPI code. This increased ease of use, along with the alternative primary use of GPU technology, means that GPU meet both requirements above to become accepted as a useful, usable mainstream technology in HPC.

As with CPUs, there is the possibility of GPUs being useful to users who require less power than a supercomputing centre provides. In fact, GPUs are perhaps more attractive to the smaller scale user due to the large computing power they can bring to a single desktop. In addition to this small form factor, the original target of the home market brings with it the added advantage of usability. The average home computer user or even gamer does not want to have to use hardware which requires significant configuration or administration. As a result, to install a GPU capable of fast computation requires simply a spare slot in a standard home PC. The requirements for a complex backbone are removed as all networking is supplied by the mother board. As can be seen in figure 2.7(b), the installation of 3 working GPUs required minimal extra cabling and the administration is limited to installing a driver.

It was only in the early years of the 21^{st} century that GPUs became suitable for general purpose parallel computing. Early GPU worked in a linear fashion performing each step of the process necessary for producing a display one after the other. As the demands of modern computer graphics grew, driven by the lucrative computer gaming market, the separate hardware to process the graphics became more and more complicated, growing the number of simple arithmetic cores they contained until GPUs became the hugely parallel devices of today. Their architecture through all this development moved further away from the architecture used by CPUs in favour of an approach based on SIMD applications. As explained in section 2.2, this means they have many simple compute cores able to carry out the same instruction on many different pieces of data at once, an inherently parallel process. This type of parallelism does have its limitations. When working with a MIMD type of system, there exist many more possibilities for parallel approaches as different sub-routines could be executed by different processors at the same time. With the SIMD approach sub-routines must be run one at a time with the large amounts of data to operate on being vital to exploit parallel computing. In order



(a) (b)

Figure 2.7: (a) A single Nvidia Tesla GPU aimed at the HPC market. (b) A single workstation housing 3 GPUs and 2 CPUs. Although of similar compute power, this configuration with 3 GPU in one machine is visibly less complex than the cluster shown in 2.5

to understand the implications of this for using GPUs, it is necessary to look in more detail at their architecture and specifically their memory structure. Unless otherwise stated, the detail for this section is taken from the Nvidia CUDA C Programming guide [60] and will relate to Nvidia GPUs as these are currently the dominant technology in HPC. Nvidia GPUs are utilised in work presented in later chapters.

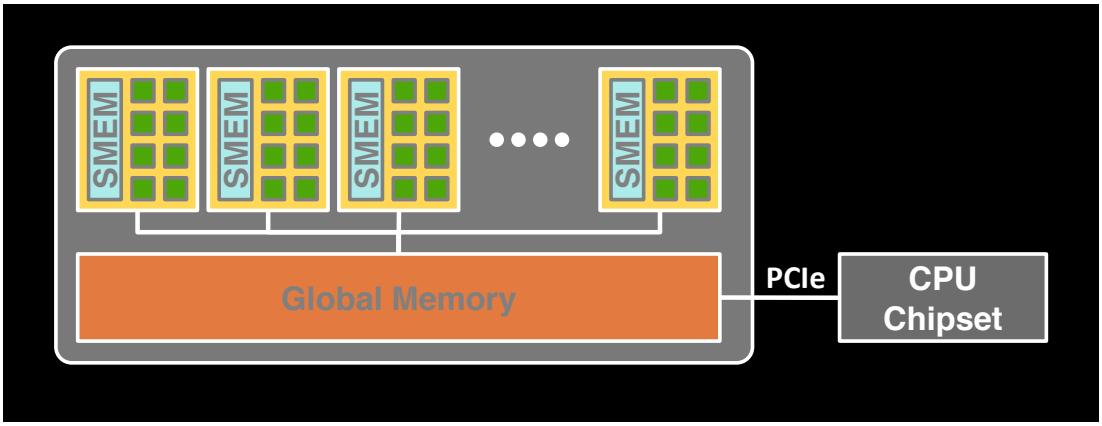


Figure 2.8: Schematic view of a GPU showing the main types of memory. The GPU is separate from the CPU and standard RAM and is accessed through the PCIE bus. Different memory on the GPU has different access times which can be used to accelerate memory access.Taken from [61]

Fig 2.8 shows a schematic of the main types of memory available on a GPU. The first thing to note is the GPU separation from the CPU system. GPUs are accelerators rather than an integral part of the CPU architecture. As such they are accessed via a Peripheral Component Interconnect Express (PCIE) bus. This view of the GPU as a peripheral means that there is no unified memory space shared between the CPU and GPU. Any data which is required by the GPU must first be copied across the PCIE bus and declared in the GPU memory. On the first appearance of GPUs this was a significant bottleneck as the speed of this transfer was slow. With the improvements of speed of PCIE this has become less of an issue with the latest currently available version reaching speeds of 15 GB s^{-1} [62] which is comparable with CPU

RAM. Even with this improvement, given the amount of data involved in modern computation, it is worth spending some time to reduce the amount of data that is transferred to and from the GPU. The ideal case is that all data necessary for computation is placed on the GPU at the beginning and left there for the duration of execution.

The largest memory block in a GPU is known as global memory. This is available to all processors on the GPU at all times. It is the slowest memory on the GPU but by far the most flexible, bearing a resemblance to RAM on a CPU system. In simulations of large systems, this is the only memory which is large enough to hold the data structures for the whole system. The bandwidth of this memory is still very quick when compared to standard DRAM with global memory bandwidth on a Tesla C2070 of 144 GB s^{-1}.

In addition to this large memory bank there are a number of smaller memory spaces known collectively as local memory. This designation as local refers to some of the internal architecture of the GPU. Inside the GPU, the different arithmetic units are grouped into streaming multiprocessors (SMs) with 32 arithmetic units making up an SM on the Tesla C2070. Each one of these SMs also has a control processor to manage it, some specialised units (such as for double precision arithmetic) and a small section of memory and caches. On the Tesla C2070 each SM contains 64 KB of configurable memory which can be shared between L1 cache and shared memory (it is the shared memory that is shown in fig 2.8). This shared memory is 'on-chip' memory, meaning that it shares the same silicon as the processor, which makes it extremely fast. The drawback with this shared memory is that it must be managed manually and variables must be declared as in shared memory. The small size of this shared memory can also make it difficult to utilise effectively as there must be a lot of juggling in order to get data into and out of shared memory when required. It is sometimes questionable whether the extra considerable development effort is worthwhile for the gains that can be made, but it is unquestionable that significant improvements in run time can be found.

This question of development time versus gains is a common problem with utilising GPU code, and it is not just in the use of this shared memory which poses this problem. There is a further type of memory, known as texture memory, which utilises hardware acceleration to maintain high access speeds. However, managing this memory is even more intensive than with shared memory and as a result, the cost of development is even larger.

Despite these costs, the use of GPUs still generates great interest due to the huge gains that are achievable. In 2010, an MD code specifically developed to be executed on a GPU was benchmarked [63] and it was found to be up to 60 times faster than on a single CPU. Intel have attempted to rebut the common misconception that GPU accelerated code can be hundreds of times faster than a CPU code on accepted benchmarks[64]. Their comparison tested a top of the range i7 processor against a 2 year old Nvidia GTX 280 on common benchmarks. They did not reveal what codes were used for the benchmark or what optimisations were carried out for either CPU or GPU. In the light of the lack of information and the commercial interests of Intel, we must be sceptical of the validity of their tests and could reasonably expect that the advantages brought by GPU acceleration would be understated. Despite this, the paper reported that GPUs are 2.5-14 times faster than CPUs across the benchmarks. It is this sometimes spectacular increase in execution speed which has made GPUs worth the effort.

In order to access the power of GPUs it is necessary to learn how to program them efficiently. Standard programming languages have no easy method to access the hardware and so until 2007 it remained largely a playground of computer scientists. In 2007, Nvidia released the first version

of CUDA, a language extension for C, which allowed people without specialist knowledge to access the hardware. Unsurprisingly, the programming model closely mirrors the architecture and is designed for applications which have high levels of parallelism built into them, allowing hundreds, if not thousands of threads to be spawned. In fact, in order to get the most out of using GPUs it is necessary to have as many threads as possible. This is required as part of the improvement in speed from using CUDA comes from hiding the latency of memory loads. While certain threads are waiting for data to be called from the memory, other threads are executing. This keeps the occupancy rate of the arithmetic cores high and allows a high throughput in the minimum amount of time.

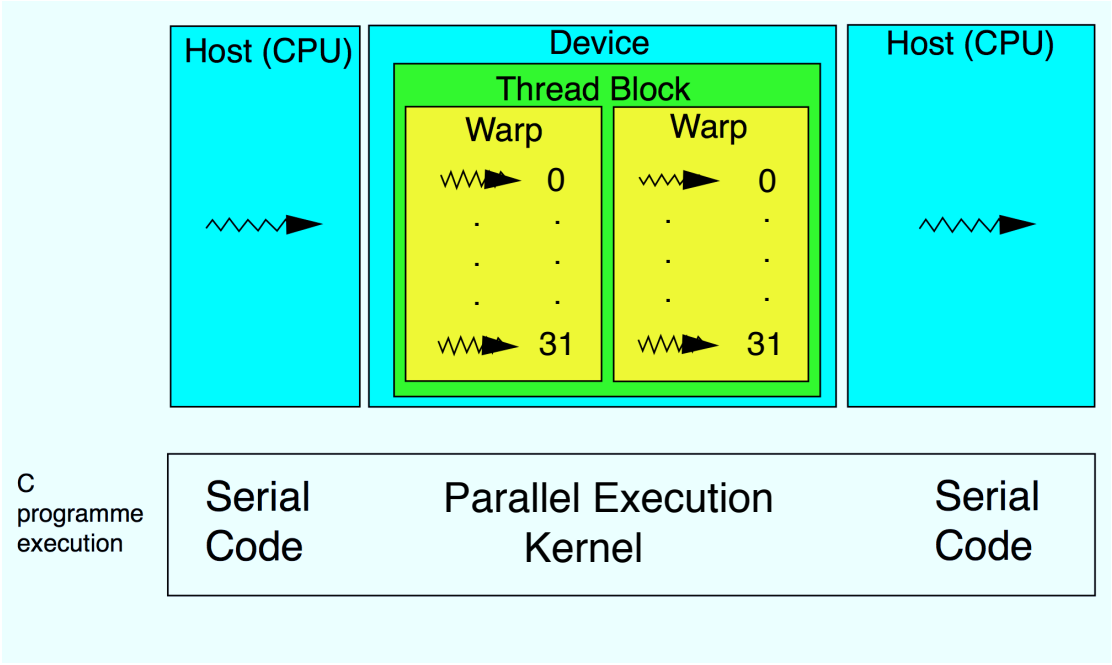


Figure 2.9: Cuda programming model. 1 thread runs on the host controlling the initiation of a parallel kernel for the GPU. Threads on the GPU are split into blocks with each block split into a number of 32 thread warps.

To invoke these threads it is necessary to use a special sub-routine known as a *kernel*. This tells the compiler that this section of code is to be executed on the GPU and passes some parameters necessary for the use of the GPU. Kernels are run as a number of threads, split into a number of *blocks*. These blocks will run in any order so it is important that individual blocks are independent, both in terms of instructions and data. In order to use the limited resources of shared memory, these blocks must be dimensioned to take account of the hardware resources available. Threads within a block are also organised into packets called *warps* which contain a maximum of 32 threads. For a schematic of the programming model see fig 2.9 All threads within a warp will be executed at the same time and it is at this level that the implementation of the algorithm can have an effect.

GPU architecture is unique in that it runs on a model that is designed to run optimally as a SIMD processor but is in fact a slightly modified version of this model. A strictly SIMD system would require that conditional statements do not cause significant branching within a set of threads as this would obviously lead to different instructions for different threads. In order to deal with this, CUDA uses a system described as Single Instruction, Multiple Thread (SIMT).

This deals with any conditional statements which cause branching by holding the threads with the divergent branch until the first set of threads have finished. This is easier to understand through fig 2.10. All threads run until the if statement. The threads with option 1 continue running while the threads which have branched to option 2 being held in the meantime. Once the option 1 branches have completed, these are held until the remaining threads have executed. This means that in some cases, code could become essentially serial for the execution of the branching. With the main motivation for using accelerators being faster executing code, this is something of a stumbling block. It is left to the programmer to manage the data such that bottlenecks such as this are avoided. If possible, data should be arranged so that all threads in a warp select the same branch at an if statement. For instance, in the example in fig 2.10, delays could be avoided if all threads had selected option 1 as there would have been no reason to pause any execution. In practice, this is very difficult to achieve. On simple MD simulations, this is achievable with careful management of neighbour lists. More will be said on these matters in chapter 3.

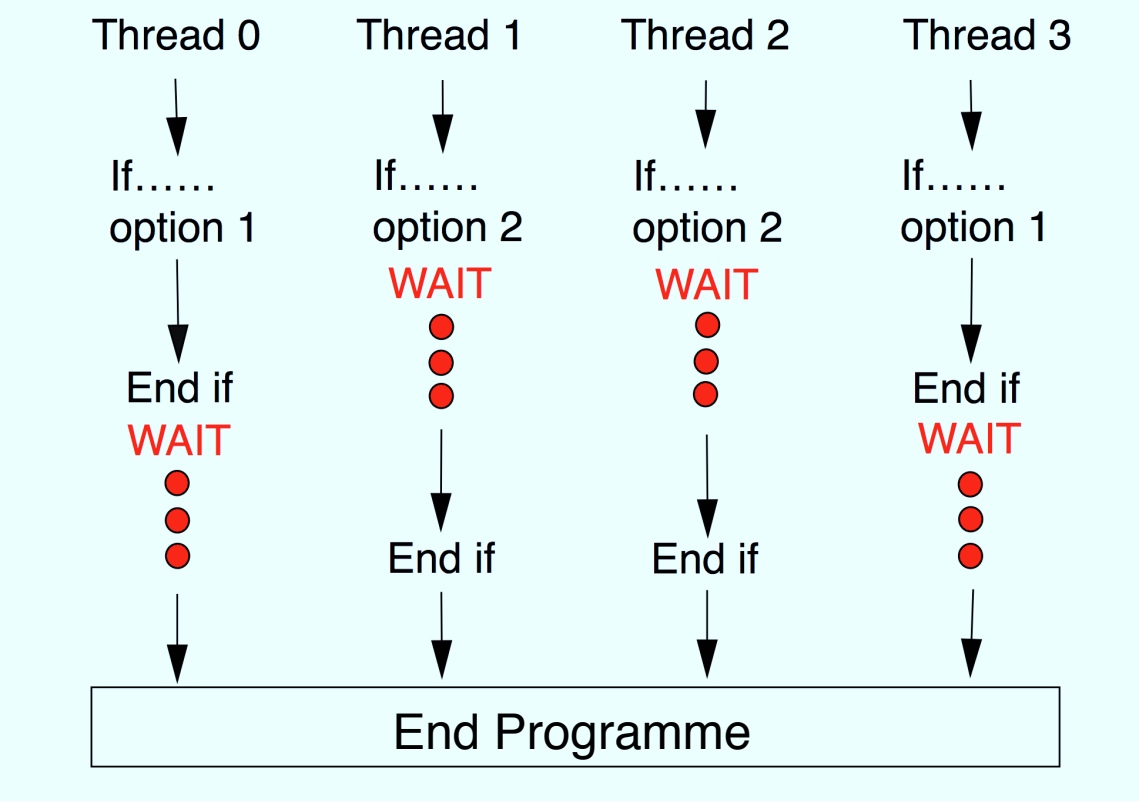


Figure 2.10: Demonstration of how CUDA handles branching within a warp. All threads run to the if statement with different threads on hold until certain branches complete. Once all branches are finished, execution continues in parallel.

As can be seen from above, programming for GPUs with CUDA is still far from straightforward and many people working in HPC do not see a place for these devices in the long term. While there is no doubt that there will be further developments in both CPU and GPU design, it is worth bearing in mind that CUDA is still a fairly immature technology. With the first version introduced in 2007, there were requirements to manage the memory in even more detail and with every new version of CUDA there come innovations to reduce the burden on

the programmer. In version 4 of CUDA a library named Thrust was included. This contained a number of routines, such as a reduction loop, which are somewhat challenging to implement on the GPU from scratch and could simply be called with existing data structures. While these standard libraries and routines may not be completely optimised for a given problem they do still harness a large part of the capability.

Away from CUDA, there are various projects in order to enable access to GPU without even knowledge of the language extension. OpenMM describes itself as 'a toolkit for molecular simulation' [65] and can be used as a library to call from existing code. It allows definition of custom force fields or potentials and is a flexible approach to fast development of simple MD simulations. More and more existing codes are also being ported to exploit GPU as well. While at their heart they utilise CUDA, that fact should be transparent to the user with the only real impact being on a system administrator to make sure libraries are present.

All of this adds up to a significant research effort being placed on finding ways to use both CUDA and GPUs in general in HPC. It is by no means certain that the fastest computers of the future will follow this model, but its adoption by a number of the large supercomputing centres means that in the medium term at least, they will remain a presence in research.

Chapter 3

Computer Simulation Methods

The use of computers themselves has pervaded all parts of science and, as shown in chapter 2, has undergone rapid and paradigm shifting changes even in the latter years of the 21^{st} century. It could be argued that the same cannot be said for the algorithms we employ to make use of computers. The Monte-Carlo (MC) algorithm, detailed below, was first proposed in 1949 [38] and the molecular dynamics(MD) approach followed around ten years later [66, 37]. Indeed, on initial inspection very little seems to have changed in the algorithms, with most simulations carried out today relying on one of these algorithms at their core.

While the central algorithms do indeed rely on the general approaches laid out so many years ago, their application has often evolved to be nearly unrecognisable. MC simulations have moved on from single simulations sampling a given probability distribution to the advanced methods using biasing for free-energy calculations [67], methods combining selection criteria, and even methods designed to yield dynamic properties [68]. For its part, MD is no longer limited to purely classical equations of motion, with both quasi-quantum [69] and fully quantum [70] approaches now possible. Coarse graining has further extended the use of these algorithms, allowing very large systems to be studied which would be unobtainable with a 'molecular mechanics' approach where every atom is included explicitly.

This chapter will firstly cover general principles of finite-size simulations, before looking in a little more detail at both MC and MD simulations, and some considerations for their implementation. The final part of this chapter will look at observables which can be calculated from the simulations for comparison with theory and experiment. While the details here will largely focus on the study of liquids, the majority of the principles may be applied to the simulation of any many-body system.

3.1 Features of finite-size simulations

Although MD and MC vary greatly in their approach, they share some general principles which govern the behaviour of both. Limitations in computing power mean that it is more often than not impossible to include as many interacting particles as would be contained in even a small test tube. Even if a simulation of that size were possible, there may remain huge difficulties of dealing with boundary conditions such as the walls of the test tube and the free surface. Often, simulations of bulk systems take a different view where it is imagined that the system of interest can be thought of as consisting of an infinite array of identical volumes or replicas and

by simulating one of these volumes, properties can be calculated for the whole system. When breaking the system up like this, there must be something in place to take care of particles moving out of the simulated volume. In line with the idea of a central box being surrounded by an infinite number of identical replicas, a particle which moves across the simulation boundary is instantly replaced with a replica particle leaving the appropriate bordering box. This is more easily visualised in fig 3.1 with particle number 1 shown on the boundary of the central box and appearing at both the top and bottom of the central box. Boundaries of this sort are known as periodic boundary conditions (PBC).

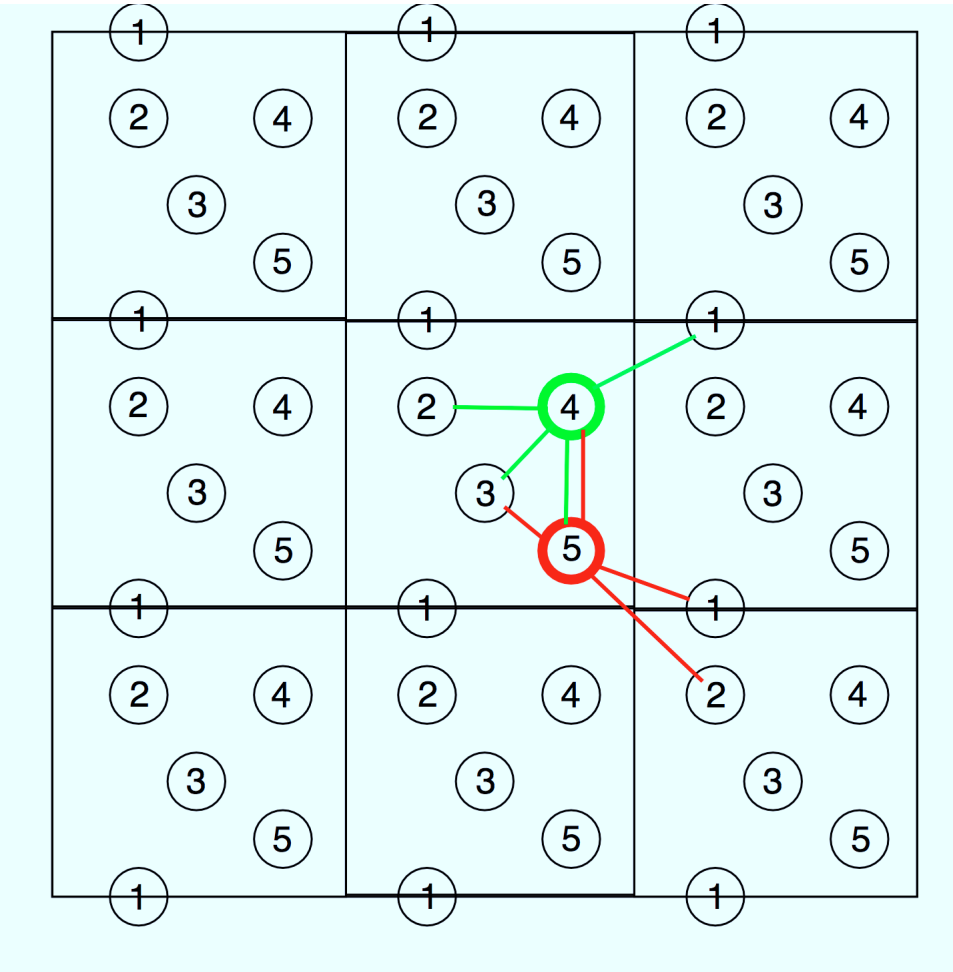


Figure 3.1: Representative view of a system to be studied by simulation with periodic boundary conditions and the minimum image convention imposed. The system is represented by identical replicas surrounding the central box which is the only one to be simulated. Particles interact with their nearest neighbours across volume boundaries.

A result of this picture of infinitely many identical boxes is the fact that there are infinitely many copies of a single particle in the system. Assuming the particles have an interaction that is longer ranged than a simple hard sphere interaction, there is the possibility of counting interactions with particles more than once. In order to avoid this, the minimum image convention can be enforced. Simply put, this dictates that particles will interact with those that are closest to themselves across the boundaries of the simulation box. Again, referring to fig 3.1, it can be seen how particles 4 and 5 would select their nearest neighbours. To comply with the minimum

image convention, it is necessary to truncate the non-bonded interactions to less than half the smallest box length. If the non-bonded interactions extended beyond this length, the particle would feel the influence from multiple copies of other particles in the system. This runs the risk of artefacts being introduced and unrealistic results and conclusions being drawn from the simulations.

The particle interactions are governed by *potentials*, otherwise known as *force fields*. In general, these are empirical in nature, designed to match experimentally determined characteristics. These potentials can be broadly split into two groups: *bonded* and *non-bonded*. As the names suggests, bonded potentials deal with intramolecular details such as bond lengths and angles, while non-bonded potentials deal with intermolecular and short/long ranged particle-particle interactions such as hard sphere, van der Waals, and Coulomb interactions. It is possible to include other factors, such as spin and other quantum effects, into the structure of the potentials and interactions but these will not be covered here as all work described here considered only the classical regime.

The simplest potentials and systems would be those that interact through a pure hard-sphere repulsion. These systems would be analogous to a box of ball bearings with no attractions between each other or the box they are in (for now, we will assume that these ball bearings are in a vacuum and that there are no gravitational effects) and they will simply bounce off of each other when they collide. In this case the potential could be written as

$$\Phi_{\mathrm{HS}} = \begin{cases} 0 & r_{ij} > \sigma \\ \infty & r_{ij} \leq \sigma \end{cases} \tag{3.1}$$

where r_{ij} is the distance between two particle centres and σ is the particle diameter. The simple form of this potential makes it easy to work with but this kind of interaction is limited in its use. At the other extreme we have the Skyrme interaction [71] which is widely used in the simulation of nuclear matter and, in its simplest two-body form (which can give some qualitative but not quantitative insight) can be written as

$$\begin{aligned} \Phi_{\mathrm{Skyrme}} = \quad & t_0(1 + x_0 P^\sigma)\delta(\vec{r}_1 - \vec{r}_2) \\ & + \tfrac{1}{2} t_1[\delta(\vec{r}_1 - \vec{r}_2)\vec{k}^2 + \vec{k}^2\delta(\vec{r}_1 - \vec{r}_2)] + t_2\vec{k}\delta(\vec{r}_1 - \vec{r}_2)\vec{k} \\ & + iW_0(\vec{\sigma}^{(1)} + \vec{\sigma}^{(2)})\vec{k} \times \delta(\vec{r}_1 - \vec{r}_2)\vec{k} \end{aligned} \tag{3.2}$$

where $\vec{k}$ and P^σ are respectively momentum and projection operators and t_n, x_0 and W_0 are fitting parameters.

The difficulty of using a potential of such complexity is clear and if it were necessary to include such complicated potentials to gain an insight into the behaviour of even simple systems the use of such models would be severely limited. Luckily, in many cases there is a middle ground between the simplicity of the hard-sphere model and the complexity of a fully quantum mechanical model. The Lennard-Jones(LJ) potential [72] captures both the medium range van der Waals attraction and the short range Pauli repulsion, while ignoring some of the complexities of a fully quantum mechanical description. The potential is commonly written as

$$\Phi = 4\epsilon \left[\left(\frac{\sigma}{r_{ij}} \right)^{12} - \left(\frac{\sigma}{r_{ij}} \right)^{6} \right] \tag{3.3}$$

where ϵ is the well depth, which can be thought of as the strength of the attraction, and r_{ij} is the distance between two particles. This potential is very widely used in simulation and with changes to ϵ and σ can be used to cover a wide range of materials. These values are generally empirically chosen, although simulations including quantum effects can be carried out and used for parameterisation. While the form given here is the canonical form, there are many modifications to this to account for different situations. For instance, in simulations where a surface is necessary in the xy plane, it is possible to integrate the x and y variables out of the standard LJ potential and thus emulate an infinitely large plane [73]. There are a large number of related potentials with similar forms, all with particular uses (For example, the Yukawa[74], Buckingham [75] and Morse[76] potentials), but it is the LJ potential which is used in the work described here.

While the LJ potential is considered a medium range potential, in order to implement it effectively in computer simulations, it is necessary to truncate the potential in some way. The most obvious and simple approach is through a simple cut-off and disregarding any interactions beyond a certain distance. This approach has the advantage of being very straightforward to implement and greatly reducing computational cost, but comes with the effect of introducing a discontinuity at the cut-off radius. This discontinuity translates into an impulse in the force calculation and does have an effect on both the distribution of particles in such a system and on the value of the potential. Disregarding the tail of the LJ potential underestimates the potential energy of the system and in order to calculate the potential correctly requires the addition of so called tail corrections. Although these tail corrections can be small, they can have significant effect on the calculation of certain properties such as phase boundaries, and any simulation studying such phenomena must include the tail corrections [77].

Building on this, with the aim of maintaining the simplicity of the truncation, it is possible to simply shift the potential so that it goes to zero at the cut-off point. This removes the sudden jumps in energy and the impulses felt by particles as they cross the cut-off but does still have some effect on the potential energy of a system, and where this value is important, tail corrections must still be applied. The truncated and shifted LJ is given by

$$U_{\text{tr-sh}}(r) = \begin{cases} U_{\text{LJ}}(r) - U_{\text{LJ}}(r_c) & r \leq r_c \\ 0 & r > r_c \end{cases} \tag{3.4}$$

Bonded potentials are also empirical potentials whose angles and lengths can be found by either experiment or quantum chemistry. In cases where experiment can struggle to provide accurate values for the structure of a molecule, a very detailed simulation of a single molecule utilising the most detailed level of theory, such as density functional theory (DFT), can be utilised. This can find the lowest energy configuration and the associated values for equilibrium bond lengths and angles.

Once the values for bond lengths and angles have been found it is not particularly straightforward to include these into a standard algorithm. The most commonly used approaches are algorithms named SHAKE [78] and RATTLE [79] with the former only constraining atom positions and the latter being an extension to constrain velocities and positions. The SHAKE and RATTLE algorithms are largely analogous to using either the Verlet or velocity Verlet integration schemes and contain the same shortcomings and advantages as these schemes. Details on the use of Verlet schemes will be included in section 3.3.

While the use of complex algorithmic approaches such as SHAKE and RATTLE are nec-

cessary for simulations which require higher accuracy, when bond angles are of no importance and the constraints can be reduced to those of simply proximity a much simpler approach can be used. The addition of an extra term to the potential in the form of a finitely extensible, non-linear elastic (FENE) [80, 81] potential places limits on the separation between two particles in a simulation while allowing them the freedom to stretch away from each other. When using this along with the LJ potential the total potential would simply be $U_{\mathrm{LJ}} + U_{\mathrm{FENE}}$ where

$$U_{\mathrm{FENE}} = -\frac{1}{2} k R_0^2 \ln\left(1 - \frac{r^2}{R_0^2}\right) \tag{3.5}$$

where k is a spring constant and R_0 is the maximum extent of the bond. When just considering just the potential, this restraint does not seem to simplify things hugely from the SHAKE or RATTLE approaches. However, when it comes to implementation, SHAKE and RATTLE require iterative solvers and extra algorithmic steps whereas the FENE potential is simply an additional term in the potential solved in a single step and with no iteration. This simplicity and ease of application brings large enough gains in computational efficiency and development time.

One property all of the potentials mentioned so far have in common is their relatively short range. If we were to confine ourselves to potentials of this kind, certain important physical effects would remain out of reach. The most common long-range interaction which plays an important role in physical phenomena is the Coulomb interaction. In certain systems, such as ionic systems, the long range nature of the Coulomb interaction can be screened as a charged fluid can act as a shield. It is these kinds of systems that the Debye-Hückel theory [82] can be applied to, simplifying calculations of the properties of the system.

Conceptually, the Debye-Hückel approach is simple to understand. In an ionic system, it would make sense that any individual ion would be surrounded preferentially by ions of the opposite charge(fig 3.2). This 'shell' of oppositely charged ions would screen an ion from others beyond the shell, hence transforming a long-ranged interaction into one of short range.

This intuitive approach implies that such a potential would be some kind of exponential function, rapidly approaching zero as the distance from the ion increased. To get a more definite form for such an equation and to tease out the assumptions and limitations it is necessary to take a more rigorous approach. In order to study systems where electrostatics are important it is necessary to solve the Poisson equation which, for a system of two types of ion with equal and opposite charges, can be written as

$$\nabla^2 \Phi(\vec{r}) = -\frac{1}{\varepsilon_r \varepsilon_0}(q n_1(\vec{r}) - q n_2(\vec{r})) \tag{3.6}$$

where $n_j(\vec{r})$ is the concentration of the j^{th} ion, ε_r is the static permittivity and ε_0 is the dielectric permittivity of the vacuum. Assuming the system to be in contact with a heat bath at temperature T, the concentration can take the form of a Boltzmann distribution:

$$n_j(\vec{r}) = \bar{n}_j \exp\left(-\frac{q_j \Phi(\vec{r})}{k_B T}\right) \tag{3.7}$$

where k_B is Boltzmanns constant and $\bar{n}$ is the average of n_j. Substituting this into Poissons

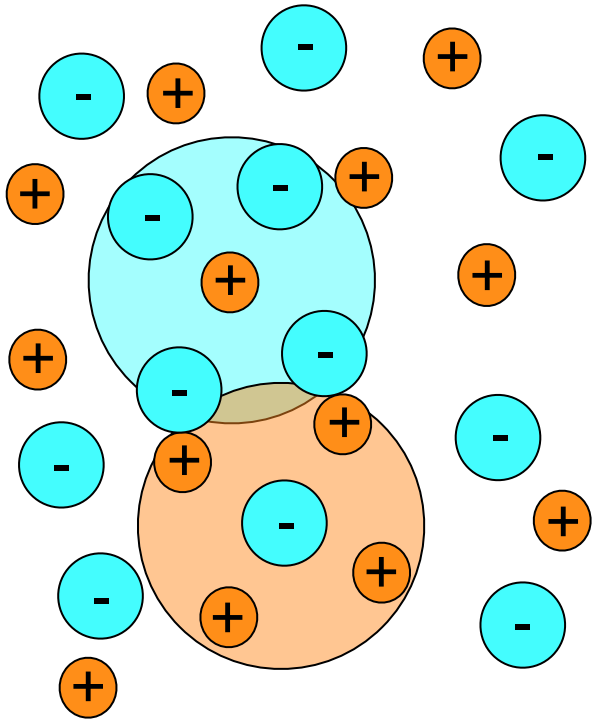


Figure 3.2: Distribution of ions in a solution. Dynamically evolving 'shells' of ions of opposite charge are arranged around ions.

equation, we find the Poisson-Boltzmann equation

$$\nabla^2\Phi(\vec{r}) = -\frac{\bar{n}q}{\varepsilon_r\varepsilon_0}\left[\exp\left(\frac{q\Phi(\vec{r})}{k_BT}\right) - \exp\left(-\frac{q\Phi(\vec{r})}{k_BT}\right)\right] \tag{3.8}$$

which can be more concisely written as

$$\nabla^2\Phi(\vec{r}) = -\frac{2\bar{n}q}{\varepsilon_r\varepsilon_0}\sinh\left(\frac{q\Phi(\vec{r})}{k_BT}\right) \tag{3.9}$$

This non-linear equation is exactly solvable in only a few special cases but it is possible to linearise this equation under certain assumptions. If we assume that the thermal energy, k_BT, is large when compared to the charge-field interaction, we can approximate sinh with a Taylor expansion giving

$$\nabla^2\Phi(\vec{r}) = -\frac{2\bar{n}q}{\varepsilon_r\varepsilon_0}\left(\frac{q\Phi(\vec{r})}{k_BT}\right) = \kappa^2\Phi(\vec{r}) \tag{3.10}$$

where

$$\kappa = \sqrt{\frac{2\bar{n}q^2}{\varepsilon_r\varepsilon_0 k_BT}}. \tag{3.11}$$

κ has units of m^{-1} and so $1/\kappa$ defines a length scale. This is the Debye length and in an electrolyte solution it defines the length beyond which charges are screened out. Therefore it defines the range of screened Coulomb interactions and allows us to use a potential which decays over a range comparable to κ^{-1}. In order to find this potential it is necessary to solve

a modified version of Poisson's equation known as the *screened Poisson equation*

$$[\nabla^2 - \kappa^2]\Phi(\vec{r}) = -\frac{q}{\varepsilon_0 \varepsilon_r}\delta_r \tag{3.12}$$

which leads to a solution of

$$\Phi(\vec{r}) = \frac{q^2}{4\pi\varepsilon_0\varepsilon_r \vec{r}}\exp(-\kappa r) \tag{3.13}$$

This has the same form as the Yukawa potential which was derived with a quantum mechanical analogue of the Debye-Hückel theorem.

The ability to truncate the long-range interactions in this way is very useful for simulation, but is obviously only applicable in ionic systems. Many interesting systems do not comply with the requirements for the application of th Debye-Hückel theory and the long-range interactions must be dealt with more fully. While there are a number of ways to deal with this (such as particle-particle/particle mesh Ewald [83], radially averaged Ewald [84], and fast multipole [85]), the most common are all based on the Ewald summation [77].

It is not immediately obvious as to why the calculation of the long-range interaction is so problematic. A naïve approach would be to simply treat the Coulomb interaction in a similar way to the van der Waals interactions and truncate at a suitable distance. For finite systems, such simple truncation methods fulfil the requirement of reducing the computational effort and indeed render the calculation possible given computational capabilities. However, it has been shown that there are significant problems with the simple truncation of the Coulomb interaction which leads to pathological artefacts in the simulations [86].

Returning to the infinite or periodic systems, the Coulomb potential could be written as

$$\Phi_{\text{Coul}} = \frac{1}{2}\sum_{i}^{N}\sum_{j,\vec{n}}{}' \frac{q_i q_j}{|\vec{r}_{ij} + \vec{n}L|} \tag{3.14}$$

where the second sum is across all particles j and all periodic images $\vec{n}$ except where $j = i$ if $\vec{n} = 0$, denoted by the prime. Following the example of Frenkel and Smit [77], Gaussian units are used to simplify the notation. This sum is only conditionally convergent and depends upon the order in which the sum is carried out which makes it unsuitable for use in calculating system properties. In order to calculate these long-range interactions we must somehow modify this in order ensure convergence.

The Ewald sum approach firstly makes use of a similar approach to the Debye-Hückel theory. As above, the charged particle is surrounded by a cloud of charged particles with a given distribution (often Gaussian) of equal magnitude and opposite sign. In a similar way to the Debye-Hückel theory, this truncates the interaction to a convergent short-range interaction in real space with the imposed charge distribution screening the real particles from longer range charges. Having artificially imposed a distribution of particles it is necessary to remove the influence of these imposed charges. This is done by adding another distribution, of the same type as previously chosen, around the real particles of the same sign and magnitude. This, in effect subtracts off the effect of the shielding around the real particles. The addition of this distribution also has the affect of making the function a smooth periodic function which is amenable to solution by Fourier transform. It is this function that makes up the reciprocal space part of the Ewald sum. Finally, there remains a term between the real particle and the added charge distribution which must be explicitly removed.

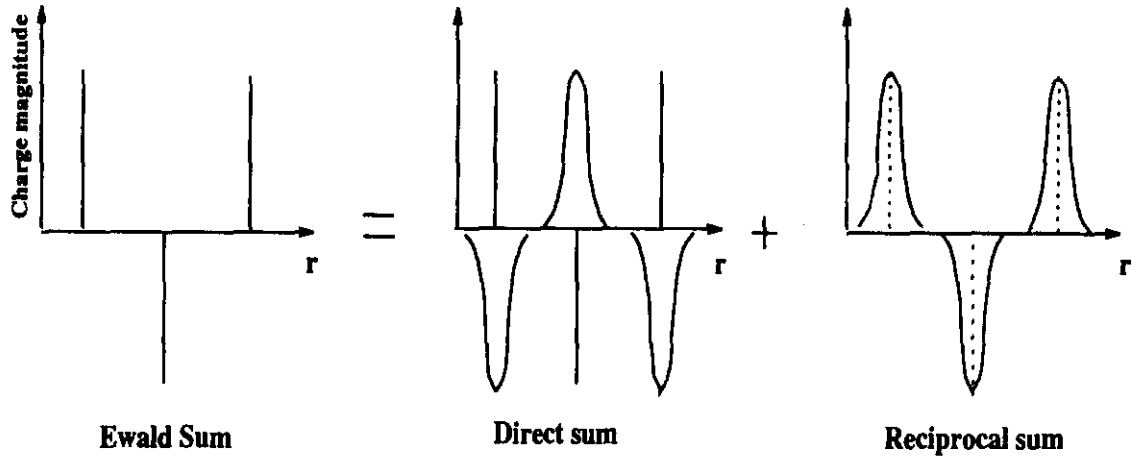


Figure 3.3: Components making up the Ewald sum for a 1D system of point charges. Gaussians are added around the point charges for the real and reciprocal space sums. Taken from [87]

This gives a final sum, with the added distributions as Gaussians, which looks like

$$\Phi_{\text{Tot}} = \Phi_{\text{Real}} + \Phi_{\text{Recip}} + \Phi_{\text{Self}} \tag{3.15}$$

with the real space term given by

$$\Phi_{\text{Real}} = \frac{1}{2}\sum_{i\neq j}^{N} \frac{q_i q_j \text{erfc}(\sqrt{\alpha} r_{ij})}{r_{ij}} \tag{3.16}$$

the reciprocal space sum by

$$\Phi_{\text{Recip}} = \frac{1}{2V}\sum_{k\neq 0} \frac{4\pi}{k^2} |\sum_{i=1}^{N} q_i \exp(i\vec{k}\cdot\vec{r}_i)|^2 \exp\left(\frac{-k^2}{4\alpha}\right) \tag{3.17}$$

and finally, the self interaction term by

$$\Phi_{\text{self}} = -\left(\frac{\alpha}{\pi}\right)^{1/2} \sum_{i=1}^{N} q_i^2 \tag{3.18}$$

where erfc is the complementary error function given by $\text{erfc}(x) = 1 - (2/\sqrt{\pi})\int_0^x \exp(-u^2)du$. There are only two parameters necessary in this approach. The first, α, describes the width of the added Gaussian distributions and affects the speed of convergence for the real-space sum. The second parameter is the number of k vectors or reciprocal-space vectors to be chosen.

There has been wide discussion over the appropriate choices of parameters for the Ewald sum. Larger values of α increase the speed of the real-space sum but there is an impact in the efficiency of the reciprocal-space sum. With higher values of α the reciprocal space sum will take longer to converge so there is a balance to be struck. Choosing $\alpha = \sqrt{\pi}/L$ where L is the periodic box length will cause the real and reciprocal space sums to converge at the same rate [88] which may seem like the obvious choice. However, techniques have been developed which can be used to accelerate the reciprocal-space sum meaning it may be beneficial to bias the

convergence towards the real space sum.

It has been suggested that there are several main considerations for the choice of parameters for the Ewald sum [87]:

1. System size is an important factor as for large systems ($N > 10^4$) even the minimum image convention becomes expensive in computational terms. In practice, the reciprocal sum is faster and so a large value for α is chosen along with a cutoff to speed up the real space sum.

2. Accuracy. At times, it may be more important to ensure high accuracy which would require more reciprocal vectors or increasing the range of the real-space sum. This would come at a cost of efficiency.

3. CPU time. The real-space sum is more costly in terms of resources and so if limited resources are available, it may be advantageous to limit the real space sum more severely. This will come at a cost in accuracy so must be considered carefully.

4. Real-space sum cut off. Assuming a cut off is in place in order to limit the real-space sum cost, α must be chosen appropriately to ensure swift convergence. This means that for a small cut off, a larger choice of α must be chosen.

In practice there is usually some relationship between box length and α and the bias in the choice of parameters leans towards a more efficent real-space sum as this is where the larger part of the computational cost resides.

The Ewald sum can also be applied to dipolar systems. In order to apply this to dipolar systems it is simply necessary to replace q_i with $\mu_i \cdot \nabla$ where μ_i is the dipole moment of the i^{th} particle. Substituting this in place we find a real-space sum of

$$\Phi_{\text{Real}} = \frac{1}{2}\sum_{i\neq 1}(\vec{\mu}_i \cdot \vec{\mu}_j)B(\vec{r}_{ij}) - (\vec{\mu}_i \cdot \vec{r}_{ij})(\vec{\mu}_j \cdot \vec{r}_{ij})C(\vec{r}_{ij}) \tag{3.19}$$

where $\vec{r}_{ij}$ is the vector between the i^{th} and j^{th} particle and we define

$$B(r) = \frac{\text{erfc}(\sqrt{\alpha}r)}{r^3} + 2\left(\frac{\alpha}{\pi}\right)^{\frac{1}{2}}\frac{\exp(-\alpha r^2)}{r^2} \tag{3.20}$$

$$C(r) = 3\frac{\text{erfc}(\sqrt{\alpha}r)}{r^5} + 2\left(\frac{\alpha}{\pi}\right)^{\frac{1}{2}}(2\alpha + \frac{3}{r^2})\frac{\exp(-\alpha r^2)}{r^2}. \tag{3.21}$$

$$\tag{3.22}$$

The reciprocal-space sum is

$$\Phi_{\text{recip}} = \frac{1}{2V}\sum_{k\neq 0}\frac{4\pi}{k^2}\left|\sum_{i=1}^{N} i\vec{\mu}_i \cdot \vec{k}\exp(i\vec{k}\cdot\vec{r}_i)\right|^2 \exp\left(\frac{-k^2}{4\alpha}\right) \tag{3.23}$$

and the self interaction term is

$$\Phi_{\text{self}} = -\frac{2\pi}{3}\left(\frac{\alpha}{\pi}\right)^{\frac{3}{2}}\sum_{i=1}^{N}\mu_i^2. \tag{3.24}$$

Before leaving the subject of the Ewald sum, it is necessary to at least comment on boundary conditions. It may seem strange to consider boundary conditions in an infinite periodic

system but it is necessary to think about certain physical phenomena of charged and dipolar systems. In the real world, any material or substance under study is not infinite. It also has a boundary to a medium which very likely has very different dielectric properties, such as free space. This boundary has a strong effect on the behaviour of materials and can completely change some responses, a case of which is shown briefly in chapter 4. In some theoretical work on the behaviour of dipolar materials this whole effect is sidestepped by assuming the shape of the material is long and thin which reduces the this effect, caused by what is known as a demagnetising field due to the reducing of the alignment of the dipoles of a material it causes (otherwise stated as reducing its magnetisation). The demagnetising field is caused by the changes in magnetic properties when moving from the magnetic material to the surrounding medium. As the magnetic field must be continuous across this boundary there is an effect at the edge of the magnetic material.

An overly simplistic but sometimes useful schematic of how this shape would reduce the effect is shown in fig 3.4. In this picture, dipolar domains are formed in both materials and around the centre line up with an external field. As these domains approach the extremities of the material the effects of the pole of the material as a whole are felt, which in the case of the spheroid are felt laterally. In the long thin sample these effects are reduced as the dipolar domains are very close to being nose to tail. A similar useful fiction popular in engineering states that for a given volume of material the lower the aspect ratio the closer the poles are so there will be a larger demagnetising effect. As with the schematic, it is not suggested this causes the demagnetising field but is rather a simple way to picture what the effect may be.

With respect to the Ewald sum, these boundary effects would correspond to the $k = 0$ term which has been excluded from the equations above. In practice, for dipolar systems it is possible to ignore this term if we assume that our material is embedded in a material of infinite dielectric constant. This is known as *conducting boundary conditions* or more colloquially as *tin foil* boundary conditions, and has been shown to best reproduce the thermodynamic limit in simulations of dipolar systems [89]. In most simulations of dipolar systems, it is in fact necessary to use these conducting boundary conditions as the small size of the simulations leads to artefacts which render the simulation useless. In order to capture the structural physical effects of a demagnetising field, it would be necessary to simulate a material at an almost macroscopic scale containing a huge number of particles. This would allow the formation of multiple domains and their influence to be correctly sampled. At present, simulations of such an enormous scale are beyond current technical ability and promise to remain so for some time.

The scale of simulations necessary to remove artefacts raises an important point. Simulating such a small piece of a material can introduce many more artefacts than just those mentioned above. Too small a simulation, either in terms of length scale of our periodic box, number of particles included in the simulation or even time[1] period covered by the simulation can introduce unrealistic correlations and unphysical results. A good first step to ensuring that there are no spatial finite-size effects is to make sure that the simulation box is more than twice the length of short-ranged interactions, such as the LJ interaction. This obviously imposes a number of particles to fill the box to the required density of the system of interest but is no guarantee that a simulation is large enough. The only way to be sure finite size effects are not causing a problem is to run some test simulations of much larger size and check for consistent results.

The possibility of finite size effects implies 'bigger is better' and indeed the ideal would be

[1] "time" here refers to either wall clock time or number of steps in the simulation

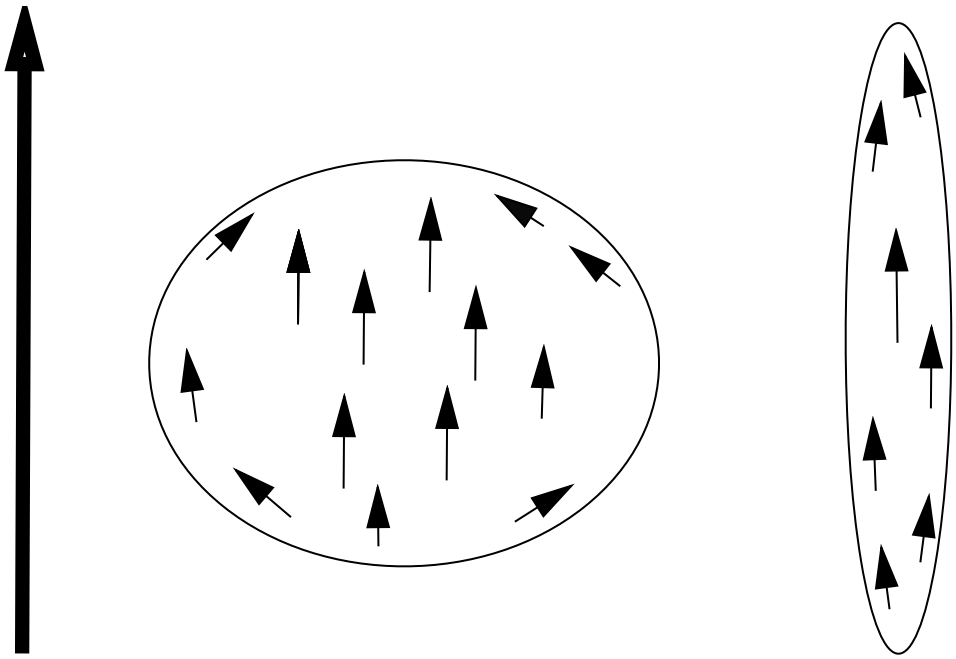


Figure 3.4: Schematic for a phenomenological description of the demagnetising field. The dipolar domains on the long thin sample are much more uniformly nose to tail. The oblate spheroid shows a much larger difference in the directions of the dipolar domains. The difference between the 'bulk' magnetisation and the 'surface' magnetisation is the demagnetising field. The large arrow is the direction of an external field.

to simulate enough particles to represent a macroscopic sample. This is never possible and in some cases even simulating a system large enough to generate useable data can be difficult when using an atomistic approach. This is particularly true in the case of a complex fluid where every solvent particle is a hard to simulate molecule. In this case there is sometimes an alternative approach of *coarse graining*. In this approach, in order to reduce the complexity of the system and the number of particles necessary to emulate the system, molecules, or even groups of molecules are bundled together into a single particle and interact with the other particles in the system through a composite potential which approximates the actual inter-molecular interaction. This approach must be used with care, as there is an inherent loss of information in this grouping of molecules together. When carrying out this grouping, it is important to consider both the chemistry and physics of the system and ensure that no important interactions are being missed. In general, if the system is chemically active, in that electrons are being exchanged between molecules or there is hydrogen bonding involved, and these effects play an important role, it is likely that a coarse grained approach will not be sufficient to capture the phenomena of interest. In fact, the question of whether a particular system would be amenable to a coarse grained approach is not a simple one. It is impossible to state a hard rule that can be applied and there must be careful consideration of what questions can be answered with this method.

The coarse graining of a system throws up an obvious question: How are the composite potentials arrived at, or parameters for potentials like LJ arrived at? This is the part of coarse graining which can cause a certain amount of controversy and dismay amongst parts of the scientific community not familiar with computational approaches. Combining molecules and assigning them what can appear like arbitrary values for inter-particle interactions is open to the charge of manipulation. Indeed, when moving farther away from the actual physical system and imposing the approximations necessary for coarse graining it would be easy to lose contact with reality. In order to avoid this, parameters for the composite potentials are either empirically derived, found from some analytic theory or found via simulations at a much higher level of theory, in some cases utilising DFT or other quantum approaches. In addition to this, it is important to validate simulations and parameters chosen against known and experimentally verifiable effects. More on what kind of system properties can be tested will be covered in section 3.4. Even with this careful validation, a coarse grained approach must be used with some care, with the initial analysis of the system all important. If the coarse grained system misses an interaction which proves to be important in its behaviour then no amount of parameterising will give rise to physically relevant results.

The final general principle in the finite-size simulations we will look at is the implementation of reduced units. In most coarse-grained simulations, the results will be in dimensionless form. For each initial system setup, a number of fundamental scales are chosen, such as a basic unit of length or energy, and all values will be normalised with respect to these values. For instance, it is common for a fundamental length scale to be chosen based on particle size, σ. Considering the number density given by $\rho = N/V$ where N is the number of particles and V is the volume of the box, the reduced density is given by

$$\rho^* = \frac{N\sigma^3}{V} \tag{3.25}$$

This process can be repeated for all parameters of a simulation leading to a set of dimensionless numbers. At first, it may not appear completely obvious as to why such a process would be applied. However, by using dimensionless units, a single simulation can be used to calculate properties for a wide range of parameters. A single set of results in their dimensionless form, can be translated back to correspond to different real systems by using different basic units.

From here, we will go on to explore the fundamentals of the two main simulation methods in use today, starting with Monte Carlo simulations.

3.2 Monte Carlo Simulations

MC simulations are named for the Monte-Carlo Casino in Monaco. It was named after the casino because it was one of the first approaches to take a completely stochastic approach to simulation, with the basic idea being to do something many times subject to random numbers and then average the outcome. The algorithm more widely used to day has become known as the Metropolis Monte-Carlo algorithm [38]. This approach, at its base, is designed to generate a required probability distribution. This can easily be seen to be directly useful for statistical mechanics and calculating properties of systems that obey and adhere to the probability distributions for the ensembles defined by statistical mechanics.

The general routine for a MC simulation of a system of particles is as follows.

1. Select a particle.
2. Select a random displacement within certain constraints (such as maximum distance) and move the particle to a new position
3. Calculate the energy of the new configuration
4. Accept or reject the new configuration according to conditions relevant to the distribution or ensemble being sampled.

This obviously skips a huge amount of detail involving the setup and consideration of acceptance criteria, some of which will be explored below, but concisely states the procedure that is followed with MC simulation. The reasons for using such an approach rather than the more intuitive MD simulations remains the same now as when it was first proposed: either the system studied cannot be solved deterministically or it is impossible to even write down a closed system of equations for the system being studied. As methods utilising MC simulations have become more sophisticated, additional applications for which MC is suitable have been identified, for example it is possible to explore rare events using MC which would require prohibitively long MD simulations to gather data on. Furthermore the use of MC extends beyond the study of purely physical systems and the general approach has been used in finance for modelling the stochastic behaviour of markets since the late 70s [90] .

The basic algorithm of MC has a large amount of flexibility with the kinds of trial moves being made and the acceptance criteria applied. In molecular simulation, the acceptance criteria relies largely on the well known Boltzmann factor given by

$$\exp\left(-\beta U_{\mathrm{i}}\right) = \exp\left(-\frac{U_{\mathrm{i}}}{k_{\mathrm{B}}T}\right) \tag{3.26}$$

where k_{B} is the Boltzmann constant, T is the temperature, and U_{i} is the potential energy of the system in state i and depends on the ensemble that is being considered. It is these acceptance criteria which form what is perhaps the most important condition in MC simulation. The naïve approach of simply making a whole series of completely random moves is unproductive as the vast majority of moves will result in a physically implausible configuration, such as one with particle overlaps. In order to avoid these situations and to make sure the sampling explores the most likely states of the system, a form of importance sampling is used. This is enforced by ensuring *detailed balance* is observed.

If the probability of a system being in some old configuration o at some time τ is $p(o,\tau)$, and we denote a probability flux from state o to some 'new' state n in a given 'time step' as $(\tau \to \tau+1)$ as $J(o \to n)$, the probability of being in state n at time $\tau+1$ is given by

$$p(o,\tau+1) = p(o,\tau) + \sum_{n}[J(n \to o) - J(o \to n)]. \tag{3.27}$$

Assuming that the system follows Boltzmann statistics we can say that at equilibrium, $p(o,\tau+1) = p(o,\tau)$, because the Boltzmann distribution is stationary. A sufficient but not necessary detailed balance condition is that

$$\sum_{n}[J(n \to o) - J(o \to n)] = 0 \quad \forall\, n \tag{3.28}$$

$$\Rightarrow J(o \to n) = J(n \to o). \tag{3.29}$$

$J(o \to n)$ can be factored in to three terms:

$$\begin{array}{ll} p(o) & \text{The probability of being in state} \\ a(o \to n) & \text{The probability of generating the trial move from } o \text{ to } n \\ acc(o \to n) & \text{The probability of accepting the trial move from } o \text{ to } n \end{array} \tag{3.30}$$

From this, we can write

$$\begin{array}{rcl} p(o)a(o \to n)acc(o \to n) & = & p(n)a(n \to o)acc(n \to o) \\ \Rightarrow \quad \dfrac{acc(o \to n)}{acc(n \to o)} & = & \dfrac{p(n)a(n \to o)}{p(o)a(o \to n)} \\ \Rightarrow \quad \dfrac{acc(o \to n)}{acc(n \to o)} & = & \dfrac{a(n \to o)}{a(o \to n)} \exp\left(-\beta[U(n) - U(o)]\right) \end{array} \tag{3.31}$$

in the canonical ensemble. It is important to note that in calculating $p(n)/p(o)$ we remove the need to calculate the partition function. This is a large advantage in MC as in general it is impossible to calculate the partition function. In order to ensure that these constraints are adhered to heavy use is made of random numbers. The Metropolis MC algorithm is as follows.

1. Pick a particle at random
2. Alter x, y, and z coordinates by displacements chosen randomly from the interval $-\Delta r_{\max} \leq \Delta r \leq \Delta r_{\max}$
3. Calculate the change in potential energy ΔU
4. Accept the move with the probability $acc = \min(1, \exp(-\beta \Delta \mathrm{U})$

   If $\Delta U < 0$ accept

   If $\Delta U > 0$ generate a random number $0 \leq \Re \leq 1$

   If $\Re \leq \exp(-\beta \Delta U)$ accept

   If $\Re > \exp(-\beta \Delta U)$ reject and retain old configuration

It turns out that detailed balance is a sufficient but not necessary condition for the MC scheme to be valid [91]. For instance, instead of choosing a particle at random it is possible to step through the particles sequentially. This violates detailed balance because $a(o \to n) \neq a(n \to o)$ and yet has no impact on the results of an MC simulation and can in fact be used to improve computational performance. Despite this, when designing a new MC algorithm it is safest to ensure that detailed balance is observed as this will make it certain that the algorithm is valid.

MC simulations are inherently serial. They require large sets of psuedo-random numbers to be generated and, in general, this is the only part of the MC process amenable to parallelisation in the traditional sense of a single simulation being spread over a distributed resource. In order to accelerate MC it is necessary to take a slightly different view point. The aim is to generate a large number of configurations from which to calculate system properties and with good statistics. Rather than attempting to accelerate an individual simulation it may be quicker to use clusters to run multiple simulations from a single configuration. The use of either differently seeded random number generators or simply generating the random numbers necessary for all simulations from a single source will ensure that a different MC trajectory is generated for each simulation.

With respect to parrallelising an MC simulation on GPUs in this way, their probabilistic nature brings another problem. As stated previously, GPUs are designed for vast numbers of threads operating on the same instruction at the same time. Conditional branching is anathema to this kind of computation as it inevitably leads to a loss of speed; multiple MC simulations will automatically cause a large amount of conditional branching, killing any potential gains.

The kind of parallelism suited to MC simulations is incompatible with GPUs and hence it requires a set of processors capable of running entire MC simulations. There are cases where GPUs have been used to accelerate MC simulations of financial markets but this has largely been limited to the generation of the vast amount of random numbers which are necessary for such simulations and, due to their confidential nature, details are scarce.

We will now move on to look at the most common deterministic simulation approach in use today: molecular dynamics.

3.3 Molecular Dynamics

MD simulations were developed for classical mechanics. There are approaches which make attempts to include quantum mechanical effects but these are rarer and hugely complex and will not be discussed here. The fundamental difference between MC and MD relates to their statistical approaches. MC takes an ensemble average from a number of statistically uncorrelated snapshots whereas MD relies on a time average over a trajectory. The MD approach relies on the *ergodic hypothesis* which can be paraphrased as follows: a systems path through phase space on a surface of constant total energy will, if considered over a long enough time, traverse all of that space. The corollary that proves useful for MD states: the time spent in any particular micro-state is proportional to its volume in phase space. It is this corollary that implies

$$\langle A \rangle_{\text{ensemble}} = \langle A \rangle_{\text{time}} \tag{3.32}$$

where A is some observable of the system and the angled brackets imply averages. This equality of ensemble averages and time averages embodies the equivalence of the approaches of MD and MC.

Having laid out the basis for assuming the validity of a time average, the approach MD takes will now be explored. The equations of motion for MD are very simple as they are just an application of Newtons' 2$^{\text{nd}}$ law

$$\vec{F}_i = \frac{d\vec{p}_i}{dt} = m_i \frac{d^2\vec{r}_i}{dt^2} \tag{3.33}$$

where i denotes the i^{th} particle, $\vec{p}$ the momentum, and $\vec{r}$ the position. The equations of motion are deterministic in nature and all that is needed to solve them is an initial distribution of particle positions, an initial distribution of particle velocities, and a potential describing the interactions between them. From this initial condition, a timestep, δt, is chosen dependent on computational limitations and certain system factors, and the algorithm of MD then proceeds to calculate the forces on each particle which directly leads to the acceleration. The new positions and velocities are then calculated and the process starts all over again for the next time step.

Despite the simplicity of this approach outlined above, there are complications. The equations of motion involving so many particles have no analytic solution and so must be solved numerically. This can be an involved process depending on the potentials involved in the system

but in general any symplectic integrator can be used with one of the most common being the *velocity Verlet* (VV) algorithm. To find the VV algorithm the positions are Taylor expanded both forwards and backwards and then some straightforward manipulation of these expansions leads to

$$\vec{r}_i(t+\delta t) = \vec{r}_i(t) + \vec{v}_i(t)\delta t + \frac{\vec{F}_i(t)}{2m_i}\delta t^2 \tag{3.34}$$

$$\vec{v}_i(t+\delta t) = \vec{v}_i(t) + \frac{\vec{F}_i(t+\delta t) + \vec{F}_i(t)}{2m}\delta t. \tag{3.35}$$

From the position of having an initial set of parameters for a system, a typical routine using these equations is as follows:

1. Compute the forces on each particle in the system according to the relevant potential.
2. Update the positions according to eq. 3.34 to give positions for the next time step.
3. Update the velocities with the forces for the current time-step.
4. Compute the forces for the new time step with the new positions from step 2.
5. Finally, finish updating the velocity with the new forces.

The main advantage this algorithm has over alternatives is in the balance between simplicity of implementation and memory requirements. Other algorithms have larger memory requirements arising from the storage of velocities from previous time steps, which also comes with a more complicated implementation.

The choice of the time step, δt, has a large effect on the behaviour of this algorithm. Its value is dependent on the system being studied, specifically the timescale or period of the fastest process being simulated. If the time step is larger than the timescale of the fastest process, the integration scheme will become numerically unstable. From a physical point of view, this is easy to understand. Considering a very simple system of two LJ spheres colliding with each other, the repulsion between them increases steeply once they pass their equilibrium positions. With too large a time step, this repulsion is not enough to keep the spheres apart, as in the space of one time step the particles "jump" a distance dependent on δt and the velocity. The result of this jump being too large would be the spheres occupying the same physical space, which is physically impossible, and at which point the simulation would break down due to infinities appearing in the force calculation. Therefore, it is important that the value of δt is chosen with the fastest processes of the system in mind.

As in MC, it is possible to sample any ensemble using an MD approach. However, for MD extra effort is necessary to access the different ensembles. For instance, in isothermal-isobaric (NPT) MC simulations, the temperature and pressure appear as extra terms in the Boltzmann factor and influence the acceptance of trial moves. In MD it is necessary to implement thermostats and barostats in order to sample the NVT and NPT ensembles. This is much more complicated than simply applying extra terms to a Boltzmann factor, and the method chosen can have an impact on the kind of systems which can be simulated.

Thermostats can be split into three categories: velocity scaling, stochastic dynamics (SD), and extended system (ES) methods. SD and ES thermostat types attempt to recreate the canonical ensemble and both have limitations. SD methods cannot recreate system dynamics accurately as they use random perturbations in order to maintain temperature. This makes

the SD methods unsuitable for the calculation of transport coefficients such as diffusion. ES methods utilise fictional extra degrees of freedom in order to remove or supply energy to the system as required to maintain a given average temperature. In general, they are much gentler on the system being studied but are much more complicated to implement.

Velocity scaling is as simple as it sounds. At each time step the velocities of the system are scaled in order for the system to meet a given temperature. The most often used form of this is the Berendsen thermostat [92]. This is really as simple as modifying Newton's equation so that there is an additional term to rescale the velocities, giving

$$m_i \frac{d^2\vec{r}_i}{dt^2} = \vec{F}_i + m_i\gamma\left(\frac{T_0}{T} - 1\right)\vec{v}_i \tag{3.36}$$

where T_0 is the target temperature and γ is a coupling constant chosen empirically. This thermostat has a number of drawbacks, the most critical being the inability to correctly sample the canonical ensemble because fluctuations are dumped too rapidly. Despite this terminal defect, the Berendsen thermostat has found a use in modern simulations. It is very useful during an equilibration stage of a simulation as it very quickly brings a system to the target temperature. Generally, thermostatic control of the system will be handed over to a thermostat which generates the correct ensemble following the equilibration.

SD methods for thermal control also make use of additional components in Newton's equations. In this case, those extra components take the form of a friction coefficient, γ, and an extra term representing an additional random force, R. This is expressed in the Langevin equation [93]:

$$\vec{F}_i - \gamma_i\vec{v}_i + \vec{R}_i(\delta t) = m_i\vec{a}_i. \tag{3.37}$$

When using this approach to model a system, the effects of any solvent are implicit and the friction coefficient and random force represent the drag and thermal jostling from the solvent respectively. Temperature control is achieved by the balance of these two extra terms with the friction force reducing particle velocities, and the stochastic force $\vec{R}_i$ representing the Brownian jostling of the solvent. These stochastic forces have the following properties:

1. They are uncorrelated with the velocities and systematic forces.
2. Their time-averages are 0
3. $\langle R_i(t), R_j(t')\rangle = 2m_i\gamma_i k_B T_0\delta(t'-t)$ where T_0 is the target temperature (This relationship ensures the desired temperature is observed).

Systems thermalised in this way are describing what is known as Langevin dynamics. A similar system is used for what is known as Brownian dynamics. If we assume that the acceleration term from eq. 3.37 is negligible we can say

$$\vec{F}_i - \gamma_i\vec{v}_i + \vec{R}_i(\delta t) = 0 \tag{3.38}$$

leaving the only inputs onto the particles as the random and inter-particle forces. This is analogous to a viscous system where inertia plays a very small role.

The approaches outlined above both suffer from the same drawback in that momentum is not conserved. This limits the systems that can be studied with this form of thermostatic control because hydrodynamic effects arising from momentum conservation are eliminated.

The final approach to maintaining temperature in a model system is the ES approach. In this approach the equations of motion for the model system are altered by the addition of an extra degree of freedom. This extra degree of freedom allows coupling to a heat bath which can be thought of as a massive pseudo-particle. This particle can then be used to either remove or add energy to the system through a loose coupling without perturbing the dynamics of the system of interest to a high degree. This approach was first introduced by Nosé [94] and was simplified by Hoover [95] and the most common implementation of ES thermostats has become known as the Nosé-Hoover thermostat (NHT).

The NHT is unusual in that it is not derived from Hamiltonian mechanics and this unconventional approach leads to a limitation with the simple form of the thermostat. As with the Berendsen thermostat, there are concerns about the ensemble sampled by the NHT. For systems which contain one conserved quantity, such as those with an external force on them removing conservation of momentum, the NHT conserves a total energy given by a modified Hamiltonian. In cases where there are more conservation laws in play, the NHT can still be used as long as the centre of mass of the system remains fixed. Where these constraints are not met, then an extension of the NHT must be used, known as Nosé-Hoover chains (NHC).

The idea behind NHC chains is a relatively simple one. The idea is to make a chain of thermostats where the first thermostat is thermostatically controlled by a further thermostat and so on down the chain for an arbitrary number of thermostats. It has been shown that the use of NHC allows the correct ensemble to be recovered. Simulations of more complex systems, where angular degrees of freedom play a role in conservation laws, can still recover the correct ensemble by rejecting 'global thermostats' and utilising a number of NHC, with the extreme cases utilising one chain per molecular degree of freedom [96].

The equations of motion as formulated by Martyna *et al.* [97] for a system driven by Nosé-Hoover chains (utilised in some of the work outlined in later sections) are given by

$$\begin{aligned}
\dot{\vec{r}}_i &= \frac{\vec{p}_i}{m_i} \\
\dot{\vec{p}}_i &= \vec{F}_i - \vec{p}_i \frac{p_{\xi_i}}{Q_i} \\
\dot{\xi}_i &= \frac{p_{\xi_i}}{Q_i} \\
\dot{p}_{\xi_1} &= \left[\sum_{i=1}^{N} \frac{\vec{p}_i^2}{m_i} - N_f k_B T \right] - p_{\xi_1} \frac{p_{\xi_2}}{Q_2} \\
\dot{p}_{\xi_j} &= \left[\frac{p_{\xi_{j-1}}^2}{Q_{j-1}} - k_B T \right] - p_{\xi_j} \frac{p_{\xi_{j+1}}}{Q_{j+1}} \\
\dot{p}_{\xi_M} &= \frac{p_{\xi_{M-1}}^2}{Q_{M-1}} - k_B T
\end{aligned} \tag{3.39}$$

In this system there are M thermostats ξ_j with masses Q_j and momenta p_ξ. N_f is the number of degrees of freedom. Such a large change to the equations of motion again has an effect on the velocity Verlet scheme. On their initial introduction, it was necessary to solve these equations with an iterative approach introducing complexity and computational overhead, but later work by Tuckerman *et al.* [98] gave an explicit integration routine utilising the velocity Verlet scheme for NHC. This is the approach used in the work outlined in later sections.

3.4 Observables

While all of the techniques and technology outlined above and in previous sections is very impressive and interesting, on its own it remains useless. Unless the simulation codes can be verified and their output tested against real-world experiments, they are little more than interesting toys. It is interesting to note that while experimental techniques have reached phenomenal levels of sophistication, the basic methods which can be used to probe the structure of soft matter are fundamentally quite limited.

The most obvious method for extracting information from a simulation is simply to look at it. Although seemingly too simple to be of any use, viewing the results of simulation and choosing simple metrics (such as film depth for thin films or cluster size for polymer aggregation) can be useful. Often just viewing a movie of a simulation can yield direct insight on experimental measurements taken, for instance, with atomic force microscopy and other imaging technologies. For many systems, the point of carrying out simulations is simply to gain some insight into what is inherently impossible to view in an experiment. The choice of metrics is system dependent and entirely chosen depending on the questions the simulations are intended to address, but it is worth mentioning here that often what appear to be the simplest of measures can be the most useful.

Scattering experiments are commonly used for probing the structure of soft matter systems. These experiments produce information in two related forms: radial distribution functions (RDF) or structure factors (SF). These two measures are related to each other and, from the point of view of simulation, have the advantage of being retrievable from simulation data. The RDF is simple to understand. It is a measure of the probability of finding a particle in a shell of a given volume, a given distance from any other particle. Figure 3.5 gives a schematic view of the calculation of the RDF while mathematically it can be expressed as

$$g(r) = \frac{n(r)}{\rho 4\pi r^2 \Delta r} \tag{3.40}$$

where $n(r)$ is the number of particles found in the shell of thickness Δr at distance r from the central particle. From this calculation, the RDF is plotted as a function of particle separation, r. At short separations, less than the particle diameter (assuming particles of one size), the RDF is zero due to particle overlaps being prohibited by the short range repulsive interactions. In the RDF of a gas, there is one peak seen at around the effective particle width which quickly reduces to one, indicating a loss of correlations. For both a fluid and solid there are a number of peaks as seen in ref. [99]. For the fluid, these peaks are broad and reduce in magnitude as r increases. The width of the peaks is indicative of thermal fluctuations due to the higher temperatures of fluids. For the solid, the peaks are sharply defined indicating the localisation of particles at crystal lattice points, and small thermal fluctuations. The RDF is very easily calculated in a simulation although it is not the natural result of experiment. For direct comparison to experiment it is often more useful to calculate the closely related SF.

The SF is defined as

$$S(\vec{k}) = \frac{1}{N} \left\langle \rho_{\vec{k}} \rho_{-\vec{k}} \right\rangle \tag{3.41}$$

where $\vec{k}$ is a wave-vector given by $\vec{k} = \frac{2\pi}{L}(n_x, n_y, n_z)$ and $\rho_{\vec{k}}$ is the Fourier transform of the

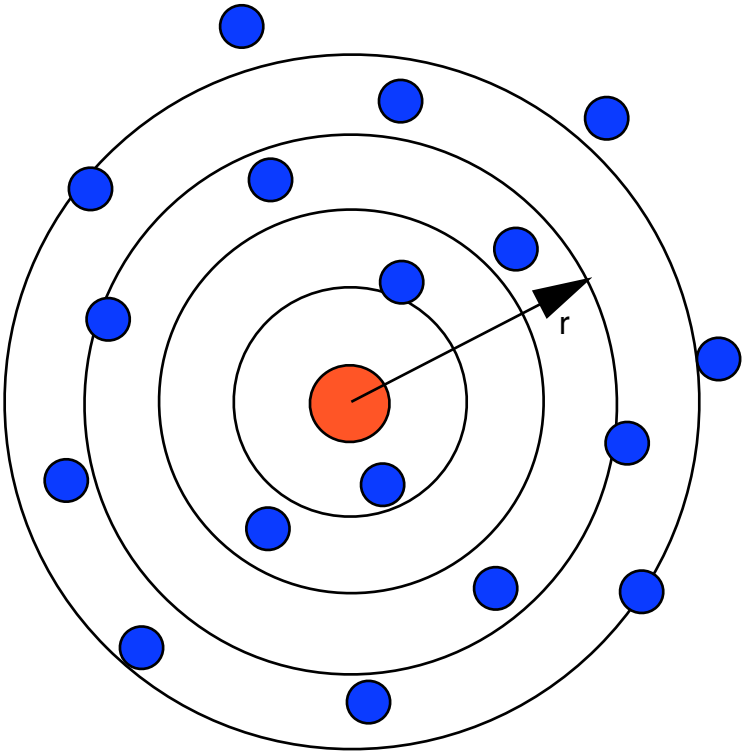


Figure 3.5: To calculate an RDF, centre a series of concentric rings on a particle so that they form shells a fixed distance apart, Δr. Over a series of snapshots the average number of particles in each shell is calculated. This is then divided by the volume of each shell and the average density of atoms in the system.

instantaneous density, $\rho(\vec{r})$. This becomes

$$S(\vec{k}) = \frac{1}{N}\left\langle \sum_{i}^{N}\sum_{j}^{N} \exp(-i\vec{k}\cdot\vec{r}_i)\exp(i\vec{k}\cdot\vec{r}_j)\right\rangle . \tag{3.42}$$

This can be shown to be equivalent to the Fourier transform of the RDF given by

$$S(\vec{k}) = 1 + \rho\int_V d\vec{r}\exp(-i\vec{k}\cdot\vec{r})[g(r)-1]. \tag{3.43}$$

In the case of a fluid, where the system can be considered isotropic, this becomes a function of the scalar value of the magnitude of the wave-vector

$$S(k) = 1 + \rho\int_V dr\exp(-ikr)[g(r)-1] \tag{3.44}$$

where $g(r)$ has become $g(r)-1$ to remove a contribution given by a fluctuation at small r from a 'self-interaction'. While these two ways of calculating the structure factor, direct and via the Fourier transform of the RDF, are in theory exactly equivalent, in practice there will be differences between the values calculated with these methods arising from well known truncation errors associated with the finite size of the system. These truncation errors can cause peaks to

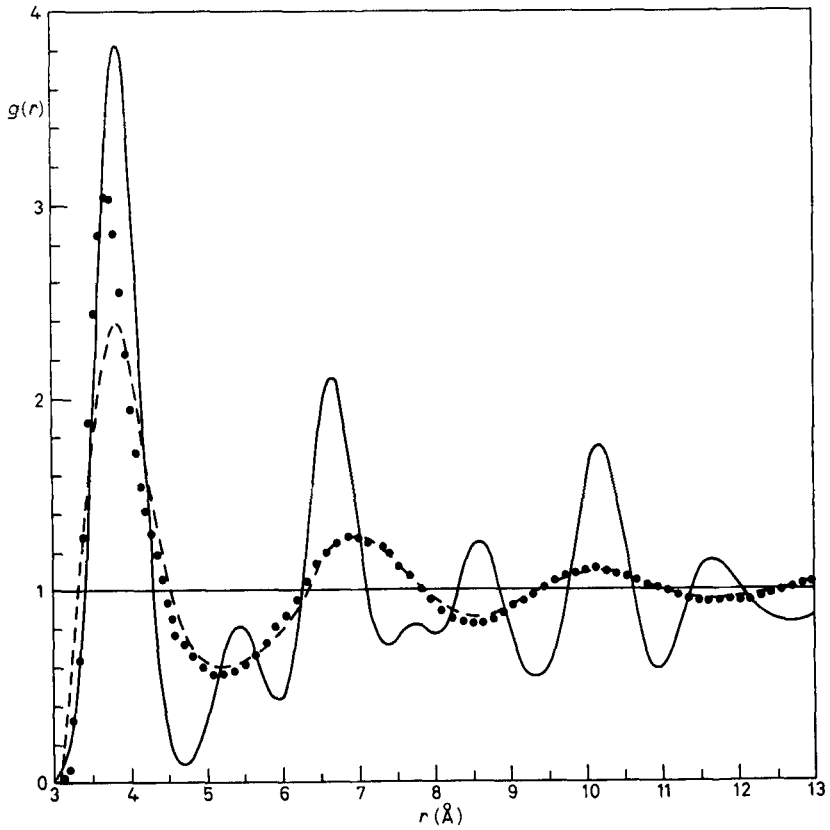


Figure 3.6: RDF for liquid argon as both a solid (solid line) and a liquid (points). The solid shows well defined sharp peaks corresponding to lattice points while the liquid shows broad peaks reducing over r. Taken from [99]

be shifted or even introduce a modulation to the plot. These errors occur because of the effect of transforming across a finite domain. If the structure factor contained data to infinity, the conversion would be exact but the loss of data from the finite domain real world experiments take place in means care must be taken to ensure no data is lost. With respect to the work detailed here, these errors are only important when making sure experimental data for comparison has been calculated correctly, as any calculation of the structure factors or RDFs have taken place directly and it has been unnecessary to transform between the two.

All of the necessary background has now been set in place and we can now move on to look at some applications of these methods. The subsequent chapters will describe work utilising all of the techniques and technology which have been explored to this point, beginning with MC simulations of a magneto-rheological gel.

Chapter 4

Simulation of Ferrogels

4.1 Introduction

A ferrogel, also known as a magnetic gel, comprises of a cross-linked, elastic polymer matrix which has material properties that respond to magnetic fields. This is achieved by embedding magnetic particles (often magnetite) within the polymer matrix [100, 101, 102]. They were first reported by Zrinyi in 1997 [103]. These first ferrogels were made using magnetite particles as the magnetic material and poly(vinyl alcohol) (PVA) as the gelling material leading to a superparamagnetic material. This first report of ferrogels showed simple, initial relationships between the deformation of a sample in a magnetic field and material properties such as concentration of magnetic particles or density of crosslinked gel and deformation.

If a magnetic gel is prepared with a polymer matrix with a large elastic modulus, the deformation of the material, or magnetodeformational effect (ME), will be very small. Of much more interest are gels with low elastic moduli which exhibit much larger responses to magnetic fields. Since this first report, a large amount of research effort has been spent on ferrogels due to their possible use in a wide range of applications. To date, they have been used in areas as diverse as 'crawling robots' [104, 105], media for localized hyperthermic treatment [106], drug delivery [107] , and artificial muscles[108].

Many applications rely on a time dependent or non-uniform magnetic field [27, 103, 102, 109] due to the enormous distortions that can be achieved, much greater than with a uniform magnetic field [110, 111, 112, 113, 114]. This difference in response to uniform and non-uniform magnetic fields can be understood from the response of dipoles to magnetic fields. In a non-uniform field, there is an additional force felt by the dipoles which is not present in a uniform field. Mathematically, the force experienced by a dipole in a magnetic field is given by

$$\vec{F} = \nabla(\vec{m} \cdot \vec{B}) \tag{4.1}$$

where $\vec{m}$ is the magnetic moment, ∇ is the gradient operator and $\vec{B}$ is the magnetic field. In a non-uniform field, this is non-zero and plays a significiant role in deformation. The effect of this force is so large that it can cause a magnetic gel to change in length by as much as 20% [108]. Conversely, in a uniform field $\vec{F} = 0$ and so the only forces felt are dipole-dipole interactions and a smaller deformational response is the result. In a uniform field with a soft ferrogel sphere, deformations of around 1.5% are observed [114].

In both the uniform and non-uniform cases, a torque will be felt by the dipoles. The torque,

$\vec{\tau}$, on a dipole in a magnetic field is described by

$$\vec{\tau} = \vec{m} \times \vec{B} \tag{4.2}$$

While this torque has no direct effect on the response of a magnetic gel due to the magnetic particle being fixed by the polymer matrix, it does have an input on the magnitude and direction of the force from the dipole-dipole interaction.

Even though the magnitude of the effect in a uniform magnetic field is much smaller, and the simple facts above rationalise the differences in the deformations there is still much to be explored and explained with a more detailed microscopic description such as magnetic and mechanical hysteresis [115].

Interestingly, deformations of different types, both contractions [116] and extensions [114], have been reported in a uniform magnetic field. These differences can be explained by exploring the mechanisms through which deformation takes place. The properties of the ferrogel are dictated by the coupling of magnetostriction (field-induced deformation leading to a change in magnetic susceptibility), magnetostatics (interactions between magnetic dipoles), and the elastic properties of the polymer-gel matrix. In the case of the uniform magnetic field, the reaction of the ferrogel is entirely driven by the interplay of these effects. Magnetostriction of the magnetic particles requires a large magnetic field and assuming that the gel has a low enough elastic modulus, the dominating cause of magnetodeformation will be the dipole-dipole interactions. As has been explained previously, dipolar systems can be strongly affected by boundary effects through the demagnetising field. The amount by which this will change the emergent effects of the dipolar interactions is strongly influenced by the shape of the sample. In order to see a contraction of the sample in [116], it was necessary to produce examples with large aspect ratios. Although the exploration of the effect of aspect ratio was not exhaustive, it was found that a larger aspect ratio allowed a larger response in an external field.

In cases where the shape of the sample under study is long and thin there will be a reduction of the effect of the demagnetising field. In this case, the deformation is seen to follow the behaviour expected for an infinitely large, homogeneous system of dipoles, with a contraction in the direction of the magnetic field and an extension perpendicular. In samples with a less extreme aspect ratio, the effect of the demagnetising field increases and the opposite deformation is seen. A theoretical study has sought to put limits on how the changes of aspect ratio will affect magnetodeformation [117]. The authors in this paper use an approach considering the energy of deformation to define the aspect ratios at which a sample will stretch and contract. They further define a *Procrustes point* which is the point at which the magnetostriction of the ferrogel sample as a whole balances the magnetostatics and no shape change under a uniform field would occur. The Procrustes point was found to be an aspect ratio of $a = 1.35$ with aspect ratios higher than this leading to a compression in the field direction and lower leading to an expansion.

One method of changing the rheological properties of ferrogels which has been investigated is to introduce some kind of anisotropy into the distribution of the embedded magnetic particles in the polymer matrix. The simplest way to do this is to synthesise the material in a magnetic field. By applying a magnetic field before cross-linking the polymers, the magnetic particles are allowed to diffuse through the sample and adopt the equilibrium conformations for the fluid phase. It is well known that dipolar particles tend to form chains under a magnetic field, and once they have been given time to adopt these anisotropic configurations, the crosslinking can

be initiated, locking the dipoles into place. This leads to a gel with anisotropic responses and properties as has been demonstrated through a number of experimental studies [118, 119, 102, 109, 120, 115]. One of these important and potentially useful properties is the elastic modulus, which would correspond to the 'stiffness' of the gel. The ability to modify the elastic response of an elastomer has important applications.

In ref [109], 10 to 30 wt% suspensions of iron-carbonyl microspheres in cross-linked polymer gel were prepared in applied fields of 400 mT and then subjected to fields in various orientations. It was found that the elastic modulus can be enhanced to varying degrees depending on the orientations of the fields during synthesis and measurement. If no field is applied during synthesis, then the elastic modulus of a 30 wt% ferrogel increases from 13 to 15 kPa (15% increase). If the fields during synthesis and measurement are parallel, then the elastic modulus in the field direction increases from 54 to 87 kPa (61% increase), while those in the transverse directions increase from 26 to 35 kPa (35% increase). If the fields during synthesis and measurement are perpendicular, then the elastic modulus in the measurement-field direction increases from 26 to 28 kPa (8% increase), while that in the synthesis-field direction increases from 54 to 60 kPa (11% increase). The increases in elastic moduli are less pronounced with lower concentrations of magnetic material. Qualitatively similar trends are seen for other ferrogels, including those containing much smaller magnetic grains [102]. These changes in properties must be down to the effects of the elastic properties of the gel and the distribution of the magnetic particles throughout the material. An obvious approach to gain an insight into the mechanisms of these changing rheological properties is via computer simulation.

Most previous computational approaches to modelling ferrogels follow a continuum mechanics approach [113, 112, 121]. The ferrogel is treated as a continuous medium with a well defined dielectric susceptibility, either field dependent or static, and the deformation of the sample is calculated through well understood continuum mechanics approaches. This model has been successful for the case of an oblate spheroid of ferrogel and for a torus and can accurately give material properties for these cases, it is limited in that no consideration is given to the microscopic structure and assumes an isotropic dispersion of magnetic particles through the carrier medium. For a large number of cases this approach yields useful, useable results but cannot be used to gain any insight on how the microscopic structure of the material can change rheological behaviour. For instance, these continuum models can successfully calculate distortions due to magnetostriction for simple spheroid shapes of ferrogels with a homogeneous distribution of particles through the carrier gel but would fail if the gel was prepared in a magnetic field causing a favoured direction for the magnetic particles.

In order to gain this insight, a model requires a level of microscopic detail. With this level of detail, the changes in dispersal of the magnetic particles through the carrier gel, and the differences in the magnetostatic forces between the dipoles when an external field is applied, can be captured and their effect on rheology explored. In the model described in the following, a microscopic representation of the internal structure of the the magnetic colloids in the ferrogel is used in conjunction with a continuum model for the elastic gel to capture the effect of changes to the dispersion of the colloids. This is the scientific question we have set out to answer: What effect does the introduction of a non-homogeneous distribution of particles have on the rheology of a ferrogel and what mechanisms effect this?

In this preliminary study, we limit the magnetic beads to monodisperse dipolar spheres. This is a simplification from the experimental situation, as it is understood that polydispersity of the

magnetic particles is an unavoidable fact [122]. Despite this simplification, the good agreement with experimental data shown below justifies this approach and to some extent shows that the polydispersity associated with the production of ferromagnetic particles has a minimal effect on the response of the ferrogel.

4.2 Model of a ferrogel

In order to correctly model the ferrogel, it is first necessary to generate the distributions of magnetic particles which will be used for the microscopic structure in the gel. These were generated using a fluid simulation of dipolar hard spheres (DHS) via a Monte Carlo (MC) simulation in the isothermal-isobaric ensemble (NPT) [93, 123]. These simulations consisted of $N = 250$ dipolar hard spheres in a cubic box of volume $V = L^3$ with a uniform magnetic field, $\vec{H}_f$ applied. In the following, the subscript "f" refers to the fluid phase. The Boltzmann distribution is given by

$$p_f \propto V^N \exp[-\beta(U_{DHS} + PV - \mu_0 \vec{H}_f \cdot \vec{M})] \tag{4.3}$$

where $\beta = 1/k_B T$, $U_{\mathrm{DHS}} = \sum_{i<j}^{N} u_{ij}$ is the DHS interaction energy and $\vec{M} = \sum_{i=1}^{N} \vec{\mu}_i$ is the magnetisation. The pair potential for two DHSs with separation vector $\vec{r}$ and dipoles $\vec{\mu}_i$ and $\vec{\mu}_j$ is given by

$$u_{ij} = \begin{cases} \infty & r \leq \sigma \\ \dfrac{\mu_0}{4\pi}\left[\dfrac{\vec{\mu}_i \cdot \vec{\mu_j}}{r^3} - \dfrac{3(\vec{\mu}_i \cdot \vec{r})(\vec{\mu}_j \cdot \vec{r})}{r^5}\right] & r > \sigma \end{cases} \tag{4.4}$$

where σ is the hard-sphere diameter. The long-range dipolar interactions were dealt with using the Ewald summation with conducting boundary conditions [93].

The MC simulation of the fluid sampled all degrees of freedom for the DHSs as both orientational and translational moves were attempted, along with volume moves. One sweep consisted of N translational or rotational moves, and one volume move. The acceptance criteria for these moves were adjusted to allow 20% of translational moves and 50% of orientational moves to be accepted. Any moves which caused an overlap between the particles were rejected immediately. Following equilibration, the configurations were output every 10^4 sweeps. These fluid runs were carried out in both zero field and with a number of different field strengths and orientations to allow different distributions of dipoles to develop.

The NPT runs allowed average conentrations, $\rho = \langle N/V \rangle$, and an average box length for the fluid, $\langle L_f \rangle$ to be calculated. These average values were then used as the starting point for the gel simulations and the configurations were scaled isotropically to match the initial box dimensions required. At least 20 particle configurations for each type of field were output and saved for use in the gel simulations.

The solid ferrogel was modelled at zero pressure in order to allow the rheological properties of the gel to drive any deformation under an applied field. Any applied external field would essentially pre-stress the gel and distort the effects of magnetostiction. The orthorhombic cell was allowed to deform independently in the box dimensions L_x, L_y and L_z. The DHS were considered to be locked into the gel matrix and so only underwent orientational moves. The

effects of the gel matrix were represented with an elastic potential

$$U_{el} = \frac{1}{2} L_0^3 G_0 \sum_{\alpha=x,y,z} \left(\frac{L_\alpha}{L_0} - 1 \right)^2 \tag{4.5}$$

where $L_0 = \langle L_f \rangle$ is the equilibrium cubic box length of the elastic matrix. This is a Hookean relationship for elastic forces and means that any volume move of the gel away from the length L_0 comes at an energetic cost given by the above equation. This energetic cost must be taken account of in the calculation of the Boltzmann factor as it forms part of the Hamiltonian of the complete system.

The limitations imposed on the box length moves mean that only affine deformations are permitted. This eliminates any consideration of shear deformation or irregular shape deformation and is a simplification of the true picture. The approach taken in this model does not easily lend itself to anything other than affine deformations and more complex distortions would be better modelled using an explicit representation of the gel. This would present it's own problems, such as matching of the boundaries if periodic boundary conditions were imposed. Periodic boundary conditions would be necessary as it would be impossible to simulate a large enough system to represent a macroscopic piece of material. Even with periodic boundary conditions, an explicitly represented gel matrix would require a large system size and would be stretching the the limits of technology.

The elastic modulus of the ferrogel was calculated by measuring fluctuations in the strain[124]

$$G_\alpha = \frac{k_b T}{\langle V \rangle_g \langle \epsilon_\alpha^2 \rangle_g} \tag{4.6}$$

where

$$\epsilon_\alpha = \frac{L_\alpha}{\langle L_\alpha \rangle_g} - 1 \tag{4.7}$$

and $\langle L_\alpha \rangle_g$ and $\langle V \rangle_g$ are, respectively, the average box lengths and volume of the gel, and not the values from the fluid-phase simulations. This is important as the fluid phase may have been prepared in a different magnetic field and so would have different average dimensions. Any comment on the deformation of the gel phase should be treated very carefully. While the rheological properties of the gel can be calculated correctly, the extreme shape dependence of the type of deformation means that the results here would only correspond to a particular shape of gel that minimised the effect of the demagnetising field. This will be expanded upon further in the results section. All of the properties calculated were averaged over at least 20 simulations of 10^6 MC sweeps, each started from an independent configuration generated in the fluid phase.

The simulations of the ferrogel in an arbitrary homogeneous magnetic field, $\vec{H}_g$, were performed according to the Boltzmann distribution

$$p_g \propto \exp[-\beta(U_{DHS} + U_{el} - \mu_0 \vec{H}_g \cdot \vec{M}) \tag{4.8}$$

where the subscript "g" refers to simulation of the gel phase. This is the probability distribution that the Monte Carlo simulation will reproduce and contains terms calculating the energetic costs of volume moves away from the equilibrium position (U_{el} given by 4.5), the interactions between the DHS (given by U_{DHS}) and the interactions with the external field (the last term in

the exponential function) . This distribution is modified from the standard NPT distribution due to the loss of a number of degrees of freedom. The DHS are locked in place and so only retain the orientational degree of freedom. The box lengths are sampled as though the gel is a solid body with embedded interaction sites, undergoing affine thermal fluctuations under zero applied pressure. This leaves the only degrees of freedom as the dipole orientations and box lengths which means the factors of V^n and the $-\beta PV$ term from the Boltzmann factor should be discarded.

It is not entirely straightforward to see that this modification of the Boltzmann factor is correct. To check this modification, it is instructive to consider what would happen if solely the gel was considered, with no DHSs and no magnetic field interactions. This would reduce the Boltzmann distribution to

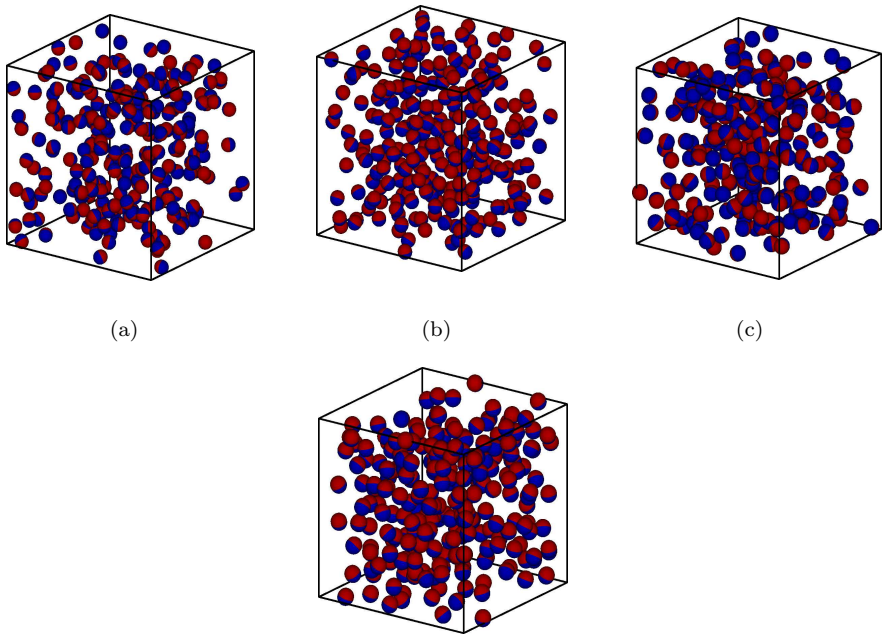
$$p_g \propto \exp(-\beta U_{el}). \tag{4.9}$$

Considering a large system where the fluctuations in the lengths are small compared to the total length, the ensemble averages are Gaussian which gives the average box lengths as

$$\langle L_\alpha \rangle = L_0. \tag{4.10}$$

The associated elastic modulus is then $G_\alpha = G_0$ which is the correct value for the pure gel.

All of the simulations carried consisted of 250 dipolar particles for both phases, snapshots of which can be seen in fig 4.1. As can be seen in the snapshots, there was no strong chaining effect and yet even this subtle ordering was shown to have a discernible effect on the rheology of the system. This is a relatively small simulation so there are obviously concerns about finite size effects. In order to check whether the use of such a small system would cause problems, spot checks were carried out with systems of 500 particles. While this check does not use a huge system itself, if there were appreciable problems with small systems, doubling the system size should produce significantly different results. In all cases doubling the system size yielded the same results within statistical error.



(a) (b) (c)

(d)

Figure 4.1: Simulation snapshots of DHS fluids. (a) $H_f^* = 0$, $\phi = 0.064$; (b) $H_f^* = (0, 0, 5)$, $\phi = 0.071$; (c) $H_f^* = 0$, $\phi = 0.096$; (d) $H_f^* = (0, 0, 5)$, $\phi = 0.106$

As is common in coarse-grained simulations, reduced units were used throughout the simulations. The DHS diameter was chosen as the characteristic length scale of the system and the thermal energy,k_BT chosen as the characteristic energy. While the experimental parameters of ferrogels vary widely, it is desirable to at least have some contact with experiment when choosing values and parameters for the basic setup of the simulations. Real ferrogels are often synthesized with a magnetic component of $10 - 30\%$ by weight. This component will be made up of roughly spherical grains of magnetite (Fe_3O_4) around 10nm mean diameter [115, 102, 125]. The mass density of magnetite is around 5 g cm$^-3$ and assuming a mass density of around 1 g cm^3 for the polymer gel, the required volume fraction of magnetic grains is $\phi = \pi\rho\sigma^3/6 \sim 0.01 - 0.1$. NPT simulations of the fluid phase were carried out with $\beta P\sigma^3 = 0.15$ and 0.25, giving volume fractions of $\phi \simeq 0.07$ and 0.1, respectively.

The magnetic properties are likewise expressed in reduced units. In order to calculate the reduced dipole moment of a particle we must know the saturation magnetisation of the constituent material. This is the maximum magnetic moment per unit volume of the bulk system and it is used to define the magnetic moment via

$$\mu = \frac{\pi M_s \sigma^3}{6} \tag{4.11}$$

where M_s is the saturation magnetisation. In magnetite, $M_s = 4.8 \times 10^5$ A m^{-1} at $T = 293$ K giving $\mu = 1.25$. The reduced dipole is given by

$$\mu^* = \sqrt{\frac{\beta\mu_0\mu^2}{4\pi\sigma^3}} \tag{4.12}$$

and in this work μ^* was taken to equal 1. This means that μ was taken as one of the fundamental

$\vec{H}_f^*$	$\beta P\sigma^3$	$\rho\sigma^3$	ϕ	$M_x/N\mu$	$\beta G_x(0)\sigma^3$	$\beta\Delta G_x\sigma^3$	$\Delta G_x/G_x(0)$	$\beta G_z(0)\sigma^3$	$\beta\Delta G_z\sigma^3$
0	0.15	0.123	0.064	0.0004	15.88	2.97	0.187	15.52	-0.06
0	0.25	0.183	0.096	-0.0003	20.92	7.76	0.371	20.66	2.01
(5,0,0)	0.15	0.136	0.071	0.81611	20.85	8.33	0.399	15.03	1.07
(5,0,0)	0.25	0.203	0.106	0.82179	27.862	24.00	0.8614	20.78	2.70
(0,0,5)	0.15	0.136	0.071	0.81611	14.90	2.70	0.181	19.692	-1.77
(0,0,5)	0.25	0.203	0.106	0.82179	19.099	8.39	0.4393	25.95	-3.80

Table 4.1: Results for model ferrogels prepared from the fluid phase in a magnetic field H_f. P is the osmotic pressure of the DHSs in the fluid phase, ρ is the concentration of DHSs, and ϕ is the corresponding volume fraction. M_x is the average magnetization in the field direction. $G_\alpha(0)$ and ΔG_α are the fitting parameters in Eqs. 6.1 and 6.2 relating the elastic moduli to the magnetization in the x direction. The figures in brackets are the estimated statistical uncertainties in the final digits.

units of my system to calculate the reduced units with. This means that $\mu_0 = 1.25$ in all the other equations to calculate the reduced units and thus in the simulations $\mu* = 1$. The reduced magnetic field is given by

$$H^* = H\sqrt{4\pi\beta\mu_0\sigma^3} \tag{4.13}$$

meaning $H^* = 1$ corresponds to approximately $H = 16$ kAm^{-1} or $B = 20$mT. H^* in the range $1 - 10$ corresponds to typical experimental fields. Finally, the reduced elastic modulus is given by

$$G^* = \beta G\sigma^3 \tag{4.14}$$

A typical polymer gel has an elastic modulus in the range $G_0 = 100-100$ kPa which corresponds to $G_0^* = 2 - 20$. In this work, a representative value of $G_0^* = 10$ is used in all polymer gel simulations.

4.3 Results

This section is broken down into sub-sections explaining the behaviour of the system at different concentrations under the same preparation and gel-phase conditions. Table 4.1 contains the parameters for the systems considered here. The choice of the magnetic field strength was influenced by the magnetisation of the system. A choice of $H^* = 5$ was found to give a high magnetisation, $M/N\mu \simeq 0.8$, and so this field strength was chosen for the ferrogel simulations. In general, the concentration of the DHSs in the fluid phase at a given pressure varies only weakly with applied field, and furthermore when the gel potential is switched on, the average concentration changes by an insignificant amount. Details for all systems studied will be reported below.

4.3.1 $\vec{H}_f^* = 0$, $\vec{H}_g^* = (H_x^*, 0, 0)$

Figure 4.2 show results for model ferrogels at two different concentrations, prepared with no magnetic field applied. In real applications, the deformation of a ferrogel in an applied magnetic field is best characterised with reference to the zero-field dimensions. Relative deformations are therefore defined by

$$\delta(\cdots) = \frac{\langle\cdots\rangle_g(H_x)}{\langle\cdots\rangle_g(0)} - 1. \tag{4.15}$$

$\delta(L_\alpha)$ and $\delta(V)$ are shown in fig 4.2 for each of the model ferrogels as a function of applied field $\vec{H}_g^* = (H_x^*, 0, 0)$. The x dimensions of the system shrink a little due to the alignment and nose-to-tail attraction of the dipoles along the direction of the field, while the transverse dimensions expand due to the corresponding side-by-side parallel repulsions. Nonetheless, these distortions are much less than 1% and are therefore insignificant (as observed in experiments). The variation in the system volume is also insignificant, being on the order of 0.1%. It is therefore safe to make the common assumption that the overall volume is fixed. The concentrations of DHSs therefore remain roughly constant with applied field, and the average concentrations shown in Figure 4.2 apply to each field strength and vary imperceptibly from the fluid-phase concentrations reported in Table 4.1. Snapshots of the DHS configurations are shown in figs 4.1(a) and 4.1(c)

The elastic moduli G_α are shown in fig 4.2. Firstly, the elastic moduli are seen to increase with increasing DHS concentration due to the influence of the interparticle interactions. Upon the application of a field, the elastic modulus in the field direction (G_x) increases with increasing field strength, while the elastic moduli in the transverse directions (G_y, G_z) remain roughly constant. The increase in G_x is more pronounced at the higher DHS concentration, in good correspondence with what is observed in experiments. To compare further with experiment, a DHS volume fraction of $\phi = 0.064$ equates to a magnetic loading of about 25 wt%. In this case, G_x^* increases by 13% from 16 to 18; in real units, this corresponds to G_x = 65-73 kPa. This enhancement is comparable to that seen in some experiments, but a more detailed comparison is precluded by the large number of variables such as particle-size polydispersity (and its relation to ferromagnetism and intrinsic superparamagnetism), polymer-gel elastic modulus, and range of magnetic field strengths that differ from study to study.

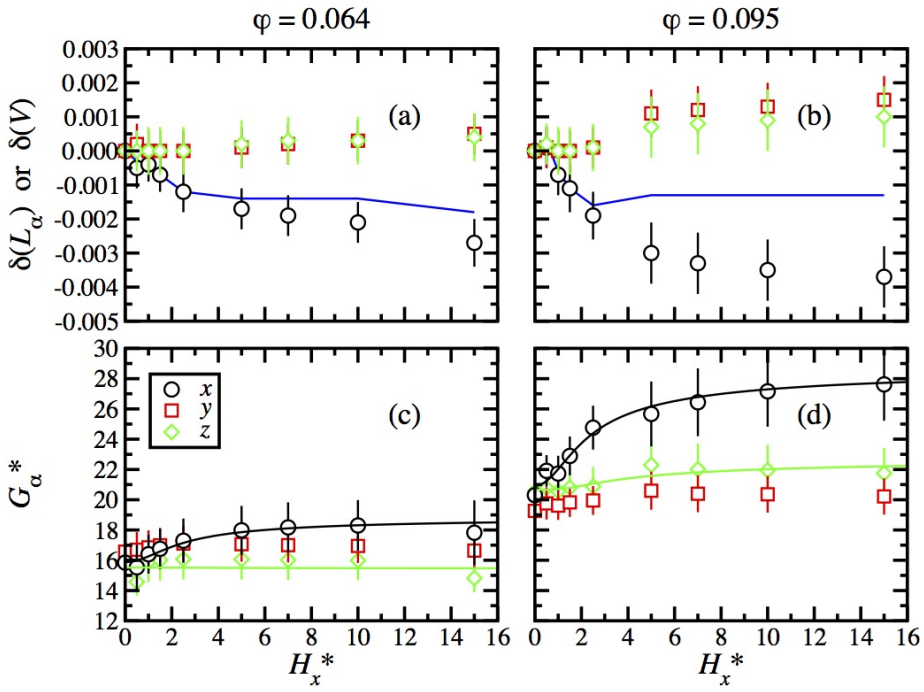


Figure 4.2: Relative deformations δ (top row) and elastic moduli G_α (bottom row) as functions of the applied magnetic field $\vec{H}_g^* = (H_x^*, 0, 0)$ for model ferrogels prepared with no magnetic field applied and at magnetic volume fractions $\phi = 0.064$ (left column) and $\phi = 0.095$ (right column). In (a) and (b) the points are $\delta(L_\alpha)(\alpha = x, y, z)$ and the line is $\delta(V)$. In (c) and (d) the points are from simulations and the solid lines are from eqs. 6.1 (black lines) and 6.2 (green lines) with $M_x(H_x)$ given by eq. 4.18

4.3.2 $\vec{H}_f^* = (5, 0, 0)$, $\vec{H}_g^* = (H_x^*, 0, 0)$

Figure 4.3 shows results for model ferrogels at two different concentrations, prepared with a magnetic field $H_f^* = (5, 0, 0)$. As for the zero-field case, the relative deformations $\delta(L_\alpha)$ and $\delta(V)$ are less than 1% and the overall DHS concentration remains almost constant over the whole range of magnetic field strengths. An applied magnetic field increases the elastic modulus in the field direction, while the elastic moduli in the transverse directions show only very slight increases; these changes are mirrored by the very small but systematic variations in δ_α. In the system with $\phi = 0.071$ (corresponding to a loading of 28 wt%) G_x^* increases by 33% from 21 to 28 (in real units 85-113 kPa). This increase is greater than that in the $H_f = 0$ case discussed previously; this is in qualitative agreement with experiment. Clearly, this has to do with the way the particles are positioned in the matrix. Figures 4.1(b) and (d) show that there are no clearly visible defined chain-like clusters. Nonetheless, the dipolar orientations are aligned by an applied field and the particle positions show enhanced nose-to-tail correlations in the field direction. Figure 4.4 shows a two-dimensional, grey scale plot of the pair correlation function. This is plotted in terms of the components orthogonal and parallel to the the field direction with $x_\perp = x\sin(\theta)$, $x_\parallel = x\cos(\theta)$ where θ is the angle of the vector between two particles with respect to the field. In 0 field (figs 4.4(a) and 4.4(c)) there is a short ranged attraction close to contact at $x = 1$ which is isotropic. When a strong field is applied (figs 4.4(b) and 4.4(d)), the isotropic nature of the attraction disappears and there is an increase in the short range attraction in the direction of the field. This shows that, despite the structure being difficult to seen in the snapshots in fig 4.1, there are chain-like, nose to tail correlations being induced by

the external magnetic field.

Interestingly, there is already a degree of anisotropy in the elastic moduli in zero field: the material is stiffer in the x direction than in the y and z directions. This is also due to the enhanced positional correlations in the x direction; even in zero field, the DHSs are still close to contact and interacting with each other strongly through both the hard-core and dipolar forces. The particles are separated further from one another in the y and z directions, however, and this leads to a relative softening of the material in those directions. The zero-field anisotropy observed here is in qualitative agreement with what is seen experimentally [115, 102, 125, 109].

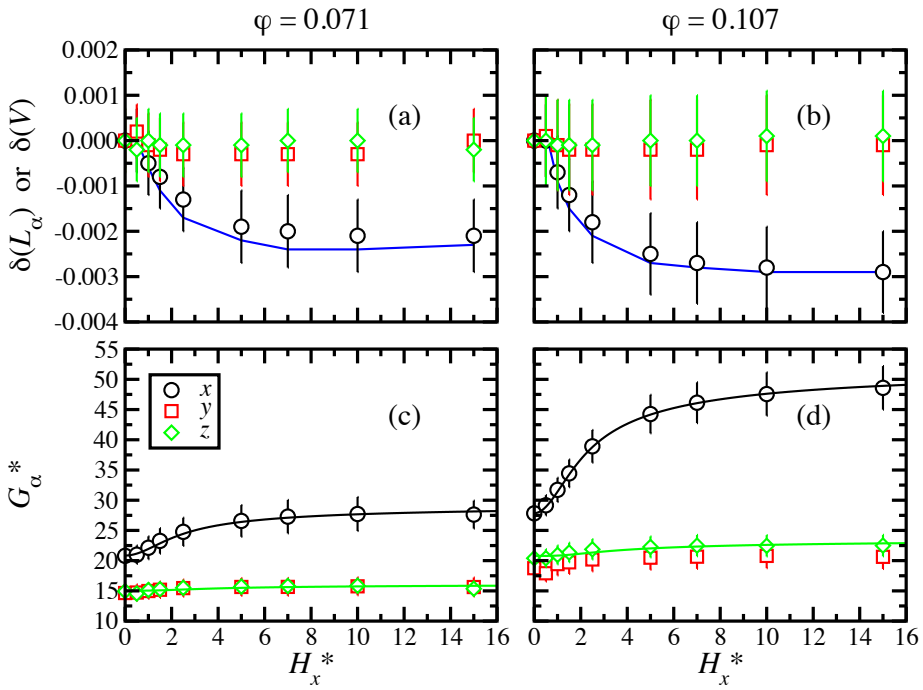


Figure 4.3: Relative deformations δ (top row) and elastic moduli G_α (bottom row) as functions of the applied magnetic field $\vec{H}_g^* = (H_x^*, 0, 0)$ for model ferrogels prepared in a field $\vec{H}_f = (5, 0, 0)$ and at magnetic volume fractions $\phi = 0.071$ (left column) and $\phi = 0.107$ (right column). In (a) and (b) the points are $\delta(L_\alpha)$ $(\alpha = x, y, z)$ and the line is $\delta(V)$. In (c) and (d) the points are from simulations and the solid lines are from eqs 6.1 (green lines) and 6.2 (black lines) with $M_x(H_x)$ given by eq 4.18

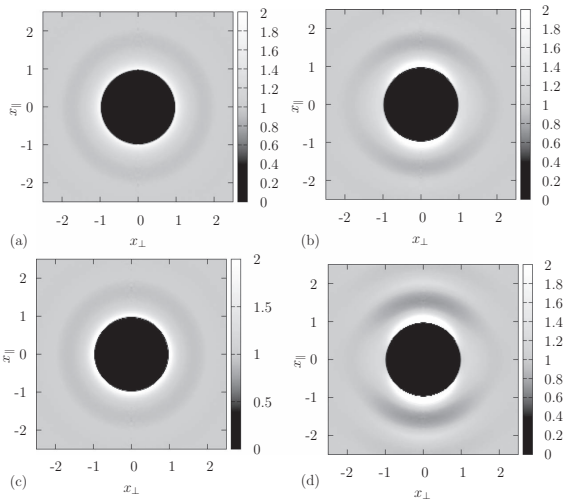


Figure 4.4: Grey scale plots of the pair correlation function $g(r_\parallel, r_\perp)$ in DHS fluids; $x = r/\sigma$ is the dimensionless distance. (a) $\vec{H}_f^* = 0$, $\phi = 0.064$; (b) $\vec{H}_f^* = (5, 0, 0), \phi = 0.071$; (c) $H_f^* = 0, \phi = 0.096$; (d) $\vec{H}_f^* = (5, 0, 0), \phi = 0.106$.

4.3.3 $\vec{H}_f^* = (0, 0, 5)$, $\vec{H}_g^* = (H_x^*, 0, 0)$

The final situation to be studied is when the field applied to the fluid phase is perpendicular to the field applied to the resulting ferrogel. The results are shown in fig 4.5 shows the results. Once again, the relative deformations of the ferrogel in an applied field are rather small, being less than 1%. In zero field, the elastic moduli show anisotropies opposite to those in the previous section: the elastic modulus in the direction of H_f (now the z direction) is greater than those in the other two directions; this is in good correspondence with experiment. As before, this is due to the presence of enhanced positional correlations in the field direction between particles close to contact. An applied magnetic field increases the elastic modulus in the field direction (x) and decreases that in the z direction. The particle positions are correlated in the z direction but the dipoles are oriented along the x axis (side-by-side parallel), which means that the dipoles repel along the z direction; this manifests itself in a very slight increase in δ_z and a reduction in the elastic modulus. Although no chain-like clusters are apparent in the simulation configurations, enhanced correlations in the field direction are certainly there as already shown in Fig. 4.4. For the ferrogel with $\phi = 0.071$ (corresponding to a loading of 28 wt%) G_x^* increases by 13% from 15 to 17 (in real units 61-69 kPa); this change is in good agreement with that typically seen in experiments. The elastic modulus and deformation in the y direction show almost no changes.

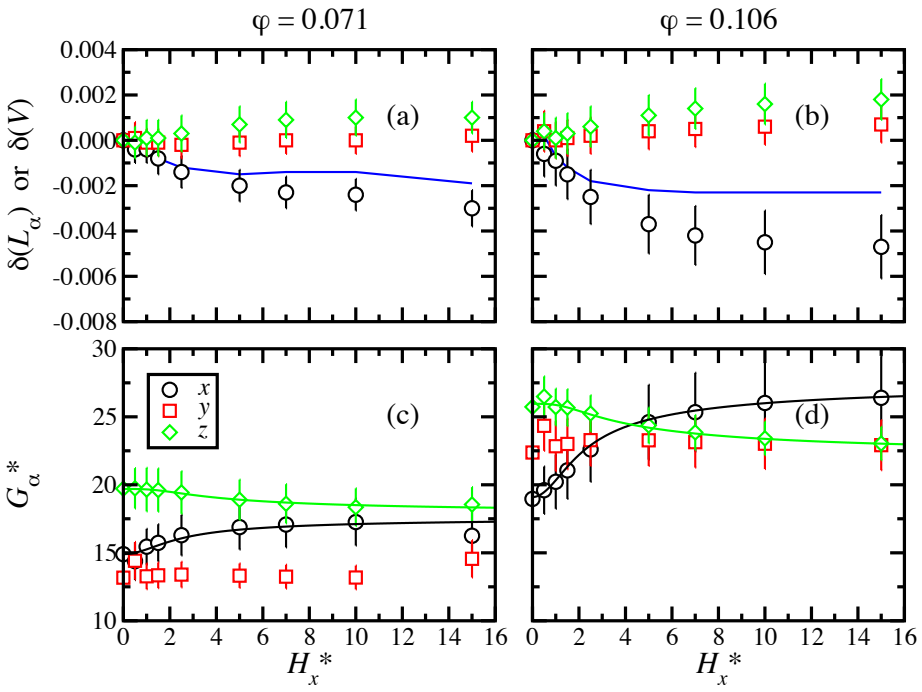


Figure 4.5: Relative deformations δ (top row) and elastic moduli G_α (bottom row) as functions of the applied magnetic field $\vec{H}_g^* = (H_x^*, 0, 0)$ for model ferrogels prepared in a field $\vec{H}_f^* = (0, 0, 5)$ and at magnetic volume fractions $\phi = 0.071$ (left column) and $\phi = 0.106$ (right column). In (a) and (b) the points are $\delta(L_\alpha)(\alpha = x, y, z)$ and the line is $\delta(V)$. In (c) and (d) the points are from simulations and the solid lines are from eqs. 6.1 (green lines) and 6.2 (black lines) with $M_x(H_x)$ given by eq. 4.18

4.3.4 The relationships between the elastic moduli, the magnetization, and the field

In order to tease out the relationships shown in the previous figures it is necessary to consider G_α both parallel and perpendicular to the applied field individually. Considering G_x, parallel to the applied field in the gel simulation, there is some previous experimental work which can guide us. In earlier work, Varga *et al.* noted that at small $H_x, G_x(H_x) - G_x(0) \propto H_x^2$, while at large $H_x, G_x(\infty) - G_x(H_x) \propto 1/H_x^2$ [115, 109]. On this basis, they proposed a fitting function equivalent to

$$G_x(H) = G_x(0) + \Delta G_x \left(\frac{H_x^2}{a_H + H_x^2} \right) \simeq G_x(0) + \Delta G_x H_x^2 + \mathcal{O}(H_x^4) \tag{4.16}$$

where a_H is a material parameter. Equation 4.16 gives excellent fits to experimental data, but it is a heuristic relationship and it would be desirable to obtain a more physically transparent form. To this end, note that at small H_x, the magnetization curve $M_x(H_x) \approx \chi H_x$ where χ is the initial magnetic susceptibility. This suggests that G_x may have a very simple dependence on the square of the corresponding magnetisation M_x^2.

In order to utilise the magnetization curve to find the dependence of G_x on M_x, we must first consider the behaviour of M_x. At large H_x, the magnetization reaches saturation as $M_x(H_x) \approx N\mu - N/\beta\mu_0 H_x$ because the field-dipole interaction dominates and so the magnetization curve

reduces to the appropriate limit of the single-particle, Langevin result:

$$M_x^L(H_x) = N\mu \left[\coth(\beta\mu_0\mu H_x) - \frac{1}{\beta\mu_0\mu H_x}\right] \tag{4.17}$$

Figure 4.6 shows the magnetisation curves for two DHS fluids and all six of the model ferrogels studied in this work. There is very little dependence on the DHS concentration and the preparation conditions. (In all cases, M_y and M_z are zero within the statistical uncertainties.) Theoretical magnetisation curves for DHS fluids are shown for comparison according to the modified mean-field theory of Ivanov and Kuznetsova [126]. In this theory, the fluid magnetization curve is given by

$$M_x(H_x) = M_x^L(H_x^{\text{eff}}) \tag{4.18}$$

where the effective field, H_x^{eff} is given by

$$H_x^{\text{eff}} = H_x + \frac{4\pi}{3}\frac{M_x^L(H_x)}{V} + \frac{(4\pi)^2}{144}\frac{M_x^L(H_x)}{V^2}\frac{dM_x^L}{dH_x}. \tag{4.19}$$

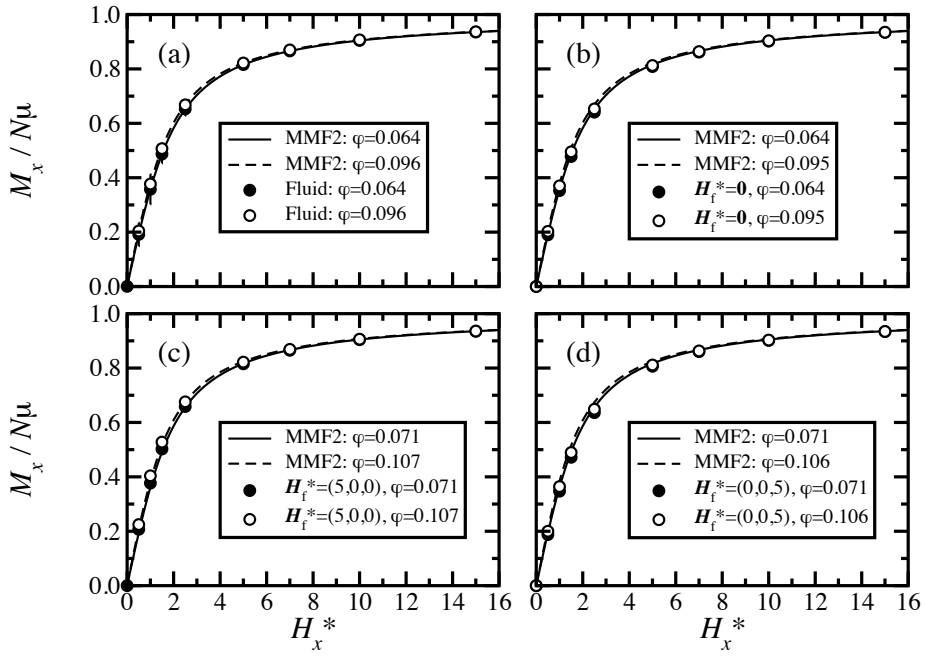


Figure 4.6: Magnetisation curves for ferrofluids and ferrogels (points) with the predictions of modified mean-field theory for the fluid phase (lines). (a) Fluids. (b) Ferrogels prepared with no magnetic field applied. (c) Ferrogels prepared in a field $\vec{H}_f^* = (5, 0, 0)$. (d) Ferrogels prepared in a field $H_f^* = (0, 0, 5)$. The volume fractions ϕ of DHSs are indicated in the legend.

Equation 4.19 has been shown to give an excellent account of the magnetisation curves in real[127, 128] and simulated[129, 130] ferrofluids, and the corresponding predictions for the magnetic susceptibility are also highly accurate [122, 126, 131]. This is confirmed for our cases in figure 4.6 with the accuracy of equation (4.18) being further confirmed for ferrogels.

Figure 4.7 shows the elastic modulus G_x plotted against M_x^2 for all six ferrogels studied in

this work. As anticipated, the results can be fitted accurately using the simple relationship

$$G_x(M_x) = G_x(0) + \Delta G_x \left(\frac{M_x}{N\mu}\right)^2 \tag{4.20}$$

where $0 \leq M_x/N\mu \leq 1$ is a dimensionless, fractional measure of the magnetisation. On symmetry grounds, $G_x(M_x)$ must be an even function of M_x and the high quality of the fits suggest that terms higher in order than M_x^2 are not required. The fit parameters $G_x(0)$ and ΔG_x are given in Table 4.1. Figure 4.7 illustrates succinctly how the degree of enhancement in G_x depends on the DHS concentration and the preparation conditions. A convenient measure for the

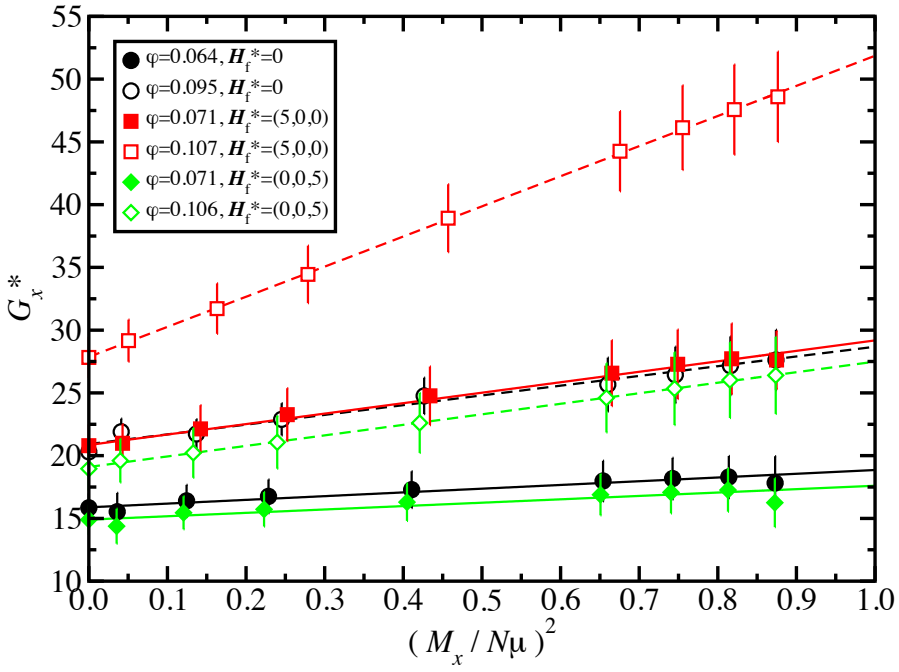


Figure 4.7: Ferrogel elastic modulus G_x as a function of the square of the magnetisation M_x^2 (points) with the fits from Eq. 6.1 (lines). The magnetic fields and volume fractions ϕ of DHSs are indicated in the legend.

maximum possible enhancement in G_x is the ratio $\Delta G_x/G_x(0)$. The values of this parameter are included in Table 4.1 and confirm that, as seen in experiments, the greatest enhancement in elastic modulus is achieved when H_f and H_g are parallel.

Substituting eq (4.18) for the magnetisation curve in eq (6.1), yields the relationship between G_x and H_x. The resulting curves are shown in figs 4.2, 4.3, and 4.5. In all cases the agreement with simulation is essentially perfect.

In the ferrogels prepared with $H_f^* = 0$ and $\vec{H}_f^* = (5,0,0)$, G_y and G_z show only very weak dependences on the applied field. In the $\vec{H}_f^* = (0,0,5)$ case, however, G_z exhibits an interesting dependence on H_x. By analogy with Eq. 6.1, a simple relationship between G_z and

M_x is sought. It turns out that, heuristically, the best expression for fitting is

$$G_z(M_x) = G_z(0) + \Delta G_z \left(\frac{M_x}{N\mu}\right)^4 . \tag{4.21}$$

Figure 4.8 shows G_z plotted against M_x^4 for all six ferrogels studied in this work. Excellent fits are obtained, although the variations in G_z for the $H_f^* = 0$ and $H_f^* = (5, 0, 0)$ cases are small. The fitting parameters are given in Table 4.1. When supplemented by the magnetisation curve from eq. (4.18), eq. (6.2) furnishes an expression for G_z in terms of H_x. The resulting curves are also shown in Figs. 4.2, 4.3, and 4.5. Once again, the agreement with simulation is essentially perfect.

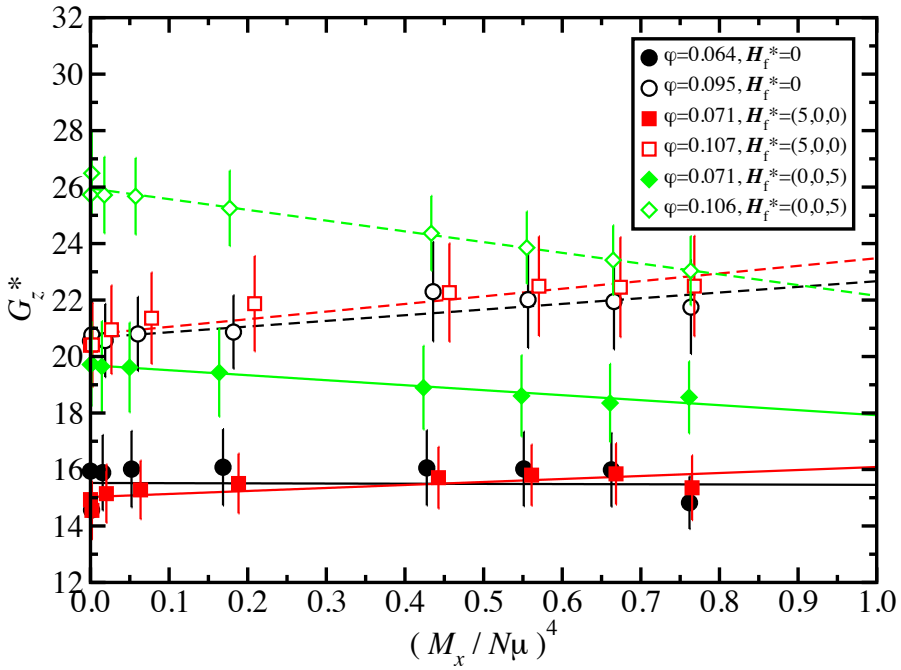


Figure 4.8: Ferrogel elastic modulus G_z as a function of the fourth power of the magnetisation M_x^4 (points) with the fits from eq. (6.2) (lines). The magnetic fields and volume fractions ϕ of DHSs are indicated in the legend

Finally, the anisotropy of a ferrogel can be characterized by a simple parameter like

$$A_{xz} = \frac{G_x - G_z}{G_x + G_z} \tag{4.22}$$

where the x and z components are selected because they show the greatest variations. Figure 4.9 shows A_{xz} for all six ferrogels. In the $H_f^* = 0$ case, the zero-field anisotropy is essentially zero within the statistical uncertainties. Applying a field increases the anisotropy, with the effect being more highly pronounced at higher concentration. In the $\vec{H}_f^* = (5, 0, 0)$ case, the zero-field anisotropy and high-field asymptote are greater than those for the $H_f^* = 0$ case. The zero-field anisotropy in the $\vec{H}_f^* = (0, 0, 5)$ case is negative because of the stiffening in the z direction. With $\vec{H}_f^* = (0, 0, 5)$ and $\phi = 0.106$ it is possible to tune the magnetic field (H_x^*) so that the elastic moduli in the x and z directions are equal. Out of the three cases, the greatest

anisotropy is achieved with $\vec{H}_f^* = (5, 0, 0)$. Combining eqs (6.1), (6.2), and (4.18), eq (4.22) can describe the dependence of the anisotropy on the applied field. Figure 4.9 shows the resulting curves, which are in excellent agreement with the simulation data.

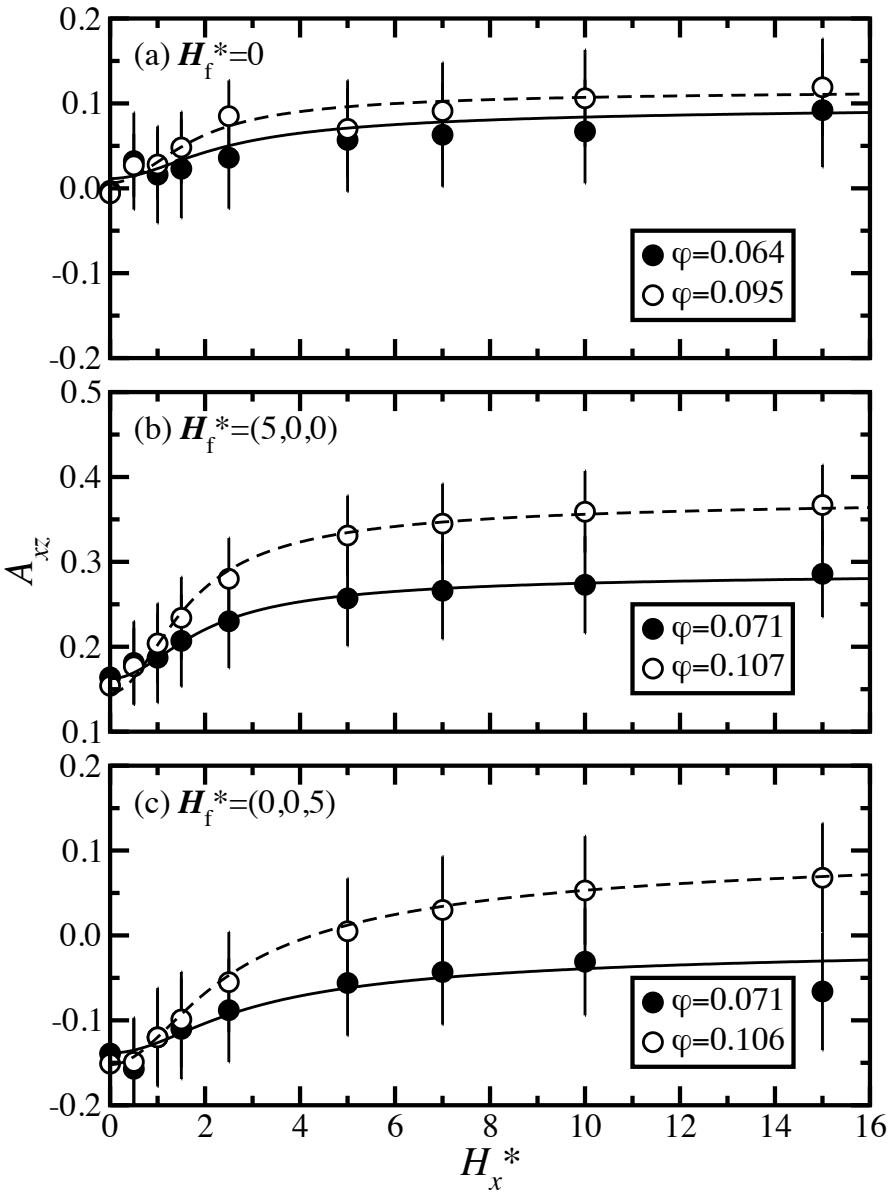


Figure 4.9: Anisotropy $A_{xz} = (G_x - G_z)/(G_x + G_z)$ of the elastic moduli in the x and z directions. (a) Ferrogels prepared with no magnetic field applied. (b) Ferrogels prepared in a field $\vec{H}_f^* = (5, 0, 0)$. (c) Ferrogels prepared in a field $\vec{H}_f^* = (0, 0, 5)$. The volume fractions ϕ of DHSs are indicated in the legend. The points are from simulations and the lines are from a combination of eqs (6.1) and (6.2) with $M_x(H_x)$ given by eq. (4.18)

4.4 Conclusion

A simple microscopic model of ferrogels has been described and its properties have been determined using MC simulations. The model represents the material as a dispersion of dipolar hard spheres trapped in an elastic matrix. In this preliminary study, a portion of ferrogel was restricted to be orthorhombic and only allowed to undergo affine deformations. Despite the simplicity of the model, it captures much of the phenomenology from experimental studies. Specifically, the elastic moduli of ferrogels can be tuned by applying uniform magnetic fields during the gel formation stage. The application of the field gives rise to a subtle enhancement of positional correlations between magnetic grains and a pronounced anisotropy to the material properties. A very significant enhancement of the elastic modulus can then be observed if the ferrogel is then placed in a parallel uniform field with less pronounced effects occurring with other arrangements of fields during and after the synthesis stage. It has been demonstrated using MC simulations that the model ferrogel mimics experimental trends quite reliably, while providing specific insights on the microscopic arrangements of the magnetic grains. These insights suggest that the introduction of a magnetic field during the production of the ferrogel causes the dipolar particles to form chain-like structures with the dipole moments lining up 'nose to tail'. These chain-like structures can provide a mechanical stiffening effect with the dipolar particles unable to slide past each other. Furthermore, these chain-like structures could also allow the magnetic interactions between the dipolar particles to play a role in the rheological properties by resisting any movement away from the energetically favourable orientations they moved to during the formation of the ferrogel. These structural facts provide an answer to the question of the mechanisms which effect the rheology of the ferrogel. In addition to these insights, the simulation results suggest a particularly simple relationship between the elastic moduli and the magnetisation. When supplemented by a magnetisation curve, this relationship leads to the dependence of the elastic moduli on the magnetic field strength, which is normally the property measured directly in experiments.

Chapter 5

Computer simulations of surface deposition of amphiphilic diblock copolymers driven by solvent evaporation

5.1 Introduction

Polymers have become ubiquitous in everyday life with applications including fibres, textiles, packaging materials, surface coatings (controlling function, friction, and adhesion), medical devices, and fluid modifiers. This adaptability stems from the ability to tune the characteristics of polymers at the molecular level in order to generate desired behaviour and responses under a given set of physical conditions. Block copolymers extend this adaptability further by allowing each block in a single polymer the possibility of reacting differently to the environment, giving rise to complex structures and responses that depend on external factors such as pH [132, 133] and exposure to solvents [134, 135]. Specific interest has grown in thin films of these polymers as they have wide use in areas as diverse as nanostructured materials [135, 136] and biotechnology [137].

The formation and structure of thin films of linear homopolymers [138, 139, 140, 141, 23], star homopolymers [142], hetero-arm star polymers [143], and diblock [144, 21, 25, 20, 145] and triblock [146] copolymers have been studied experimentally. These studies have covered chemisorbed and physisorbed polymers, and have been concerned with single-molecule properties such as the coil-globule transition, and thin-film properties such as friction. The focus here will be on the adsorption of polymers on to surfaces from solution, and the subsequent structural reorganisation of the polymers upon solvent evaporation. As such, it is the mechanism by which polymers physisorb and subsequently collapse on to the surface to form the thin film which will be examined.

In general, the study of adsorption of polymers of all kinds follows a typical experimental protocol. A clean surface (such as mica or highly ordered pyrolytic graphite) is immersed in a polymer solution - typically in good solvent conditions - and allowed to reach equilibrium adsorption. Then, the surface is removed and rinsed with solvent, and the excess solvent is

removed by rapid drying in a gas stream. Finally, the organisation and structure of the polymers are investigated using a surface technique such as atomic-force microscopy (AFM). In recent studies, this protocol has been used to study, in detail, the adsorption of star homopolymers [142] and diblock copolymers [145] on to mica surfaces. A significant result from these studies is that the basic effect of solvent evaporation is to switch from good solvent to bad solvent conditions. This means that the steady-state polymer structures after solvent evaporation are related to the structures that exist in good solvent conditions and the polymers undergo collapse and clustering due to the change in effective monomermonomer interactions. There is no experimental evidence for polymers being redistributed on the surface by the solvent layer due to mechanisms such as spinodal dewetting [147, 148, 149, 150, 151] or hole nucleation [147, 148, 152]. These mechanisms would give rise to characteristic large scale structures on the surface and specific variations with molecular weight that are not observed in the experiments [142, 145]. The lack of such structures strongly implies that the polymer conformations are not driven by the solvent evaporation. However, it remains the case that these processes cannot be viewed experimentally and the only evidence is inferred from steady-state observations. It is here that molecular simulations can go some way to filling that gap allowing the mechanisms to be visualised directly.

There have been many studies of the adsorption of linear homopolymers, with the key work being carried out with off-lattice bead-spring models [153, 154, 155, 156, 157, 158] although more accurate coarse-grained parametrisations are available for a variety of polymer systems [159, 160]. The attraction to the bead-spring models rather than to the more sophisticated parameterisations remains their simplicity and the ability to access large time and length scales, while retaining a sufficient degree of chemical resolution and realism in the polymer molecular structure.

Recent work has used simulations of linear and star homopolymers in order to make a direct link with AFM measurements and to gain insights on molecular-scale aspects of polymer adsorption [142, 161, 162]. These simulations utilised a Langevin dynamics approach with bead-spring models to correlate the experimental measurements with molecular characteristics such as polymer size and functionality, solvent quality, and surface interactions. The use of Langevin dynamics (discussed previously in section 3.3) meant that an implicit solvent was used, reducing the computational cost of long simulations. The effect of the solvent and its evaporation was captured via Brownian forces and a Stokes-law drag acting on the beads and by a switching of the interactions between the beads, emulating good- and bad-solvent conditions. While in good solvent, the monomer-bead interactions are chosen as repulsive mimicking the effect of the solvent separating the monomers. The solvent evaporation switches these interactions to attractive mimicking bad solvent conditions. In contrast with these studies, the work presented here utilises an explicit solvent, albeit a simplified 'atomic' model with multiple molecules represented by a large single bead. This allows a more faithful representation of the solvent-evaporation process, and gives an accurate picture of the solvation and structure of the physisorbed polymer molecules. The associated increase in computational cost is offset by exploiting GPU acceleration with a bespoke MD code (GPU programming is discussed in section 2.4).

The protocol and simulation for the current study was largely motivated by ref [145] and its method and results are summarised here. Poly(isoprene-b-ethylene oxide) (PI-PEO) diblock copolymers were synthesised with 29 wt% PI, corresponding to a PI monomer fraction of $x_{\mathrm{PI}} =$

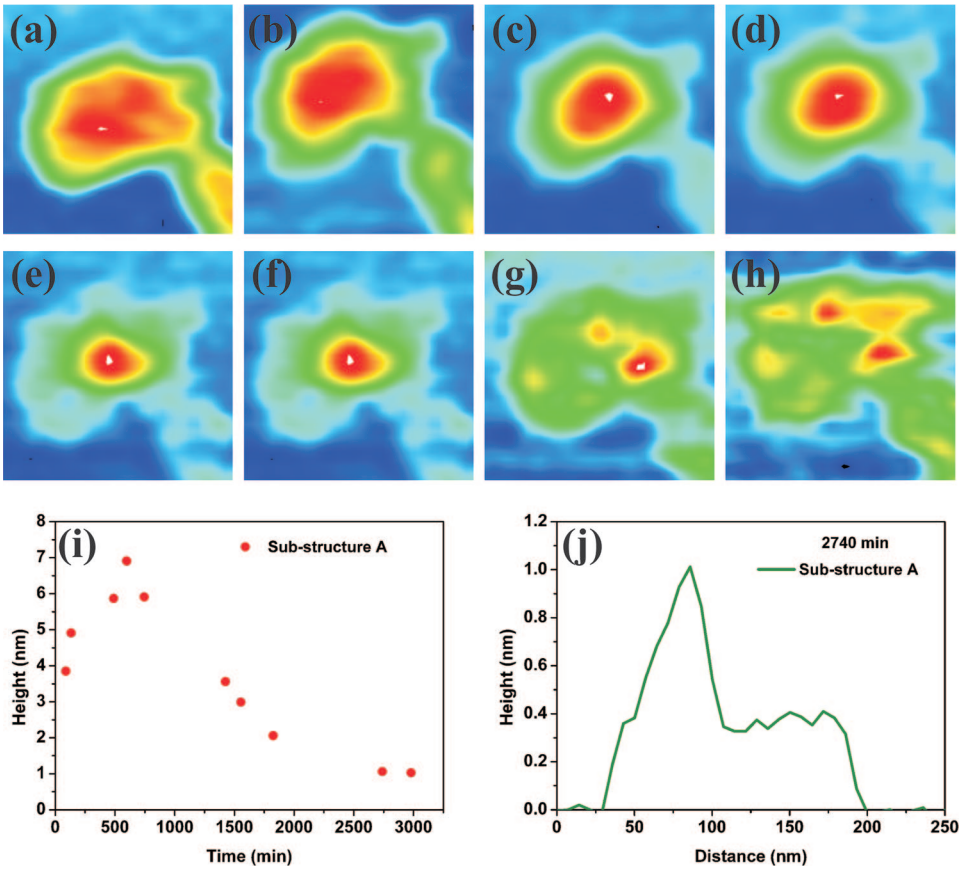


Figure 5.1: High-contrast AFM images for the case of the secondary substructure in structure A after (a) 133, (b) 489, (c) 598, (d) 745, (e) 1425, (f) 1555, (g) 1825, and (h) 2740 min since the sample was prepared; (i) plot of the corresponding maximum height evolution of the substructure: the height initially increased but later decreased; (j) a height profile of the substructure taken from an AFM image after 2740 min since the sample was prepared showing a maximum height of about 1 nm but also flat areas of about 0.4 nm. Taken from ref.[145]

0.21. These polymers were dissolved in ultrapure deionised water at concentrations of 2×10^{-3} g/mL , well above the critical micelle concentration of 1.44×10^{-5} g/mL. A drop of this solution was then placed on to freshly cleaved mica surfaces, which were rinsed and dried under a stream of nitrogen. The sample was then imaged using AFM in tapping mode. The PI-PEO diblock copolymers were seen initially to form flat polymer islands which were weakly adsorbed on the substrate. They then displayed an exponential-type growth of height with time with their lateral shape becoming circular. A study of one island can be seen in fig 5.1. This shows the evolution of one structure on the cleaved mica surface over the lifetime of the experiment. This shows that this structure increased in height up to 598 min and then decreased (fig 5.1i) and that it took on a bimodal conformation with areas of height 1nm and a wider area of height 0.4 nm. One possible explanation for this behaviour is a change in the affinity of mica to water, and a concomitant decrease in the thickness of the water layer due to evaporation.

It has been shown that in ambient air, mica adsorbs water from the environment [163, 164, 165, 166, 167, 168, 169] and this adsorbed water is shown to form two layers. The first is a structured layer about 0.2 nm thick [164, 165] and a thicker, bulk like layer ranging from a monolayer to about 2 nm thick [164, 165, 166, 169], denominated *phase II*. The affinity of mica

to water decreases with time due to processes such as the adsorption of organic contaminants that are always present under ambient conditions [165, 167].

The PEO blocks of the synthesised block copolymers are hydrophilic and are expected to extend into both the phase I and phase II layers. The PI blocks are hydrophobic and are thought to be floating on top of the phase II layer. This structure corresponds to the flat islands seen at short times. As the mica becomes less hydrophilic with time, the water layer thins and hence the PEO blocks are confined to a smaller volume within phase II. As a result, the PEO blocks on the edge of the island are forced to spread laterally within the phase-I layer and in contact with the surface. The PI blocks remained floating on top of the diminished phase-II layer in a smaller cap. This process is thought to give rise to the long-time growth in height. A cartoon of this process is shown in fig 5.2. The aims of this work are to reproduce and gain insight

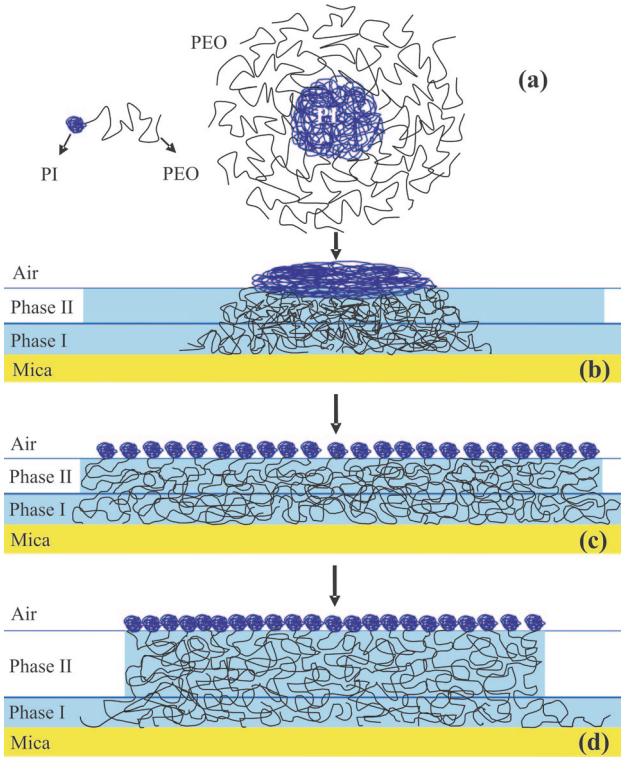


Figure 5.2: Schematic drawing of the proposed process for deformation of a deposited PI-PEO micelle on a freshly cleaved mica surface forming a flat brushlike polymeric island within a phase II water layer. Taken from ref.[145]

on the effect of the solvent evaporation on the structure of diblock copolymers adsorbed on surfaces such as mica, and to explore the approach to steady-state conditions, which may not necessarily be at thermodynamic equilibrium.

Techniques such as Monte Carlo simulations of lattice models provide valuable information on the equilibrium structures and thermodynamics of block-copolymer systems [170, 171], but they cannot capture the complex dynamical processes that are expected to be important in the current situation. Instead, off-lattice, bead-spring models of polymers in an explicit atomic solvent are studied using MD simulations. Using a simulation protocol consisting of equilibrium, evaporation, and steady-state phases, the effects of the polymer architecture (PI:PEO ratio) and solvent selectivity are surveyed systematically. Although not yet explored in the experiments

[145], these parameters are easily varied in simulations. The results clarify the roles played by the solvent in the polymer deposition process.

5.2 Simulation model and methods

The study of the system described above poses a significant computational challenge due to the large number of particles necessary for an explicit solvent. The explicit solvent is itself necessary, as an important part of the studied system is the liquid layer and its evaporation. The inclusion of an interface with the evaporation of the solvent leads to long computation times as systems at or near coexistence equilibrate slowly [172]. In order to deal with this a bespoke code has been developed to make use of the computational power of Nvidia GPUs via the language extension CUDA for C (see section 2.4). This allows large simulations to be carried out for many millions of time steps in a reasonable amount of time (on the order of a week). In this section we outline the model used and the simulation protocol.

5.2.1 Molecular models

The PI-PEO/solvent/mica system is modelled as follows. The PI units are hydrophobic while the PEO units are hydrophilic. Hence, the PEO units should have a more significant attraction to the mica surface. If the solvent is water, then it will have a strong attraction to the PEO units and the surface, but the PI-water interaction will be less favourable; in this sense, the solvent is selective. One might conceive of another solvent that has no strong preference for PI or PEO; this will be referred to as a non-selective solvent. In either case, the solvent molecules experience mutually attractive interactions.

A useful approach when considering systems of linear polymers is to model them as chains of coarse-grained beads connected by springs [80]. Each model diblock copolymer is comprised of N_b beads of equal mass, m, connected by non-linear infnitely extensible (FENE) springs defined by the potential

$$u_\mathrm{FENE} = -\frac{1}{2}kR_0^2 \ln\left(1 - \frac{r^2}{R_0^2}\right) \tag{5.1}$$

where r is the bead-bead separation, R_0 is the maximum possible bead-bead separation and k is a spring constant. The beads in each polymer may either be of type A (hydrophobic, e.g. PI) or of type B (hydrophilic, e.g. PEO). The solvent is represented as a system of single beads of type C. The different chemical identities of these beads are represented by the non-bonded interactions, all of which are expressed in terms of the Lennard-Jones (LJ) potential

$$u_\mathrm{LJ}(r) = 4\epsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right] \tag{5.2}$$

where ϵ and σ are energy and range parameters, respectively. Attractive bead-bead interactions are given by the cut and shifted potential $u_\mathrm{att}(r \leq r_c) = u_\mathrm{LJ}(r) - u_\mathrm{LJ}(r_c)$ and $u_\mathrm{att}(r > r_c) = 0$, with $r_\mathrm{c} = 2.5\sigma$. In a similar way, the repulsive bead-bead potential $u_\mathrm{rep}(r)$ is defined by the Weeks-Chandler-Andersen potential [173], $u_\mathrm{rep}(r \leq r_0) = u_\mathrm{LJ}(r) - u_\mathrm{LJ}(r_0)$ and $u_\mathrm{rep}(r > r_0) = 0$, where $r_0 = \sqrt[6]{2}\,\sigma$ is at the minimum of $u_\mathrm{LJ}(r)$. In all cases, the AA, BB, CC, and BC interactions are attractive, and the AB interaction is repulsive. In a selective solvent, the AC interaction is repulsive, while in a non-selective solvent it is attractive. The surface was taken to be structureless, parallel to the xy plane, and with $z < 0$. The bead-surface interactions were

	Selective Solvent				Non-selective Solvent			
	A	B	C	S	A	B	C	S
A	$u_{\rm att}(r)$	$u_{\rm rep}(r)$	$u_{\rm rep}(r)$	$\phi_{\rm rep}(z)$	$u_{\rm att}(r)$	$u_{\rm rep}(r)$	$u_{\rm att}(r)$	$\phi_{\rm rep}(z)$
B	$u_{\rm rep}(r)$	$u_{\rm att}(r)$	$u_{\rm att}(r)$	$\phi_{\rm att}(z)$	$u_{\rm rep}(r)$	$u_{\rm att}(r)$	$u_{\rm att}(r)$	$\phi_{\rm att}(z)$
C	$u_{\rm rep}(r)$	$u_{\rm att}(r)$	$u_{\rm att}(r)$	$\phi_{\rm att}(z)$	$u_{\rm att}(r)$	$u_{\rm att}(r)$	$u_{\rm att}(r)$	$\phi_{\rm att}(z)$

Table 5.1: Table of interactions for beads of type A (solvophobic), B (solvophilic), C (solvent), and the surface, S. For interactions between beads A-C, $u_{\rm att}(r)$ is the Lennard-Jones (12,6) potential cut-and-shifted at $r = 2.5\sigma$, and $u_{\rm rep}(r)$ is the Lennard-Jones (12,6) potential cut-and-shifted at the minimum $r = \sqrt[6]{2}\,\sigma$ (the WCA potential). For interactions involving the surface, $\phi_{\rm att}(z)$ is the full Lennard-Jones (9,3) potential, and $\phi_{\rm rep}(z)$ the Lennard-Jones (9,3) potential cut-and-shifted at the minimum $z = \sqrt[6]{2/5}\,\sigma$.

dealt with through an effective potential [73] based on integrating the LJ interactions with a homogeneous distribution of sites within the surface. This is given by

$$\phi(z) = \frac{2\pi\epsilon_{\rm s}}{3}\left[\frac{2}{15}\left(\frac{\sigma}{z}\right)^9 - \left(\frac{\sigma}{z}\right)^3\right] \tag{5.3}$$

where z is the perpendicular distance of the bead from the surface and $\epsilon_{\rm s}$ controls the strength of the bead surface attraction. The attractive bead-surface potential is given by $\phi_{\rm att}(z) = \phi(z)$ while the repulsive bead-surface potential is given by $\phi_{\rm rep}(z \le z_0) = \phi(z) - \phi(z_0)$ and $\phi_{\rm rep}(z > z_0) = 0$, where the cut-off $z_0 = \sqrt[6]{2/5}\,\sigma$ is at the minimum of $\phi(z)$. In all cases, the interaction between A beads and the surface is repulsive, while B beads and C beads experience attractive interactions with the surface. The interaction potentials for selective and non-selective solvents are summarised in Table 5.1. The only difference between the two solvents is the interaction between A beads and C beads: in a selective solvent, this interaction is repulsive; in a non-selective solvent, this interaction is attractive.

Each polymer consisted of $N_{\rm b} = 50$ beads. Polymers with formula $A_{12}B_{38}$ and monomer fractions $x_{\rm A} = 0.24$ and $x_{\rm B} = 0.76$ approximate the polymers studied in ref. [145] with $x_{\rm PI} = 0.21$ and $x_{\rm PEO} = 0.79$. Two more molecular architectures were studied, with formulas $A_{25}B_{25}$ and $A_{38}B_{12}$. For simplicity, $\epsilon_{\rm s}$ was set equal to ϵ. The FENE parameters were $R_0 = 1.5\sigma$ and $k = 30\epsilon\sigma^{-2}$. MD simulations were performed in the NVT ensemble using a chain of Nośe-Hoover thermostats, as described by Martyna *et al.* [97] The equations of motion were integrated using the velocity-Verlet scheme [98] with timestep $\delta t/\tau = 0.002$, where $\tau = \sqrt{m\sigma^2/\epsilon}$ is the basic unit of time. In all cases, the target temperature was $T^* = k_{\rm B}T/\epsilon = 0.8$. This is below the vapour-liquid critical temperature of a bulk atomic fluid with interaction potential $u_{\rm att}(r)$. The critical density is $\rho_{\rm c}\sigma^3 = 0.3211(5)$ and the critical temperature is $T_{\rm c}^* = 1.0795(2)$ [174].

5.2.2 Simulation protocol

All simulations were carried out with $N_{\rm p} = 100$ polymers (giving a total number of polymer beads $N_{\rm A} + N_{\rm B} = 5000$) and $N_{\rm C} = 27768$ solvent beads in an $L \times L \times H$ cuboidal box. Periodic boundary conditions (PBCs) were applied in the x and y directions. All bead interactions with the top surface of the box ($z > H$) were made repulsive. Following a high-temperature randomisation at $T^* = 2$, the temperature was reduced to $T^* = 0.8$. At $T^* = 0.8$ the bulk coexistence densities for the vapour and liquid phases are $\rho\sigma^3 \simeq 0.029$ and 0.73, respectively [175, 176]. L was made greater than the total length of the polymers to simplify the initial crystalline packing of the system and to avoid PBC artefacts. Initial values of $L = 70\sigma$ and

$H = 27.5\sigma$ were chosen to give a liquid layer of sufficient thickness to solvate the polymers, in equilibrium with a thick layer of vapour to eliminate the effects of the box boundary. The total density of the solvent beads was $\rho\sigma^3 = 0.206$ and, from the lever rule, this gives liquid and vapour layers of thickness approximately 7σ and 20σ, respectively.

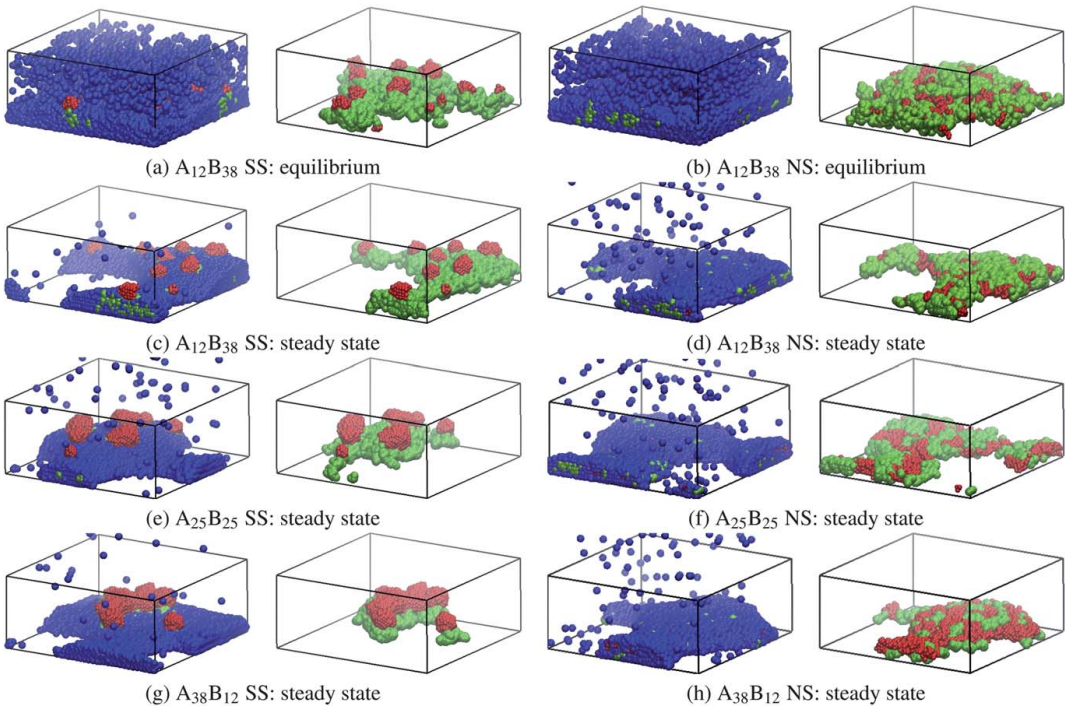


Figure 5.3: Snapshots from various stages of the simulations of polymers in a selective solvent (SS) (a, c, e and g) and non-selective solvent (NS) (b, d, f and h) at $T^* = 0.8$: (a and b) during equilibrium; (c-h) steady-state conditions after evaporation. A beads are red, B beads are green, and C beads (solvent) are blue. In each case, snapshots are shown with and without solvent included. The box dimensions are those during the equilibrium stages of the simulations.

For each polymer type, the simulation consisted of four distinct stages. (a) (High-temperature randomisation) Initially, a short run ($\sim 7 \times 10^4$ time steps) at a high temperature of $T^* = 2$ was carried out in order to reduce any artefacts introduced by the initial configuration. (b) (Equilibrium) The system was equilibrated at $T^* = 0.8$ over a long period of $\sim 2 \times 10^6$ time steps ($\sim 4000\tau$). (c) (Evaporation) Evaporation was carried out by increasing H incrementally at a rate of 0.05% per 100 time steps over $\sim 2 \times 10^5$ time steps ($\sim 400\tau$). (d) (Steady state) The system was then maintained under constant conditions until an apparent steady-state structure was reached. Simulation snapshots from the equilibrium and steady-state stages of simulations of $A_{12}B_{38}$ polymers in selective and non-selective solvents are shown in Fig. 5.3. Although the simulated molecules and system dimensions are small compared to those in experiments, they should give a qualitative picture of the dominant molecular-scale processes.

5.3 Results

This section is organised by system characteristic (such as film height or radius of gyration) and results for all systems studied will be presented in each section.

5.3.1 Equilibrium density profile

Fig. 5.4 shows the local density of each bead type at equilibrium, prior to solvent evaporation. In each case, the solvent forms a liquid-like layer near the surface. The local solvent density shows strong layering near the surface, and with increasing distance z shows damped oscillations around an average density of about $\rho\sigma^3 \simeq 0.6$, before falling rapidly as the liquid-vapour interface is traversed. By fitting a simple interfacial profile

$$\rho(z) = \frac{1}{2}(\rho_{\rm vap} + \rho_{\rm liq}) + \frac{1}{2}(\rho_{\rm vap} - \rho_{\rm liq}) \tanh\left(\frac{z - z_{\rm i}}{\xi}\right) \tag{5.4}$$

which ignores the oscillations near to the surface, rough estimates can be obtained for the average vapour and liquid densities $\rho_{\rm vap}$ and $\rho_{\rm liq}$, and the position ($z_{\rm i}$) and width (ξ) of the interface. These are given in Table 5.2 and the fits are shown in Fig. 5.4

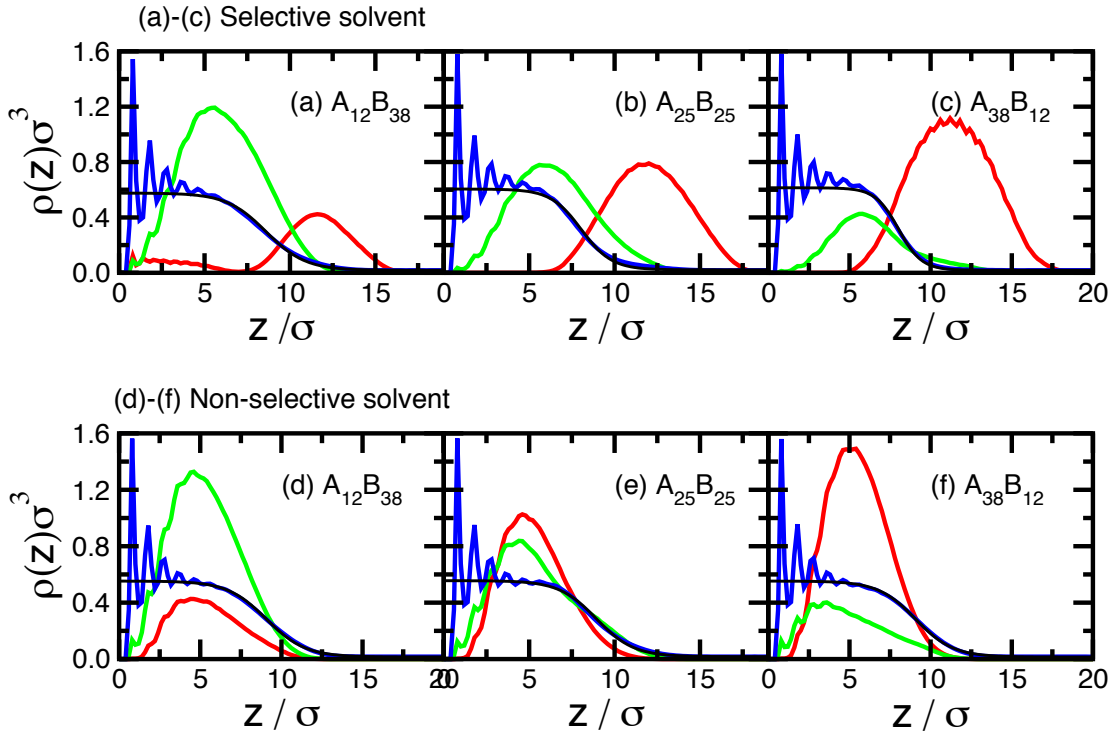


Figure 5.4: Local density $\rho(z)$ during the equilibrium stage for A beads (red), B beads (green), and solvent (C) beads (blue). The densities $\rho_{\rm A}(z)$ and $\rho_{\rm B}(z)$ have been multiplied by 10 for clarity. The thin black lines are fits to the solvent density using eqn (5.4). Results are shown for a selective solvent (a-c) and for a non-selective solvent (d-f).

The apparent vapour and liquid densities are different from their bulk values due to the presence of the polymers and the surfaces. $z_{\rm i}$ can be identified with the thickness of the liquid-solvent layer, which is around $8 - 9\sigma$ in each case, and slightly higher than that expected from the lever rule due to the solvation of the polymers. It is important to note that ξ is the apparent interfacial width only for the specific system size being considered. From simulations of polymer mixtures in slit-pores, Werner *et al.* found very significant finite-size effects where the concentration profile varied strongly not only with the confined-film thickness, but also with the lateral dimension of the simulation cell (corresponding to L here) [177, 178].

System	$\rho_{\mathrm{vap}}\sigma^3$	$\rho_{\mathrm{liq}}\sigma^3$	z_{i}/σ	ξ/σ
$A_{12}B_{38}$/SS	0.018(15)	0.576(28)	8.60(33)	2.31(58)
$A_{25}B_{25}$/SS	0.021(15)	0.605(27)	8.05(28)	1.79(48)
$A_{38}B_{12}$/SS	0.019(15)	0.614(26)	8.02(24)	1.43(41)
$A_{12}B_{38}$/NS	0.015(15)	0.553(27)	9.07(35)	2.40(61)
$A_{25}B_{25}$/NS	0.018(15)	0.556(26)	8.93(33)	2.19(57)
$A_{38}B_{12}$/NS	0.015(15)	0.553(26)	9.07(34)	2.31(59)

Table 5.2: Estimates of the vapour-liquid interfacial properties of the solvent at equilibrium: ρvap and ρ_{liq} are the vapour and liquid densities, respectively, z_{i} is the position of the interface, and ξ is the interfacial width, all as fitted by eqn 5.4. SS and NS denote selective and non-selective solvents, respectively

The A and B beads are distributed according to the selectivity of the solvent. With a selective solvent, the solvophilic B beads are located preferentially near the surface, while the solvophobic A beads are expelled from the liquid layer. With a non-selective solvent, the A and B density profiles are similar, with peak positions in the region of $z = 5\sigma$. The B beads show a little more structure near the surface due to the attractive interactions with the surface. Snapshots of $A_{12}B_{38}$ polymers equilibrated in selective and non-selective solvents are shown in Figs. 5.3(a) and (b), respectively. In selective solvent, the polymers form structures where the solvophobic A beads are clustered together to form caps that float on the islands formed by the B beads. In non-selective solvent, the A beads are still clustered, but these clusters are dispersed within the layer of B beads. The repulsive interactions between A beads and selective-solvent beads lead to a greater degree of clustering and protrusion from the liquid layer.

The properties of the solvent layer reported in Table 5.2 vary weakly but systematically amongst the different cases. For a given polymer, the liquid layer is thicker, the interface is broader, and the liquid density is lower for a non-selective solvent than for a selective solvent. This is due to the full solvation of the polymer in the non-selective solvent layer. With a non-selective solvent, the properties of the liquid layer are not strongly dependent on the polymer. With a selective solvent, the liquid layer gets thicker, the interface gets broader, and the liquid density decreases as the proportion of solvophilic polymer beads is increased, reflecting the greater degree of polymer solvation.

5.3.2 Film height

Following the equilibration stage, the solvent is evaporated by extending the box in the z direction causing dewetting of the surface and a collapse of the polymers. This section will focus on the effects of this process on the polymers. The collapse of the polymer film is reflected in the average bead height h, and those resolved in to different types of polymer beads. Figure 5.5 show the time dependence of h through the equilibrium, evaporation, and steady-state stages of the simulations.

Figures 5.5(a)-(c) show the results for polymers in a selective solvent. In all cases, and throughout each stage of the simulation, the solvophobic A beads are further from the surface than the B beads. The average height of all beads is a simple weighted average of the A and B bead heights. Upon solvent evaporation, the bead heights drop immediately and rapidly due to thinning of the uppermost part of the liquid-solvent film. Significantly, on a longer time scale, the bead heights increase with time. This is most clear for $A_{12}B_{38}$, in which h shows a rather

rapid increase at around $t \simeq 7000\tau$. This coincides with the solvent beginning to dewet the surface which appears to cause the polymers to pile up rather than extend across the surface and can be seen as similar to the exponential increase in heights seen in experiments [145]. This effect is a small one in the simulation and with longer simulation runs this effect may have been clearer as the trend is still slightly upwards when the simulations end. To make this clearer the figure has been included on a full page. This is still a subtle effect but the upward trend is noticeable.

In each case, h eventually approaches a steady-state value. For each polymer, the average steady-state heights of the A and B beads are 9-10σ and 4-5σ, respectively. The steady-state structures of all polymers in selective solvents are shown in Fig. 5.3(c), (e) and (g). For $A_{12}B_{38}$ polymers, a comparison of Fig. 5.3(a) and (c) shows the overall flattening of the B beads on to the surface when the solvent has evaporated and is no longer fully wetting the surface. Figures 5.3(c), (e) and (g) show that, irrespective of the composition, the A beads form clustered caps that sit on top of the B beads. Figures 5.5(d)-(f) show that, in a non-selective solvent, the A and B beads are dispersed evenly through the polymer layer at all times. At the start of solvent evaporation, h drops due to the depletion of the topmost part the liquid-solvent layer, but then increases again as further solvent dewetting causes beads to pile up. A comparison of fig. 5.3(b) and (d) shows the overall flattening of $A_{12}B_{38}$ accompanying the onset of solvent evaporation. Figure 5.3(d), (f) and (h) show that the A beads are clustered, but that the clusters remain dispersed within the islands of B beads on the surface. The steady-state structures shown in fig 5.3 (d), (f) and (h) are qualitatively similar to those at the equilibrium stage.

5.3.3 Radius of Gyration

The dimension of a single polymer is indicated by the radius of gyration R_{g}. A two-dimensional radius of gyration is defined as

$$R_{\mathrm{g}}^2 = \left\langle \frac{1}{N^2} \sum_{i}^{N} \sum_{j}^{N} |x_i - x_j|^2 + |y_i - y_j|^2 \right\rangle \tag{5.5}$$

with a three-dimensional radius of gyration found by including a term for the z dimension. In both the two and three-dimensional case, the sum was either over all A beads in the polymer, all B beads in the polymer, or all beads in the polymer. The two-dimensional quantity gives an indication of the lateral spread of the molecules with respect to the surface (xy plane) with the three-dimensional quantity clarifying the overall extension of the polymer. In this case, the three-dimensional graphs, shown in fig. 5.6, closely matches the track of the two-dimensional graph, shown in fig. 5.7, implying that the extension in the z-direction does not play a large role in the calculation of the radius of gyration and the simpler two-dimensional graph contains all the important information about the system.

The time dependence of the two-dimensional R_{g} is shown in Fig. 5.7. In all cases, the radii of gyration of A and B beads increase with the respective monomer fractions. For $A_{25}B_{25}$ in a selective solvent, R_{g} of the B beads is slightly higher than that of the A beads during the equilibrium stage, presumably due to the attractive interaction between B beads and the surface, and a flattening on the surface. In non-selective solvent the A and B beads of $A_{25}B_{25}$ exhibit essentially the same radii.

Generally, for both types of solvent, the onset of solvent evaporation leads to a rapid decrease

in R_g for the majority bead type, while R_g for the minority bead type stays roughly constant. For example, in $A_{12}B_{38}$, the B beads contract laterally, shown by the reduction in the 2D radius of gyration for these beads, while the A beads - which are already clustered and perched on top of the B beads - remain much as they were. This lateral contraction of the B beads corresponds to a coiling of the B bead polymer. In combination with the simultaneous and rapid reduction in film height, these data indicate an overall collapse of the polymers on to the surface. On longer time scales, there is a slow decrease which correlates with the piling up effect seen in the bead heights. Overall, the picture is that the conformations of individual polymers undergo rapid collapse at the onset of solvent evaporation, and afterwards decrease towards slowly steady-state values as further solvent dewetting occurs.

5.3.4 Characteristic Length

So far, the picture is that solvent evaporation leads to an initial rapid collapse of the polymers, followed by a slow lateral contraction coinciding with a piling up as more of the solvent evaporates. The simulation snapshots in Fig. 5.3 show that the polymer molecules aggregate. Coarsening in the surface plane is monitored by a characteristic length l given by [179]

$$l = \frac{2\pi}{\langle k \rangle} = \frac{2\pi \sum_k S(\vec{k})/g(k)}{\sum_k kS(\vec{k})/g(k)} \tag{5.6}$$

where $\vec{k} = 2\pi(nx, ny)/L \quad (nx, ny = 0, \pm 1, \pm 2, ...)$ is a two-dimensional wave-vector, $k = |\vec{k}|, S(\vec{k})$ is the structure factor, and $g(k)$ is the number of wave-vectors with length k. $S(k)$ is given by

$$S(\vec{k}) = \rho(\vec{k})\rho(-\vec{k}) \tag{5.7}$$

where $\rho(\vec{k}) = \sum_{i=1}^{N} \exp(-i\vec{k} \cdot \vec{r}_i)$ is an instantaneous Fourier component of the particle density. A high-k cut off of $2\pi/\sigma$ was applied to the sum over wave-vectors. This calculation defines a length scale of the structures formed by the polymers. Characteristic lengths were computed separately for the A beads, the B beads, and the A and B beads combined; the results are shown in Fig. 5.8. The characteristic length for a given bead type increases with its monomer fraction. In non-selective solvents there is a slight downward drift in l during the equilibration stage, reflecting very slow collective relaxation to equilibrium. Upon solvent evaporation, l increases quite sharply reflecting a growing length scale in the xy plane. This signals that low wave-vector density correlations in the polymer film are growing as the solvent evaporates and dewets the surface. Two processes may contribute to this effect: the agglomeration of loosely associated polymers in the film and an overall increase in the local concentration of beads within the polymers. For a given bead type, the characteristic lengths in selective and non-selective solvents are very similar to one another. There was a fluctuation in l for the A beads in the $A_{38}B_{12}$ polymers in selective solvent [fig. 5.8(c)] at $t \simeq 9000\tau$. This coincides with slight jumps seen at the same time in R_g. The bead height, h, shows no fluctuation. These features coincided with the onset of rapid solvent evaporation, and the ultimate steady-state values appear to have been unaffected.

5.3.5 Solvent evaporation, dewetting, and polymer restructuring

Figures 5.5-5.7 all show rapid responses in the structure of the polymer film at the onset of solvent evaporation, followed by slower variations as more and more solvent evaporates and eventually dewets the surface. Clearly, then, this two-stage process is tied to the evaporation mechanism of the solvent. The solvent molecules near the liquid-vapour interface begin to evaporate first, leading to a desolvation of the top-most parts of the polymers and an immediate reduction in the film height, as shown in fig.5.5. At a later stage, the solvent dewets the surface, apparently due to a heterogeneous nucleation effect since the dewetting of the surface originates on the perimeters of polymer clusters, and then spreads out. This late-stage desolvation of the polymers leads to an increase in the height h (fig.5.5) and a decrease in the lateral radius of gyration R_{g} (fig.5.7). The overall picture is that, upon solvent evaporation, the polymers first flatten rapidly, and then slowly contract laterally and increase in height. The steady-state snapshots in fig.5.3(c)-(h) show that there is a residual film of solvent on the polymers, but that the bare surface is dewetted. This is only a monolayer, however, and this appears to have no further bearing on the development of the polymer structure.

5.4 Conclusion

Computer simulations have been used to study the deposition of amphiphilic diblock copolymers on a smooth surface driven by solvent evaporation. Coarse-grained models of the polymers were constructed to reflect different ratios of solvophilic (and surfacephilic) and solvophobic (and surfacephobic) components, including a ratio studied experimentally [145]. In contrast with earlier simulations of similar systems, the solvent was modelled explicitly, allowing an investigation of solvent quality (whether it is selective for one of the polymer components or non-selective), and the coupling between the solvent dewetting processes and polymer structure. The significant computational cost associated with the explicit solvent was offset by using a bespoke MD code written for GPUs.

The simulation protocol was designed to mimic the situation of the polymers solvated by a thin liquid film in equilibrium with its vapour, followed by solvent evaporation and the approach to a steady state. The structure of the thin film was elucidated by examining individual density profiles of the solvent molecules, and the solvophilic and solvophobic components of the polymers. In general, the polymers form clusters with the surfacephilic groups providing a flat base on top of which the surfacephobic groups form clusters. The evolution of the polymer structure upon solvent evaporation was monitored by measuring molecular heights, radii of gyration, and a characteristic length which characterises coarsening of the structure. In general, the deposition mechanism upon solvent evaporation consists of two stages. The solvent molecules near the liquid-vapour interface are the first to evaporate, and this leads to an immediate flattening of the polymers. Next, the solvent dewets from the surface via a nucleation process originating near the perimeters of the polymer clusters. This leads to a lateral contraction and a slight increase in the polymer height. This lateral contraction is due to the loss of solvent surrounding the polymer islands. All of this supports the general mechanisms put forward in ref [145]. These state that the evaporation of the solvent from the surface of the mica leads to the solvophilic parts of the polymer to maximise the contact with the remaining solvent trapped beneath the polymers. This leads to a reduction in their extent across the surface of the mica and raises the height of the polymer islands. In this movement of

the polymers, it is the polymers that drive the structures formed by trapping solvent beneath them.

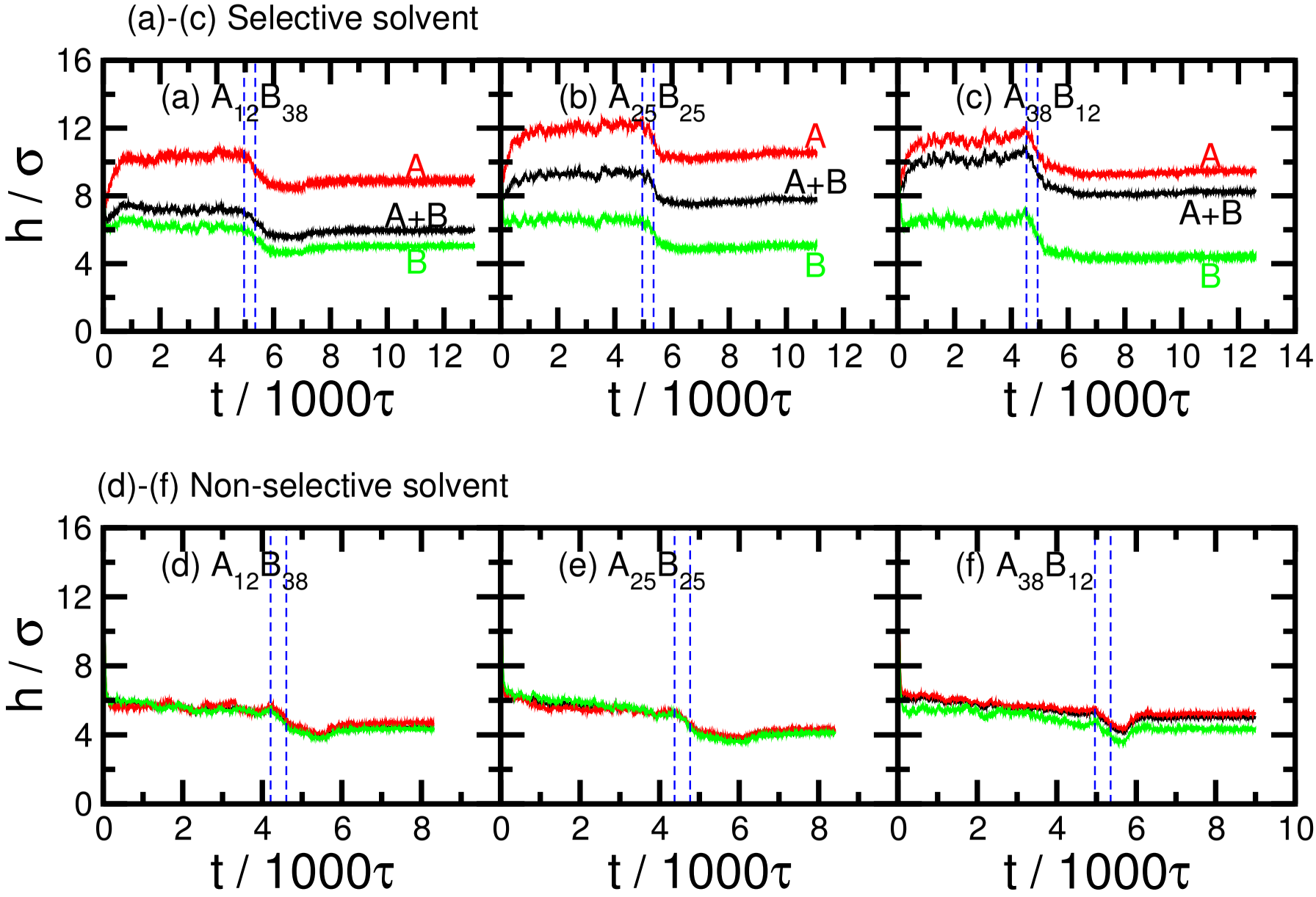


Figure 5.5: Average bead height h for A beads (red), B beads (green), and all A and B beads (black). Results are shown for a selective solvent (a-c) and for a non-selective solvent (d-f). The blue dashed lines indicate the beginning and end of the solvent-evaporation stage.

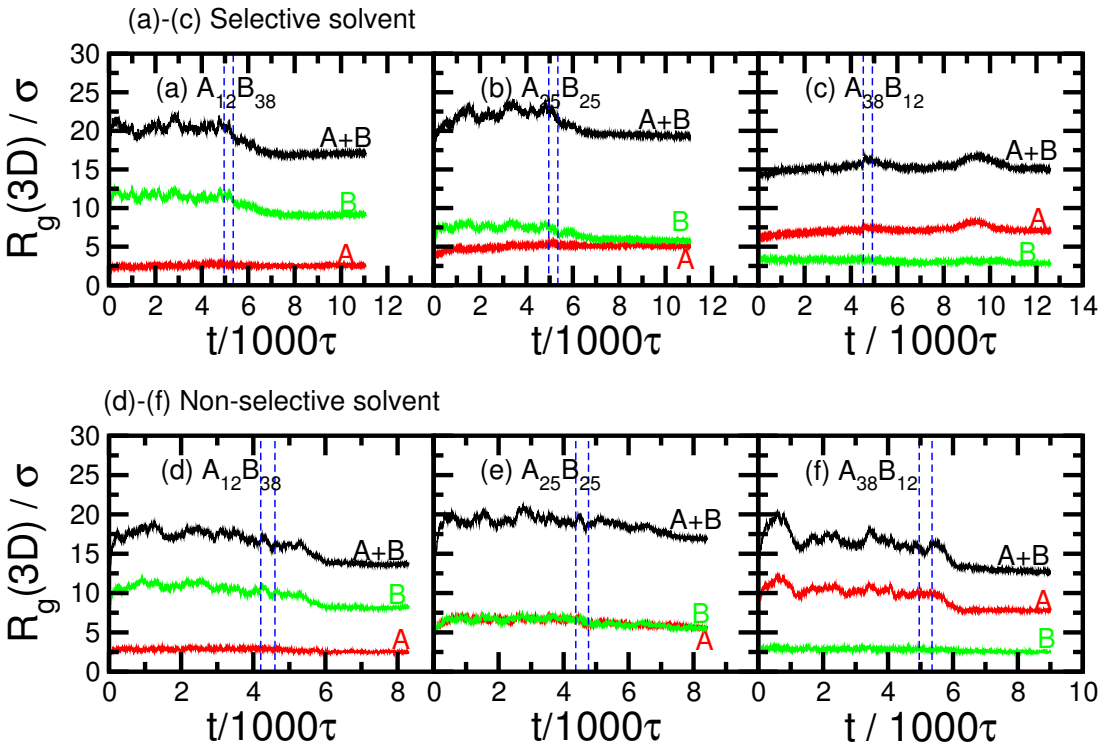


Figure 5.6: Three-dimensional radius of gyration R_g for A beads (red), B beads (green), and all A and B beads (black). Results are shown for a selective solvent (a-c) and for a non-selective solvent (d-f). The blue dashed lines indicate the beginning and end of the solvent-evaporation stage.

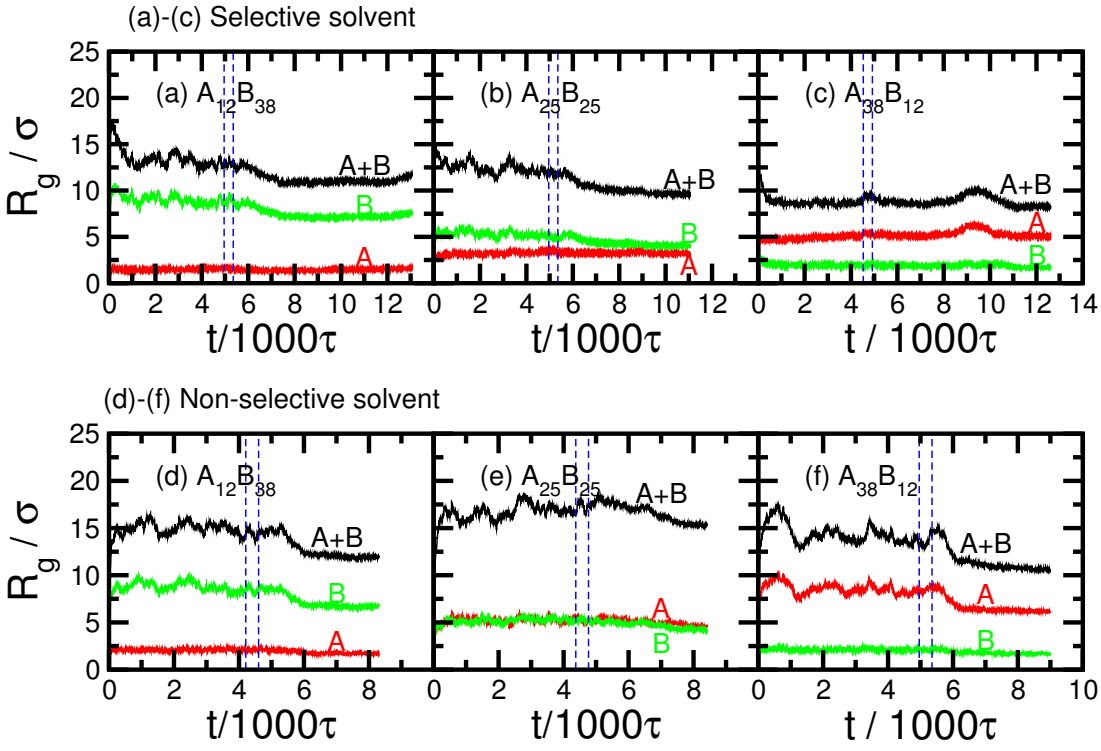


Figure 5.7: Two-dimensional radius of gyration R_g for A beads (red), B beads (green), and all A and B beads (black). Results are shown for a selective solvent (a-c) and for a non-selective solvent (d-f). The blue dashed lines indicate the beginning and end of the solvent-evaporation stage.

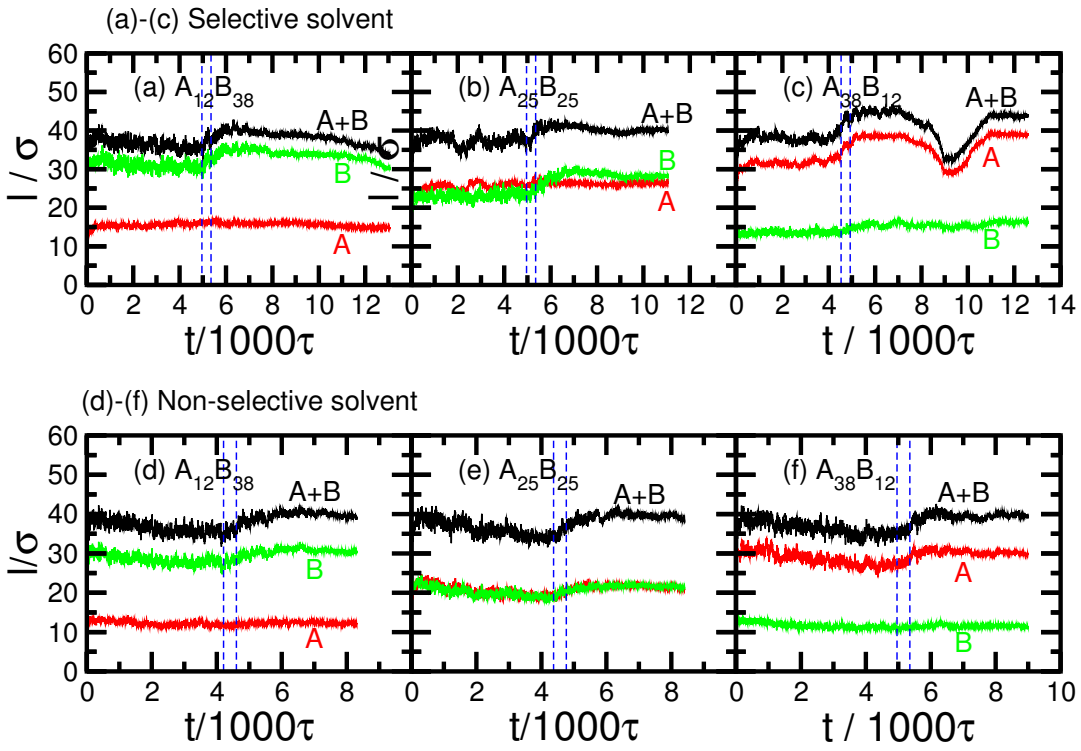


Figure 5.8: characteristic lengths

Chapter 6

Conclusions and perspectives

This final chapter sums up the work presented in the previous results chapters. The work presented here represents a fairly disparate body of work with little connecting one project to the other. As such this chapter will simply summarise the work which has been carried out, outline improvements and the future work which has begun.

The first results chapter reports on published work [1] describing a simple microscopic model of ferrogels using MC simulations. The model represents the material as a dispersion of dipolar hard spheres trapped in an elastic matrix. In this preliminary study, a portion of ferrogel was restricted to be orthorhombic and only allowed to undergo affine deformations. Despite the simplicity of the model, it captures much of the phenomenology from experimental studies. Specifically, the elastic moduli of ferrogels can be tuned by applying uniform magnetic fields during the gel formation stage. The application of the field gives rise to a subtle enhancement of positional correlations between magnetic grains and a pronounced anisotropy to the material properties. A very significant enhancement of the elastic modulus can then be observed if the ferrogel is then placed in a parallel uniform field with less pronounced effects occurring with other arrangements of fields during and after the synthesis stage. It has been demonstrated using MC simulations that the model ferrogel mimics experimental trends with quite reliably, while providing specific insights on the microscopic arrangements of the magnetic grains. The experimental trends reproduced are those regarding the changes in rheology of a ferrogel in a uniform magnetic field. These trends were shown in the paper by Varga *et al.*[109] and showed that the rheology of the ferrogel could be changed by manufacturing the gel in the presence of a magnetic field. The simulations reproduced this change in elastic modulus and showed that the cause of this is the structure imposed before polymerisation while the dipolar particles are in a dipolar fluid. While in a dipolar fluid, the magnetic particles form chains with the dipole moments 'nose-to-tail' and move close to contact. This close structure gives them little space to move around each other stiffening the gel with the interactions of the dipoles adding to this effect.

In addition to these insights, the simulation results suggest a particularly simple relationship between the elastic moduli and the magnetisation. The moduli in the direction of the field applied during production was found to be given by

$$G_x(M_x) = G_x(0) + \Delta G_x \left(\frac{M_x}{N\mu}\right)^2 \tag{6.1}$$

where $0 \leq M_x/N\mu \leq 1$ is a dimensionless, fractional measure of the magnetisation and $G_x(0)$ and ΔG_x are fitting parameters. That there should be a relationship between the elastic moduli and the magnetisation which was an even function is easily explained by symmetry considerations. A further relationship for the orthogonal axis to the applied field was found to be

$$G_z(M_x) = G_z(0) + \Delta G_z \left(\frac{M_x}{N\mu}\right)^4 . \tag{6.2}$$

where again $G_z(0)$ and ΔG_z are fitting parameters. This relationship was found heuristically and an explanation for this is still being sought. When supplemented by a magnetization curve, these relationships lead to the dependence of the elastic moduli on the magnetic field strength, which is normally the property measured directly in experiments. This gives a direct connection between the simulations performed and the production of ferrogels with desired properties.

A number of improvements to the model can be envisaged. An extension of the model to the case of paramagnetic particles should be attempted. Although ferrogels are often made with nanometer sized ferromagnetic or intrinsically superparamagnetic particles, some more recent applications may require materials made from larger particles that do not possess single magnetic domains.

Another feature of real ferrogels is the polydispersity of the magnetic grain sizes and dipole moments. In other contexts, magnetic polydispersity has been shown to have quite significant effects [122]. The model might also be extended to take in to account shear deformations and non-affine deformations. An explicit model of the cross-linked polymer gel component would help accommodate local non-affine displacements of the magnetic grains within a simulation box of triclinic geometry.

Finally, some comment should be made on shape-dependence. The assumptions of an infinite system and the inability to fully simulate effects of the demagnetising field mean that limited value can be placed on the deformation measurements of this study. While the rheology changes and magnitudes of the changes in box length match experiment, the lack of a way to account for the demagnetising field mean that little could be stated about the direction of changes in size of a finite macroscopic sample. It would be desirable to be able to simulate either a system of significant enough size to enable the effect of a demagnetising field to be included. This is currently far beyond current technical ability leaving the only approach to find alternative methods which capture this effect, attempts at which have been made [13].

Despite these shortcomings and the simplicity of the current model, good agreement with experiment has been achieved. This preliminary study, therefore represents a decent platform from which to explore further links between the microscopic and macroscopic properties of ferrogels.

The second results chapter reports on work previously published [2]. Computer simulations have been used to study the deposition of amphiphilic diblock copolymers on a smooth surface driven by solvent evaporation. Coarse-grained models of the polymers were constructed to reflect different ratios of solvophilic (and surfacephilic) and solvophobic (and surfacephobic) components, including a ratio studied experimentally [145]. In contrast with earlier simulations of similar systems, the solvent was modelled explicitly, allowing an investigation of solvent quality (whether it is selective for one of the polymer components or non-selective), and the coupling between the solvent dewetting processes and polymer structure. The dewetting process is an important part of the simulation as the experiment motivating this work was carried out

on a freshly cleaved mica surface. Mica is known to lose affinity to water over time and it is this which causes the solvent dewetting. The significant computational cost associated with the explicit solvent was offset by using a bespoke MD code written for GPUs.

The simulation protocol was designed to mimic the situation of the polymers solvated by a thin liquid film in equilibrium with its vapour, followed by solvent evaporation and the approach to a steady state. The structure of the thin film was elucidated by examining individual density profiles of the solvent molecules, and the solvophilic and solvophobic components of the polymers. In general, the polymers form clusters with the surfacephilic groups providing a flat base on top of which the surfacephobic groups form clusters. The evolution of the polymer structure upon solvent evaporation was monitored by measuring molecular heights, radii of gyration, and a characteristic length which characterises coarsening of the structure. In general, the deposition mechanism upon solvent evaporation consists of two stages. The solvent molecules near the liquid-vapour interface are the first to evaporate, and this leads to an immediate flattening of the polymers. Next, the solvent dewets from the surface via a nucleation process originating near the perimeters of the polymer clusters. This leads to a lateral contraction and a slight increase in the polymer height. The overall picture supports the general mechanisms put forward in ref [145].

This is only a first attempt at simulating polymer deposition by solvent evaporation. There are at least three effects that have not been addressed with the coarse-grained model and simulation protocol adopted here. Firstly, the system has been maintained at a constant temperature using an artificial thermostat, whereas in reality, a temperature gradient would be established in the polymer-solvent film due to evaporative cooling by the solvent. Secondly, the coarse-grained model is a very crude representation of the various interactions between polymer, surface, and solvent. There are several specific, chemical details that may play significant roles in the polymer deposition process, including the structure and hydration of the surface, the cause of solvent evaporation from the surface, and the precise changes in interactions as the polymers crossover from good-solvent to bad-solvent conditions upon solvent evaporation. Thirdly, given the limitations on simulation length scale and time scale, it has not been possible to survey the effects of varying the rate of solvent evaporation with respect to polymer relaxation rates: this could well be a parameter that influences the structure of the adsorbed polymer film.

Despite these limitations, and the fact that the simulations are unavoidably limited to short length scales and time scales, the results show how solvent evaporation can control the slow restructuring of diblock copolymers on a surface. The mechanisms uncovered show that it is the evaporation of the solvent from the mica surface that causes the hydrophilic polymers to move into the remaining solvent forcing the hydrophobic ends upwards. These are precisely the effects measured in experiments [145].

The work presented in this thesis comes from the completed projects. Further work has been carried out to modify the GPU code developed to deal with different types of polymer and magnetic swimmers. The first extension of the code has been carried out to generate and simulate polymers with more complicated structures than simple block-copolymers. Specifically, attention has been turned to bottle brush polymers[180] and the GPU code is in the process of being modified to be able to simulate collections of these polymers with charged sections in an ionic solvent. A prototype of this code has been developed and it remains to be optimised to be able to simulate a large enough system to generate results of interest. In addition to this,

work has been carried out to simulate the behaviour of magnetic swimmers [181] produced by an experimental group in Barcelona. This work has recently passed the proof of concept stage and will form part of a PhD to be carried out in Barcelona.

Bibliography

[1] D. S. Wood and P. J. Camp. Modeling the properties of ferrogels in uniform magnetic fields. *Phys. Rev. E*, 83(1):011402, 1 2011.

[2] Dean S. Wood, Vasileios Koutsos, and Philip J. Camp. Computer simulations of surface deposition of amphiphilic diblock copolymers driven by solvent evaporation. *Soft Matter*, 9:3758, 2013.

[3] Takayuki Uekusa, Shusaku Nagano, and Takahiro Seki. Unique molecular orientation in a smectic liquid crystalline polymer film attained by surface-initiated graft polymerization. *Langmuir*, 23(8):4642–4645, 2007.

[4] A. R. Khokhlov and A. N. Semenov. Liquid-crystalline ordering in the solution of long persistent chains. *Physica A: Statistical Mechanics and its Applications*, 108(2):546–556, 1981.

[5] J. David Carlson and Mark R. Jolly. Mr fluid, foam and elastomer devices. *Mechatronics*, 10(4):555–569, 2000.

[6] Yadong Wang, Guillermo A. Ameer, Barbara J. Sheppard, and Robert Langer. A tough biodegradable elastomer. *Nature biotechnology*, 20(6):602–606, 2002.

[7] Soft condensed matter physics in molecular and cell biology, 2006. http://site.ebrary.com/id/10143493, Online; accessed 27 May 2013

[8] A. P. Philipse and D. Maas. Magnetic colloids from magnetotactic bacteria: chain formation and colloidal stability. *Langmuir*, 18(25):9977–9984, 2002.

[9] Jure Dobnikar, Alexey Snezhko, and Anand Yethiraj. Emergent colloidal dynamics in electromagnetic fields. *Soft Matter*, 2013.

[10] Teemu Murtola, Alex Bunker, Ilpo Vattulainen, Markus Deserno, and Mikko Karttunen. Multiscale modeling of emergent materials: biological and soft matter. *Phys Chem Chem Phys*, 11(12):1869–92, 3 2009.

[11] S. Poblete, M. Praprotnik, K. Kremer, and L. Delle Site. Coupling different levels of resolution in molecular simulations. *The Journal of Chemical Physics*, 132, 3 2010.

[12] Matej Praprotnik, Luigi Delle Site, and Kurt Kremer. Multiscale simulation of soft matter: from scale bridging to adaptive resolution. *Annu Rev Phys Chem*, 59:545–71, 2008.

[13] Oleg V. Stolbov, Yuriy L. Raikher, and Maria Balasoiu. Modelling of magnetodipolar striction in soft magnetic elastomers. *Soft Matter*, 7(18):8484–8487, 2011.

[14] Hai Lin and Donald G. Truhlar. Qm/mm: what have we learned, where are we, and where do we go from here? *Theoretical Chemistry Accounts*, 117(2):185–199, 2007.

[15] Y. Efendiev, V. Ginting, T. Hou, and R. Ewing. Accurate multiscale finite element methods for two-phase flow simulations. *Journal of Computational Physics*, 220(1):155–174, 2006.

[16] L. H. Baekeland. The synthesis, constitution, and uses of bakelite. *Industrial & Engineering Chemistry*, 1(3):149–161, 1909.

[17] Li Yan Qiu and You Han Bae. Polymer architecture and drug delivery. *Pharmaceutical research*, 23(1):1–30, 2006.

[18] Kell Mortensen and Wyn Brown. Poly (ethylene oxide)-poly (propylene oxide)-poly (ethylene oxide) triblock copolymers in aqueous solution. the influence of relative block size. *Macromolecules*, 26(16):4128–4135, 1993.

[19] Benjamin Chu. Structure and dynamics of block copolymer colloids. *Langmuir*, 11(2):414–421, 1995.

[20] Junchai Zhao, Shizhe Tian, Qiang Wang, Xiaobo Liu, Shichun Jiang, Xiangling Ji, Lijia An, and Bingzheng Jiang. Nanoscopic surface patterns of diblock copolymer thin films. *Eur Phys J E Soft Matter*, 16(1):49–56, 1 2005.

[21] J. P. Spatz, M. Möller, M. Noeske, R. J. Behm, and M. Pietralla. Nanomosaic surfaces by lateral phase separation of a diblock copolymer. *Macromolecules*, 30(13):3874–3880, 1997.

[22] R. A. Segalman. Patterning with block copolymer thin films. *Materials Science and Engineering*, 48:191–226, 12 2005.

[23] Sergiy Minko, Anton Kiriy, Ganna Gorodyska, and Manfred Stamm. Single flexible hydrophobic polyelectrolyte molecules adsorbed on solid substrate: transition between a stretched chain, necklace-like conformation and a globule. *J Am Chem Soc*, 124(13):3218–9, 4 2002.

[24] Y. Roiter and S. Minko. Afm single molecule experiments at the solid-liquid interface: In sity conformation of adsorbed flexible polyelectrolyte chains. *Journal of the American Chemical Society*, 127:15688–25689, 2005.

[25] Jiro Kumaki and Takeji Hashimoto. Conformational change in an isolated single synthetic polymer chain on a mica surface observed by atomic force microscopy. *J Am Chem Soc*, 125(16):4907–17, 4 2003.

[26] M. Zrinyi, L. Barsi, and A. Buki. Deformation of ferrogels induced by nonuniform magnetic fields. *The Journal of Chemical Physics*, 104(21):8750–8756, 1996.

[27] M. Zrinyi, L. Barsi, and A. Buki. Ferrogel: a new magneto-controlled elastic medium. *Polymer Gels and Networks*, 5:415–427, 1997.

[28] P. I. C. Teixeira, J. M. Tavares, and M. M. Gama. The effect of dipolar forces on the structure and thermodynamics of classical fluids. *Journal of Physics: Condensed Matter*, 12:R411, 2000.

[29] D. Levesque and J. J. Weis. Orientational and structural order in strongly interacting dipolar hard spheres. *Physical Review E*, 49(6):5131–5140, 6 1994.

[30] J. J. Weis and D. Levesque. Ferroelectric phases of dipolar hard spheres. *Physical Review E*, 48(5):3728–3740, 11 1993.

[31] P. Jund, S. G. Kim, D. Tománek, and J. Hetherington. Stability and fragmentation of complex structures in ferrofluids. *Phys Rev Lett*, 74(15):3049–3052, 4 1995.

[32] S. C. McGrother and G. Jackson. Island of vapor-liquid coexistence in dipolar hard-core systems. *Phys Rev Lett*, 76(22):4183–4186, 5 1996.

[33] A. Gil-Villegas, S. C. McGrother, and G. Jackson. Chain and ring structures in smectic phases of molecules with transverse dipoles. *Chemical physics letters*, 269(5-6):441–447, 1997.

[34] R. D. Sanchez, J. Rivas, P. Vaqueiro, M. A. López-Quintela, and D. Caeiro. Particle size effects on magnetic properties of yttrium iron garnets prepared by a sol–gel method. *Journal of magnetism and magnetic materials*, 247(1):92–98, 2002.

[35] R. Das, A. Gupta, D. Kumar, S. H. Oh, S. J. Pennycook, and A. F. Hebard. Dipolar interactions and their influence on the critical single domain grain size of ni in layered ni/al2o3 composites. *Journal of Physics: Condensed Matter*, 20:385213, 2008.

[36] R. F. Butler and S. K. Banerjee. Theoretical single-domain grain size range in magnetite and titanomagnetite. *Journal of Geophysical Research*, 80(29):4049–4058, 1975.

[37] B. J. Alder and T. E. Wainwright. Studies in molecular dynamics. i. general method. *The Journal of Chemical Physics*, 31:459, 1959.

[38] N. Metropolis and S. Ulam. The monte carlo method. *Journal of the American statistical association*, 44(247):335–341, 1949.

[39] Hector:uk national supercomputing service.

[40] Top 500 supercomputer lists.

[41] M. J. Harris, G. Coombe, T. Scheuermann, and A. Lastra. Physically-based visual simulation on graphics hardware. In *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS conference on Graphics hardware*, pages 109–118. Eurographics Association, 2002.

[42] Top 500 Supercomputer List Nov 2012 http://www.top500.org/list/2012/11/, Online; accessed 1 Apr 2013

[43] Ananth Grama. *Introduction to parallel computing.* Addison-Wesley, Harlow, England ; New York, 2003.

[44] Gregory R. Andrews. *Foundations of multithreaded, parallel, and distributed programming.* Addison-Wesley, Reading, Mass., 2000.

[45] D. B. Kirk and W. H. Wen-mei. *Programming massively parallel processors: a hands-on approach.* Morgan Kaufmann, 2010.

[46] M. J. Flynn. Some computer organizations and their effectiveness. *Computers, IEEE Transactions on*, 100(9):948–960, 1972.

[47] J. Backus. Can programming be liberated from the von neumann style?: a functional style and its algebra of programs. *Communications of the ACM*, 21(8):613–641, 1978.

[48] Geoffrey Blake, Ronald G. Dreslinski, Trevor Mudge, and Krisztián Flautner. Evolution of thread-level parallelism in desktop applications. *SIGARCH Comput. Archit. News*, 38(3):302–313, 6 2010.

[49] Myrinet Overview http://www.myricom.com/scs/myrinet/overview/, Online; accessed 24 Jan 2013

[50] S. Plimpton. Fast parallel algorithms for short-range molecular dynamics. *Journal of Computational Physics*, 117(1):1–19, 1995.

[51] J. J. Dongarra, S. W. Otto, M. Snir, and D. Walker. An introduction to the mpi standard. *Communications of the ACM*, 1995.

[52] W. Gropp, E. Lusk, N. Doss, and A. Skjellum. A high-performance, portable implementation of the mpi message passing interface standard. *Parallel computing*, 22(6):789–828, 1996.

[53] Intel ark.

[54] W. C. Feng and T. Scogland. The green500 list: Year one. In *Parallel & Distributed Processing, 2009. IPDPS 2009. IEEE International Symposium on*, pages 1–7. IEEE, 2009.

[55] W. Feng and H. Lin. The green500 list: year two. In *Parallel & Distributed Processing, Workshops and Phd Forum (IPDPSW), 2010 IEEE International Symposium on*, pages 1–8. IEEE, 2010.

[56] S. Craven and P. Athanas. Examining the viability of fpga supercomputing. *EURASIP Journal on Embedded systems*, 2007(1):13–13, 2007.

[57] T. Takagi and T. Maruyama. Accelerating hmmer search using fpga. In *Field Programmable Logic and Applications, 2009. FPL 2009. International Conference on*, pages 332–337. IEEE, 2009.

[58] M. Chiu and M. C. Herbordt. Efficient particle-pair filtering for acceleration of molecular dynamics simulation. In *Field Programmable Logic and Applications, 2009. FPL 2009. International Conference on*, pages 345–352. IEEE, 2009.

[59] D. H. Jones, A. Powell, C. S. Bouganis, and P. Y. K. Cheung. Gpu versus fpga for high productivity computing. In *Field Programmable Logic and Applications (FPL), 2010 International Conference on*, pages 119–124. IEEE, 2010.

[60] N. Coorperation. Nvidia cuda c programming guide 4.0, 2011.

[61] GPU History and CUDA Programming Guide http://code.google.com/p/stanford-cs193g-sp2010/source/browse/trunk/lectures/lecture_2/gpu_history_and_cuda_programming_basics.pdf?r= Online; accessed 21 Aug 2011

[62] PCIe Base 3.0 Specification http://www.pcisig.com/members/downloads/specifications/pciexpress/PCI_E Online; accessed 25 Jan 2012

[63] J. Anderson, A. Keys, C. Phillips, T. Dac Nguyen, and S. Glotzer. Hoomd-blue, general-purpose many-body dynamics on the gpu. *Bulletin of the American Physical Society*, 55, 2010.

[64] V. W. Lee, C. Kim, J. Chhugani, M. Deisher, D. Kim, A. D. Nguyen, N. Satish, M. Smelyanskiy, S. Chennupaty, and P. Hammarlund. Debunking the 100x gpu vs. cpu myth: an evaluation of throughput computing on cpu and gpu. In *ACM SIGARCH Computer Architecture News*, volume 38, pages 451–460. ACM, 2010.

[65] Open MM project overview https://simtk.org/home/openmm, Online; accessed 15 Feb 2013

[66] B. J. Alder and TEf Wainwright. Phase transition for a hard sphere system. *The Journal of Chemical Physics*, 27(5):1208–1209, 1957.

[67] Glenn M. Torrie and John P. Valleau. Nonphysical sampling distributions in monte carlo free-energy estimation: Umbrella sampling. *Journal of Computational Physics*, 23(2):187–199, 1977.

[68] Kurt E. Sickafus, E. A. Kotomin, and Blas P. Uberuaga. Radiation effects in solids, 2007.

[69] W. Ebeling, G. E. Norman, A. A. Valuev, and I. A. Valuev. Quasiclassical theory and molecular dynamics of two-component nonideal plasmas. *Contributions to Plasma Physics*, 39(1-2):61–64, 1999.

[70] Hans Feldmeier, Konrad Bieler, and Jürgen Schnack. Fermionic molecular dynamics for ground states and collisions of nuclei. *Nuclear Physics A*, 586(3):493–532, 1995.

[71] Peter Ring and Peter Schuck. *The nuclear many-body problem*. Springer, 2005.

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

[73] Scott W. Sides, Gary S. Grest, and Mark J. Stevens. Large-scale simulation of adhesion dynamics for end-grafted polymers. *Macromolecules*, 35(2):566–573, 2002.

[74] A. P. Copestake and R. Evans. Charge ordering and the structure of ionic liquids: screened coulomb versus coulomb interionic potentials. *Journal of Physics C: Solid State Physics*, 15:4961–4974, 1982.

[75] R. A. Buckingham. The classical equation of state of gaseous helium, neon and argon. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 168(933):264–283, 1938.

[76] Philip M. Morse. Diatomic molecules according to the wave mechanics. ii. vibrational levels. *Physical Review*, 34(1):57, 1929.

[77] D. Frenkel and B. Smit. *Understanding Molecular Simulation*. Academic Press Inc, 2nd edition, 2002.

[78] Jean-Paul . P. Ryckaert, Giovanni Ciccotti, and Herman JC Berendsen. Numerical integration of the cartesian equations of motion of a system with constraints: molecular dynamics of¡ i¿ n-alkanes. *Journal of Computational Physics*, 23(3):327–341, 1977.

[79] Hans C. Andersen. Rattle: A velocity version of the shake algorithm for molecular dynamics calculations. *Journal of Computational Physics*, 52(1):24–34, 1983.

[80] Gary S. Grest and Kurt Kremer. Molecular dynamics simulation for polymers in the presence of a heat bath. *Phys. Rev. A*, 33(5):3628–3631, 5 1986.

[81] Gary S. Grest, Kurt Kremer, and T. A. Witten. Structure of many arm star polymers: a molecular dynamics simulation. *Macromolecules*, 20(6):1376–1383, 1987.

[82] Luis M. Varela, Manuel Garci a, and Vi ctor Mosquera. Exact mean-field theory of ionic solutions: non-debye screening. *Physics reports*, 382(1):1–111, 2003.

[83] G. Rajagopal and R. J. Needs. An optimized ewald method for long-ranged potentials. *Journal of Computational Physics*, 115(2):399–405, 1994.

[84] E. Yakub and C. Ronchi. An efficient method for computation of long-ranged coulomb forces in computer simulation of ionic fluids. *JOURNAL OF CHEMICAL PHYSICS*, 119(22), 12 2003.

[85] Josh Barnes and Piet Hut. A hierarchical 0 (n log iv) force-calculation algorithm. *nature*, 324:4, 1986.

[86] Peter J. Steinbach and Bernard R. Brooks. New spherical-cutoff methods for long-range forces in macromolecular simulation. *Journal of Computational Chemistry*, 15(7):667–683, 2004.

[87] A. Y. Toukmaji and J. A. Board Jr. Ewald summation techniques in perspective: a survey. *Computer Physics Communications*, 95:73–92, 1996.

[88] B. R. A. Nijboer and F. W. De Wette. On the calculation of lattice sums. *Physica*, 23(1):309–321, 1957.

[89] M. Widom and H. Zhang. Comment on "long-ranged orientational order in dipolar fluids". *Physical Review letters*, 74(13):2616, 1995.

[90] Phelim P. Boyle. Options: A monte carlo approach. *Journal of Financial Economics*, 4(3):323–338, 1977.

[91] Vasilios I. Manousiouthakis and Michael W. Deem. Strict detailed balance is unnecessary in monte carlo simulation. *The Journal of chemical physics*, 110:2753, 1999.

[92] Herman JC Berendsen, J. Pl M. Postma, Wilfred F. van Gunsteren, ARHJ DiNola, and J. R. Haak. Molecular dynamics with coupling to an external bath. *The Journal of chemical physics*, 81:3684, 1984.

[93] M. P. Allen and D. J. Tildesley. *Computer simulation of liquids*, volume 18. Oxford university press, 1989.

[94] S. Nosé. A molecular dynamics method for simulations in the canonical ensemble. *Molecular Physics*, 100:191–198, 11 1983.

[95] W. J. Hoover. Canonical dynamics: Equilibrium phase-space distributions. *Phys. Rev. A*, 31(3):1695–1697, 3 1985.

[96] Mark E. Tuckerman, Yi Liu, Giovanni Ciccotti, and Glenn J. Martyna. Non-hamiltonian molecular dynamics: Generalizing hamiltonian phase space principles to non-hamiltonian systems. *The Journal of Chemical Physics*, 115(4):1678–1702, 2001.

[97] D. J. Tobias G. J. Martyna and M. L. Klein. Constant pressure molecular dynamics algorithms. *Journal of Chemical Physics*, 101(4177), 5 1994.

[98] G. J. Martyna, M. E. Tuckerman, D. J. Tobias, and M. L. Klein. Explicit reversible integrators for extended systems dynamics. *Molecular Physics*, 87(5):1117–1157, 1996.

[99] S. Franchetti. Radial distribution functions in solid and liquid argon. *Il Nuovo Cimento B Series 11*, 26(2):507–521, 1975.

[100] R. Hernandez, A. Sarafian, D. Lopez, and C. Mijangos. Viscoelastic properties of poly(vinyl alcohol) hydrogels and ferrogels obtained through freezing – thawing cycles. *Polymer*, 46:5543–5549, 2004.

[101] D. Yoruk T. Caykara and S. Demirci. Preparation and characterization of poly(n-tert-butylacrylamide-co-acrylamide) ferrogel. *Journal of Applied Polymer Science*, 112:800–804, 2009.

[102] Z. Varga, G. Filipcsei, and M. Zrinyi. Electric and magnetic field-structured smart composites. *Macromolecular Symposia*, 227:123–134, 2005.

[103] M. Zrinyi, L. Barsi, D. Szabo, and H. G. Killian. Direct observation of abrupt shape transition in ferrogels induced by nonuniform magnetic field. *JOURNAL OF CHEMICAL PHYSICS*, 106(13), 4 1997.

[104] K. Zimmermann, V. A. Naletova, I. Zeidis, V. Bohm, and E. Kolev. Modelling of locomotion systems using deformable magnetizable media. *Journal of Physics: Condensed Matter*, 18(S2973), 9 2006.

[105] K. Zimmermann, V. A. Naletova, I. Zeidis, V. A. Turkov, E. Kolev, M. V. Lukashevich, and G. V. Stepanov. A deformable magnetizable worm in a magnetic field - a prototype of a mobile crawling robot. *Journal of Magnetism and Magnetic Materials*, 311:450–453, 12 2007.

[106] L. Lao and R. Ramanujan. Magnetic and hydrogel composite materials for hyperthermia applications. *Journal of Materials Science:Materials in Medicine*, 15:1061–1064, 2004.

[107] T-Y . Y. Liu, S-H . H. Hu, T-Y . Y. Liu, D-M . M. Liu, and S-Y . Y. Chen. Magnetic-sensitive behavior of intelligent ferrogels for controlled release of drug. *Langmuir*, 22:5974–5978, 5 2006.

[108] R. Ramanujan and L. Lao. The mechanical behavior of smart magnet-hydrogel composites. *Smart Materials and Structures*, 15:952–956, 2006.

[109] Z. Varga, G. Filipcsei, and M. Zrinyi. Magnetic field sensitive functional elastomers with tuneable elastic modulus. *Polymer*, 47:227–233, 2006.

[110] Yu L. Raikher and O. V. Stolbov. Deformation of an ellipsoidal ferrogel sample in a uniform magnetic field. *Journal of Applied Mechanics and Technical Physics*, 46(2):434–443, 2005.

[111] Yu L. Raikher, V. V. Rusakov, W. T. Coffey, and Yu P. Kalmykov. Dynamic susceptibilities of an assembly of dipolar particles in an elastic environment. *Physical Review E*, 63(031402), 2 2001.

[112] Yu L. Raikher and O. V. Stolbov. Magnetodeformational effect in ferrogel samples. *Journal of Magnetism and Magnetic Materials*, 258:477–479, 2003.

[113] Yu L. Raikher and O. V. Stolbov. Magnetodeformational effect in ferrogel objects. *Journal of Magnetism and Magnetic Materials*, 289:62–65, 2005.

[114] C. Gollwitzer, A. Turanov, M. Krekhova, G. Lattermann, I. Rehberg, and R. Richter. Measuring the deformation of a ferrogel sphere in a homogeneous magnetic field. *The Journal of Chemical Physics*, 128(164709), 4 2008.

[115] G. Filipcsei, I. Csetneki, A. Szilágyi, and M. Zrínyi. Magnetic field-responsive smart polymer composites. *Advanced Polymer Science*, 206:137–189, 2007.

[116] J. E. Martin, R. A. Anderson, D. Read, and G. Gulley. Magnetostriction of field-structured magnetoelastomers. *Phys. Rev. E*, 74(051507), 11 2006.

[117] Konstantin Morozov, Mark Shliomis, and Hiroshi Yamaguchi. Magnetic deformation of ferrogel bodies: Procrustes effect. *Phys Rev E Stat Nonlin Soft Matter Phys*, 79(4 Pt 1):040801, 4 2009.

[118] L. V. Nikitin, G. V. Stepanov, L. S. Mironova, and A. N. Samus. Properties of magnetoelastics synthesized in external magnetic field. *Journal of magnetism and magnetic materials*, 258:468–470, 2003.

[119] L. V. Nikitin, G. V. Stepanov, L. S. Mironova, and A. I. Gorbunov. Magnetodeformational effect and effect of shape memory in magnetoelastics. *Journal of magnetism and magnetic materials*, 272:2072–2073, 2004.

[120] L. V. Nikitin, D. G. Korolev, G. V. Stepanov, and L. S. Mironova. Experimental study of magnetoelastics. *Journal of Magnetism and Magnetic Materials*, 300(1):e234–e238, 2006.

[121] E. Jarkova, H. Pleiner, H. W Müller, and H. R. Brand. Hydrodynamics of isotropic ferrogels. *Physical Review E*, 68:041706, 2003.

[122] A. Ivanov, S. Kantorovich, E. Reznikov, C. Holm, A. Pshenichnikov, A. Lebedev, A. Chremos, and P. J. Camp. Magnetic properties of polydisperse ferrofluids: A critical comparison between experiment, theory, and computer simulation. *Physical Review E*, 75(061405), 2007.

[123] D. Frenkel and B. Smit. *Understanding molecular simulation: from algorithms to applications*. Academic Press, Inc., 1996.

[124] M. Parrinello and A. Rahman. Strain fluctuations and elastic constants. *The Journal of Chemical Physics*, 76(5):2662–2666, 1982.

[125] Z. Varga, G. Filipcsei, and M. Zrinyi. Smart composites with controlled anisotropy. *Polymer*, 46:7779–7787, 2005.

[126] A. O. Ivanov and O. B. Kuznetsova. Magnetic properties of dense ferrofluids: An influence of interparticle correlations. *Phys. Rev. E*, 64(4):041405, 9 2001.

[127] Atul Bharde, Debabrata Rautaray, Vipul Bansal, Absar Ahmad, Indranil Sarkar, Seikh Mohammad Yusuf, Milan Sanyal, and Murali Sastry. Extracellular biosynthesis of magnetite using fungi. *Small*, 2(1):135–41, 1 2006.

[128] F. Gazeau, F. Boué, E. Dubois, and R. Perzynski. Static and quasi-elastic small angle neutron scattering on biocompatible ionic ferrofluids: magnetic and hydrodynamic interactions. *Journal of Physics: Condensed Matter*, 15(15):S1305, 2003.

[129] Tamás Kristóf and István Szalai. Magnetic properties and structure of polydisperse ferrofluid models. *Phys Rev E Stat Nonlin Soft Matter Phys*, 68(4 Pt 1):041109, 10 2003.

[130] J. P. Huang, Z. W. Wang, and C. Holm. Computer simulations of the structure of colloidal ferrofluids. *Phys Rev E Stat Nonlin Soft Matter Phys*, 71(6 Pt 1):061203, 6 2005.

[131] A. O. Ivanov and O. B. Kuznetsova. Magnetogranulometric analysis of ferrocolloids: Second-order modified mean field theory. *Colloid Journal*, 68(4):430–440, 2006.

[132] G. B. Webber, E. J. Wanless, V. Bütün, S. P. Armes, and S. Biggs. Self-organized monolayer films of stimulus-responsive micelles. *Nano Letters*, 2(11):1307–1313, 2002.

[133] G. B. Webber, E. J. Wanless, S. P. Armes, Y. Tang, Y. Li, and S. Biggs. Nano-anemones: Stimulus-responsive copolymer-micelle surfaces. *Advanced Materials*, 16(20):1794–1798, 2004.

[134] Chen Xu, Xuefeng Fu, Michael Fryd, Song Xu, Bradford B. Wayland, Karen I. Winey, and Russell J. Composto. Reversible stimuli-responsive nanostructures assembled from amphiphilic block copolymers. *Nano Lett*, 6(2):282–7, 2 2006.

[135] J. Yoon C. Park and E. L. Thomas. Enabling nanotechnology with self assembled block copolymer patterns. *Polymer*, 44:6725–6760, 7 2003.

[136] O. K. C. Tsui. *Polymer thin films*, volume 1. World Scientific Pub Co Inc, 2008.

[137] H. L. Khor, Y. Kuan, H. Kukula, K. Tamada, W. Knoll, W, M. Moeller, and D. W. Hutmacher. Response of cells on surface-induced nanopatterns: Fibroblasts and mesenchymal progenitor cells. *Biomacromolecules*, 8(5):1530–1540, 2007.

[138] V. Koutsos, E. W. Van der Vegte, E. Pelletier, A. Stamouli, and G. Hadziioannou. Structure of chemically end-grafted polymer chains studied by scanning force microscopy in bad-solvent conditions. *Macromolecules*, 30(16):4719–4726, 1997.

[139] V. Koutsos, E. W. Van Der Vegte, P. C. M. Grim, and G. Hadziioannou. Isolated polymer chains via mixed self-assembled monolayers: Morphology and friction studied by scanning force microscopy. *Macromolecules*, 31(1):116–123, 1998.

[140] V. Koutsos, E. W. Van der Vegte, and G. Hadziioannou. Direct view of structural regimes of end-grafted polymer monolayers: A scanning force microscopy study. *Macromolecules*, 32(4):1233–1236, 1999.

[141] Anton Kiriy, Ganna Gorodyska, Sergiy Minko, Werner Jaeger, Petr Stepánek, and Manfred Stamm. Cascade of coil-globule conformational transitions of single flexible polyelectrolyte molecules in poor solvent. *J Am Chem Soc*, 124(45):13454–62, 11 2002.

[142] E. Glynos, A. Chremos, G. Petekidis, P. J. Camp, and V. Koutsos. Polymer-like to soft colloid-like behaviour of regular star polymers adsorbed on surfaces. *Macromolecules*, 40:6947–6958, 2007.

[143] A. Kiriy, G. Gorodyska, S. Minko, M. Stamm, and C. Tsitsilianis. Single molecules and associates of heteroarm star copolymer visualized by atomic force microscopy. *Macromolecules*, 36(23):8704–8711, 2003.

[144] J. C. Meiners, A. Quintel-Ritzi, J. Mlynek, H. Elbs, and G. Krausch. Adsorption of block-copolymer micelles from a selective solvent. *Macromolecules*, 30(17):4945–4951, 1997.

[145] E. Glynos, S. Pispas, and V. Koutsos. Amphiphilic diblock copolymers on mica: Formation of flat polymer nanoislands and evolution to protruding surface micelles. *Macromolecules*, 41:4313–4320, 2008.

[146] X. Li, Y. Han, and L. An. Surface morphology evolution of thin triblock copolymer films during spin coating. *Langmuir*, 18(13):5293–5298, 2002.

[147] R. Seemann, S. Herminghaus, and K. Jacobs. Dewetting patterns and molecular forces: a reconciliation. *Phys Rev Lett*, 86(24):5534–7, 6 2001.

[148] R. Seemann, S. Herminghaus, and K. Jacobs. Gaining control of pattern formation of dewetting liquid films. *Journal of Physics: Condensed Matter*, 13:4925, 2001.

[149] Eli Ruckenstein and Rakesh K. Jain. Spontaneous rupture of thin liquid films. *J. Chem. Soc., Faraday Trans. 2*, 70:132–147, 1974.

[150] G. Reiter, A. Sharma, A. Casoli, M. O. David, R. Khanna, and P. Auroy. Thin film instability induced by long-range forces. *Langmuir*, 15(7):2551–2558, 1999.

[151] A. Sharma and R. Khanna. Pattern formation in unstable thin liquid films under the influence of antagonistic short-and long-range forces. *The Journal of chemical physics*, 110:4929, 1999.

[152] U. Thiele, M. Mertig, and W. Pompe. Dewetting of an evaporating thin liquid film: Heterogeneous nucleation and surface instability. *Physical review letters*, 80(13):2869–2872, 1998.

[153] K. Binder, A. Milchev, and J. Baschnagel. Simulation studies on the dynamics of polymers at interfaces. *Annual Review of Materials Science*, 26(1):107–134, 1996.

[154] A. Milchev and K. Binder. Static and dynamic properties of adsorbed chains at surfaces: Monte carlo simulation of a bead-spring model. *Macromolecules*, 29(1):343–354, 1996.

[155] N. Källrot and P. Linse. Dynamic study of single-chain adsorption and desorption. *Macromolecules*, 40(13):4669–4679, 2007.

[156] N. Ka llrot, M. Dahlqvist, and P. Linse. Dynamics of polymer adsorption from bulk solution onto planar surfaces. *Macromolecules*, 42(10):3641–3649, 2009.

[157] P. Linse and N. Ka llrot. Polymer adsorption from bulk solution onto planar surfaces: Effect of polymer flexibility and surface attraction in good solvent. *Macromolecules*, 43(4):2054–2068, 2010.

[158] P. Linse. Effect of solvent quality on the polymer adsorption from bulk solution onto planar surfaces. *Soft Matter*, 8:5140, 2012.

[159] Goundla Srinivas, Dennis E. Discher, and Michael L. Klein. Self-assembly and properties of diblock copolymers by coarse-grain molecular dynamics. *Nat Mater*, 3(9):638–44, 9 2004.

[160] Nicodemo di Pasquale, Daniele Marchisio, and Paola Carbone. Mixing atoms and coarse-grained beads in modelling polymer melts. *J Chem Phys*, 137(16):164111, 10 2012.

[161] A. Chremos, E. Glynos, V. Koutsos, and P. J. Camp. Adsorption and self-assembly of linear polymers on surfaces: a computer simulation study. *Soft Matter*, 5:637, 2009.

[162] Alexandros Chremos, Philip J. Camp, Emmanouil Glynos, and Vasileios Koutsos. Adsorption of star polymers: computer simulations. *Soft Matter*, 6(7):1483–1493, 2010.

[163] J. Israelachvili and R. Pashley. The hydrophobic interaction is long range, decaying exponentially with distance. *Nature*, 300(5890):341–2, 11 1982.

[164] J. Hu, X. D. Xiao, D. F. Ogletree, and M. Salmeron. Imaging the condensation and evaporation of molecularly thin films of water with nanometer resolution. *Science*, 268(5208):267–9, 4 1995.

[165] J. Hu, X. D. Xiao, D. F. Ogletree, and M. Salmeron. The structure of molecularly thin films of water on mica in humid environments. *Surface science*, 344(3):221–236, 1995.

[166] W. Cantrell and G. E. Ewing. Thin film water on muscovite mica. *The Journal of Physical Chemistry B*, 105(23):5434–5439, 2001.

[167] Chiara Spagnoli, Katja Loos, Abraham Ulman, and Mary K. Cowman. Imaging structured water and bound polysaccharide on mica surface at ambient temperature. *J Am Chem Soc*, 125(23):7124–8, 6 2003.

[168] L. Cheng, P. Fenter, K. L. Nagy, M. L. Schlegel, and N. C. Sturchio. Molecular-scale density oscillations in water adjacent to a mica surface. *Phys Rev Lett*, 87(15):156103, 10 2001.

[169] D. Beaglehole and H. K. Christenson. Vapor adsorption on mica and silicon: entropy effects, layering, and surface forces. *The Journal of Physical Chemistry*, 96(8):3395–3403, 1992.

[170] Marian E. Gindy, Robert K. Prud'homme, and Athanassios Z. Panagiotopoulos. Phase behavior and structure formation in linear multiblock copolymer solutions by monte carlo simulation. *J Chem Phys*, 128(16):164906, 4 2008.

[171] T. Chen, H. Liu, and Y. Hu. Monte carlo simulation for the adsorption of diblock copolymers. i. in nonselective solvent. *The Journal of Chemical Physics*, 114:5937, 2001.

[172] M. Mecke, J. Winkelmann, and J. Fischer. Molecular dynamics simulation of the liquid–vapor interface: The lennard-jones fluid. *The Journal of chemical physics*, 107:9264, 1997.

[173] D. Chandler J. D. Weeks and H. C. Andersen. Role of repulsive forces in determining the equilibrium structure of simple liquids. *The Journal of Chemical Physics*, 54(12), 6 1971.

[174] Wei Shi and J. Karl Johnson. Histogram reweighting and finite-size scaling study of the lennard–jones fluids. *Fluid Phase Equilibria*, 187:171–191, 2001.

[175] B. Smit. Phase diagrams of lennard-jones fluids. *Journal of chemical physics*, 96(11):8639–8640, 1992.

[176] Philip J. Camp and Michael P. Allen. Phase coexistence in a pseudo gibbs ensemble. *Molecular Physics*, 88(6):1459–1469, 1996.

[177] Andreas Werner, Friederike Schmid, Marcus Müller, and Kurt Binder. Anomalous size-dependence of interfacial profiles between coexisting phases of polymer mixtures in thin film geometry: a monte-carlo simulation. *arXiv preprint cond-mat/9708153*, 1997.

[178] A. Werner, F. Schmid, M. Müller, and K. Binder. Intrinsic profiles and capillary waves at homopolymer interfaces: A monte carlo study. *Physical Review E*, 59(1):728, 1999.

[179] Vivien M. Kendon, Michael E. Cates, Ignacio Pagonabarraga, J. C. Desplat, and Peter Bladon. Inertial effects in three-dimensional spinodal decomposition of a symmetric binary fluid mixture: a lattice boltzmann study. *Journal of Fluid Mechanics*, 440:147–203, 2001.

[180] P. M. Claesson, R. Makuska, I. Varga, R. Meszaros, S. Titmuss, P. Linse, J. Skov Pedersen, and Cosima Stubenrauch. Bottle-brush polymers: adsorption at surfaces and interactions with surfactants. *Adv Colloid Interface Sci*, 155(1-2):50–7, 3 2010.

[181] Pietro Tierno, Ramin Golestanian, Ignacio Pagonabarraga, and Francesc Sagués. Controlled swimming in confined fluids of magnetically actuated colloidal rotors. *Phys Rev Lett*, 101(21):218304, 11 2008.