

A Model for Polymer Membranes



Richard Broadbent
Department of Physics
Imperial College London

A thesis submitted for the degree of

Doctor of Philosophy

24-12-2014

Abstract

Separation processes are widely used throughout the chemical and pharmaceutical industries. Polymer membranes have the potential to significantly improve both energy usage and the costs of separation processes by reducing reliance on fractional distillation. For this to occur methods to control the porosity of the polymer membranes must be identified. The P84 molecule is a relatively complex co-polymer with numerous strongly interacting rigid groups, with a persistence length of over 1.1 nm, and the region in which filtration pores form in the membrane is typically 50–80nm thick, whilst the pores of interest within the membrane are typically less than 0.5 nm in size. P84 membranes are used commercially to separate molecules from organic solvents, in a process called organic solvent nanofiltration. Recent experiments with membranes produced from the P84 polyimide molecule found that altering the solvent used in the initial stage of manufacture radically altered the size of the sub-nanometre pores in the filtration region of the membrane. This effect was not expected, and could not be explained by the available models for polymer membrane formation.

I present here a model as well as key results developed during my investigation of the formation of P84 polymer membranes. The model uses a mixture of fully atomistic molecular dynamics simulations of a single P84 molecule in solvent and coarse grained Monte Carlo simulations containing hundreds of complete polymer molecules. It demonstrates that the experimentally observed changes in pore sizes in P84 membranes can be explained by the differing interaction energies between the solvents and the polymers. I further present a new method

for coarse graining aromatic polymers in molecular dynamics simulations which has been shown to permit the time step to be increased from 1 fs to 5 fs whilst maintaining all-atom accuracy.

To my friends and family without your support and encouragement I
would never have gotten this far.

This thesis describes work carried out at Imperial College London,
under the Supervision of Adrian P. Sutton, Arash A. Mostofi,
Andrew G. Livingston, and James S. Spencer.

The following research is my own work. Any research conducted by
or in collaboration with others has been duly noted and
appropriately referenced as such. All information which is the
intellectual property of others has been fully cited.

The copyright of this thesis rests with the author and is made
available under a *Creative Commons Attribution Non-Commercial
No Derivatives* licence. Researchers are free to copy, distribute or
transmit the thesis on the condition that they attribute it, that they
do not use it for commercial purposes and that they do not alter,
transform or build upon it. For any reuse or redistribution,
researchers must make clear to others the licence terms of this work

Richard James Broadbent, Imperial College London 2014

Acknowledgements

This work has been made possible by the support of the Centre for Doctoral Training on Theory and Simulation of Materials at Imperial College funded by EPSRC [grant number EP/G036888/1]. I would also like to thank the the Thomas Young Centre for their support. This work was carried out on computer facilities provided by King Abdullah University of Science and Technology (KAUST) Supercomputing Laboratory (KSL), the UK HPC Materials Chemistry Consortium funded by EPSRC [grant number EP/F067496], and the Imperial College London HPC service without these facilities this research could not have been completed.

Contents

Contents	7
List of Figures	11
List of Tables	13
Acronyms	14
Glossary	16
Symbols	18
1 Introduction	20
1.1 Industrial Separation	20
1.2 Membranes	21
1.3 Modelling	21
2 Techniques	23
2.1 Monte Carlo	23
2.1.1 The Monte Carlo Method	24
2.1.2 Monte Carlo Modelling of Polymers	26
2.1.2.1 The Representation	26
2.1.2.2 The Move	28
2.2 Molecular Dynamics	30
2.2.1 The Molecular Dynamics Method	33
2.2.2 Molecular Dynamics Modelling of Polymers	34

2.3	Density-Functional Theory	38
2.3.1	The Schrödinger Equation	38
2.3.2	Applied Density-Functional Theory	40
3	Polymer Membranes	45
3.1	Organic Solvent Nanofiltration Membranes	46
3.1.1	Thin Film Composite Membranes	46
3.1.2	Integrally Skinned Membranes	47
3.2	Phase Inversion	47
3.2.1	Controlling the Porosity	49
3.2.2	Imaging Membrane Structures	50
3.3	Modelling Polymers	53
3.3.1	Dilute Polymer Solutions	53
3.3.2	Concentrated Polymer Solutions	55
3.4	Modelling Membranes	56
3.4.1	Membrane Performance	56
3.4.2	Membrane Structure	57
4	Coarse Graining	59
4.1	Method	61
4.1.1	Two Dimensional Rigid Bodies	62
4.1.2	Coarse Grained Representation of Aromatic Polymers	62
4.1.3	Constraint Algorithms	65
4.1.4	Forces on Rigid Groups	72
4.1.5	Force Fitting	73
4.2	Results	76
4.3	Conclusions	87
5	Analysis of Molecular Dynamics Simulations	88
5.1	Expectations	89
5.2	Qualitative Tests	91
5.3	Quantitative Tests	93
5.3.1	The Micro-Canonical Ensemble	94
5.3.2	The Canonical Ensemble	99

6	P84 Membranes	103
6.1	The P84 Molecule	103
6.2	Solvents	106
6.3	Modelling P84 Membranes	107
7	Molecular Dynamics of P84	110
7.1	P84 in the Dreiding Force Field	110
7.2	Parametrising a Custom P84 Force Field	111
7.3	Molecular Dynamics of P84 in the OPLS-AA forcefield	114
7.4	P84 Oligomers in Dilute Solution	116
7.5	P84 at Experimental Concentrations	120
7.5.1	Theory	121
7.5.2	Simulation Details	122
7.5.3	Results	125
7.5.4	Conclusion	130
8	Monte Carlo Simulations of P84 Membranes	131
8.1	Layers Model	132
8.2	Two Dimensional Lattice Model	135
8.3	Three Dimensional Lattice Model	137
8.3.1	Single Polymers	138
8.3.2	Multiple Polymers	139
8.3.3	Dense Polymer Systems	145
8.3.3.1	Estimation of Pore Size	145
8.3.3.2	Comparison with STEM Images	147
8.3.4	Directional Biasing	147
9	Conclusions	153
9.1	Coarse Graining	154
9.2	Modeling P84 Membranes	154
9.3	Further Work	155
	References	157
A	Monte Carlo	180

B Supplemental Energy Drift Graphs	184
------------------------------------	-----

List of Figures

2.1	Lattice polymer reptation	31
2.2	A caged polymer configuration	32
2.3	A schematic diagram of bonding terms	37
3.1	Membrane structure	48
3.2	Rejection curves for P84 membranes	51
4.1	A schematic demonstration of the coarse graining procedure . . .	63
4.2	PEEK chemical structure	66
4.3	Energy drift	78
4.4	Energy drift against SHAKE tolerance	79
4.5	Energy drift for an under converged SHAKE tolerance	80
4.6	Performance	82
4.7	Radius of gyration	84
4.8	Hinge angles	85
4.9	Dihedral angles key	86
4.10	Dihedral angles	86
5.1	Voids in DMF	92
5.2	Circle vs. normal distribution	95
5.3	An inconsistent energy drift	98
5.4	A further inconsistent energy drift	100
6.1	The P84 molecule	105
6.2	The DMF molecule	106
6.3	The dioxane molecule	106

List of Figures

6.4	Multi-scale coupling	108
7.1	P84 un-parametrised dihedral angles	115
7.2	Convergence of radius of gyration	119
7.3	Angular correlation against separation for P84	127
7.4	Angular correlation against separation for separations of less than 15 nm	128
7.5	Persistence length against separation cutoff	129
8.1	Layers model	133
8.2	Configurations	134
8.3	Thickness change	134
8.4	Solvent polymer interaction	140
8.5	Single polymers compared to pairs of polymers	142
8.6	Behaviour with respect to density	143
8.7	Persistence length	144
8.8	Projections	148
8.9	STEM images of P84 membranes	148
8.10	Directionally biased persistence length	150
B.1	Supplemental energy drift plot 1	185
B.2	Supplemental energy drift plot 2	186
B.3	Supplemental energy drift plot 3	187
B.4	Supplemental energy drift plot 4	188
B.5	Supplemental energy drift plot 5	189
B.6	Supplemental energy drift plot 6	190
B.7	Supplemental energy drift plot 7	191
B.8	Supplemental energy drift plot 8	192

List of Tables

4.1	Coarse graining parameters	66
7.1	Energy drifts	125
7.2	Temperature drifts	126
7.3	Simulation Pressures	126
B.1	P84 oligomers	184

Acronyms

aMD *ab initio* molecular dynamics

AA all-atom

AA-RB all-atom rigid bonds

B88 Becke 1988

BTDA benzophenone tetracarboxylic acid dianhydride

CG coarse grained

COMPASS condensed-phase optimized molecular potentials for atomistic simulation studies

DFT density-functional theory

dioxane 1,4-dioxane

DMF dimethylformamide

dSFMT double-precision SIMD optimised Mersenne twister

FH Flory-Huggins

GAFF generalised AMBER forcefield

GGA generalised gradient approximation

HK Hohenberg and Kohn

KS Kohn and Sham

LDA local-density approximation

LINCS linear constraint solver

LJ Lennard-Jones

MC Monte Carlo

MD classical molecular dynamics

MDA 4,4'-diaminodiphenylmethane

MgO magnesium oxide

MH Metropolis-Hastings Monte Carlo

MILC-SHAKE matrix inverted linearized constraints SHAKE

MILCH-SHAKE MILC hybridized with SHAKE

MP2 second order Møller Plesset perturbation theory

MSD mean squared displacement

NEMD non-equilibrium molecular dynamics

NMR nuclear magnetic resonance

ODE ordinary differential equation

OPLS-AA optimised potential for liquid simulation all atom

OSN organic solvent nanofiltration

P-LINCS Parallel-LINCS

PBE Perdew, Burke, and Enzerhof

PEEK polyether ether ketone

PERM pruned-enriched Rosenbluth method

PIMs polymers of intrinsic microporosity

PME particle mesh Ewald summation

POEMS parallelizable open source efficient multibody software

PRNG pseudorandom number generator

PS polystyrene

PW91 Perdew and Wang 1991

RO reverse osmosis

RR Rosenbluth and Rosenbluth

SAW self-avoiding *random* walk

SEM scanning electron microscope

STEM scanning transmission electron microscope

TDA toluenediamine

TEM transmission electron microscope

vdW van der Waals

WELL well equidistributed long-period linear

XC exchange-correlation

Glossary

dope A liquid mixture of macro-molecule and solvent.

good A solvent in which a macro-molecule is easily dissolves.

poor A solvent in which a macro-molecule does not dissolve.

Θ temperature The temperature, for a given solvent polymer pair, where the polymer recovers the ideal, non-interacting, scaling behaviour.

co-polymer A polymer with two distinct monomer units.

cx2 Imperial College High Performance Computing Centre's supercomputer it is a heterogeneous machine consisting of 8, 12 and 16 core Intel Xeon nodes.

HECToR The UK national supercomputer from 2008-2013. In its final phase it was a Cray XE6 with 90112 cores

macro-molecule A large molecule. These need not be composed of a set of repeated monomer units they could be completely non-repeating. However, in this work a macro-molecule will generally refer to a co-polymer or other polymeric system which is formed from one or more distinct monomer units.

monomer A molecular unit which is joined with other monomer units to form a polymer.

P84 A co-polyimide composed of benzophenone tetracarboxylic acid dianhydride (BTDA) 4,4'-diaminodiphenylmethane (MDA) (20%) and BTDA toluene-diamine (TDA) (80%).

permeate The fluid which permeates through the membrane.

phase inversion A widely used technique for manufacturing membranes.

polymer A macro-molecule comprised of a repeated set of monomer units.

RATTLE An iterative constraint solver for velocities used in MD simulations.

retentate Everything which does not permeate through the membrane.

SETTLE An analytic solution for the SHAKE RATTLE procedure for a rigid three mass object in MD simulations.

Shaheen The 65536 core Bluegene/P supercomputer at King Abdullah University of Science and Technology.

SHAKE An iterative constraint solver for positions used in MD simulations.

solution A fluid phase mixture of macro-molecule and solvent.

solvent A chemical used to form a fluid solution or *dope* containing larger molecules which would otherwise be solid under the given atmospheric conditions.

TIP3P A three-site water model for use in MD simulations, it is the standard model for use with the OPLS-AA forcefield.

WIGGLE An iterative constraint solver for accelerations used in MD simulations.

Symbols

$\beta \equiv (k_B T)^{-1}$

$\chi_{\mathbf{FH}}$ The Flory-Huggins parameter

$\Delta\varepsilon$ The solvent polymer interaction parameter.

Δt time step

H The Hamiltonian of a system. This defines the energy of a configuration with positions $\{\mathbf{q}\}$ and momenta $\{\mathbf{p}\}$.

k_B Boltzmann's constant

l_p The persistence length of a polymer, it characterises the decay in correlation between the direction of segments along a polymer molecule.

ν The scaling exponent of a polymeric system

T Temperature

Θ Theta solvent

\mathcal{Z} The partition function

Chapter 1

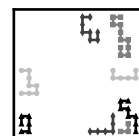
Introduction

Why are we interested in polymer membranes, and why would a model be useful? To answer these questions and understand the scope of this work we will briefly consider separation at an industrial scale, the role of membranes, and the potential impact of modelling on this field moving forwards.

1.1 Industrial Separation

“Separation processes account for 40–70% of capital and operating costs in industry” [1]. This gives us some idea of the importance of separation processes to our daily lives, from the refining of oil, plastics and fuels to the creation of medicines, industrial separation is everywhere.

One of the main processes used in industrial separation is fractional distillation where a liquid is heated and the various gasses are then extracted based on their molecular weight and re-condensed. Fractional distillation is highly energy intensive and requires huge facilities such as oil refineries to produce large quantities of liquid. In contrast membrane based facilities have a demonstrated ability to separate vast quantities of fresh water from sea water using only a fraction of the energy of fractional distillation or even the membrane filtration plants of a decade ago[2]. Therefore considerable effort has been spent on membrane filtration.



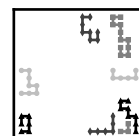
1.2 Membranes

Originally developed in the 1960s, polymer membranes have been in widespread use for many decades[3, 4, 5]. The need to reduce energy consumption both to reduce costs and greenhouse gas emissions has resulted in increased interest in membranes in the last two decades[1]. In particular there has been a large volume of work on organic solvent nanofiltration (OSN) membranes[6, 7, 8, 9, 10, 11, 12, 13, 14, 15]. These permit the separation of organic molecules from organic solvents based on their size and molecular weight.

Developing a new product, particularly one tailored on a nano-scale like an OSN membrane, is expensive. Therefore, the ability to understand the influence of manufacturing techniques, or to predict the behaviour of a membrane before it has been manufactured, has the potential to substantially reduce the costs associated with designing membranes. Recent work has shown that the models currently used to understand the formation of phase inversion OSN membranes do not match the trends seen experimentally. In particular models typically expect that the evaporation time in manufacturing dominates the final porosity, whilst recent experiments have shown that certain membranes are insensitive to this whilst being exceptionally sensitive to the composition of the solvents used in their production[6, 7, 8, 9, 10]. A new approach to modelling the formation of these membranes is therefore required.

1.3 Modelling

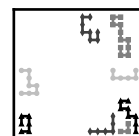
The creation of a new model is a complex undertaking. Polymer membranes have numerous features which can be modelled including porosity, diffusivity of different species, and degree of cross-linking. This work focusses on building a model for the formation of phase inversion membranes. This model is designed to predict the molecular structure of phase inversion membranes from the inter- and intra-molecular interactions of the chemicals involved. The phase inversion process takes place over seconds and is governed by interactions between polymers and solvents. Accurately modelling a system containing many large strongly interacting molecules over a period of seconds requires both the precision of atomistic and



quantum modelling techniques and the ability of coarse grained representations to rapidly sample configurations. Therefore, a model using Monte Carlo (MC), classical molecular dynamics (MD), and density-functional theory (DFT), was designed. MC¹ allows the study of large collections of complete polymer molecules, thereby providing access to the pore sizes and distributions. Whilst MD provides detailed information about the behaviour of small collections of molecules. DFT provides more accurate energetics and can permit the parameterisation of custom forcefields for molecules where accurate forcefields do not exist.

Coupling three computational techniques together is clearly a challenging task. However, despite this, significant progress has been made here towards a complete model taking atomistic information and using it to understand the trends seen in the production of commercial OSN membranes. In the course of developing this model a new way to coarse grain polymers with aromatic backbones was developed[16]. Furthermore, it can be seen from this work that the trends in pore size and distributions seen in the dense surface layer of P84 membranes match those seen in the initial polymer *dope*. This research represents the beginning of a larger body of work which will aid in our understanding of the behaviour of membranes during both manufacture and operation.

¹Example configurations from the MC model, described in Sec. 8.2, are provided in the lower right hand corners of the pages of this work. These are best viewed as a flip book to show the evolution of the polymers through the simulation.



Chapter 2

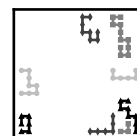
Techniques

Since their invention computers have been used to simulate models of physical systems[17]. The techniques used cover a wide range of topics from understanding wind damage during a hurricane[18], to the loads on a jet engine due to the loss of a fan blade[19]. Techniques are similar to theories in that they are only applicable for certain systems; quantum mechanical approaches yield the correct structure when modelling simple molecules[20], however, they would not be practicable or useful to model a complete suspension bridge as the physics of interest occurs on a much larger length scale. Systems moving on different length and time scales require different approaches to modelling. When choosing which simulation techniques to use it is, therefore, important to consider both the length and time scales involved in the system of interest. During the course of my research I have used the MC method, MD, and DFT. This chapter will present an overview of these three techniques and some of their strengths and weaknesses when applied to polymeric systems.

2.1 Monte Carlo

The MC method is an approach to studying systems by using random numbers to sample the possible configurations¹. From the earliest days of computer mod-

¹ A configuration is a set of variables which uniquely identify the state of a system, e.g., for a system comprising a single point mass connected to a perfect harmonic spring, the point mass' position and velocity describe its configuration.



elling MC has been used to solve problems which are too complicated to solve algebraically[17]. Applications of the MC technique are highly varied with it being used for the study of systems as disparate as the relative risks posed by a virus to farmed Scottish rainbow trout[21], the fretting fatigue life of components made from Ti-6Al-4V[22], as well as some of the most exact numerical solutions to the non-relativistic Schrödinger equation[23]. The discussion in this section will be limited to the general principle of MC modelling, and how it is applied to polymers. For reference a basic introduction to the advantages of MC sampling and the Metropolis-Hastings Monte Carlo (MH) method[17, 24] is provided in App. A, a more complete description is available in e.g., Refs [25, 26]

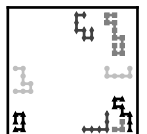
2.1.1 The Monte Carlo Method

A MC simulation consists of two parts, a sampling scheme and a system. There are many sampling schemes, see for instance Refs [27, 28] however, the MH algorithm is the most widely used and is the principal technique used in my MC simulations. Sampling using the MH algorithm involves adding states to a Markov chain based on their relative probabilities. Markov chains are sets of conformations where the probability of making a transition to a trial configuration is dependent only on the current configuration of the system and the trial configuration¹. In physical systems the Markov chain generally samples a Boltzmann distribution of energies in the system. The relative probability between two states is given by the Boltzmann weight $\exp(-\beta\Delta E_{ij})$, where ΔE_{ij} is the change in energy between state i and state j and $\beta = (k_B T)^{-1}$, where k_B is Boltzmann's constant and T is the temperature. The Markov chain is then generated by a random walk with the acceptance probability

$$P_A(j|i) = \min[1, \exp(-\beta\Delta E_{ij})], \quad (2.1)$$

which is the probability to accept the move to state j given that the system is in state i . Following this step the state of the system is added to the Markov

¹This is defined mathematically as $P(\mathbf{X}_{n+1} = \mathbf{x}|\mathbf{X}_n) = P(\mathbf{X}_{n+1} = \mathbf{x}|\{\mathbf{X}_i\})$, where $\{\mathbf{X}_i\}$ is the set of all previous configurations of the Markov chain, \mathbf{x} is the trial configuration, and \mathbf{X}_n is the current configuration in the chain.

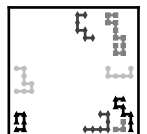


chain. This process ensures that the sampling frequency obeys a Boltzmann distribution. Averaging a micro-state observable over the Markov chain produces an ensemble average of that observable. For further details see, e.g., Refs [25, 26]

In all MC simulations it is by definition necessary to stochastically sample the phase space, doing so requires the use of random numbers. Unfortunately computers are by their nature deterministic, given identical input they will produce identical output. Typically large quantities of random numbers ($\gg 10^9$) are required for a MC simulation, therefore, generating these before the simulation and reading them in is impractical. The solution to this problem is to use a special piece of computer code known as a pseudorandom number generator (PRNG). These PRNGs generate streams of numbers which are not random[29, 30, 31] but obey certain statistical properties which are expected of random numbers. No generator is perfect and using a poor generator such as the linear congruential generators commonly included in standard programming libraries can result in incorrect sampling[32]. Using an appropriate generator which generates a statistically reasonable stream of pseudorandom numbers is essential for MC simulations. Therefore, in my research I have primarily used the double-precision SIMD optimised Mersenne twister (dSFMT)[29, 31] generator which is optimized and improved form of the Mersenne twister algorithm. A more recent algorithm known as well equidistributed long-period linear (WELL)[30] has shown improved uniformity in higher dimensional spaces as well as better recovery from a poor seed. However, it is substantially slower and less widely cited, it was therefore not used in my MC program.

The MH algorithm requires some means of generating a new configuration of the system (state j in Eq. (2.1)): this is known as a *move*. A move in a MH algorithm is system specific and will be discussed in the context of polymers in the following subsection (Sec. 2.1.2). In general it is desirable to use a valid move¹ which causes a large change in the parameters defining phase space, whilst maintaining a high acceptance probability. This allows efficient sampling of a large region of the phase space at a reasonable computational cost.

¹In my research I have always used moves which obey detailed balance; for discussion of validity of MH moves and the detailed balance condition see, e.g., Ref. [25].



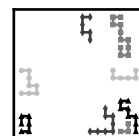
2.1.2 Monte Carlo Modelling of Polymers

Polymers were one of the first systems to be modelled using MC computer simulations[33]. There are two primary challenges when designing MC simulations of polymers: finding or creating a representation of a polymer, which includes the necessary information for the property of interest; and choosing a suitable move for that representation. These two challenges are inherently linked as a move suitable for one representation of a polymer will not necessarily translate to another representation. The representation and MC move will be addressed in turn.

2.1.2.1 The Representation

Polymer models come in many forms or representations. These vary from the highest resolution models where the electrons in a polymer are treated quantum mechanically through all-atom (AA) models, to soft core *blob* models, where a polymer is treated as chain of soft spheres, or even a single soft sphere. The resolution used to represent a polymer system depends on which properties of the polymer are of interest. Higher resolution models usually incur a large computational cost relative to lower resolution models as they have a larger space of possible conformations. Therefore, it is advisable to use the lowest resolution model that will include the necessary interactions. The discussion that follows applies, unless otherwise stated, to models of polymers that range in resolution from AA to a series of non-interacting connected segments.

When modelling polymers it is necessary to generate an initial configuration. This process is usually referred to as ‘building’ or ‘growing’ a molecule. If the representation is completely non-interacting this can be accomplished by randomly placing one polymer segment at a certain orientation then placing another segment on its end with a random orientation, and another segment after that, and so on until the desired length is reached. However, chains built in this manner are likely to include overlapping segments which for interacting representations are often excluded from the space of allowed configurations or included with a very large energy. Configurations with a large energy relative to the thermal energy



(β^{-1}) have minimal impact on the partition function,

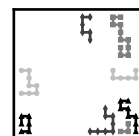
$$\mathcal{Z} = \int_{\mathbb{R}^{6N}} \exp(-\beta H(\{\mathbf{q}\}, \{\mathbf{p}\})) d\mathbf{q}^{3N} d\mathbf{p}^{3N}. \quad (2.2)$$

Here H is the hamiltonian for the system, and $\{\mathbf{q}\}$ and $\{\mathbf{p}\}$ are respectively the positions and the momenta of the N objects in the system. Therefore, configurations with high energies should ideally be generated infrequently as they will generally have a very small acceptance probability (see Eq. (2.1)). As every failed attempt to grow a polymer chain expends computational effort, and the number of chains which must be sampled increases with chain length, it will become impossible to sample long chains if the acceptance rate is too low. Therefore, a significant amount of effort has been directed towards finding algorithms to grow polymer chains efficiently.

One of the earliest computer models for a polymer chain worked by generating polymer chains on a lattice, with a self-avoiding *random* walk (SAW)[33]. This was subsequently extended to allow its use for systems containing multiple polymer molecules[34]. The SAW model of a chain represents a polymer as a series of connected segments on a lattice, each of these segments may be oriented to join any nearest neighbour site on the lattice. The self avoidance is enforced by forbidding placing a segment onto an occupied lattice site. This model is generally used to sample the scaling behaviour of polymers[35] or systems where the polymers are weakly interacting with one another[36]. The SAW model, is widely used in polymer simulation and is a topic of great interest in many areas, for instance in polymer adsorption at surfaces[36], and has been extended to allow the segments, or sites, to interact via a potential and then be sampled using MH[37].

Models of polymers can be far more sophisticated than the SAW model. OPLS-AA¹ is a forcefield designed for atomistic MC simulations of organic molecules[38]. These AA models can also be used in MD simulations of polymers using force-fields, as discussed in Sec. 2.2.2.

¹ Optimised potential for liquid simulation all atom (OPLS-AA).



2.1.2.2 The Move

The probability of randomly growing a SAW polymer chain decreases rapidly with increasing chain length and increasing density. Therefore Rosenbluth and Rosenbluth (RR)[33] developed a procedure for growing a polymer chain that only allows segments to be placed on unoccupied sites, which introduces a bias into the sampling. RR corrected for this by introducing a weight onto each conformation based on the available positions when each segment is placed[33]. This technique was also generalised to multiple chains[34] and more complex models using potentials[37]

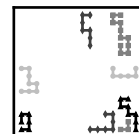
Increasing computer power in the 1980s resulted in large, > 100 segments, polymer systems being simulated for the first time. This led to the discovery that certain rare conformations with large RR weights dominate the sample space[39]. This issue was solved by modifying the bias used in the original sampling process of RR to sample according to the Boltzmann distribution, thereby, removing the intrinsic bias and the need for weighting the configurations[37]. The bias is removed by adding an acceptance criteria,

$$P(j|i) = \min \left[1, \frac{W_j}{W_i} \right], \quad (2.3)$$

where W_i is the RR weight of configuration i , into the RR procedure. The approach was later generalised to allow the construction and sampling of continuously deformable (non-lattice) polymers[40]. This technique is often used within programs, such as *Amorphous Cell*^{®1}, to generate initial polymer conformations for MD simulations. Although it should be noted that in Materials Studio V6.1 *Amorphous Cell*[®] does not make any correction for the inherent bias in a RR process.

The space of configurations for a polymer system with atomistic resolution is vast (\mathbb{R}^{3N} where N is the number of atoms in the polymer). Therefore, as the majority of these configurations will represent highly unphysical conformations, it is sensible to bias the sampling towards the more likely areas of phase space. Techniques which bias the system prior to a RR or MH step exist; see, e.g.,

¹A part of the Materials Studio[®] version 6.1 software suite by Accelrys[®] (<http://accelrys.com/products/materials-studio/>).

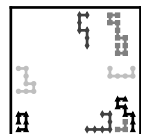


Ref. [25] for a discussion of how these can be implemented. However, even in non-atomistic models the conformational phase space is very large as polymers of interest can consist of over a hundred monomers[6]. Polymers of this length in dense systems are computationally expensive to grow; therefore, it would be useful to attempt to explore the region of phase space near a polymer before, or instead of, growing a new polymer.

Many different schemes for making conformational changes to (or moving) polymers exist[41, 42, 43, 44, 45, 46, 47]. A brief overview of a selection of moves is presented, however, this is only a small selection of many possible MH moves for polymers. Furthermore, a discussion of moves at atomistic resolution will not be included as these have not been required for the applications presented in subsequent chapters. However, the process is broadly similar except that in general parts of the model are made rigid with only selected atoms permitted to be chosen as the move sites; a discussion of the use of AA moves is included in Ref. [48].

The choice of polymer move is highly system-dependent and must be chosen with care. The following is a brief overview of some of the more popular moves. Refs. [42, 43] describe a collection of three types of moves, known as: *end-bond*, *kink-jump*, and *crankshaft* for a square lattice; these were later generalised to a cubic lattice in [44]. The moves were some of the earliest to be used in conjunction with the MH algorithm to allow sampling of the phase space around an existing polymer structure and allow any configuration of polymer to be sampled. However, they only apply on a square or cubic lattice and different moves are necessary on other lattice types[42, 43, 44]. The popular *pull moves*[49, 50, 51] must be used with caution as one of their primary assumptions, reversibility, has been shown not to hold on certain lattice types[41]. The *pivot* move[45] takes a site in the polymer and pivots one part of the polymer around this pivot point. The move is very effective in dilute systems, however, the moves become increasingly unlikely in highly dense systems. The *reptation move*[46, 47] is easily transferable between lattices, simple to code and highly effective in sampling dense configurations. It has been used extensively in my research.

The *reptation move* has sometimes been called *slithering snake*[47]. It involves removing one or more monomers from one end of a polymer molecule,



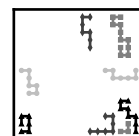
then adding the same number of monomers to the other end, see Fig. 2.1 for an example of this process. This process allows a polymer to move through high density conformations and, for a polymer composed of identical monomers, it allows movements analogous to the reptation model of polymer motion, see, e.g., Ref. [52] for a discussion of the reptation model of polymer dynamics. Further details will be discussed in Chap. 8. Some limitations of this algorithm are that it is slow to make large conformational changes to the system, and some configurations, such as when two ends of the polymer are caged by the polymer as shown in Fig. 2.2, cannot be effectively sampled. However, relative to the vast number of available configurations, these are a small subset and are extremely unlikely to occur on a lattice with a high coordination number, such as a cubic lattice[47].

The exact choice of model, representation and moves, for the MC simulation will determine the limitations of the simulation. One general limitation of MC is that only the degrees of freedom which are altered by the move can change the state of the system. Therefore, if a move is chosen which does not include a key degree of freedom the simulation will fail to accurately represent the underlying physics. This becomes problematic in more fine-grained models, such as atomistic simulations, as incorporating every possible alteration would require vast numbers of possible moves[53]. As the resolution of a simulation is decreased, the number of necessary moves decreases. However, it may become necessary to build an interaction scheme for coarse grained (CG) representations as general models such as SAW might not be directly applicable.

2.2 Molecular Dynamics

MD is a method for evolving objects interacting via a potential energy function known as a forcefield¹. MD can be used to simulate a wide range of materials in a variety of environments and, as a result, it has been used to simulate countless systems and provide insight into numerous physical phenomena, including systems which cannot be directly observed in experiments such as the melting of magnesium oxide (MgO) at core-mantle boundary pressures[54] and the dynamics of protein folding[55]. This discussion of MD will be limited to forcefields used in

¹the topic of *ab initio* molecular dynamics (*aMD*) will be discussed in Sec. 2.3.2.



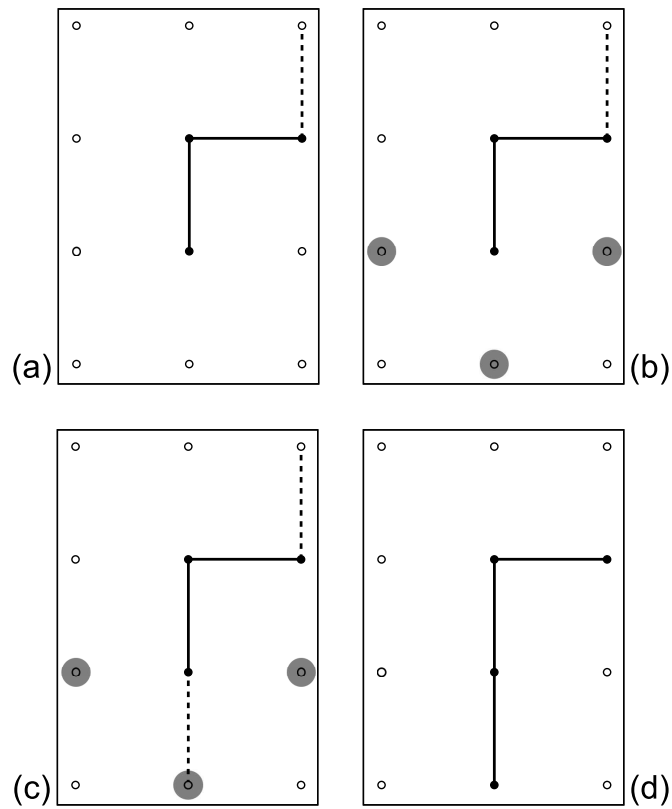
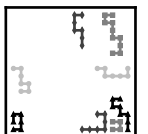


Figure 2.1: A schematic demonstration of reptation of a 3 monomer polymer on a square lattice. A segment end is chosen at random and the last element is removed, shown as dashed in (a), the available sites at the other end of the polymer are then tested (b), one of the available sites is selected (c), the move is accepted based on Boltzmann weighting (d). This conformation can then be added to the ensemble average.



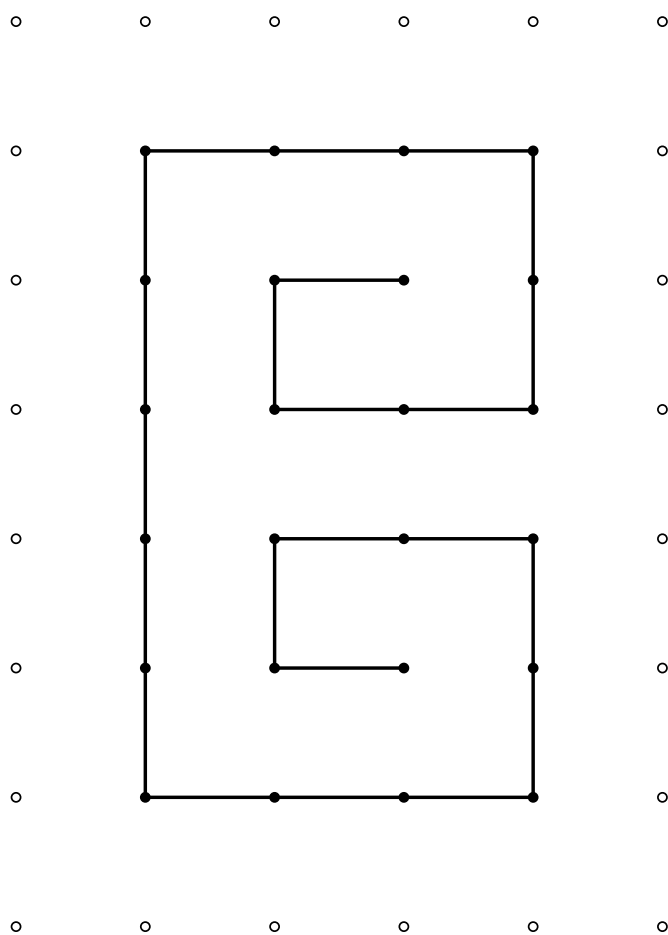
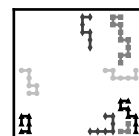


Figure 2.2: A demonstration of a cage configuration at both ends of a polymer on a square lattice. The *reptation move* can neither escape nor generate a configuration with a cage at both ends.



soft matter systems and the modelling of polymeric systems. For a more complete introduction to MD the interested reader is advised to consult e.g., Refs. [25, 26].

2.2.1 The Molecular Dynamics Method

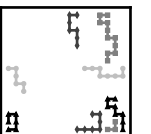
The study of classical systems generally involves solving the ordinary differential equation (ODE) defined by Newton's second law:

$$\mathbf{F}_i \left(\{\mathbf{R}\}, \{\dot{\mathbf{R}}\} \right) = m_i \ddot{\mathbf{R}}_i, \quad (2.4)$$

where \mathbf{F}_i is the force on a particle i of mass m_i and $\{\mathbf{R}\}$, $\{\dot{\mathbf{R}}\}$, and $\{\ddot{\mathbf{R}}\}$ are respectively the positions, velocities, and accelerations of all the particles in the system. For a simple system, such as two masses connected via a spring, this can be solved analytically; as the problem becomes more complex, an analytic solution becomes intractable. MD is an alternative approach where Eq. (2.4) is numerically integrated with respect to time. This yields a trajectory for the system which can then be analysed. There are two parts to a MD simulation, the integration algorithm and the representation of the physical system of interest.

There are many different algorithms for integrating Eq. (2.4). The two most common are *leapfrog* and *velocity verlet*. These algorithms are very similar and are discussed in detail in Refs. [25, 26]. The procedure used is to move forwards in time in a series of small steps. The error in the position following an integration step in either *leapfrog* or *velocity verlet* is $O(\Delta t^4)$ where Δt is the size of the time step. Therefore, a decrease in Δt results in a large increase in the accuracy of the integration; however, decreasing Δt will also increase the computational cost of the simulation as more steps will be needed to simulate a given amount of time. A balance between the two conflicting desires of accurate integration and long simulations is therefore found by monitoring properties such as energy and temperature drift over a simulation.

Standard integration algorithms for Eq. (2.4) generate a micro-canonical (constant energy, volume, and number of particles) ensemble. In general it is desirable to be able to simulate a canonical (constant temperature, volume, and number of particles) or even a constant pressure and temperature ensemble. While there is

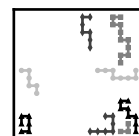


no perfect means to accomplish this, the general approach is to use algorithms to approximate the extended ensemble. These algorithms are known as thermostats (constant temperature) or barostats (constant pressure). In general they work by altering the positions and or velocities of the particles during the integration step of a MD simulation. Thermostats are often named after the lead author of the paper proposing them. Some available thermostats are: *Andersen*[56], *Berendsen*[57], *Bussi* (also called velocity rescaling or stochastic velocity rescaling)[58], *Nosé-Hoover*[59, 60], and *Langevin*[61]. All these thermostats have advantages and disadvantages. In my research I have primarily used the last three (*Bussi*, *Nosé-Hoover*, and *Langevin*) to initialise a system. I then produced results using a micro-canonical ensemble whilst monitoring the temperature drift. This approach reduces the influence of the thermostat on the dynamics whilst allowing a specific temperature to be modelled. There is a similar variety of barostats. However, as all simulations presented in this thesis were conducted in constant volume ensembles, these shall not be discussed further here. A more thorough discussion of barostats and thermostats may be found in, for example Refs. [25, 26]

The system for a MD simulation is in two parts: the physical system and its representation in MD. There are times when these two parts are indistinguishable; as for instance if the physical system is a collection of point masses connected by linear springs, then it can be represented perfectly. However, if the system comprises atoms the two parts are quite distinct and care should be taken in interpreting a physical system based on MD simulations of a representation of it, e.g., the average length of a diatomic molecule in OPLS-AA will not increase with temperature whilst this is known to happen in the physical system. The need for a representation in MD simulation arises because the underlying nature of the interactions in physical systems is not generally classical but quantum mechanical, even if the dynamics are almost invariably treated classically.

2.2.2 Molecular Dynamics Modelling of Polymers

For an atomistic system the first step is usually to approximate an atom as a point mass. A forcefield is then used to determine the interactions between the point masses. Many forcefields exist for molecular systems, and some of the more

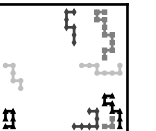


common can be found in Refs. [38, 62, 63, 64, 65, 66]. These forcefields use a functional form that associates an energy with the lengths of each covalent bond, the angle between two covalent bonds, the dihedral angle formed by three sequential bonds twisting about the axis provided by a central bond, and the improper dihedral formed by three covalent bonds leaving a central atom. These are shown schematically in Fig. 2.3. Further terms known as *non-bonded* interactions are associated with: the electrostatic interaction between two charged atoms, Pauli repulsion, and van der Waals (vdW) interactions. The Pauli repulsion is a quantum mechanical effect which states that two electrons cannot occupy the same spin state, it results in an exponential repulsion between atoms at short distances, for reasons of computational efficiency this is generally approximated with an r^{-n} term in the forcefield. vdW is an effect where one atom has a dipole induced between its nucleus and its electron cloud, this induces a dipole in another atom and the two atoms begin to interact. Adding together all these components as well as the kinetic energies of every atom gives us the total energy of the system.

My work has primarily used the OPLS-AA forcefield[38], the which has potential energy (V) given by:

$$\begin{aligned}
V = & \sum_{\text{bonds}} k_r (r_{ij} - r_0)^2 \\
& + \sum_{\text{angles}} k_\theta (\theta - \theta_0)^2 \\
& + \frac{1}{2} \sum_{\text{dihedrals}} \sum_{i=1}^4 k_i \left(1 - (-1)^i \cos(i(\phi - \phi_0)) \right) \\
& + \frac{1}{2} \sum_{\text{impropers}} \sum_{i=1}^4 k_i \left(1 - (-1)^i \cos(i(\phi - \phi_0)) \right) \\
& + \sum_{i=1}^{N-1} \sum_{j=i+1}^N f_{ij} \left(\frac{q_i q_j}{4\pi\epsilon_0 r_{ij}} + \left(\frac{A_{ij}}{r_{ij}} \right)^{12} - \left(\frac{B_{ij}}{r_{ij}} \right)^6 \right), \quad (2.5)
\end{aligned}$$

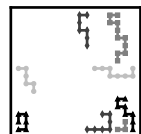
where r_{ij} is the separation between two atoms, θ is the angle between two covalent bonds, ϕ is the dihedral (or improper dihedral) angle formed by three covalent bonds, k_r , k_θ , and k_i are the spring constants associated with covalent bonds,



angles, and dihedral angles (or improper dihedral angles), respectively, and r_0 , θ_0 , and ϕ_0 are the respective equilibrium values. $A_{ij} = \sqrt{A_{ii}A_{jj}}$, $B_{ij} = \sqrt{B_{ii}B_{jj}}$, A_{ii} is an empirically determined constant for an atom of type i and is parametrised to replicate the Pauli repulsion associated with that atom, B_{ii} is a similarly determined constant representing the vdW interaction, these two constants are generally fitted simultaneously to ensure reasonable equilibrium spacing between a wide variety of molecules[38]. q_i and q_j are the charges on the atoms, ϵ_0 is the permittivity of free space, and N is the number of atoms in the system. f_{ij} is a topologically determined constant which ensures that interactions between bonded atoms are represented through the *bonded* rather than *non-bonded* terms: it is 0 if the minimum number of bonds which separate the atoms i and j is < 3 , 0.5 if there are precisely three bonds separating the atoms, and 1 otherwise, including if the atoms are on different molecules. These parameters determine how the representation of a molecular system in OPLS-AA interact within a MD simulation.

The force in Eq. (2.4) is then defined as $\mathbf{F}_i \equiv -\nabla_i V$. Other forcefields, such as condensed-phase optimized molecular potentials for atomistic simulation studies (COMPASS)[66], extend this form by adding higher order corrections to couple the bond, angle, dihedral angle, and improper dihedral angle energies together. Forcefields are typically fitted to a mixture of experimental and quantum mechanical simulation data and optimized to reproduce certain properties of interest[38, 63, 64, 65, 66]. As there can be no perfect analogue between quantum and classical mechanics due to non-local correlation[67], the parameters will be a reasonable approximation only for certain properties in regimes similar to those used for the parametrisation; therefore, transferability of parameters is an issue. Furthermore, if terms that are necessary to describe the physics of the system are not included in the functional form of the forcefield, then the results obtained will not be representative of the physical system. Despite these issues MD can provide very-useful insight into a system’s behaviour[68].

A key advantage of MD is that given a good forcefield it provides accurate data for large systems on reasonable simulation time scales[55]. As a result, MD simulations of polymeric systems have become widespread[69, 70, 71]. The main limitation however is that polymeric systems can evolve over very long time



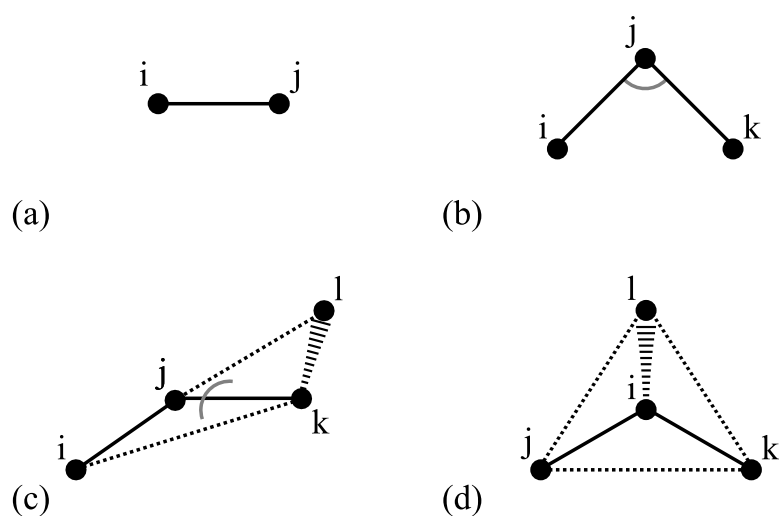
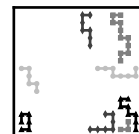


Figure 2.3: A schematic diagram of the various bonded terms in a polymer forcefield of the OPLS-AA form. A bond energy term is associated the distance between i and j in (a). The angle $\widehat{i-j-k}$, shown in grey, in (b) gives rise to an energy term in a MD forcefield. The dihedral angle energy for the system in (c) is given by the angle, shown in grey, between the two triangles defined by $i-j-k$ and $j-k-l$. The improper dihedral angle in (d) is the angle between the two triangles defined by $i-j-k$ and $j-k-l$. This is algebraically the same as a dihedral angle, however, the topological distinction between the two is clear from the arrangements in (c) and (d).



scales[72] that are not currently accessible to large systems in atomistic MD. Therefore, it can be necessary to use techniques such as Monte Carlo to extend the time scales available to modelling. Furthermore, if a forcefield is incomplete, or highly accurate energetics of a small part of a system are required, it is necessary to use a more detailed model for the interactions to obtain either the parameters or the data of interest.

2.3 Density-Functional Theory

Quantum mechanics is a highly-established theory which has provided insight into fields as disparate as encryption[73] and understanding the interaction between organic and inorganic materials in cuttle fish bone[74]. Accurately modelling quantum-mechanical systems¹ such as materials is an ongoing challenge in science. Quantum mechanical methods, in particular DFT, have been widely used in materials modelling to predict and understand the behaviour of materials[75, 76]. This introduction to DFT will provide some context to the complexity of electronic structure, as well as an overview of DFT with particular focus on its assumptions and limitations. A more detailed and in-depth discussion can be found in Refs [77, 78].

2.3.1 The Schrödinger Equation

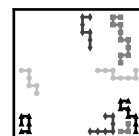
The Schrödinger equation defines the behaviour of quantum objects. It defines the wave function of a quantum system and is incredibly successful at predicting the behaviour of atomic systems. It can be written as:

$$i\frac{\partial}{\partial t}|\Psi\rangle = \hat{\mathcal{H}}|\Psi\rangle, \quad (2.6)$$

where $|\Psi\rangle$ is the state-vector of the system and $\hat{\mathcal{H}}$ is the Hamiltonian operator which defines the energy of the system². The outward simplicity of this equation belies its true subtlety and complexity, $|\Psi\rangle$ includes the details of every component

¹Systems where the underlying interactions are not quantum mechanical generally involve gravity, such as the motion of planets

²Here and throughout this section the unit system is taken to be one in which $\hbar = 1$.



in the system and couples them all together. This fully coupled system is generally intractable, therefore, a series of approximations based on the nature of materials are made before attempting to solve the simplified equation.

The first stage in making an approximation is to consider the system. Materials are composed of atoms, atoms are themselves composed of a nucleus and electrons. The dominant interaction between nuclei and electrons is the Coulomb interaction¹. This makes $\hat{\mathcal{H}}$ a time independent operator, therefore, we can use separation of variables and solve the time independent Schrödinger equation

$$E |\psi\rangle = \hat{\mathcal{H}} |\psi\rangle, \quad (2.7)$$

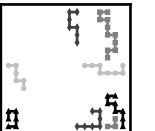
where E is the total energy of the system and $|\psi\rangle$ is the time independent state-vector of the system. The next step is to analyse the motion of the electrons and nuclei, as a proton is ~ 1800 times heavier than an electron. Born and Oppenheimer made the approximation that the motion of the electrons and the motion of the nuclei could be decoupled[77, 79]². The quantum behaviour of the electrons is often referred to as *electronic structure*. This approximation is generally valid and significantly simplifies the process of generating a solution to the time independent Schrödinger equation. The next approximation which is often made is to change the treatment of the nuclei from quantum mechanical to classical. This is referred to as the *classical nuclei* approximation and is widely used³.

Following these approximations the Schrödinger equation has been reduced to a time invariant equation, the solution to which is a wave-function for the electrons which depends parametrically on the positions and charges of the nuclei. Whilst this is a significant reduction in complexity the problem is still exponentially scaling with the number of electrons and as such immensely complicated

¹Electrons and nuclei are relatively light and energetic, therefore, gravitational interactions are neglected, and the *strong* and *weak* forces only apply at very short ranges ($\lesssim 10^{-14}$ m).

² Whilst the original derivation[79] is interesting, readers are advised that it is a little inaccessible as it is written German, and the approximation is only given incidentally whilst attempting to solve a different problem. Therefore, the interested reader is advised to consult e.g., Appendix C of Ref. [77], for a more direct derivation of the approximation.

³Treating the nuclei quantum mechanically is sometimes necessary[80] but is beyond the scope of this work.



to solve[23]. Godby[81] gives the following example to understand the problem of exponential scaling:

storing the occupancy probability on a $10 \times 10 \times 10$ lattice of all 14 electrons of a single silicon atom. A $10 \times 10 \times 10$ lattice contains 10^3 points, therefore, assuming that the occupancy probability corresponding to a single electron can be stored as a double precision number (16 bytes), the data storage for the first electron would be 16×10^3 bytes. As a silicon atom has 14 electrons and the configuration of every electron is coupled to that of every other electron, the second electron requires the same amount of storage as for the first electron for every site the first could be in making the storage requirement $16 \times 10^3 \times 10^3$ bytes. Once all 14 electrons are considered the total storage requirement becomes $16 \times (10^3)^{14} \approx \times 10^{43}$ bytes, or $\sim 10^{33}$ DVD's[81], or a single DVD scaled up to have a diameter of $\sim 10^{16}$ m ≈ 1 light year.

As a result a direct attempt to solve this approximate Schrödinger equation without using additional approximations and mathematical manipulations is unlikely to succeed for a real material system which in principle might involve hundreds or even thousands of electrons. Many techniques exist that address the problem of solving the Schrödinger equation, and none are perfect[82]. DFT, which I have used, maintains an excellent balance between accuracy and computational cost.

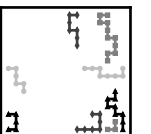
2.3.2 Applied Density-Functional Theory

The electronic structure of a material is determined by the electronic wave-function. As discussed in the previous section, the electronic wave-function is very difficult to obtain directly, even using approximations. DFT addresses this by working with the charge density and a system of non-interacting electrons.

The story of DFT begins with Hohenberg and Kohn (HK) who proved mathematically that[83]:

1. The ground state charge density of a system of electrons is isomorphic¹ to

¹This is a mathematical term describing two sets where there is an invertible mapping from one to the other.



the external potential acting on it¹.

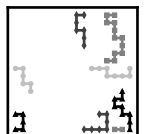
2. There exists a functional of the electronic charge density, valid for any external potential, for which the global minima is the ground state of the system.

These two results are fundamental to DFT, therefore, the proofs are widely reproduced, see e.g., Ref. [77]. The second HK theorem relies on a variational principle proof, however, the principle that there exists an external potential for which the charge density is the ground state (v -representability) does not always hold for every possible trial charge density. Therefore, a solution found by an algorithm which used such densities violating v -representability need not converge to a physical solution. This issue was solved by Levy[84] who showed that it was sufficient for the electronic density to be positive definite and obtainable from an antisymmetric N -body wave-function (N -representability). This weaker condition makes it practical and theoretically sound to use the variational principle to find the ground state charge density.

The HK theorems are existence proofs which give no indication of how this energy functional might be found. Therefore, the work in Ref. [83] might have remained a mere mathematical curiosity rather than becoming amongst the most highly cited research of all time². The reason for this massive interest is the work of Kohn and Sham (KS)[85]. KS proposed solving a system of non-interacting electrons in an external potential formed by the nuclei and the electrons themselves. This approach gives rise to a form for the energy functional of the second HK theorem. Moving into the Schrödinger formalism where the wavefunction is described as a mathematical function and neglecting spin for simplicity, we define a set of N single particle wavefunctions (ϕ_i), which are defined to have the same electron density ($n(\mathbf{r})$) as the time independent state vector ($|\psi\rangle$). These are the antisymmetric N -body wave-functions in Levy's formalism. The time

¹Here and throughout this thesis potential energy functions which differ only by an additive constant are considered degenerate.

²On 2013-08-14, the journal *Physical Review* listed over 15000 articles citing Ref. [83]



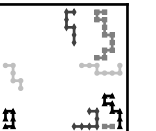
independent schrödinger equation can, therefore, be re-expressed as

$$\epsilon_i \phi_i(\mathbf{r}) = \left(\frac{1}{2m} \nabla^2 + V(\mathbf{r}) + \int d\mathbf{r}' \frac{e^2 n(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} + V_{XC}[n(\mathbf{r})] \right) \phi_i(\mathbf{r}), \quad (2.8)$$

where m is the mass of an electron, e is the charge on an electron, ∇ is the 3 dimensional gradient operator, ϵ_i is the energy associated with ϕ_i , the total energy is calculated as $E = \sum_{i=1}^N \epsilon_i$, and V_{XC} is the exchange-correlation (XC) functional. The KS approach calculates most of the interactions in a mean-field manner, then groups the corrections to the mean-field interactions into the XC functional which is system independent. Furthermore, only the electron-electron interaction and the XC functional are non-local.

The exact XC functional should in principle be independent of the system depending only on the number of electrons and the density. Unfortunately, the exact XC functional is unknown. This might at first appear to be an insurmountable barrier to using this approach. However, in many cases the contribution of the XC functional to the system is small. Therefore, it is possible to make approximations to the exact XC functional. These approximations, whilst by no means perfect, are often surprisingly accurate[86]. Approximations to the exact XC functional fall broadly into four classes: local-density approximation (LDA), generalised gradient approximation (GGA), hybrid, and non-local. LDA[87] is the simplest approximation which models the energy as that of a uniform electron gas of the same density. There are many forms of GGA including: Perdew, Burke, and Enzerhof (PBE)[88], Becke 1988 (B88)[89], and Perdew and Wang 1991 (PW91)[90]. These differ from LDA as they include corrections based on the gradient of the density. The hybrid functionals mix portions of LDA, GGA, and potentially results from other quantum simulation techniques to match various known results[91]. The final class of functional move away from the local (LDA) or semi-local (GGA) approximations and incorporate non-local contributions from the density, so as to include non-local interactions such as vdW[92, 93]. These non-local functionals are comparatively recent developments and much work remains to be done before they will produce accurate forces compared to higher level theories[20].

The work of HK and KS, combined with the use of approximate XC func-



tionals provides a solid theoretical footing for DFT. The next stage is to express the KS wave-functions in terms of a basis of functions. The two most common basis functions are plane waves and gaussians. Plane waves have the advantage that increasing the number of basis functions will improve the precision uniformly; whilst gaussian basis sets sacrifice this systematic improvability in return for their integrals having an analytic form and increased precision near the atoms. As the basis set cannot be infinite, this process introduces the finite basis set approximation[94, 95]. This approximation manifests as a limit to the maximum kinetic energy of the electrons and a finite resolution for the KS wave-functions. This approximation is also present in other forms of quantum mechanical simulation[23, 96]. If a chosen basis set is not able to express the KS wave-function the results of the simulation may be unphysical. Therefore, it is essential to choose the basis set with care.

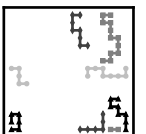
The main reason for using DFT to study the structure of materials is that the bonding between atoms is due to the electrons and electrons are inherently quantum objects. However, not all electrons take part in bonding: generally the outer valence electrons are most important[77]. Therefore, to reduce the number of electrons in the system and hence the necessary basis set size and computational cost, it is beneficial to approximate the interactions of the valence electrons with the core (non-valence) electrons in an approximate manner. The standard approach to this is to replace the Coulomb potential used for the nuclei with a *pseudopotential* which models the interaction due to both the nuclei and the core electrons[77]. This approximation is one that is not strictly necessary to conduct a DFT simulation, however, it was used in all simulations I conducted and is widely used[94].

Thanks to these approximations simulations using $O(10^4)$ electrons[97] are now possible. However, macro-scale systems involving billions of atoms are still well beyond our present abilities. Therefore, periodic boundary conditions are used to approximate an infinite system. This approximation, which is also often used in MD simulations, is widely used and accepted[72, 76, 97].

Whilst this plethora of approximations used in DFT may seem vast the physical motivation behind each of them is clear and the results are astounding. Since the 1990s DFT has become probably the most widely used quantum mechani-



cal simulation technique with thousands of new papers published every year[98]. DFT is able to predict bond lengths and structural parameters with accuracies of typically $< 2\%$ [86]. Furthermore, dramatic progress has been made in topics such as excited states[77, 98] and vdW interactions[20, 92, 93] which are not part of the original theory.



Chapter 3

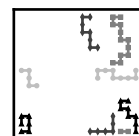
Polymer Membranes

The use of polymer membranes has been widespread for many decades[3, 4, 5]. The primary use of membranes is in separation processes[13, 14, 15]. Membrane separation can be used in areas such as dialysis[99] and reverse osmosis (RO)[100]¹. Industrial membranes are used on a large scale, and to be commercially viable they must be comparatively cheap to manufacture, long lasting, whilst maintaining a high flux of the permeate.

One of the largest issues in membrane systems when deployed in commercial systems is fouling[2]. This topic will no doubt be a significant feature of many further papers and theses concerning membranes as it is considered key to the widespread deployment of membranes[1]. However, it has not been a feature of my research. Therefore, it will not form a significant part of further discussions on membranes in this thesis.

My research has focussed on integrally skinned, asymmetric, phase inversion membranes for OSN applications. Therefore, this chapter will discuss OSN membranes including a discussion of two significant membrane classes, thin film and integrally skinned. It then focuses on the method of phase inversion, before covering the modelling of polymers and polymer membranes.

¹RO is a process for removing salt from water using membranes



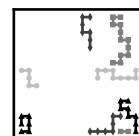
3.1 Organic Solvent Nanofiltration Membranes

OSN is a process for separating molecules or nanoparticles from organic solvents[101]. A significant difficulty in making OSN membranes is that by definition organic solvents weaken the intermolecular interactions in organic molecular systems often causing them to break apart entirely and form a molecular dope. Therefore, to be effective the membrane must be: resistant to the organic solvents in use, mechanically strong so as to survive the pressure gradients used, and able to select the molecules of interest from the other parts of the system.

3.1.1 Thin Film Composite Membranes

Thin film composite membranes are formed in three parts: a structural layer providing mechanical strength, a highly porous support layer, and a thin selective surface layer. This class of membrane is typically used in RO[102], however, it can also be used in OSN[11, 12]. The advantage of this class of membrane is that the structural and support layers can be highly porous as they do not necessarily participate in the filtration[102]. Furthermore, the thinness of the active layer brings with it the advantages of high uniformity and selectivity. However, problems such as delamination where the component layers separate are difficult to avoid[103].

The structural layer is generally a woven mesh which provides only structural support[102]. The manufacturing process for the support membrane is generally phase inversion, which will be discussed in Sec. 3.2. The surface layer is often manufactured by interfacial polymerisation[102], dip coating[104] or, recently, polymers of intrinsic microporosity (PIMs)[11]. For both interfacial polymerisation and dip coating the final porosity is dependent both on the molecules used and the conditions of their growth. In contrast the porosity of PIMs arises from the choice of molecules, which are naturally contorted in such a manner that, as the polymer is deposited on the support layer, it must leave nano-scale pores. They are, therefore, less susceptible to environmental conditions during their manufacture[11]. The multi-stage manufacturing process, making both a support and surface membrane, generally increases the time and cost of their manufacture[104].



3.1.2 Integrally Skinned Membranes

Unlike thin film membranes integrally skinned membranes consist of a single membrane which acts as both support and filter. These membranes fall broadly into two classes: symmetric and asymmetric. Symmetric membranes have a uniform pore structure throughout the membrane, whilst asymmetric have regions with different pore size and structure[105]. For scanning electron microscope (SEM) images of cross-sections through examples of the types of integrally skinned membranes see Fig. 3.1. OSN requires small pores, therefore, integrally skinned OSN membranes typically have an asymmetric structure[6, 7, 8, 9]. The smallest pores in an asymmetric membrane are often smaller than those in an equivalent symmetric or thin film membrane, reducing the flux; however, the larger pores in the remainder of the structure compensate for this generally resulting in a flux somewhere between equivalent symmetric and thin film membranes.

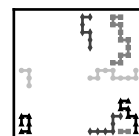
The advantage of integrally skinned membranes is that they are cheap to manufacture and are not subject to problems such as delamination. However, membranes with graded pore structures can suffer from reduced flux of the permeate[12], i. e. it is more difficult for the permeate to pass through the membrane than if it had a more open support structure.

Asymmetric membranes can be further classified according to their substructure: uniform-pore¹, graded-pore, and finger-pore; see Fig. 3.1. The membranes of interest in my research have a finger-pore structure[6]. However, much of the discussion of asymmetric integrally skinned membranes applies to all three classes of asymmetric membranes.

3.2 Phase Inversion

The phase inversion process of membrane manufacture is one of the oldest methods of manufacturing membranes[3]. It is still widely used[6, 102] as it is simple to use in large scale manufacturing processes. The process can be described in terms of three component system: P, S_A, and S_B, where the polymer P is soluble

¹The uniformity refers to the structure of the pores behind the surface layer



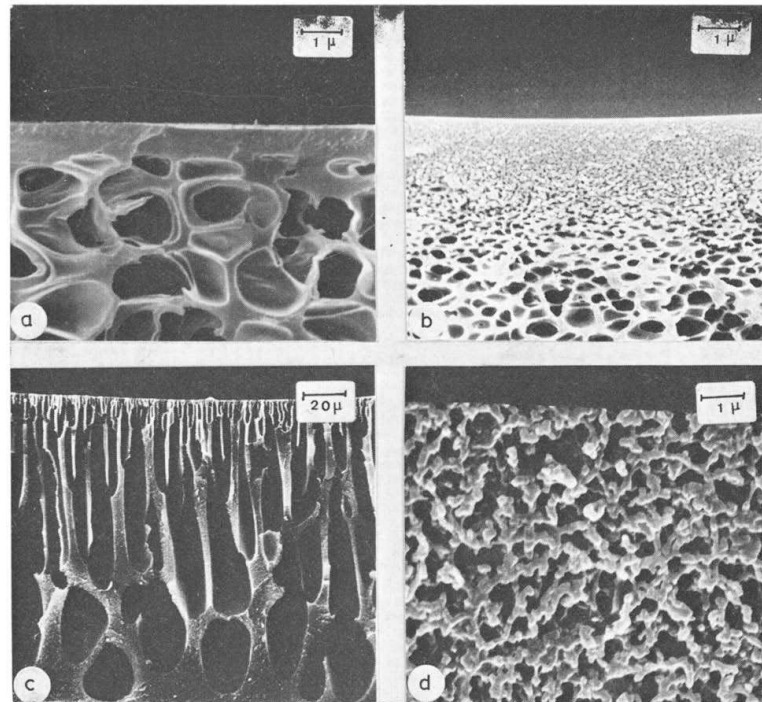
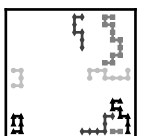
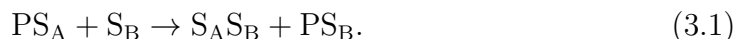


Figure 3.1: Scanning electron micrographs of membrane cross sections with typical structures: a) Asymmetric membrane with uniform-pore substructure; b) Asymmetric membrane with a graded-pore substructure; c) Asymmetric membrane with a finger-pore substructure; d) Symmetric microporous membrane without a skin. Reprinted from *Desalination*, 21, H. Strathmann and K. Kock, The formation mechanism of phase inversion membranes, 241–255, 1977, with permission from Elsevier, license number 3393141117308.



in solvent S_A but not soluble in solvent S_B , and solvent S_A is highly miscible with solvent S_B . A *dope* of polymer P in solvent S_A is spread onto a backing layer¹. The *dope* on the backing layer is then immersed in a large amount of solvent S_B . The entropy of mixing causes a chemical potential gradient which drives solvent S_A out of the polymer P into the bath of solvent S_B , whilst simultaneously driving solvent S_B into the polymer matrix left behind by solvent S_A . Since the polymer P is not soluble in solvent S_B it forms a solid membrane and the remaining solvent S_B can be rinsed or evaporated from the membrane [6, 105]. Solvent S_B , therefore, acts as a precipitant for the the polymer P . The process can be written as



The swapping of the solvent surrounding the polymer P is termed *phase inversion*.

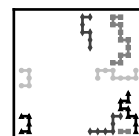
For a phase inversion membrane to be useful in OSN the polymer used must be chemically resistant to the solvents used in the separation process. Recently, therefore, polyimide macro molecules such as P84[6, 7, 8, 9] and Matrimid[7, 8, 9, 106, 107, 108] have become popular. To make these polymer membranes practical it is necessary to be able to tailor their porosity to the target system.

3.2.1 Controlling the Porosity

There are many ways to control the porosity of a membrane. One of the most popular techniques is to alter the time allowed for evaporation between spreading the polymer dope on the backing material and immersing it in the precipitant bath[107, 109]. However, recently it has been shown that the final porosity of P84 phase inversion membranes does not vary with evaporation time or with the volatility of the solvents used[7, 8, 9]. Conversely, it has been shown that for these membranes the final porosity is strongly dependent on the solvent used to produce the initial polymer dope[6, 7, 8, 9].

For P84 membranes the initial solvent is usually a mixture of two solvents dimethylformamide (DMF), which is polar, and 1,4-dioxane (dioxane), which is non-polar. DMF has been shown to be a *good* solvent for P84, whilst dioxane is a *poor* solvent[7, 8, 9]; in this context *good* implies that the polymer will spread out

¹The backing layer provides support until the membrane is formed when it can be removed.



so as to maximise its surface area in contact with the solvent whilst *poor* implies that the polymer will do the converse and minimise its surface area in contact with the solvent¹[7, 8, 9, 110]. By varying the ratio of DMF to dioxane in the initial dope it is possible to change the porosity of the membrane; increasing the proportion of DMF results in a membrane that is more porous and allows larger molecules to pass through, whilst increased amounts of dioxane results in a more dense membrane that is only porous to smaller molecules[6, 7, 8, 9].

The porosity of a membrane is generally defined by its rejection curve. Rejection is defined by

$$R(\%) = 100 \left(1 - \frac{C_r}{C_p} \right), \quad (3.2)$$

where C_r and C_p are the concentrations of the target molecule in the retentate and permeate respectively[111]. Rejection is generally plotted for molecules of different weights under the conditions of either constant pressure or constant flux. Example rejection curves for polystyrene (PS) of various molecular weights² are shown in Fig. 3.2, from which the effect of the initial solvent ratio can be clearly seen[6]. In actuality the rejection curve characterises the porosity and pore connectivity as these quantities are inherently linked. Often models for the pore size make assumptions such as the pores being uniform cylinders through the membrane[112, 113]. However, the nano-scale structure of a membrane is difficult to establish, and is one of the topics of this thesis.

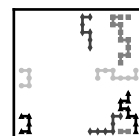
3.2.2 Imaging Membrane Structures

In the early days of phase inversion membranes, the membranes were thought to have a $> 50\%$ failure rate³. This anomalously high failure rate was eventually discovered to be due to the asymmetry in phase inversion membranes[114]. Had the original manufacturers been able to image the nano-scale structure of the

¹ Due to the polar nature of DMF the solubility parameter often used to describe solvent quality is the Hansen solubility parameter. This is actually a set of three parameters covering dispersion δ_d , dipolar δ_p , and hydrogen bonding δ_h interactions[110].

²PS in this context refers to styrene oligomers of a specific molecular weight not polymeric styrene of unknown molecular weight.

³The membranes were experimentally determined to be permeable to both salt and water and pure water.



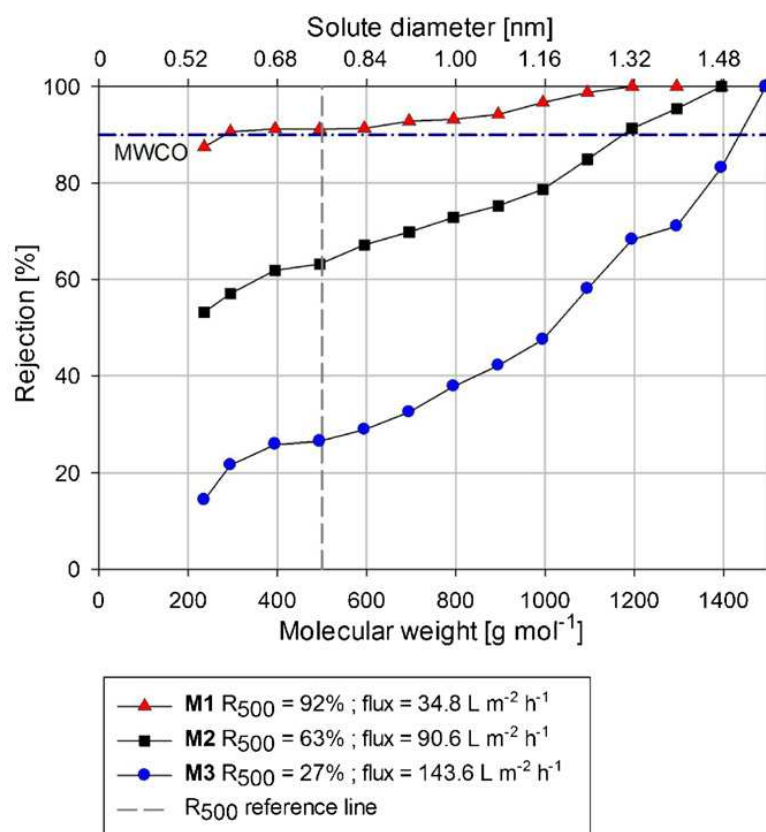
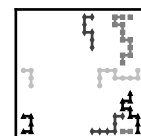


Figure 3.2: Rejection curves for PS oligomers of different molecular weights at concentrations of 1 gL^{-1} in toluene for three P84 membranes. The membranes **M1**, **M2**, and **M3** were manufactured from 24 wt% P84 in DMF:dioxane ratio of 1:2, 2:1, 4:1 respectively. The experiments were carried out under constant flux. For complete experimental details see the original paper, Ref. [6]. Reprinted from Journal of Membrane Science, 413–414, Joanna Stawikowska and Andrew G. Livingston, Nanoprobe imaging molecular scale pores in polymeric membrane, 1–6, 2012, with permission from Elsevier, license number 3393131159500.

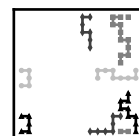


membranes this might have been understood almost immediately.

Experimental imaging on the nano-scale is challenging as visible light has a wavelength of 380–750 nm. Therefore, visible light cannot currently resolve objects smaller than hundreds of nanometres in size[10, 115]. Atomic force microscopy observes surfaces with nanometre resolution by moving a cantilevered tip over it and monitoring deflections of the cantilever. However, this approach yields details only about the surface and has a number of limitations including the inability to resolve features smaller than the tip and the potential to damage the sample during the characterisation[116]. An alternative technique is electron microscopy, which can resolve objects on a sub-nanometre scale. It requires atoms with a large electron scattering cross section¹, which makes imaging carbon based polymer nano-structures difficult[117]. To circumvent this, nanoparticles of heavy metals, such as gold can be used to coat the surface of cross section slices through membranes which can then be imaged by SEM to provide insight into the micrometre scale structure of membranes[6, 105], see Fig. 3.1. However, these images whilst useful in understanding membranes do not show the pores used in nanofiltration as these are on a far smaller scale[6]. Furthermore, the high energy of the incident electrons in transmission electron microscope (TEM) can damage polymer samples which has the potential to significantly influence the observation[117].

Recently sub-nanometre resolution scanning transmission electron microscope (STEM) images of phase inversion membranes have been produced[6]. These images were generated by infusing samples of the membrane with osmium dioxide (OsO_2) nanoparticles then imaging the nanoparticles, which have high atomic numbers and perform a similar function to the gold particles used in SEM above. This process allows the characterisation of the size of OsO_2 clusters in the membrane, and thereby gives an indication of the pore size and distribution. However, there is no guarantee that the OsO_2 has reached all the available pores, and it gives no indication of the form of the pathways through the membrane. Furthermore, the OsO_2 may be invasive meaning that the structure is significantly altered by its presence. Another issue is that the filtration is inherently dynamic, the permeate may induce swelling during operation, and the pressure gradient used in

¹i.e. high atomic numbers.



filtration are not included, therefore, the structures might not be representative of the membrane in operation[6]. Any of these effects have the potential to affect the pore size and distribution relative to these experimental results. Despite this, these STEM images show different distributions of pore sizes for different initial solvents and will be discussed in the context of my model in Chap. 8.

3.3 Modelling Polymers

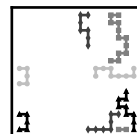
Due to the difficulties in directly imaging and characterising polymers, modelling has the potential to provide insight into polymers which cannot be readily achieved through experiments. The modelling of polymers and other macro-molecular systems has been a significant challenge in science over the last century. A significant difficulty is the large conformational space of macro-molecular systems. This combines with the wide range of length and time scales to make these systems difficult to model theoretically or computationally.

In general polymers exist either in solution, amorphous, or crystalline states with no solvent present. Polymers in solution are subdivided into dilute or concentrated regimes, and the solvents are characterised as *good*, or *poor*. The phase inversion membranes which are of the topic of my research are formed from a concentrated polymer dope where the solvent is a mixture of *good* and *poor* solvents[6]. However, as models for polymers in dilute and concentrated solutions are related both will be discussed. The next subsections cover analytic and computational models for polymers in dilute and concentrated solutions. The topics of crystalline and amorphous phases of polymers as well as polymer melts are covered in Sec. 3.4.

3.3.1 Dilute Polymer Solutions

When considering dilute polymers it is important to recognise that these regimes represent configurations with very low densities of polymers¹.

¹ When modelled as a self-avoiding random walk, the crossover from concentrated to semi-dilute occurs when the volume concentration of monomer units approaches $N^{1-3\nu}$, where ν is the scaling exponent of a radius of gyration of a polymer, and N is the number of monomer units[52]. For a system composed of 100 segment oligomers in solution, and using the approximation



At the simplest level, models of dilute polymer solutions do not include solvent-polymer, inter- or intra-polymer interactions. One such model is known as the *bead spring* model[52]. This assumes the polymer segments do not interact and have an energy given by a harmonic potential associated with their length. This model can be solved analytically for many properties. However, the absence of excluded volume effects due to the lack of inter- and intra-polymer interactions results in the *bead spring* model underestimating the scaling exponent (ν) of the size of polymeric systems with increasing numbers of segments[52]. Excluded volume can be added to the model in a mean field manner, resulting in reasonable agreement with numerical simulations of self-avoiding polymers[52].

A common dilute solution polymer model is the Rouse model[118]. This uses the *bead spring* model as its representation of a polymer and incorporates Langevin dynamics to model the effect of the solvent. In its original form does not accurately reproduce properties such as the scaling of the diffusivity with polymer length. However, a correction to the dynamics such that interactions between neighbouring beads are included results in the correct scaling[119]. Simulations of self-avoiding *bead spring* polymers, using this model have been shown to produce accurate predictions for dilute solutions[120].

Isolated polymers can be simulated with on and off lattice MC techniques[35, 121, 122] using advanced derivatives of the RR procedure, e.g., the pruned-enriched Rosenbluth method (PERM)[123]. Other groups have used MD simulations of both AA[124] and CG[125] representations to study polymers. Probably the largest amount of work on the simulation of dilute macro-molecular systems is in biology where MD has been extensively used to analyse the behaviour of proteins and other biological macro-molecules[38, 55, 64, 126]. Another interesting recent area of research is modelling which allows the resolution to be altered during the simulation, thereby allowing both AA and CG molecules in a single simulation[127]. This hybrid approach is likely to have a big impact in the study of polymers in dilute solution in the future as it offers significant computational cost reductions for the solvent-solvent interactions which dominate this type of system[127]. However, when studying concentrated polymer systems the solvent-solvent interaction ceases to be dominant and, therefore, the advantage of these

$\nu = 0.6$ for the scaling exponent, the volume concentration for the crossover is 2.5%.



adaptive resolution techniques is reduced.

3.3.2 Concentrated Polymer Solutions

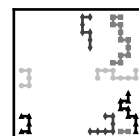
Many industrial processes, including phase inversion, use concentrated polymer solutions[6]. Therefore, a large effort, both theoretical and computational, has been exerted to study these systems[125, 128, 129, 130].

Anyone who has ever tried to separate out a tangled mass of strings¹ has discovered the motion of a single string is coupled to that of the others. Furthermore, the longer the strings the more difficult it is to separate them, and the more the motion of one string is blocked by other strings. This is a common problem when handling extended objects such as strings and polymers. The reptation model provides a mechanism for the motion of such extended objects and correctly predicts how the diffusivity alters with length[131]. This model predicts the experimentally observed scaling of the diffusion rate for polymers in concentrated solutions[120] and provides insight into the motion of polymers in concentrated solutions[132, 133]. However, it does not explain the mixing of polymer solutions.

Thermodynamics is a highly established tool for studying mixtures of different types of fluids, and it was used to create the Flory-Huggins (FH) model for polymer solvent solutions[128, 129]. This model is essentially an analogue to the ideal solution model for fluids[134] with a correction for the polymer solvent interaction which is characterised in the FH parameter (χ_{FH}). This correction transforms the model from an ideal solution model into a *regular solution* model[134]. All the polymer solvent interactions in the model are encoded in χ_{FH} . Calculating χ_{FH} can be accomplished using MD simulations to find the enthalpy of solvation[135, 136, 137]; it can also be determined experimentally using calorimetry[135]. The use of the FH parameter in lattice models of polymers is discussed further in Chap. 8.

A principal difficulty in studying polymers in solution is that even for concentrated solutions treating the solvent explicitly incurs a large computational cost. Implicit solvation models have been developed for systems such as proteins

¹This often forms if multiple lengths of string are left uncoiled in a pot in the kitchen drawer.



in water, but these do not always reproduce the same dynamics or free energy landscapes as explicit solvent models[138]. Furthermore, implicit solvation models generally only simulate the effect of good solvents and are specific to a given solvent molecule[139]. Membranes and polymer melts, in contrast, do not generally include solvents, therefore, their dynamics can often be studied without this complexity.

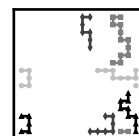
3.4 Modelling Membranes

There are two primary topics in the modelling of a membrane: modelling its structure and its performance. The focus of my research has been developing a model for the structure of a membrane. To understand the importance of generating membrane structures it is sensible to consider not only the current goal of understanding the generation of membrane structures, but also future applications of these structures. There follows a brief overview of membrane performance prior to a discussion of existing procedures for generating and characterising membrane structures.

3.4.1 Membrane Performance

In the context of filtration membranes performance is characterised by two quantities: the flux and the rejection. These factors depend on the molecules being separated as well as the membrane being used[6, 7, 8, 9]. There are numerous models for the passage of molecules through membranes[112, 113, 140, 141]. As this is not the focus of my research only two models will be discussed here, the non-equilibrium molecular dynamics (NEMD) method[141], and small molecule gas diffusion[140].

The NEMD approach of Frentrup *et. al.* [141] simulates fluid flow through a membrane by placing a sample of membrane in an elongated periodic box. The sample is kept at constant temperature and its position restrained. The box is packed with the fluid at the desired temperature, but with no thermostat applied. On one side of the membrane an acceleration towards the membrane is applied to fluid molecules in a region of the simulation box. The result is a flow



of molecules towards the membrane, the passage of which can be observed[141]. Clearly an accurate representation for the membrane structure would help this model provide reliable information about the flow of fluids through it.

Diffusion-based schemes do not attempt to model the flow of molecules through a membrane, but rather the diffusion of molecules between pores. To do this they use a sample of the membrane which fills a periodic box. A small molecule is inserted in a pore in the membrane, and the system is integrated forward in time using MD. During the simulation the membrane is sometimes treated as being rigid, which will clearly affect the results but offers considerable computational savings. When the inserted molecule moves between two pores a large change in its mean squared displacement (MSD) is expected. Analysing these inter pore movements provides insight into the diffusion of the particle through the membrane[140, 142].

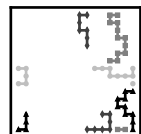
The diffusion rate (D) over an energy barrier, of height E , is governed by the Arrhenius equation[140, 142],

$$D = D_0 e^{-\beta E}, \quad (3.3)$$

where D_0 is the pre-exponential factor which includes a contribution from the activation entropy. As the energy in MD simulations is dependent on the positions of the atoms in the system, the resulting diffusion rate will be strongly reliant on the membrane structure used and whether it is permitted to move during a simulation.

3.4.2 Membrane Structure

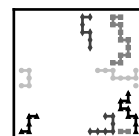
There are many techniques for building conformations of polymers, which are expected to be representative of membranes[48, 140, 142]. A common approach used in membrane simulations is to build a model for the polymer in bulk and then assume that this is representative of the membrane[48]. This approach is often undertaken by generating initial conformation based on RR sampling without using the RR weighting factors, see Sec. 2.1.2.1. Thermostats and barostats are applied to equilibrate the polymer to the target temperature and density[140, 142]. However, since the RR sampling procedure produces incorrect sampling



unless care is taken and the correct weightings used[37, 39, 40], the resulting conformations may be highly biased.

An approach which produces a more realistic conformation for the equilibrium structure of a polymer uses *pivot moves* in MC simulations of the polymer molecule to generate an initial structure for the system[48, 143, 144, 145, 146, 147, 148, 149]. This approach also permits the introduction of free surfaces which allow the polymer to either stand up, or lie down at the surface. The result is a structure which can reflect the physical structure of a membrane. For further details consult the review included in the tutorial by Neyertz[48]. One issue with membrane structures created by this approach is that even when, as in Ref. [149], more than 10^5 atoms are used the resulting membrane is only around 10 nm thick. Such a simulation would treat only 10–20% of the total depth of the nano-porous region of a P84 membrane, and the molecular weights used to generate P84 membranes are three times larger than those in Neyertz’s simulation[6, 149]. To build a more realistic model for P84 using this technique would be extremely computationally demanding. Another issue with using this technique is that it incorporates no solvent effects into the structure. However, as discussed earlier, solvent effects have been shown to be dominant in determining the structure and properties of P84 membranes[6, 7, 8, 9].

The relationship between solvent interactions and porosity seen in P84 phase inversion membranes, is explained by considering the process as a three phase system[105]. This view suggests that the structure will be determined in part by the reaction kinetics, so that it will not necessarily be an equilibrium structure for the polymer[105]. This is consistent with the observation that P84 membranes become non-porous when they are annealed[6, 7, 8, 9]. Any realistic model for the structure of these P84 membranes will need to include a consideration of both the solvent and the kinetics of the phase inversion process. It is clear that the latest experimental results for P84 OSN membranes are not explained by the current models of membrane formation used in manufacturing[6, 7, 8, 9, 10]. Furthermore, the latest models typically neglect the solvent-polymer interaction which appears dominant in the formation process.



Chapter 4

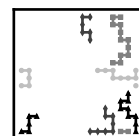
Coarse Graining

The following research is an expansion of the work presented in Ref. [16], including a discussion of some interesting physics resulting from our coarse graining approach.

The dynamics of polymers span a wide range of time scales. Carbon-hydrogen covalent bonds have a vibration period of ~ 10 fs[150], while macromolecules such as proteins undergo conformational changes over microseconds[72]. In practice, the maximum time scale accessible to molecular dynamics (MD) is limited by the time step required to capture the fastest degree of freedom in the system. It is desirable, therefore, to include only the degrees of freedom that are physically relevant to the problem under consideration.

In polymers the fastest oscillations are usually associated with covalent bonds between heavy atoms (e.g., carbon, nitrogen, oxygen) and hydrogen. However, conformational changes are always associated with considerably longer time scales. Consequently these fastest oscillations are often suppressed by applying constraints to fix bond lengths at their equilibrium values. This is usually accomplished by the SHAKE[151, 152] or LINCS[153, 154] constraints algorithms. Often, it is desirable to go further and to constrain other bond lengths and angles, enabling time steps of up to 2 fs[155]. While this approach can result in an increase in the duration of time that may be simulated, parallel scalability and algorithm stability can be affected when the constraints are highly coupled[154].

An alternative approach to applying constraints is the use of generalised-coordinates representing rigid multi-body elements[156], whereby atoms are grouped

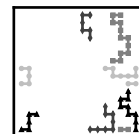


into rigid dynamical units. The removal of fast degrees of freedom may also be achieved through the introduction of “virtual sites” for the hydrogen atoms, whereby the position of each hydrogen atom is defined by the positions of nearby heavy atoms, which can allow time steps of up to 7 fs[150]. A drawback of this approach is that the moment of inertia tensor of the molecule is not conserved[150] (e.g., the principal values for benzene can have errors greater than 10%) which may affect the large scale dynamics of the system.

In this chapter, I present a coarse graining approach that addresses this limitation for the case of aromatic groups in polymers. Each aromatic group is mapped onto a rigid triangle, defined by three vertices, *in a manner that conserves key dynamical quantities*: namely the total mass, the centre of mass, and the moment of inertia tensor. As a result, this triangle has the same dynamical response as a rigid, planar aromatic group. The fast degrees of freedom associated with bond vibrations and out-of-plane buckling modes *within* each group are removed. At the same time the slower degrees of freedom, which are responsible for conformational changes and associated with variations in bond and torsion angles *between* adjacent groups are retained. The fixed relationship between the coordinates of the vertices of the triangle and the true atomic positions associated with the aromatic group uniquely determines the force and torque on the triangle for a given set of positions and a given AA force-field. My approach is a means of implementing coupled rigid body dynamics[157], but critically does not require altering the core integration routine of an existing MD code. This CG method is equally applicable to aromatic groups that form the polymer backbone as well as those present in side chains.

I have tested the approach across a wide range of temperatures and system sizes for polyether ether ketone (PEEK), a widely used industrial polymer[158, 159, 160]. Structural and dynamical properties are found to be in excellent agreement with both AA and all-atom rigid bonds (AA-RB) simulations, and the increased MD time step (~ 5 fs) enabled by the method results in a five-fold increase in the time-scale of a given simulation as compared to AA approaches.

The reduction in degrees of freedom has the advantage that various intramolecular bonded interactions are constant. Constant interactions in MD result in a shift in the potential energy but do not alter the dynamics, therefore, when



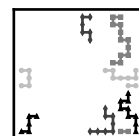
parametrising a force-field these terms can be neglected. As additional parameters require more data when fitting a force-field, reducing the necessary parameters is clearly advantageous.

The remainder of this chapter is organised as follows. An introduction into the principles of rigid dynamics and their application to coarse graining aromatic polymers is presented in Sec. 4.1 to obtain a CG representation of planar units. The application to PEEK is used to illustrate this approach. An overview of constraint algorithms and the advantages of reduced degrees of freedom when fitting force-fields is provided in Sec. 4.1.3 and Sec. 4.1.5. A comparison of the results obtained from AA, AA-RB, and CG simulations of PEEK for a variety of polymer lengths and temperatures, is given in Sec. 4.2. The chapter concludes, with an analysis of the coarse graining technique and its applicability to other aromatic polymers, in Sec. 4.3.

4.1 Method

Considerable computational savings may be realised by removing the fast degrees of freedom normally associated with planar units in molecules, such as aromatic groups. Any rigid two-dimensional object has six independent quantities that govern its dynamics. For a planar object in the x - y plane, where x and y denote Cartesian axes, these are the position coordinates associated with the centre of mass, $\mathbf{R}^0 = (X^0, Y^0)$, the total mass M_T , and the three independent components of the moment of inertia tensor \mathbf{I} , denoted I_{xx} , I_{yy} and I_{xy} ¹. As a set of rotations and translations can always be found such that a planar object lies in the x - y plane this applies to all planar objects. Any two rigid bodies having these properties in common will exhibit the same dynamical response when subjected to the same set of forces. The coarse graining approach, described below, explicitly conserves these properties.

¹In accordance with the perpendicular axis theorem $I_{zz} = I_{xx} + I_{yy}$ and $I_{xz} = I_{yz} = 0$.



4.1.1 Two Dimensional Rigid Bodies

Consider a two-dimensional rigid object composed of N atoms¹. Let the x and y axes be aligned with the principal axes of \mathbf{I} and let the origin be at \mathbf{R}^0 , without loss of generality. Therefore,

$$\mathbf{R}^0 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad (4.1)$$

$$M_T = \sum_{i=1}^N m_i, \quad (4.2)$$

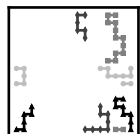
$$\mathbf{I} = \begin{pmatrix} \sum_{i=1}^N m_i y_i^2 & 0 \\ 0 & \sum_{i=1}^N m_i x_i^2 \end{pmatrix}, \quad (4.3)$$

where the i^{th} atom has position coordinates (x_i, y_i) and mass m_i . In order to evolve the positions of this rigid body using MD a naive approach might be to apply a set of constraints among the atoms such that only the rigid body degrees of freedom remain. However, this introduces $3(N-2)$ constraints, the enforcement of which becomes unstable as N increases owing to the coupled nature of the constraints. Furthermore, constraints on light atoms such as hydrogen become unstable as the time step increases[150]. It is desirable, therefore, to develop a coarse graining scheme that exhibits stability at large time steps while preserving dynamical properties.

4.1.2 Coarse Grained Representation of Aromatic Polymers

Consider the case of a para-substituted aromatic group in a polymer backbone, shown schematically in Fig. 4.1. An example of a polymer that exhibits this structural feature is the macromolecule PEEK, shown in Fig. 4.2(a). Assuming planarity, rigidity and symmetry, and using the notation of Eqs. (4.1)-(4.3), the

¹In principle these ‘atoms’ could be any point masses, but to avoid confusion they will be referred to as atoms henceforth.



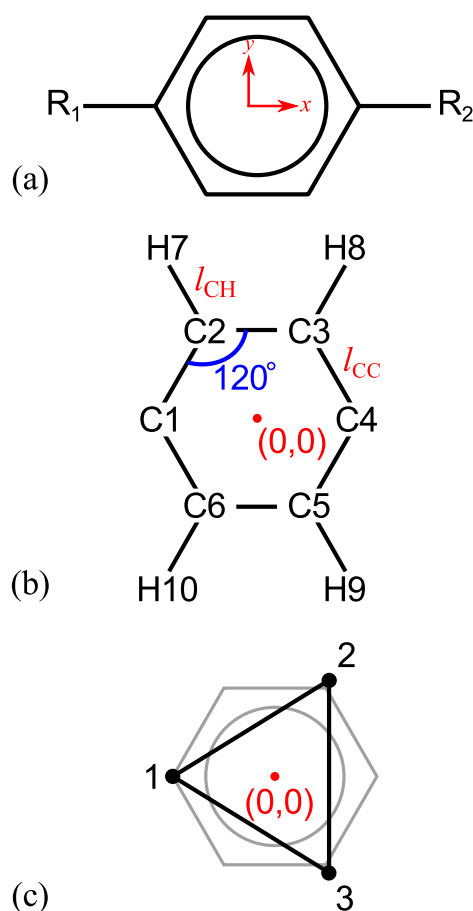
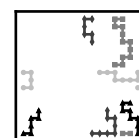


Figure 4.1: A schematic demonstration of the coarse graining procedure of a para-substituted aromatic group in an organic molecule. (a) shows the coordinate system and chemical formula for the aromatic group with the remainder of the aromatic molecule indicated by R_1 and R_2 . (b) shows the relevant bond angles and lengths where l_{CC} and l_{CH} are the equilibrium lengths of the carbon-carbon and carbon-hydrogen bonds. The atoms in this AA representation are labelled for reference. (c) shows the resulting triangle of the CG representation with the vertices labelled 1, 2, and 3 to match the text. The underlying molecule in grey is the AA group being replaced. All three figures are to scale using the parameters from the OPLS-AA force-field[38]. The positions and masses of the triangle vertices in (c) are given in Tab. 4.1. The orientations of the triangles in the PEEK monomer are arbitrary as it would have been equally possible to place vertex 1 at the position of C4 at the beginning of the coarse graining procedure. Fig. 4.2 shows the orientation used in all CG simulations.



AA representation of the aromatic unit (Fig. 4.1(b)) is characterised by

$$\mathbf{R}^0 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad (4.4)$$

$$M_T = 6m_C + 4m_H, \quad (4.5)$$

$$I_{xx} = 3m_C l_{CC}^2 + 3m_H (l_{CC} + l_{CH})^2, \quad (4.6)$$

$$I_{yy} = 3m_C l_{CC}^2 + m_H (l_{CC} + l_{CH})^2, \quad (4.7)$$

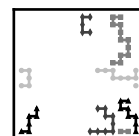
$$I_{xy} = 0, \quad (4.8)$$

where m_C and m_H are the masses of a carbon and hydrogen atom, respectively, and l_{CC} and l_{CH} are the lengths of a carbon-carbon and carbon-hydrogen bond in the unit, respectively. The choice of coordinates in Fig. 4.1(a) is convenient as these axes are the eigenbasis for \mathbf{I} , as can be seen from Eq. (4.8).

The six independent quantities expressed in Eqs. (4.4)-(4.8) can be conserved exactly by a CG representation composed of three point masses at the vertices of a triangle (Fig. 4.1(c)). These vertices, which I label $\alpha \in \{1, 2, 3\}$, are associated with *nine* parameters: position coordinates (X_α, Y_α) and masses M_α . However, since there are only *six* independent constraints embodied in Eqs. (4.4)-(4.8), there is not a unique solution.

Further constraints are added by the requirement that the group be connected via distance constraints to the rest of the organic molecule, which leads to the convenient choice that one of the vertices of the triangle should coincide with the position of an atom connecting the aromatic group to the rest of the polymer. Placing vertex 1, therefore, directly at the position of the carbon atom labelled C1 in Fig. 4.1(b), thus specifying $X_1 = -l_{CC}$ and $Y_1 = 0$, reducing the number of free parameters associated with the vertices of the triangle to seven.

Furthermore, by symmetry, $X_3 = X_2$, $Y_3 = -Y_2$, and $M_3 = M_2$, thus simultaneously satisfying Eq. (4.8) and the requirement that $Y^0 = 0$ (Eq. (4.4)), while also reducing the number of free variables associated with the CG representation



to four:

$$X^0 = \frac{-M_1 l_{CC} + 2M_2 X_2}{M_T}, \quad (4.9)$$

$$M_T = M_1 + 2M_2, \quad (4.10)$$

$$I_{xx} = 2Y_2^2 M_2, \quad (4.11)$$

$$I_{yy} = l_{CC}^2 M_1 + 2X_2^2 M_2. \quad (4.12)$$

Solving these equations for M_1 , M_2 , X_2 and Y_2 in terms of the known AA quantities (Eqs. (4.4)-(4.8)), yields

$$M_1 = \frac{M_T I_{yy}}{I_{yy} + l_{CC}^2 M_T}, \quad (4.13)$$

$$M_2 = \frac{1}{2}(M_T - M_1), \quad (4.14)$$

$$X_2 = l_{CC} \left(\frac{M_T}{2M_2} - 1 \right), \quad (4.15)$$

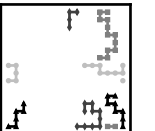
$$Y_2 = \sqrt{\frac{I_{xx}}{2M_2}}. \quad (4.16)$$

Taking equilibrium parameters for the AA representation from the OPLS-AA force-field[38] results in the positions and masses associated with the CG representation given in Tab. 4.1.

The CG representation shown in Fig. 4.1(c) and defined by the parameters given in Tab. 4.1 has identical dynamical properties to the original rigid AA aromatic group of Fig. 4.1(b). However, in order to use the CG representation in an MD simulation of a polymer such as PEEK, it is also necessary to constrain the ‘bond lengths’ associated with the CG representation. For the central aromatic group in the PEEK monomer, shown in Fig. 4.2(b), these constraints are denoted 1–2, 1–3, 2–3, 1–O₁, and 1–O₂ and are enforced using the SHAKE[151, 152] algorithm.

4.1.3 Constraint Algorithms

Constraints are used for a variety of purposes in simulations, however, the most widely used is to fix bond lengths in MD. The discussion of constraints presented



Vertex α	X_α (nm)	Y_α (nm)	M_α (g mol ⁻¹)
1	-0.140	0.000	25.871
2	0.072	0.133	25.114
3	0.072	-0.133	25.114

Table 4.1: The positions (X_α, Y_α) and masses M_α of the vertices α of a triangle with the same dynamics as a rigid aromatic group along the backbone of PEEK.

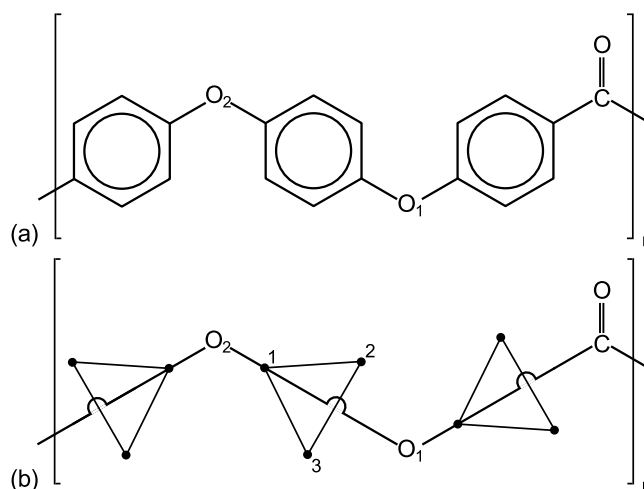
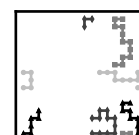


Figure 4.2: The chemical structure of PEEK, and the corresponding CG representation. The suffixes on the oxygen atoms are for identification purposes only — they do not indicate molecular or atomic oxygen. The constraints used for the middle triangle connect the pairs: O₂-1, 1-2, 1-3, 2-3, and 1-O₁. The constraints for the other triangles are analogous to this, and the oxygen in the ketone group is connected with a single constraint to the carbon atom.



here will be restricted to the case of linear constraints; however, some of the techniques described herein are applicable to other forms of constraints. Linear constraints are very widely used in molecular simulations and are an interesting mathematical and computational challenge in MD.

A linear constraint in a simulation is any constraint which can be written as:

$$|\mathbf{r}_\alpha - w_{\alpha\beta}\mathbf{r}_\beta|^2 - l_{\alpha\beta}^2 = 0. \quad (4.17)$$

Where $\mathbf{r}_\alpha, \mathbf{r}_\beta \in \mathbb{R}^3$ are the positions of two objects within the system¹, l_{ij} is the constraint length, and $w_{\alpha\beta}$ is a weighting between the two positions. Please note the Einstein summation convention is not used in this section. In MD simulations $w_{\alpha\beta} = 1$ and will be taken as such from this point forward therefore the linear constraint equation becomes

$$\sigma_{\alpha\beta}(\mathbf{r}_\alpha, \mathbf{r}_\beta) = |\mathbf{r}_\alpha - \mathbf{r}_\beta|^2 - l_{\alpha\beta}^2 = 0. \quad (4.18)$$

There are many algorithms used to implement constraints in MD. The majority use either generalised-coordinates[156] or Lagrange multipliers[151, 152, 153, 154, 161, 162, 163] to permit numerical integration of Eq. (2.4).

The generalised-coordinate approach works in one of two ways which are mathematically equivalent. The first is to take the position vectors of all the objects in the system ($\{\mathbf{r}_\alpha\}$) and express them as a single global position vector ($\mathbf{R} \in \mathbb{R}^{3N}$). The basis of $\mathbf{R} \in \mathbb{R}^{3N}$ is simply the combination of the individual basis vectors of the original coordinates. There is now a vector in this space which defines the separation between objects α and β . Therefore, that vector can be made one of the basis vectors of the system and the positions of everything in the system can be re-expressed in this new basis. The constraint along this vector can then be enforced by either removing the components of the force and velocity along this vector or removing the vector from the system. Updating the positions and velocities with a standard MD integration scheme results in the constrained positions and velocities. The disadvantage of this scheme is that the basis vectors must be recalculated at every step of the MD simulation and a naive implementation of

¹If this is a CG the positions could in principle be for either atoms or vertices.



the algorithm scales cubically with the number of objects in the system.

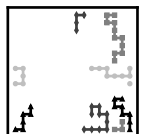
The alternative generalised-coordinate approach is that adopted in the parallelizable open source efficient multibody software (POEMS)[156], where the rigid constraints are evolved explicitly as multi-body elements. This works by explicitly defining the position of one part of the system relative to another, e.g., to constrain a diatomic molecule the position of one atom could be specified in normal Cartesian coordinates, the second atom could then be expressed in polar coordinates with a fixed radius centred on the first atom. The result of this is that the basis automatically evolves with the MD integration. However, calculations of the forces becomes more complex as this often requires calculating separations which requires reference to the global position scheme which can require a complicated remapping for systems involving many coupled constraints[156]. Both these generalised-coordinate approaches allow more complicated constraints than the simple linear constraints of Eq. (4.17), however, that is beyond the scope of this work.

The Lagrange multiplier approach generates a constraint force which corrects an unconstrained integration step such that after the force has been applied to the unconstrained step the integration has exactly followed the constrained path. To understand where the various Lagrange multiplier based constraint algorithms¹ break down it is necessary to understand their derivation. The derivation presented here follows the structure given in Ref. [162] which provides an overview of both the SHAKE, and RATTLE algorithms as well as the MILC-SHAKE algorithm. Throughout this section the equations will be for simulations in a micro-canonical ensemble, however, it has been widely shown that thermostats and barostats can be applied during the unconstrained part of the simulation, provided the degrees of freedom of the system are correctly accounted for[165, 166, 167].

In an unconstrained MD simulation the acceleration of an object is given by:

$$\frac{d^2\mathbf{r}_\alpha(t)}{dt^2} = -\frac{1}{m_\alpha}\nabla_\alpha U(\{\mathbf{r}_\gamma(t)\}), \quad (4.19)$$

¹SHAKE[151], RATTLE[152], WIGGLE[164], matrix inverted linearized constraints SHAKE (MILC-SHAKE)[162], MILC hybridized with SHAKE (MILCH-SHAKE)[163], SETTLE[161], linear constraint solver (LINCS)[153], and Parallel-LINCS (P-LINCS)[154]



where t is time, m_α is the mass of object α , ∇_α is the gradient operator with respect to the position \mathbf{r}_α , and U is the force-field of the system. For a constrained system Eq. (4.18) is included yielding:

$$\frac{d^2 \mathbf{r}_\alpha(t)}{dt^2} = -\frac{1}{m_\alpha} \nabla_\alpha \left(U(\{\mathbf{r}_\gamma(t)\}) + \sum_{\beta \rightarrow \alpha} \lambda_{\alpha\beta} \sigma_{\alpha\beta}(\mathbf{r}_\alpha, \mathbf{r}_\beta) \right), \quad (4.20)$$

where $\lambda_{\alpha\beta}$ is the Lagrange multiplier and $\beta \rightarrow \alpha$ denotes the summation over all objects β constrained to object α . After stepping Δ forward in time, the positions are (to order Δ^2):

$$\mathbf{r}_\alpha(t + \Delta) = \tilde{\mathbf{r}}_\alpha(t + \Delta) + \frac{\Delta^2}{2m_\alpha} \mathbf{F}_\alpha^C(t), \quad (4.21)$$

where $\tilde{\mathbf{r}}_\alpha(t + \Delta)$ is the new position for object α generated using a standard MD integration scheme without the constraints applied and $\mathbf{F}_\alpha^C(t)$ is the constraint force defined by:

$$\mathbf{F}_\alpha^C(t) = \frac{1}{\Delta^2} \sum_{\beta \rightarrow \alpha} \lambda_{\alpha\beta} \nabla_\alpha \sigma_{\alpha\beta}(\mathbf{r}_\alpha, \mathbf{r}_\beta) = \frac{2}{\Delta^2} \sum_{\beta \rightarrow \alpha} \lambda_{\alpha\beta} (\mathbf{r}_\alpha - \mathbf{r}_\beta). \quad (4.22)$$

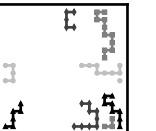
Substituting Eq. (4.22) into Eq. (4.21) results in

$$\mathbf{r}_\alpha(t + \Delta) = \tilde{\mathbf{r}}_\alpha(t + \Delta) + \frac{1}{m_\alpha} \sum_{\beta \rightarrow \alpha} \lambda_{\alpha\beta} (\mathbf{r}_\alpha - \mathbf{r}_\beta). \quad (4.23)$$

Using the fact that the new positions $\{\mathbf{r}_\alpha(t + \Delta)\}$ must also satisfy all the constraints in the system defined by Eq. (4.18) we obtain,

$$l_{\alpha\beta}^2 = \left[(\tilde{\mathbf{r}}_\alpha - \tilde{\mathbf{r}}_\beta) + \frac{1}{m_\alpha} \sum_{\gamma \rightarrow \alpha} \lambda_{\alpha\gamma} (\mathbf{r}_\alpha - \mathbf{r}_\gamma) + \frac{1}{m_\beta} \sum_{\eta \rightarrow \beta} \lambda_{\beta\eta} (\mathbf{r}_\beta - \mathbf{r}_\eta) \right]^2. \quad (4.24)$$

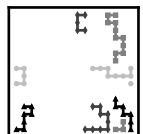
The SHAKE, MILC-SHAKE, and MILCH-SHAKE then eliminate terms of order λ^2 and then iterate the solution to the linearised problem until all the constraints are satisfied to the desired tolerance. The SHAKE algorithm finds the Lagrange multipliers for each constraint sequentially, which can result in insta-



bilities and convergence failures for systems with very large constraint violations in $\{\tilde{\mathbf{r}}_\alpha(t + \Delta)\}$. The MILC-SHAKE is explicitly designed to simulate systems where the heavy atoms are connected to at most two other heavy atoms, and uses a direct matrix inversion of the resulting tridiagonal or cyclic-tridiagonal matrix equation for the linearised $\lambda_{\alpha\beta}$, which is then iterated until convergence. The direct inversion results in faster convergence and increased stability, however, the technique is limited to linear chains[162]. MILCH-SHAKE applies a MILC-SHAKE step to the heavy atoms then a SHAKE step to the hydrogen/side groups and iterates to convergence. This hybrid scheme is both faster and more stable than SHAKE, however, it is not suitable for use when the side groups have similar or greater mass than the objects in the linear chain[163]. The RATTLE and WIGGLE routines are SHAKE like algorithms which can be applied to the velocities and accelerations during the MD integration step. In general RATTLE is also used in most MD codes when using SHAKE for constraints, to reduce constraint violations in $\{\tilde{\mathbf{r}}_\alpha(t + \Delta)\}$ and ensure correct velocities are computed and thereby improve energy conservation[152, 162]. For an unconnected triangle the SETTLE algorithm has been developed this is an analytic solution for the Lagrange multipliers and is often used to accelerate simulations involving water[154, 155, 161, 168, 169, 170].

Unfortunately the iteration stage of these schemes inhibits domain and intramolecular parallelisation as it leads to an unknown amount of communication between the different processes. Furthermore, the affect of the constraints can be highly non-local, which also increases the communication overhead. The LINCS and particularly the P-LINCS algorithms avoid this by using a subtly different scheme for solving Eq. (4.20). The scheme projects out the forces which act in the direction of the bond at time t , then further projects out the effects lengthening of the bond caused by forces not acting in the direction of the bond. This scheme in principle must also be iterated to ensure the correct set of projections is used as a Lagrange multiplier. However, the scheme makes a much better initial guess than the SHAKE based schemes which results in far fewer iterations being necessary¹. The procedure requires inverting a $K \times K$ matrix where

¹In the simulations discussed in Chap. 6 and Chap. 5 three LINCS iterations were found to provide excellent constraint conservation, whilst in the simulations of Sec. 4.2 more than 100



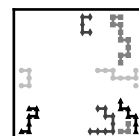
K is the number of constraints in the system, for large K this would become extraordinarily expensive¹. Therefore, the matrix is altered into a form where an expansion can be used as an approximation to the inversion, for details see Ref. [154]. Knowing the number of iterations and the order of the series expansion for the matrix inversion makes the communication pattern both predictable and local, which enables the P-LINCS algorithm to be used in massively parallel MD programs[154, 155, 168, 169, 170]. The series expansion only works for systems with low connectivity such as systems where only bonds are constrained, an altered, higher order, expansion scheme can be used to permit isolated angles to be constrained[154, 155]. However, when the series expansion is used for systems with high connectivity, such as the CG strategy presented in this chapter, the expansion is divergent and the approach moves the constrained objects in a highly un-physical manner[153, 154]. This is explicitly demonstrated for the PEEK molecule in Sec. 4.2.

An effect of using a Lagrange multiplier approach for constraints in the CG procedure is that these constraints must constrain sites with non-zero mass. This is due to the factors of m^{-1} in Eq. (4.24). For the PEEK molecule this gives rise to the constraint set-up of Fig. 4.2(b). However, for other molecules it can give rise to more complicated positions for the vertices or even the need for additional vertices which can result in decreased stability of the constraint algorithms.

As a result of these considerations the SHAKE algorithm was chosen for use with the CG approach. This allows stable integration of the constraints, however, a single molecule cannot be simulated in parallel and the constraints limit the time step for the simulations (details in Sec. 4.2). An alternative strategy would be to build a hybrid scheme similar to MILCH-SHAKE, where the constraints on the backbone of the polymer, e.g., constraints O_2-1 and $1-O_1$ of vertex 1 in Fig. 4.2(b), are constrained using 1 step of MILC-SHAKE and then the rigid triangles, such as $\{1, 2, 3\}$ in Fig. 4.2(b), are constrained with SETTLE. Iterating this should result in a very stable approach permitting fast and accurate constraining of these highly coupled systems. It is unclear whether a SHAKE step would also be required to constrain the carbonyl oxygen in PEEK. If no

SHAKE iterations were necessary.

¹Direct inversion of this matrix is an $O(K^3)$ operation.



alternative could be found this would reduce the stability of the algorithm and increase the computational overhead of the algorithm. This approach would be an interesting follow up to the technique, but no development of this has yet been undertaken. Another alternative which was considered was the replacement of the series expansion in LINCX with an alternative matrix inversion but this would inhibit the parallelisation and require new code being written. Therefore, little effort in that direction was undertaken.

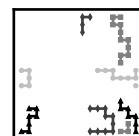
The leapfrog and velocity verlet algorithms for MD simulations are designed to be reversible. This helps ensure that they maintain reasonable energy conservation during an MD simulation¹. In theory constraints can be used in MD simulations without affecting this as they are simply another force acting on the objects. However, in practice the force is only calculated approximately, therefore, at every integration step there is a random extra force in the integration, and the integration algorithm is no longer reversible. This effect can result in substantial energy drifts. In Sec. 4.2 we observe an increase in absolute energy drift of three orders of magnitude when the fractional SHAKE tolerance is increased from 10^{-10} to 10^{-7} . Therefore, care must be taken to ensure that the constraint forces are calculated with sufficient accuracy to ensure reasonable dynamics.

4.1.4 Forces on Rigid Groups

The procedure set out in Sec. 4.1.2 gives a CG representation composed of point masses that define a set of rigid, planar triangles with the same centre of mass, total mass and moment of inertia as a rigid, planar AA representation. The algorithms necessary to evolve the resulting coupled triangles in MD simulation are discussed in Sec. 4.1.3. However, it is also necessary to ensure that the CG triangles experience the same force and torque of interaction with other objects as the groups they replace. This is achieved via the concept of ‘virtual sites’[171], in a manner similar to the generalised-coordinates approach of POEMS[156].

The virtual sites are the positions of the atoms in the AA representation of the aromatic group, and their position coordinates are defined in terms of their

¹Exactly what reasonable energy conservation means is somewhat subjective and will be discussed in Chap. 5.



fixed relationship with the vertices of the triangle in the CG representation. In the case of rigid planar groups, the transformation between AA and CG representations is a simple linear mapping (more complex mappings are possible[150, 171] but are not required for this CG approach). Given the vertices of a triangle at $\mathbf{r}_1, \mathbf{r}_2$ and \mathbf{r}_3 , the position of an arbitrary point (\mathbf{Q}) in the plane defined by the triangle is given by

$$\mathbf{Q} = (1 - a - b)\mathbf{r}_1 + a\mathbf{r}_2 + b\mathbf{r}_3, \quad (4.25)$$

for some values of a and b that can be calculated for each virtual site once and for all using simple vector geometry. Performing this mapping on a set of (a, b) pairs derived from the equilibrium atomic coordinates, the positions of the atoms in the AA representation may be calculated quickly and easily from the positions of the CG masses at each time step.

Given an AA force-field the forces ($\mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3$) on the triangle vertices due to a force (\mathbf{G}) acting on an atomic site are found by taking the derivative of Eq. (4.25):

$$\mathbf{f}_1 = (1 - a - b)\mathbf{G}, \quad (4.26)$$

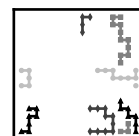
$$\mathbf{f}_2 = a\mathbf{G}, \quad (4.27)$$

$$\mathbf{f}_3 = b\mathbf{G}. \quad (4.28)$$

Thus, the task of generating a separate force-field for the CG representation of the molecule is avoided, and standard AA force-fields which have been developed and tested for the same macromolecule can be used without alteration. Furthermore, the automatically generated atomic coordinates can be used in conjunction with a variety of existing tools for analysing MD trajectories. In the event that a complete force-field for the desired molecule is unavailable the reduction in the number of degrees of freedom enabled by this method correspondingly reduces the number of terms that need to be parametrised.

4.1.5 Force Fitting

The use of parameters in force-fields fitted to DFT simulations of organic molecules has been shown to improve the accuracy with which nuclear magnetic resonance (NMR) structural data is reproduced by MD simulations[172]. However, a direct

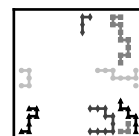


application of this technique to polymeric systems with large monomers in many other industrial polymeric molecules, results in hundred of parameters which would need to be found. Unlike in biological systems where the complexity of the molecule often originates from the ordering and distribution of a set of small building blocks¹, industrial macromolecules can be constructed from any chemically feasible monomer unit. Therefore, many industrial polymers do not have a chemically representative force-field available. Non-chemically representative force-fields typically bias the molecule into areas of phase space which are not representative of the physical system. This CG procedure was originally conceived as a means to reduce the number of parameters needed when fitting a custom force-field for industrially relevant molecules.

There are many schemes which can be used to generate parameters for force-fields. Many force-fields are fitted at least in part to experimental data[173]. However, as quantum mechanical simulation techniques and computers have improved these have become widely used in the parametrisation of force-fields[38]. These techniques often use computationally expensive techniques such as second order Møller Plesset perturbation theory (MP2)[174], which limits the system size to comparatively small molecules, and does not currently allow the matching of forces[38]. Using DFT, which is less computationally expensive, allows the simulation of larger systems and through its ability to calculate forces, more data can be obtained from a single simulation. Unfortunately, as discussed in Sec. 2.3, DFT does not currently provide the correct vdW interactions between molecules, unless extra terms are introduced[92, 93]. However, it has been shown that by retaining the original Lennard-Jones (LJ) parameters from the generalised AMBER forcefield (GAFF) and fitting the remaining terms, excellent agreement with experimental structures can be obtained[172]. Therefore, an investigation into force fitting for polymeric molecules was undertaken.

The procedure considered is sometimes known as iterative force matching. It has been widely used, particularly in the construction of highly accurate solid state force-fields[54, 175, 176], but also in the construction of molecular

¹This is inherently an incredibly complex problem in itself as the configuration space of ordering the different building blocks is factorially scaling, and the blocks are strongly interacting making sampling even a single point in this configuration space immensely difficult!



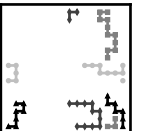
forcefields[172]. The method starts by conducting a MD simulation with an initial, often poor, force-field, from which a series of decorrelated configurations are selected.

These configurations are used as the input coordinates for a set of single point DFT simulations. The difference between the values of observables (typically atomic forces and energy differences) calculated in DFT and MD simulations is then minimised with respect to the force-field parameters. This is usually accomplished by minimising

$$Q = \sum_i^{\text{configurations}} \sum_j^{\text{observables}} w_{ij} \frac{(\mathbf{A}_{ij}^{\text{MD}} - \mathbf{A}_{ij}^{\text{DFT}})^2}{(\mathbf{A}_{ij}^{\text{DFT}})^2}, \quad (4.29)$$

where Q is a measure of the quality of the fit, $\mathbf{A}_{ij}^{\text{MD}}$ and $\mathbf{A}_{ij}^{\text{DFT}}$ are the vectors of the observables of type j in configuration i in MD and DFT respectively, and w_{ij} is a weighting arising from the different numbers of components in the various observables, e.g., if there are more forces than energy difference associated with a given frame, the fractional difference between the forces should be weighted more strongly than the energy differences. Alternative forms of Eq. (4.29) can be used, such as using the sum of the fractional deviations in each component of the vector \mathbf{A}_{ij} [54, 175, 176]. However, Eq. (4.29) is generally sufficient. Since both the gradient and Hessian matrix of Q with respect to the force-field parameters can be calculated, a minimum of Eq. (4.29) can be calculated using standard minimisation algorithms such as steepest descents[177]¹, conjugate gradients[178], or L-BFGS[179]. These do not guarantee finding the global minimum. Techniques such as simulated annealing using a MH approach can be more appropriate for systems where no reasonable guess of the initial parameters can be made [175, 176]. Once the optimal parameter set has been found these are used to conduct another MD simulation, and the procedure is iterated until convergence[172, 175, 176]. There have been some criticisms of this approach but, these usually focus on highly un-physical systems[180]. Techniques such as iterative Boltzmann inversion, and inverse MC, have also been used in deriving CG force-fields, which require that

¹For non-fluent German speakers the discussion in Ref. [178] or Ref. [179] might be more accessible.



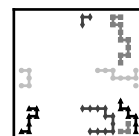
dynamical simulations be conducted in the higher accuracy regime[181]. This is generally not practicable when comparing to quantum mechanical data.

The complete force-field for large molecular systems, even ones with simple monomers such as PEEK, generally have vast numbers of parameters. The AMBER force-field for PEEK has 116 parameters. This can be reduced to 90 by considering that charges in chemically similar environments should be the same. However, by eliminating the degrees of freedom using the CG procedure, maintaining the original LJ parameters and periodicity of the dihedral and improper dihedral angles, and defining the rigid geometries from their equilibrium DFT positions, the number of terms can be reduced to just 38 parameters. Therefore, this technique has the potential to offer substantial computational savings when undertaking force fitting. However, in the process of developing the technique it became apparent that the inhibited parallelisation combined with the complex construction of vertices required for molecules such as P84 and Matrimid, would reduce the force-field’s practical value. This combined with the computational effort as well as human time involved in executing the many thousands of DFT simulations required per molecule resulted in the decision not to proceed with force fitting of large molecular systems and focus the research on developing strategies for using the results of MD simulations produced by existing force-fields.

4.2 Results

The CG representation of PEEK was compared with AA and AA-RB simulations for four different lengths of polymer comprising 4, 8, 12, and 16 monomers (the monomer unit is shown in Fig. 4.2). The molecules were terminated with a hydrogen atom on the left and a phenyl ring group after the ketone on the right. The chains therefore consisted of 13, 25, 37, and 49 aromatic groups, respectively.

All simulations were conducted using the GROMACS[154, 155, 168, 169, 170] MD suite (v4.5.5 with double precision). The polymers were modelled using the OPLS-AA force-field[38], with 1.1 nm cut-off radius, smoothly tapered in the final 0.05 nm. The leapfrog integration algorithm and, for simulations carried out in a canonical ensemble, a stochastic velocity rescaling thermostat[58] was used. For

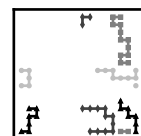


the CG and AA-RB simulations constraints were enforced to a fractional tolerance of 10^{-10} with the SHAKE[151, 152] algorithm. To ensure adequate statistical sampling, for each system 16 evenly-spaced, uncorrelated configurations were taken from a 2 ns simulation in a canonical ensemble at 1400 K and used as the initial starting points for further simulations. These initial configurations were then simulated at 300 K, 500 K, 700 K, 1000 K, and 1400 K. First, the polymers were equilibrated in a canonical ensemble for 1 ns, followed by a further 1 ns in a micro-canonical ensemble. Production runs were then carried out in a micro-canonical ensemble with samples taken every 16.8 ps over a total duration of 5 ns for the 4- and 8-monomer conformations, and over a duration of 10 ns for the 12- and 16-monomer conformations. The sampling interval was chosen because it was the smallest interval for which derived properties for each polymer were unaffected by doubling the interval. All properties presented below were averaged over the 16 independent simulations of each polymer molecule and the resulting trajectories were analysed using the MDAnalysis toolkit[182].

First consider the 8-monomer polymer. Fig. 4.3 shows the energy drift at 300 K for AA, AA-RB and CG simulations as a function of MD integration time step. For the CG simulations, the SHAKE algorithm was unable to converge for time steps larger than 7 fs, while for the AA simulations, time steps larger than 1.25 fs resulted in unstable dynamics (the vibration period of carbon-hydrogen bonds in the system is approximately 11 fs).

The fractional tolerance of 10^{-10} used with the SHAKE algorithm for all constraints renders the integration algorithm almost perfectly reversible. With the small time steps where the AA and AA-RB simulations remain stable the energy drift of the CG simulations is significantly smaller. At larger time steps, where the AA and AA-RB simulations become unstable, the energy drift of the CG simulations becomes comparable to that of the stable AA simulations. The energy drift of the CG simulations increases as the constraint tolerance is increased, as expected (see Fig. 4.4). Nevertheless, even when the fractional constraint tolerance was increased by 3 orders of magnitude to 10^{-7} in the CG simulations the energy drift with a 5 fs time step was $-30 \pm 1 \text{ meVns}^{-1}$. Therefore, a SHAKE

¹For comparison this energy drift is less than 25% of the drift per degree of freedom obtained for a box of 820 simple point charge (SPC) water molecules using analytic constraints, with a



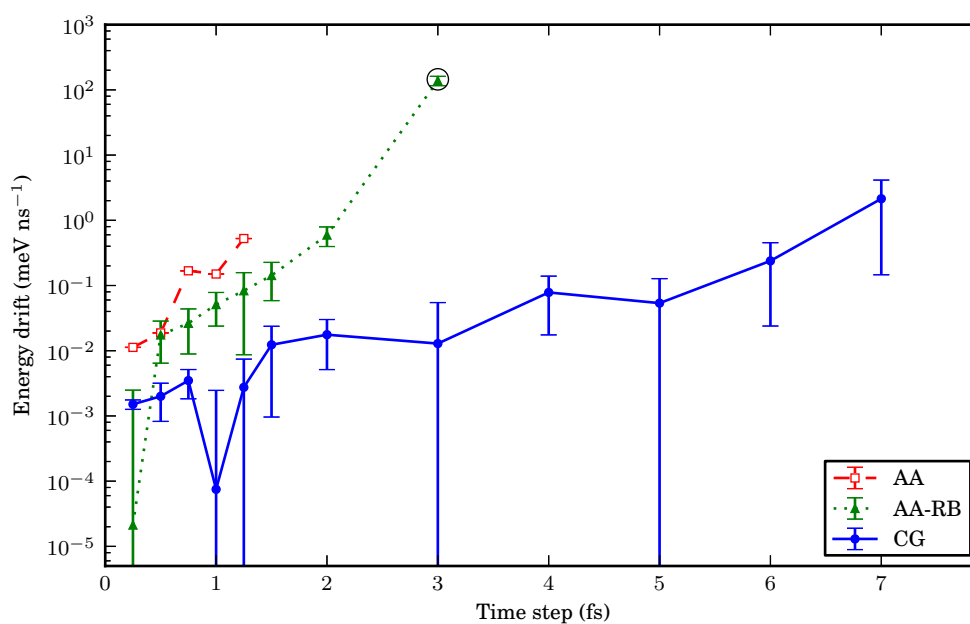


Figure 4.3: The absolute drift in total energy averaged over 16 independent 5 ns micro-canonical simulations of an 8-monomer polymer at 300 K for AA, AA-RB, and CG representations. As can be seen, with the exception of the 3 fs time-step AA-RB simulations (circled), all simulations showed excellent energy conservation.



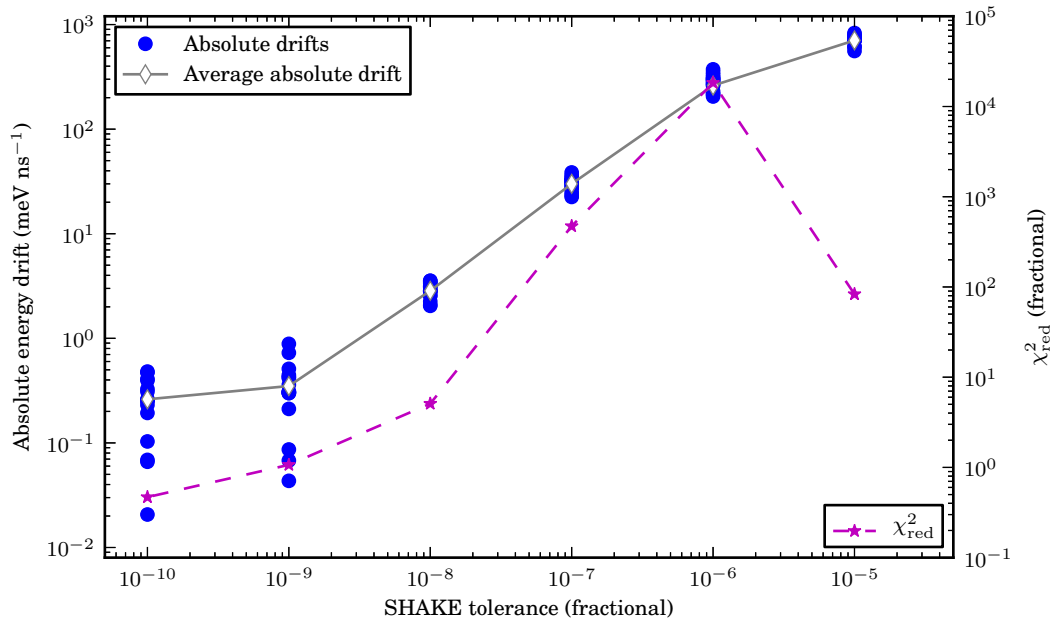
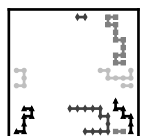


Figure 4.4: The absolute drift in total energy for micro-canonical simulations of an CG 8-monomer polymer at 300 K. The absolute drifts of 16 independent simulations at each of the six different constraint tolerances are shown as blue circles. The average of these data points is given by the grey diamonds, and is calculated by χ^2 analysis. The χ^2_{red} values for these averages (purple stars) are shown and are plotted against the right hand scale. The large drop in χ^2_{red} value at a SHAKE tolerance of 10^{-5} is caused by the non-linear energy drift experienced by these molecules, see Fig. 4.5. This results in very large errors on the initial linear fit to the energy, which reduces the χ^2_{red} value.

tolerance in the range $[10^{-10}, 10^{-7}]$ is suitable for most users. Simulations were also conducted using a SHAKE tolerance of 10^{-5} . However, as can be seen in Fig. 4.5, not only is the change in energy over the simulation very large, but the change is highly non-linear this demonstrates the breakdown of the integration routine for CG systems with large constraint tolerances.

The LINCS algorithm was tested for a wide range of parameters (expansion order $\{4, 6, 8, 10, 12, 14, 16, 18, 20\}$, iterations $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$) with time steps of $\{1, 2, 3, 4, 5\}$ fs. For each combination of parameters 16 starting configurations were used, each with a 4 fs time step and increased hydrogen masses to reduce the energy drift[150].



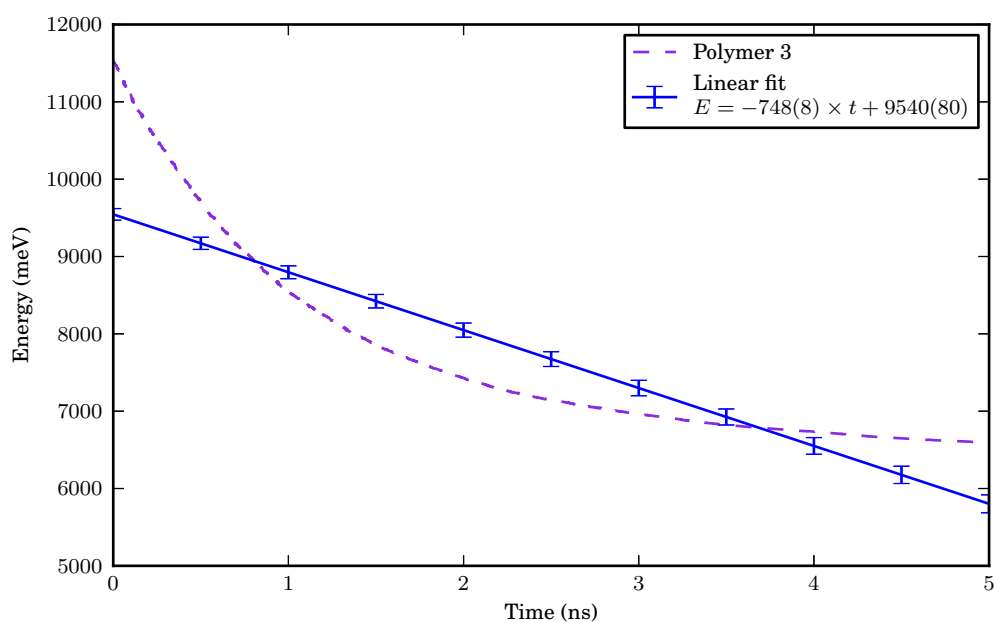
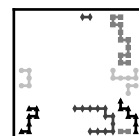


Figure 4.5: The total energy of a single 8-monomer polymer during a 5 ns micro-canonical simulation with SHAKE tolerance 10^{-5} (dashed purple line). The polymer was started from a structure equilibrated in a micro-canonical ensemble with SHAKE tolerance 10^{-10} for 1 ns, the starting temperature was 300 K. A least squares linear fit to this data is also provided (solid blue line) to emphasise the non-linearity of this energy drift. This shows the break down of the leapfrog integration algorithm for large values of the SHAKE tolerance.

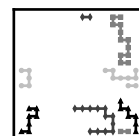


urations were taken from CG simulations successfully completed using SHAKE with the same time step. However, none of the 7200 simulations attempted remained stable for more than 4 MD steps. This was caused by the highly coupled nature of the constraints which produce eigenvalues too large for the expansion used in LINCS. Therefore, the SHAKE algorithm was used for all constraints throughout this chapter and no further simulations using the LINCS algorithm were conducted.

The AA-RB simulations using a 3 fs time step (circled in Fig. 4.3) results in an energy drift of 140 ± 20 meVns⁻¹. This is 10 times the equipartition thermal energy of the system ($k_B T/2 \approx 13$ meV). It is caused by the rapid movement of the hydrogen atoms which cannot be accurately reproduced with this time step. At 300 K a time step of 2 fs provides good energy conservation and stable dynamics. However, at 1000 K and above the rapid movement of the hydrogen atoms led to failures by SHAKE to converge, or it produced erroneous constraint forces and concomitant spurious large changes in the total energy of the system. Both the SHAKE convergence failures and spurious energy changes are absent with a time step 1 fs in the AA-RB simulations.

Fig. 4.6 shows the computational performance of simulations of the 8-monomer polymer at 300 K, as a function of the integration time step. It can be seen that for the same time step, CG simulations have performance comparable to AA simulations, and greater than AA-RB simulations. But the real advantage of the CG representation is that a larger time step can be used than with both the AA and AA-RB representations without sacrificing stability or energy drift. This performance enhancement is limited only by the maximum time step that can be used before the SHAKE algorithm fails to converge in a reasonable number (250) of iterations. At higher temperatures the largest time step that can be used is reduced. For example, at 1400 K a time step of less than 6 fs must be used. For all the simulations that follow, I used a 5 fs time step for CG, and a 1 fs time step for AA and AA-RB. With these parameters it can be seen in Fig. 4.6 that the performance increase of the CG approach over the AA approach is a factor of five.

In Fig. 4.7 I report the radius of gyration, a key property of the molecular structure, as a function of temperature for polymers ranging in size from 4 to 16



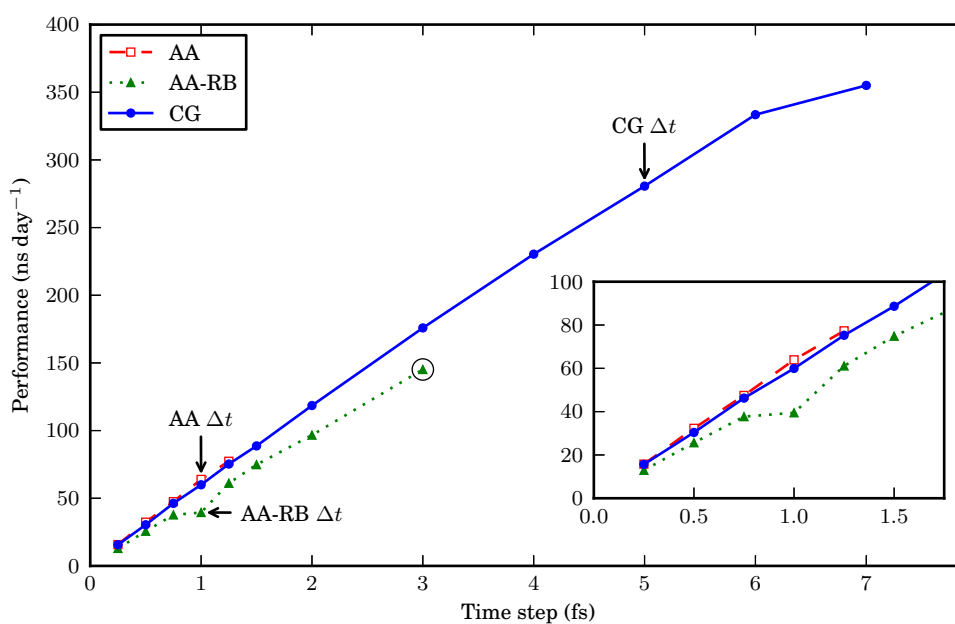
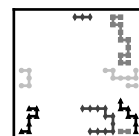


Figure 4.6: The performance averaged over 16 independent 5 ns micro-canonical simulations of an 8-monomer molecule at 300 K for AA, AA-RB, and CG representations. The AA simulation becomes unstable with a time step of 1.5 fs, and the energy drift of the AA-RB simulation with a 3 fs time step is too large. The arrows indicate the time steps used in the rest of the paper. The simulations were performed on one core of an Intel Xeon E5-2650.



monomers. As can be seen, the CG simulations are in very good agreement with both the AA and AA-RB results.

In Fig. 4.8 I show the excellent agreement between AA, AA-RB, and CG representations for the “hinge” angles, labelled O_1 , O_2 and C in Fig. 4.2. In the ground state these angles are 120° . The data shown in Fig. 4.8 is for the 16-monomer system at 500 K, which has been chosen because it displays the worst agreement between the AA and CG simulations of all the molecules and temperatures simulated; in the case of the best agreement all data points are indistinguishable. The agreement was characterised by the reduced χ^2 value,

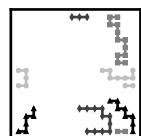
$$\chi_{\text{red}}^2 = \frac{1}{N_a - 1} \sum_{i=1}^{N_a} \frac{(\theta_i^{CG} - \theta_i^{AA})^2}{(\alpha_i^{CG})^2 + (\alpha_i^{AA})^2}, \quad (4.30)$$

where θ_i^{CG} and α_i^{CG} are the mean value and standard error, respectively, of the i^{th} angle, and $N_a = 3$ is the number of angles. Smaller values of χ_{red}^2 indicate better agreement between data sets.

The relative orientation of the aromatic groups in PEEK is largely defined by the eight dihedral angles identified in Fig. 4.9. There is a choice between two symmetrically equivalent atoms on each aromatic group for each dihedral angle. According to the definition of a dihedral angle, angles are measured in each of the four quadrants of a circle. The symmetry of the aromatic group results in the distribution in each quadrant being related by symmetry to the distribution in all the other quadrants. Therefore, to aid comparison between the simulations the measured dihedral angle $\theta \in (-180, 180]$ is mapped, into the first quadrant, using the relation:

$$\Phi = \begin{cases} |\theta| & |\theta| < 90^\circ \\ 180^\circ - |\theta| & |\theta| \geq 90^\circ \end{cases}, \quad (4.31)$$

where $\Phi \in [0, 90]$ is the resulting dihedral angle defining and characterising the orientation. In the case of the AA and AA-RB simulations the aromatic group can buckle resulting in a discrepancy between the two possible dihedral angles. This discrepancy is resolved by recording the average of the two possible dihedral angles after they have been transformed by Eq. (4.31). This reduces the effect of the buckling inherent to AA and AA-RB simulations of aromatic groups.



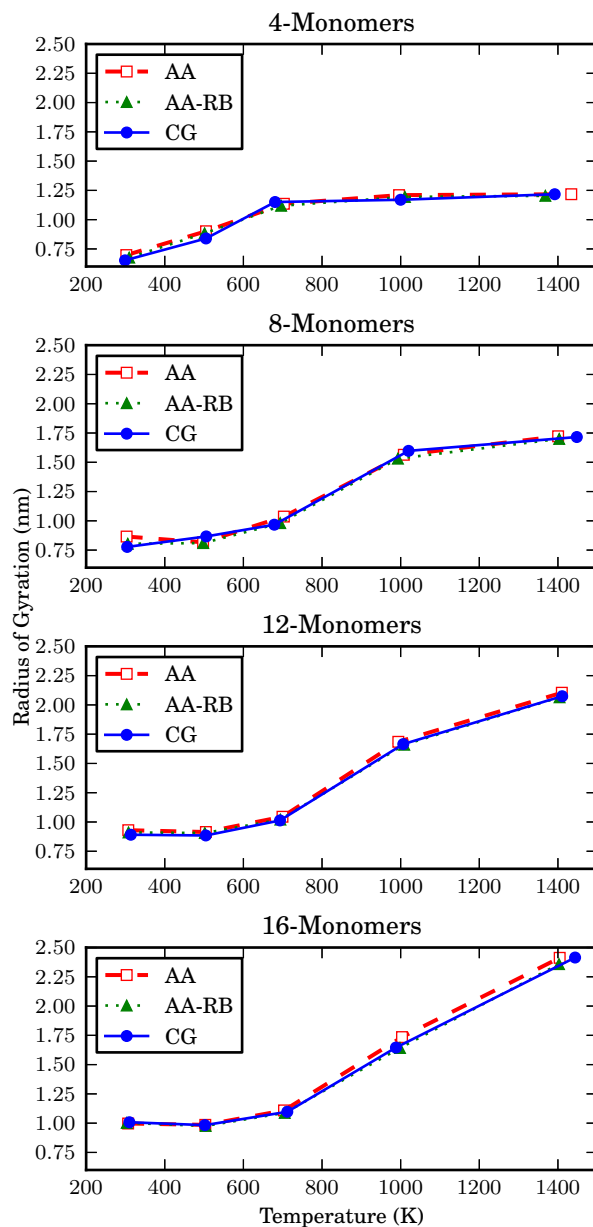


Figure 4.7: The radius of gyration for PEEK molecules, comprising 4, 8, 12 and 16 monomers, as a function of temperature. The CG simulations were performed with a time step of 5 fs, the AA and AA-RB simulations with a time step of 1 fs. Each data point is the average over 16 independent simulations. Straight lines have been drawn through the data to guide the eye, the AA-RB and CG data is shown as discrete symbols. Some data points lie on top of each other. Error bars are comparable to the size of the symbols.



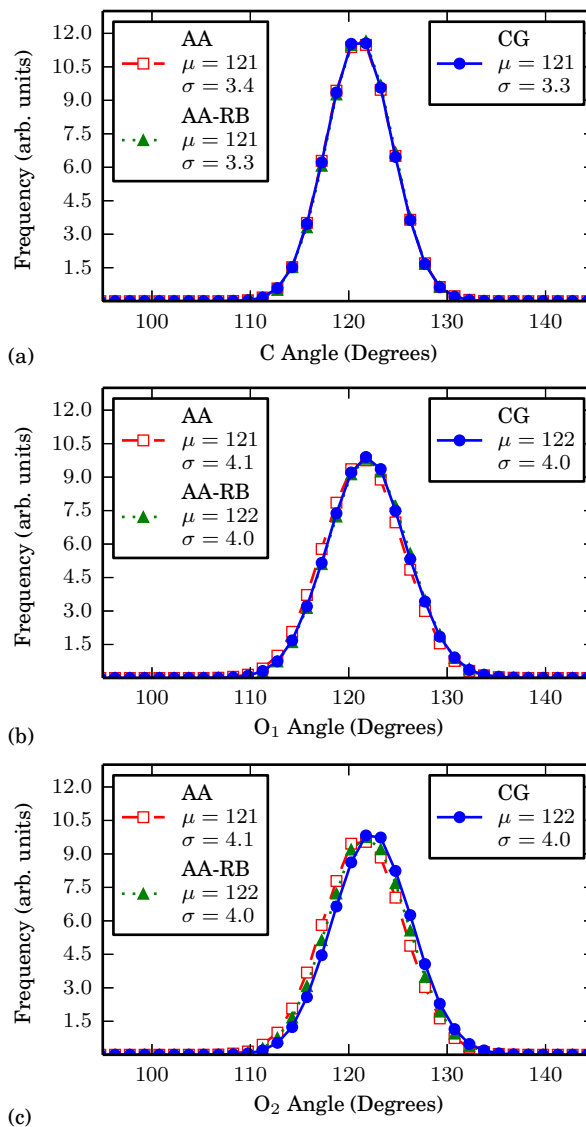
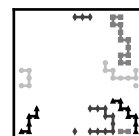


Figure 4.8: Normalised distributions of the hinge angles labelled C (a), O₁ (b) and O₂ (c) in Fig. 4.2, for the 16-monomer system at 500 K, simulated in the AA (hollow squares), AA-RB (solid triangles), and CG (solid circles) representations. Some data points lie on top of each other. Lines have been added to the data as a guide to the eye. The error bars are comparable to the line width. This set of simulations has the largest discrepancy, determined by χ^2 analysis, between AA and CG representations. For many of the other systems and temperatures the AA, AA-RB, and CG representations are indistinguishable on the scale of the plots.



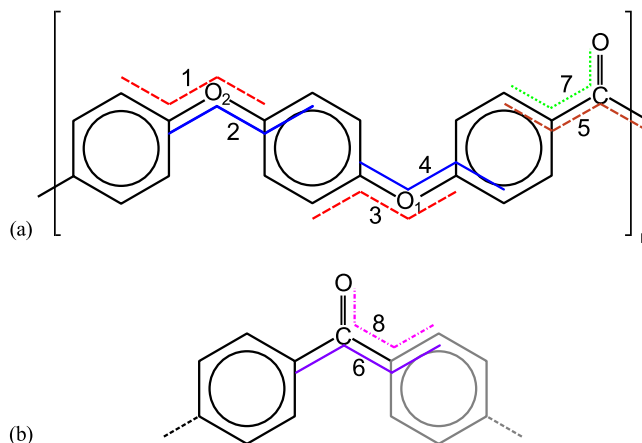


Figure 4.9: The dihedral angles monitored during the MD simulations of PEEK. The dihedral angles are identified by a line connecting each of the four atoms involved and are identified by a number 1-8. In (a) we see the dihedral angles which are contained within the monomer these are 1-5 and 7. Whilst (b) shows the dihedral angles 6 and 8, which involve the first aromatic group of the next monomer (coloured grey) as it connects to the ketone. The choice of atom when two atoms from the same aromatic group are used is made by mapping both possible angles into the interval $[0, 90]$ then taking the average.

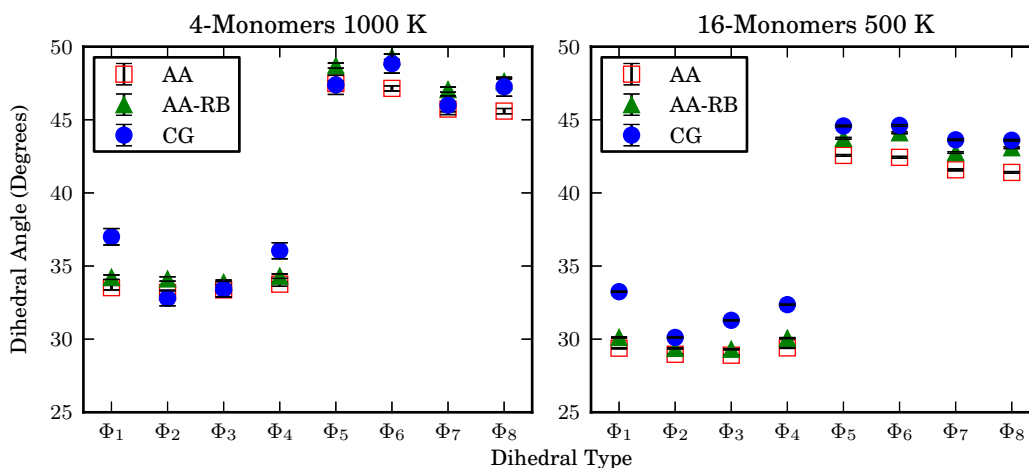


Figure 4.10: The mean values of the dihedral angles Φ_i labelled in Fig. 4.9, in AA (hollow squares), AA-RB (solid triangles), and CG (solid circles) representations. The 16-Monomer 500 K and 4-Monomer 1000 K simulations have respectively the largest and smallest χ_{red}^2 values of all the configurations simulated.



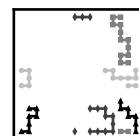
In Fig. 4.10 I present the observed values for these eight dihedral angles for the 16-monomer system at 500 K and the 4-monomer system at 1000 K. These two systems were chosen because they represent respectively the highest and lowest χ_{red}^2 values for the dihedral angles. All angles are averaged over all dihedral angles of that type in the simulation, and over 16 independent simulations at that temperature. It can be seen that the CG dihedral angles are in excellent agreement with the AA and AA-RB simulations and that the maximum deviation is less than 6° , demonstrating that the orientation of the aromatic groups is preserved in the CG representation.

4.3 Conclusions

I have formulated and tested a coarse graining approach to constrain planar groups of atoms along polymer backbones to move as rigid objects during molecular dynamics simulations. A key feature of my method is that it preserves essential dynamical properties of each group that is coarse grained, namely the centre of mass, the total mass and the moment of inertia. Furthermore, the concept of ‘virtual sites’ is used to map forces from an all-atom force-field on to the coarse grained representation.

I have tested the approach by coarse-graining the aromatic groups along the backbone of the industrially important polymer PEEK over a wide range of temperatures and system sizes. The agreement with all atom and all atom rigid bond simulations is excellent for a number of important parameters that characterise the polymer structure. The principal limitations of this technique are those of the SHAKE algorithm: stability at large time steps and poor parallelisation.

The systematic removal of fast degrees of freedom that are irrelevant to molecular conformational changes enables the simulation to focus on the most interesting and pertinent degrees of freedom. By removing the fast out of plane vibrations associated with aromatic groups, it is possible to use an integration time step of 5 fs in the coarse grained simulations, as compared to 1 fs in the all atom simulations, for the same computational cost.



Chapter 5

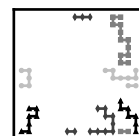
Analysis of Molecular Dynamics Simulations

The use of MD simulations in materials modelling is commonplace, see Sec. 2.2. This has led to the development of numerous MD packages including amongst many others AMBER[183], CHARMM[184], DL_POLY[185], GROMACS[154, 155, 168, 169, 170], LAMMPS[186], and NAMD[187]¹. These large MD packages often contain many hundreds of thousands of lines of computer code. Furthermore, the complex systems which these codes allow us to model frequently require extremely large and complicated input files² and generate vast quantities of output data. It is therefore essential that both the input and output of MD simulations are carefully checked to ensure that not only is the right system being modelled but also that the simulation performed the desired calculation to the desired precision.

Nothing made by humankind is perfect[134]. This is especially true of large and complicated projects. Large computer programs such as MD codes will always have parts where they do not perform as expected, these are often known as bugs. In general, bugs can be classified into two types. First there are the

¹This list covers only the MD programs I have used as well as CHARMM and NAMD which have been added as the CHARMM forcefield is widely used in bio-molecular simulations, and NAMD is widely used for massively parallel MD simulations.

²If the coordinates of the sites cannot be generated automatically for instance by duplicating a smaller box, the coordinates of each site must be specified. For large systems (> 50000 atoms) this can result in megabytes of data in addition to all the other input variables.



kind that cause the simulation to fail or produce nonsensical results; these are generally easier to identify. Then there are those that give incorrect results that are nonetheless superficially plausible; these are, therefore, generally much more difficult to identify. The presence of bugs does not necessarily mean that a computer program should not be used only that care should be taken in how and when it is used.

This chapter gives an overview of some of the tests and checks which have helped me to identify bugs both in programs I have written to generate input files and within the GROMACS MD package. During the course of my research into polymers more than 10 million core hours of computer time has been used on MD simulations. The aim of this chapter is to share my experience about the tests that can and should be run on results from a MD simulation. It is not meant to provide an exhaustive list of tests that will guarantee a correct answer. My sincere belief is that no finite set of tests will guarantee the validity of an answer, and that as the number of tests increases the probability of there being an error in the testing increases. Therefore, it is advisable to choose with care a set of tests which will indicate whether a simulation is behaving sensibly and which can also be conducted in a sensible amount of time. This chapter provides specific examples of where these tests have identified issues with MD simulations during my research. These issues have in total resulted in the loss of over 9 months of research time from this project. The rest of the chapter is organised into three sections. The first section covers the importance of testing against expected results and describes certain key assumptions about results which will be used in the subsequent sections. The second section of this chapter discusses qualitative tests, focussing on where they have been used to identify issues in simulations of P84 in solution and CG PEEK. Finally qualitative tests for MD simulations will be discussed, the particular focus here are problems with large simulations of P84 in solution.

5.1 Expectations

Testing is a key part of the scientific method and in order to test a theory or in this case a MD simulation it is necessary to have some expectations of the

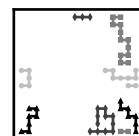


results. In this context an expectation does not necessarily mean knowing the answer to the question being posed, but rather certain aspects of the solution. By comparing the results of the simulation with these expectations we can gain insight into whether the results obtained are valid.

MD simulations use finite precision arithmetic, typically double (8 bytes) or single (4 bytes) precision numbers and, over time, a numerical integration scheme will always result in disagreement with an analytic function. This occurs due to the accumulation of rounding errors. Therefore, even for simple systems where analytic solutions may be known, MD simulations will after some finite time be different from the analytic result. Despite this inherent problem, by understanding the properties of the integration algorithm as well as the system of interest it is possible to generate sensible expectations.

MD algorithms perform an integration, therefore, expectations should concern the average behaviour over the integration of an observable rather than the value of any one individual observation during the integration. Furthermore, MD simulations are made up of many small components and it is often advisable to test these components both in isolation, as much as possible, and then combine them. This allows you to narrow down where any issues are occurring, which makes them easier to find, and sometimes to avoid issues by choosing a different algorithm or different parameters. Therefore, I usually run test simulations using small test systems prior to running tests on larger systems. An example of this is given in Sec. 5.2, where an 8681 atom system was used to identify thermostat issues prior to simulations of a complete ≈ 90000 atom system. Testing in this manner enables certain parameters and simulation procedures to be designed and tested at a reduced computational cost¹. However, as will be discussed in Sec. 5.2 and Sec. 5.3, testing and analysing large systems is also essential as certain problems only occur during full scale simulations.

¹MD simulations typically scale as $O(N \log N)$ with the number of particles in the system (N).

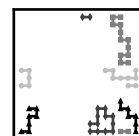


5.2 Qualitative Tests

Qualitative testing is a topic that can make people uncomfortable, owing to its association with guessing, intuition, and other topics that seem to lack the rigour of quantitative analysis. However, it must also be understood that these tests can often be the first indication that something is going wrong in a simulation and can provide insight that is not easily obtained from a quantitative analysis. The line between qualitative and quantitative analysis is somewhat ill defined, and whilst some of the issues discussed in this section could be placed in Sec. 5.3, I have typically made the distinction based on whether the issue was discovered as a result of numerical values falling out of an expected range when computing an observable or as a result of looking at trends in a visual representation of the data.

When testing a simulation I often visualise the output coordinates of the system. Whilst it is often difficult to discern detailed information from a visualisation, e.g., observing a single misplaced atom out of ten thousand is not easy, it can reveal anomalies that should be further investigated.

For example, whilst preparing to conduct large scale simulations of solvated P84, I simulated a test system consisting of 611 DMF molecules and a 137 atom section of P84. The simulation was performed at 300 K in a canonical ensemble with temperature conservation enforced by means of a Nosé-Hoover[59, 60] chain thermostat in GROMACS v4.5.5. The density of this box was 861 kgm^{-3} which is around 87 % of the experimental density of 18 wt% P84 in a 3:1 mixture of DMF:dioxane, and the temperature was well conserved. However, on visualisation it became apparent that large voids were forming in the solvent, see Fig. 5.1. This behaviour was not expected at these conditions, therefore, I decided to investigate further. Continuing the simulations, without a thermostat and using the Langevin thermostat both showed the voids filling almost immediately. Therefore, the thermostat was identified as the likely cause, and testing was conducted using time coupling parameters, 0.1–5 ps for Nosé-Hoover chains of lengths from 1–30. These simulations all showed similar void formation, and also identified that the average pressure of a simulation cell varied with the number of cores on which the simulation was run. I used this information to file two bug reports with



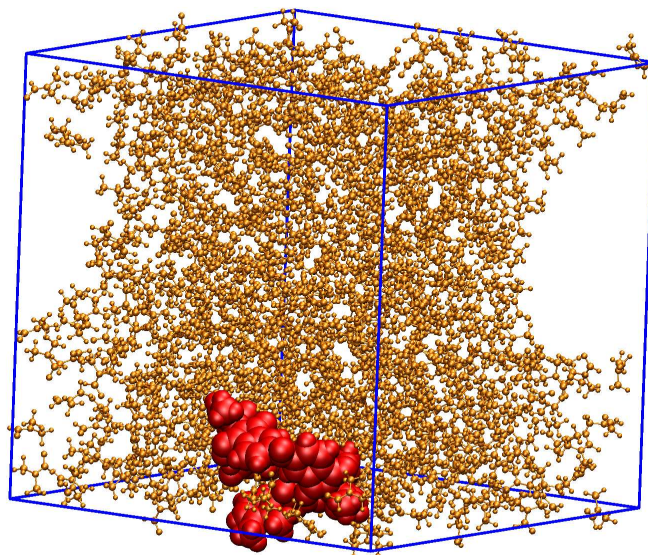
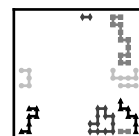


Figure 5.1: A visualisation of a periodic box of 611 DMF molecules (orange, ball and stick models) and a small (137 atom) section of P84 (red, space filling spheres). Voids are clearly visible at the left and right edges of the periodic simulation box.

the GROMACS team, reports #1003 and #1012, these were fixed when various core routines were updated in preparation for a new release of the code. This highlights the importance of these visual checks as they can be an early indication of something going wrong in the simulation.

Another qualitative test is that the behaviour of equivalent components within a system should be similar. Whilst conducting the initial testing on the CG PEEK molecule described in Chap. 4, eight dihedral angles were monitored for each monomer of the polymer (see Fig. 4.9 for definitions of the dihedral angles). It was found that despite a symmetry which ensured that all four possible quadrants of dihedral angles 6 and 8 were geometrically equivalent and should occur with equal frequency, checks showed that for the terminal monomer dihedral angle 6 was always in the region $-180 < \Phi_6 \leq -90$ and dihedral angle 8 was always in the region $90 < \Phi_8 \leq 180$. This observation was incompatible with the expectations

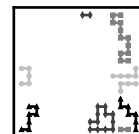


based on the symmetry of the system. Furthermore, the values obtained for dihedral angles 6 and 8 not in the terminal monomer were evenly distributed between the four quadrants. An analysis of the input file led to the discovery of a single misplaced constraint which had been missed in previous checks of the file. Once corrected all equivalent dihedral angles had a similar distribution of values between the four symmetrically equivalent quadrants.

These tests should give an overview of some of the things which should be considered at a qualitative level when analysing an MD simulation. One thing which is important to remember is to have a physically motivated hypothesis of what the results will show before starting the simulation. This is important because it is far too easy to convince yourself that what is in front of you is the correct physical result if you do not have a clear picture of what to expect before you start.

5.3 Quantitative Tests

Quantitative tests are generally loved by scientists. In particular, when the solution to a problem is known one can simply check if the result from the simulation is in agreement and know that one is reproducing at least some aspect of the system accurately. However, in a dynamical simulation such as MD this is almost never the case. Therefore, the most important test for a simulation is the check that the observable(s) of interest are sampled by the simulation in a sensible manner. There is no simple test for this which will work for every observable. It is therefore necessary to consider the observable carefully and consider what distribution of observations might be expected. Checking the distribution of points and not just the average and standard deviation is essential. To understand why this is the case consider a set of data points uniformly distributed on a circle of unit radius. Given a sufficient sampling of the circle, these will have an average position at the centre of the circle and a standard deviation in both the Cartesian directions of $1/\sqrt{2}$. This is the same average and distribution as would be obtained from a set of x and y coordinates drawn from a normal distribution with standard deviation $1/\sqrt{2}$. However, it is clear from Fig. 5.2 that these two distributions are sampling the Cartesian plane in a very different manner. For



this reason the distribution and fluctuations of an observable must be considered. If an observable has sampled according to an inappropriate distribution it may indicate that something is wrong with either the code or the underlying procedure used for the simulations. Another issue which can occur is that the simulation is sampling in a sensible manner but has yet to obtain a sufficient quantity of data. If this is the case, then either the simulation should be extended in duration or alternative techniques that accelerate the exploration of phase-space such as meta-dynamics[188], or MC should be considered.

5.3.1 The Micro-Canonical Ensemble

The micro-canonical ensemble is the simplest ensemble to simulate in MD, as it maintains constant particle number, volume, and internal energy. Testing for changes in particle number or volume is trivial as these should not vary at all in the MD simulation. However, no perfect numerical scheme is known for moving the particles forward in time whilst conserving energy. Therefore, it is necessary to test the behaviour of the energy in view of the imperfect integration algorithm. When using the *leapfrog* and *velocity verlet* algorithms, the positions at future points in time are accurate to $O(\Delta t^2)$ where Δt is the time step. Despite this accuracy both algorithms do result in fluctuations in the total energy of the system from one time step to the next and a drift in the total energy over longer times. Higher order schemes such as fourth order Runge-Kutta[179] typically have a smaller energy drift and usually do not have significant fluctuations; however, these require additional force evaluations, which can significantly increase computational cost, and lack time reversibility, which can result in non-physical behaviour at large time scales. Therefore, they are far less widely used than *leapfrog* and *velocity verlet*, and will not be discussed further.

For both the *leapfrog* and *velocity verlet* algorithms: the total energy drift and amplitude of the fluctuations increase and the computational cost decreases with the size of the time step. Therefore, a balance must be sought between the increased precision of a smaller time step and the reduced computational cost of a large time step which can allow increased accuracy by sampling a larger proportion of the available configurations. The fluctuations in the total energy



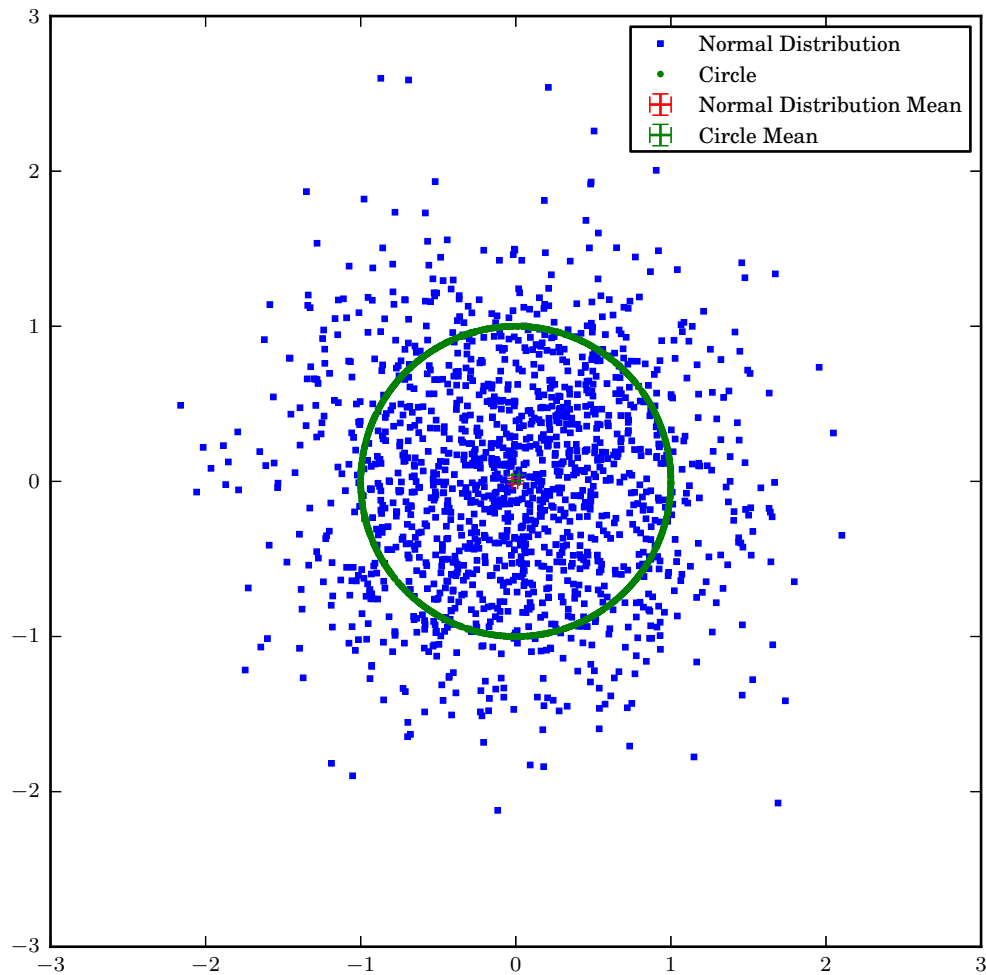


Figure 5.2: Two sets of 1500 randomly generated points. The blue squares are drawn from a two dimensional normal distribution with standard deviation $1/\sqrt{2}$, whilst the green circles have been drawn uniformly from the unit circle. The resulting mean positions and standard deviations are in perfect agreement. However, the two sets of randomly generated points clearly do not sample the plane in a similar manner.



are easily reduced as they scale as $O(\Delta t^2)$ [26] and for these I tend to ensure that they are small relative to any energy-based observation being made¹. There is no universal value for the acceptable energy drift in a simulation as this will depend on the observable properties of interest and how they are affected by the drift. When considering an AA simulation it is sensible to look at the energy drift per atom, and a sensible value to compare this with is the thermal energy of the simulation which, in the absence of constraints, is $\frac{3}{2}k_B T$ per atom. If the total energy drift per atom over the duration of a simulation is a substantial fraction of the thermal energy, then it is an indication that the properties from a simulation may not have been calculated with sufficient precision.

The average rate of drift in total energy of a simulation should be a constant for a given simulation, if this is not the case it can be an indication that something is not behaving as expected. An example of this occurred whilst simulating a 392 atom section of P84 solvated in 7389 DMF molecules (a total of 89060 atoms) in GROMACS v4.6.2. The system was initially equilibrated using a Langevin thermostat, at 250 K, for 5 ns followed by a 5 ns micro-canonical equilibration resulting in a simulation temperature of 279 K, (full details of the preparation and cut-off settings used are discussed in Chap. 6). As can be seen in Fig. 5.3, for the first almost 190 ns there is an acceptable drift of $2.21 \times 10^{-4} \text{ meV ns}^{-1} \text{ atom}^{-1}$, then there is an abrupt change for the final 7 ns to $-6.4 \times 10^{-3} \text{ meV ns}^{-1} \text{ atom}^{-1}$. On further investigation, it was found that this discontinuity in the energy drift coincided with the continuation of the simulation from a checkpoint file². This indicates that there is an error with the underlying MD simulation software. The cause of this error has yet³ to be identified, however, it appears to be limited to the GROMACS v4.6.x program. This data shown in Fig. 5.3 is part of a series of 40 simulations, the energy drift graphs from the full range of simulations are available in App. B. These simulations represented in excess of 5 million core hours of computer time and were run over the course of 6 months on four different computer platforms. The discovery of this bug in the simulation software

¹As total energy is degenerate with regards to an additive constant this concept should only be applied to observations of energy differences.

²These continuations were necessary at the time taken by the simulation exceeded the maximum wall time on the computer.

³November 2013



invalidates all the results from these simulations and significantly reduced the amount of data I was able to gain from MD simulations of P84. I failed to spot this problem with the simulations earlier as I relied too heavily on the idea that if the first part of one simulation had behaved in a sensible manner that the other similar simulations would perform equally sensibly and would, therefore, not require detailed monitoring as they were run. This was a serious mistake and I now actively encourage people to learn from this and monitor the detailed behaviour of the data as results come in. The simulations which were lost to this bug represented more than 6 months of time and were conducted on three high performance computers¹, repeating them with an alternative version of GROMACS which does not suffer from this bug was not possible due to the amount of time and computational resources it would have consumed at a late stage in the project. The root cause of the bug has not been identified or to the best of my knowledge rectified. One of the main causes of this is its intermittent nature which leads me to believe that it is likely due to the load balancing routines. However, this has not been confirmed, and new release of GROMACS v5.0, which included substantial changes to all core routines, has now been developed and this bug which is in a version of the code no longer under active development, is unlikely to be resolved.

Given two MD simulations of similar systems generated using the same procedures one might presume that the simulations would demonstrate similar characteristics and that it would, therefore, only be necessary to check fundamental properties of a MD simulation such as energy conservation in one of the pair. A second system comprised of a 392 atom section of P84 solvated in 7384 DMF molecules (a total of 89000 atoms) was prepared and equilibrated in the same manner as the system shown in Fig. 5.3. The two simulations differ only in the distribution of monomer units within the P84 oligomer and the number of solvent molecules present which was altered to maintain equal densities. After equilibration the second system had a temperature of 278 K. The simulations used the same input parameters, and were run simultaneously using the same executable on the same computer. However, as can be seen in Fig. 5.4, the behaviour of

¹It represents almost the entire amount of computer time that had acquired for this project through four successful resource applications.



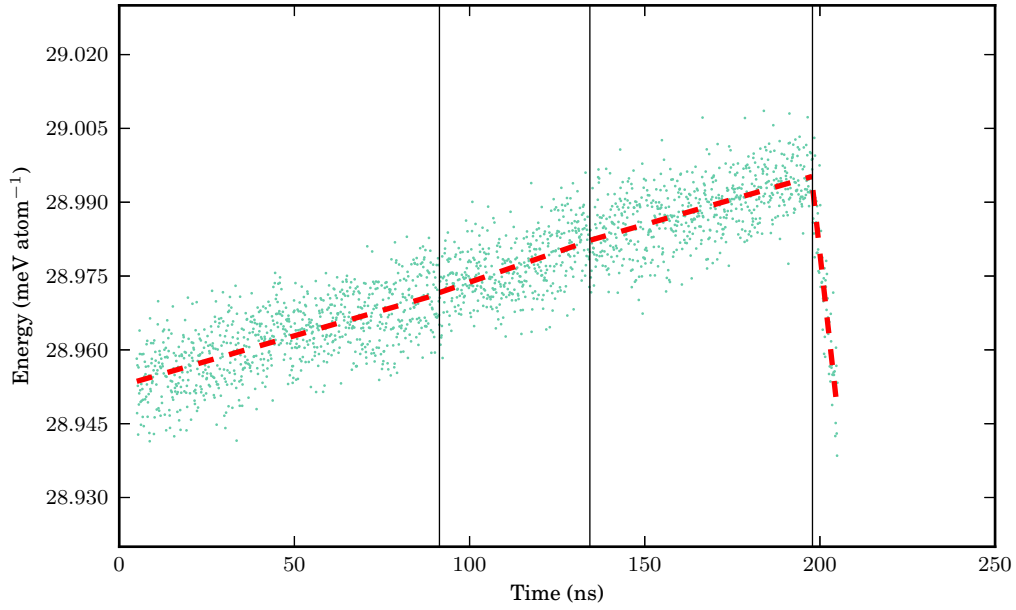


Figure 5.3: The energy drift per atom of a 89060 atom, 200 ns, micro-canonical, MD simulation. The raw data is shown as light green points, times where the simulation was continued from a checkpoint file are indicated with thin vertical black lines, and a least squares linear fit to the energy in each region is indicated with a dashed red line. The initial 5 ns of micro-canonical simulation was used to equilibrate the polymer and is therefore not shown. The gradients of the four regions are from left to right: 2.06×10^{-4} , 2.5×10^{-4} , 2.0×10^{-4} , and $-6.4 \times 10^{-3} \text{ meV ns}^{-1} \text{ atom}^{-1}$. Clearly there is a significant alteration in the behaviour of the simulation after 197 ns.



the energy during the simulations is not similar. The energy drift shown in the graph has three clear regions with gradients $-6.921 \times 10^{-3} \text{ meV ns}^{-1} \text{ atom}^{-1}$, $2.0 \times 10^{-4} \text{ meV ns}^{-1} \text{ atom}^{-1}$, and $-6.68 \times 10^{-3} \text{ meV ns}^{-1} \text{ atom}^{-1}$. The gradient changes occur at the points where the simulation was continued from checkpoint files. Had only the first 190 ns of Fig. 5.3 been analysed it might have been concluded that the energy conservation for the polymer in solution at 280 K was reasonable and the anomalous behaviour of the energetics of the second simulation might not have been observed. For this reason it is essential to always check that the basic properties of a MD simulation are behaving sensibly even if you have tested similar systems using the same code on the same hardware before.

5.3.2 The Canonical Ensemble

The term temperature is one that is often used when talking about MD simulations. In a MD simulation temperature is defined by the relation

$$\left\langle \sum_i^{\text{atoms}} \frac{\mathbf{p}_i^2}{2m_i} \right\rangle = \frac{N_{\text{DoF}}}{2} k_B T, \quad (5.1)$$

where \mathbf{p}_i and m_i are the momentum and mass of atom i , and N_{DoF} is the number of degrees of freedom in the simulation. The term temperature is also sometimes used for values obtained from instantaneous rather than average kinetic energies, and any instances of this in this work will be clearly identified. Modelling a canonical ensemble is more complex than the micro-canonical ensemble as it involves coupling the system to a heat bath which can add or remove kinetic energy from the system thereby approximating a constant temperature within the simulation. The average in Eq. (5.1) gives an indication of the fact that the instantaneous temperature of a canonical ensemble will experience fluctuations, and the magnitude of these fluctuations is known to be [26]

$$\langle \mathcal{T}^2 \rangle_{NVT} - \langle \mathcal{T} \rangle_{NVT}^2 = \frac{2T^2}{N_{\text{DoF}}}, \quad (5.2)$$

where $\langle \rangle_{NVT}$ is an average in the canonical ensemble and \mathcal{T} is the instantaneous temperature. This information can be used to demonstrate that a simulation in



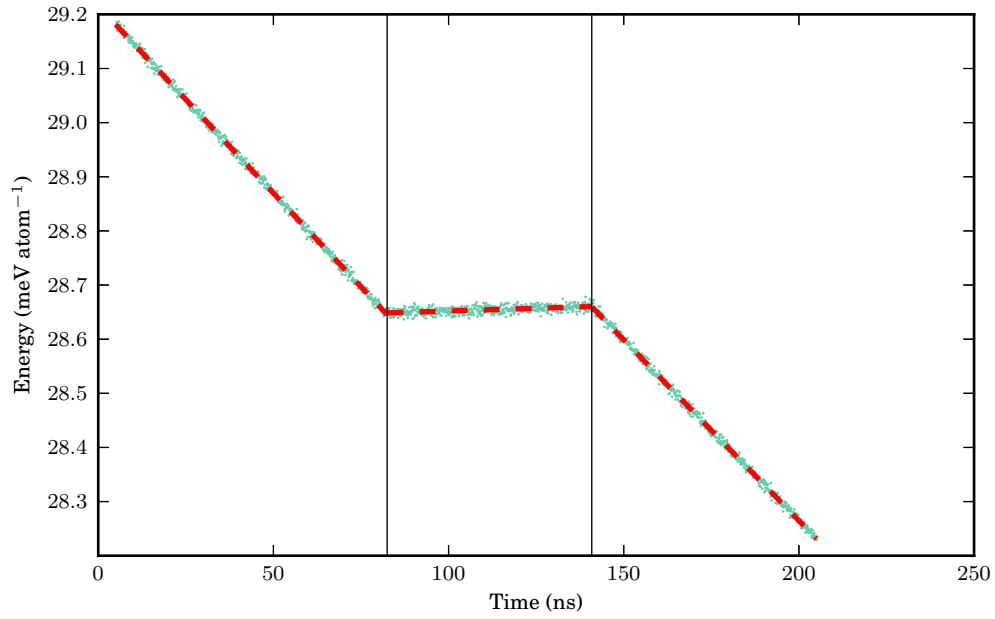


Figure 5.4: The energy drift per atom of a 89000 atom, 200 ns micro-canonical, MD simulation. The raw data is shown as light green points, times where the simulation was continued from a checkpoint file are indicated with thin vertical black lines, and a least squares linear fit to the energy in each region is indicated with a dashed red line. The initial 5 ns of micro-canonical simulation was used to equilibrate the polymer and is therefore not shown. The gradients of the three regions are -6.921×10^{-3} , 2.0×10^{-4} , and -6.68×10^{-3} meV ns⁻¹ atom⁻¹. Clearly the behaviour differs significantly between the three regions and is not similar to that displayed in Fig. 5.3.



the canonical ensemble is giving rise to velocity fluctuations that are physically meaningful. Similarly, for a micro-canonical ensemble, the fluctuations are given by[26]

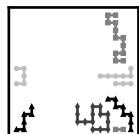
$$\langle \mathcal{T}^2 \rangle_{NVE} - \langle \mathcal{T} \rangle_{NVE}^2 = T^2 \left(\frac{2}{N_{\text{DoF}}} - \frac{k_B}{C_V} \right), \quad (5.3)$$

where $\langle \rangle_{NVE}$ is an average in the micro-canonical ensemble, and C_V is the constant volume heat capacity. Whilst C_V is usually unknown for a system it can be calculated for a small system then checked that the behaviour is consistent in a larger system using the extensivity¹ of C_V . If the temperature fluctuations are not consistent with the expected distribution it is often caused by insufficient equilibration, or an inappropriate set of parameters for the thermostat. It should be noted that N_{DoF} has been used rather than the number of atoms (N_{atoms}) as constraints on the system will reduce the number of degrees of freedom relative to the standard $3N_{\text{atoms}}$.

My experience has been that typically library thermostats perform a reasonable job at maintaining sensible distributions of instantaneous kinetic energies within a simulation. However, extreme care must be taken when selecting the strength of coupling with the heat bath. If the coupling is too strong the system is dominated by the velocities from the heat bath rather than the system. I observed this when in the course of testing the Langevin thermostat in LAMMPS I used the same value for the coupling parameter in the input file for a 500 K simulation of a 145 monomer long PEEK molecule I had previously used in GROMACS. The two pieces of software use the coupling parameter in the input file in a different manner, and the resultant simulation caused the PEEK atoms to oscillate about their positions with no significant movement of any clusters of atoms. I was able to rapidly identify this as an issue with the thermostat, and when the parameter was set to a more sensible value (3 orders of magnitude lower) the polymer rapidly transitioned from an unstable elongated structure to a more stable balled up structure.

In this chapter we have outline some of the possible checks which can indicate potential problems with a MD simulations. There are numerous additional tests and checks which can and often should be run on a simulation to ensure that it

¹Extensive quantities are quantities which are directly proportional to system size.



is accurately reproducing the physics of interest. The purpose of this chapter has not been to give a list of tests that should be run on a simulation but rather an idea of the mindset with which simulations should be approached. The principle I have used throughout my analysis of simulations is this, *if you are uncertain or uncomfortable about something assume that it is broken until you have proved otherwise.*



Chapter 6

P84 Membranes

The study of P84 membranes is immensely complex. The properties of P84 originate from a broad range of length scales from the square meters of membranes which are produced, to the angstrom scale pores which exist in the surface layer. Furthermore, the membranes are produced in seconds by a process involving atomistic interactions yet have an operational life of many years. To begin modelling P84 membranes it is necessary to consider which parts of the life cycle of a P84 membrane are most important. Experiments demonstrate that the porosity and maximum molecular weight cut-off¹ of P84 membranes are determined by the solvents used in the initial manufacture[6, 7, 8, 9, 10]. The focus of this work is the structure generated by the phase inversion process.

The phase inversion process is discussed in Sec. 3.2. For P84 the process takes place over a period of around one second. Modelling the entire process *ab initio* is clearly both impossible and not sensible. It is therefore important to consider the P84 molecule itself, the solvents used during phase inversion, and the assumptions the properties of these components allow us to make.

6.1 The P84 Molecule

P84 is a polyimide macro-molecule. It is composed of benzophenone tetracarboxylic acid dianhydride (BTDA) units bonded together with toluenediamine

¹Molecular weight cut-off is the weight to which an oligomer can be polymerised before it will no longer pass through the membrane



(TDA) and 4,4'-diaminodiphenylmethane (MDA) units as shown in Fig. 6.1. As a macro-molecule P84 does not define a single unique molecule but rather a class of molecules with different molecular weights. The distribution of masses within a collection of P84 molecules, or indeed other molecules formed by polymerisation, is characterised by the polydispersity (\mathcal{D}_M). Polydispersity is defined by

$$\mathcal{D}_M \equiv \frac{\overline{M}_w}{\overline{M}_n}, \quad (6.1)$$

where the number averaged molecular weight is given by

$$\overline{M}_n = \frac{\sum_i^{N_p} m_i}{N_p}, \quad (6.2)$$

the mass-average molecular weight is defined as

$$\overline{M}_w = \frac{\sum_j^{N_m} n_j m_j^2}{\sum_j^{N_m} n_j m_j}, \quad (6.3)$$

and m_i is the mass of polymer i , N_p is the number of molecules, n_j is the number of molecules with mass m_j , and N_m is the number of distinct molecular masses[189]. For P84 $\mathcal{D}_M = 1.46$ with $\overline{M}_n = 62000 \text{ g mol}^{-1}$ [190], indicating very little scatter in the distribution of molecular masses. The average molecular mass of a P84 repeat unit is $423.59 \text{ g mol}^{-1}$. Therefore, a typical P84 molecule is composed of approximately 146 repeat units.

The aromatic structure of P84 causes electrons to delocalise over large areas of the BTDA, MDA, and TDA sub-units. The delocalised electrons inhibit bending and rotation of the groups, resulting in P84 having a highly rigid structure compared to polymers such as PEEK, see Chap. 4, or polyethylene. This rigidity will slow conformational changes in the molecule and help ensure the long term stability of the membrane[7, 8, 9].

As with many polyimide molecules¹ P84 is chemically stable. P84 is highly miscible with polar solvents such as DMF; however, it is insoluble in non-polar

¹Imide groups consist of a trigonal nitrogen connected to two carbonyl groups and third organic group, and can be found at either end of the BTDA groups in Fig. 6.1.



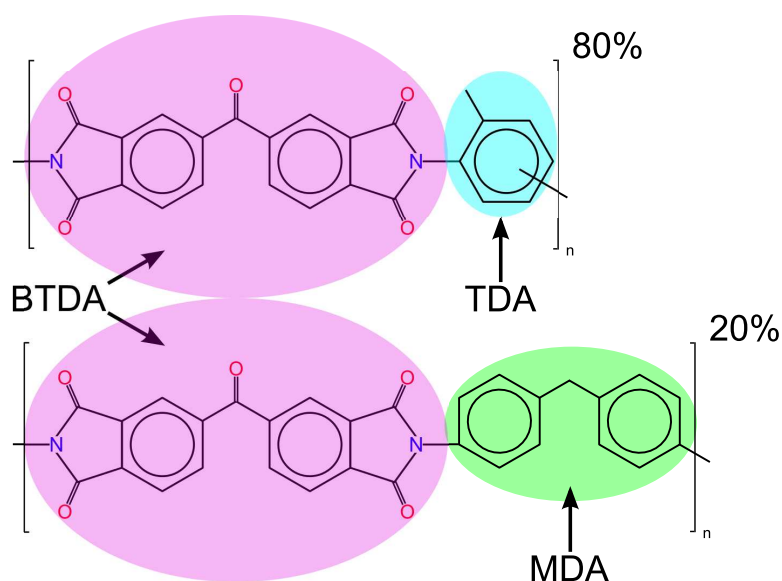
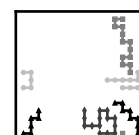


Figure 6.1: The chemical structure of P84 provided by the manufacturer (Evonik GmbH). The subcomponents benzophenone tetracarboxylic acid dianhydride (BTDA), toluenediamine (TDA), and 4,4'-diaminodiphenylmethane (MDA), have been labelled for reference. The exact proportions of the possible bonding configurations between BTDA and TDA is confidential and is, therefore, not shown. However, the proportions of MDA and TDA units in P84 have been widely published[6, 7, 10] and are therefore included.



solvents including water and dioxane[6, 7, 8, 9, 10]. Therefore, intermediate solvents for P84 are constructed by mixing DMF and dioxane together.

6.2 Solvents

DMF is a widely used organic solvent[6, 7, 8, 9, 10]. The chemical structure, shown in Fig. 6.2, results in a highly polar molecule. DMF has a molecular mass of 73 g mol^{-1} , and under standard atmospheric conditions forms a liquid with a density of 944 kg m^{-3} . P84 is highly soluble in DMF which is why the latter can be used to break down P84 membranes. In contrast dioxane, shown in Fig. 6.3, is non-polar and does not act as a solvent for P84. Dioxane has a molecular mass of 88 g mol^{-1} , and under standard atmospheric conditions forms a liquid with a density of 1033 kg m^{-3} . DMF and dioxane are miscible and are used as solvent and co-solvent respectively when forming P84 membranes[190].

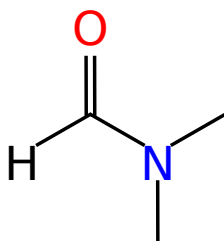


Figure 6.2: The chemical structure of dimethylformamide (DMF).

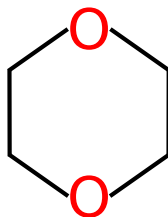


Figure 6.3: The chemical structure of 1,4-dioxane (dioxane).

Both DMF and dioxane are hydrophilic, and water is used as the non-solvent during phase inversion of P84. DMF is non-volatile, dioxane is more volatile



and some dioxane may evaporate from the membrane during casting in ambient conditions. However, it has been shown that this is a very small quantity during typical casting times¹ and has no discernible impact on P84 membrane porosity[7, 8, 9, 190].

6.3 Modelling P84 Membranes

The physical properties of P84 and the solvents used in the production of the membrane allow us to make an assumption about the behaviour of the system during phase inversion. The rate of diffusion of water into the surface of the polymer dope occurs on a time scale much faster than required for conformational changes of a rigid molecule such as P84. This assumption results in the surface layer of the final P84 membrane having a structure broadly similar to that of the polymer in solvent. This dense layer would inhibit diffusion of water molecules through it slowing the solvent exchange in the back of the membrane; thereby allowing the polymer to diffuse in that region. The resulting structure is a dense interface layer with a more open support layer which in agreement with the experimental observations. It is clear from experimental observations that the membrane structure is more open and has larger pores when formed using a good solvent and more closed with smaller pores when formed using a poor solvent, which agrees with what we expect for a polymer in solution. This assumption is supported by the self diffusion coefficients of the solvents, DMF $1.49 \text{ nm}^2 \text{ ns}^{-1}$ [191], dioxane $1.09 \text{ nm}^2 \text{ ns}^{-1}$ [192], water $2.23 \text{ nm}^2 \text{ ps}^{-1}$ [193]. These numbers indicate that in a single nano-second the slowest of molecules would be expected to demonstrate a significant displacement from its starting position, whilst it is clear from Sec. 7.5.3 that in 300 ns a complete P84 molecule was unable to fully equilibrate. Therefore, the assertion that diffusion of water into the polymer dope occurs faster than the speed of conformational changes in P84 appears reasonable.

¹Casting is the process where the polymer dope is spread onto a backing material prior to phase inversion. This is typically $< 10 \text{ s}$ during manufacture of P84 membranes. However, some membranes are manufactured with long casting times to increase solvent evaporation[7, 8, 9, 190].



The challenge is therefore to model the behaviour of large collections of P84 molecules in solution. Whilst this problem is no longer one of kinetics, finding the equilibrium structure of a strongly interacting polymer in solution is not trivial. The polymer dope is typically left to equilibrate for 24 hrs[6], which is far beyond the range of current MD techniques. However, the importance of the polymer solvent interactions shows that a high level of accuracy is required for this problem. The multi-scale model coupling DFT, MD and MC, shown in Fig. 6.4, is therefore the most appropriate approach.

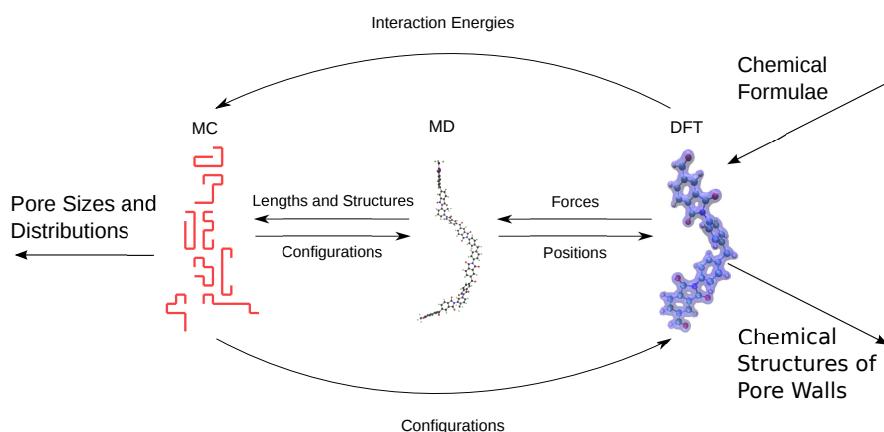
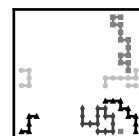
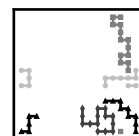


Figure 6.4: A schematic diagram demonstrating the multi-scale model to be used in coupling the different length scales in this work. The arrows indicate the directions of information transfer in the coupling process, whilst the labels indicate what is passed in that direction.

When coupling different techniques it is important to consider what information will be passed between them. The scheme shown in Fig. 6.4 indicates the final goal. However, a complete implementation is not necessary to understand the formation of the membrane structures, but will be useful in the generation of atomistic structures for considering other topics such as the flux of molecules through the membrane structure. Therefore, for this work, the energies and forces, which would ideally come from DFT, will be in the first instance taken from MD simulations using established forcefields, and the path from the MC simulations to the MD and DFT representations has not been developed. Information about the detailed structure of pores will remain a topic for future research. In Chap. 7 we establish the length and energy scales associated with



P84 membranes from MD simulations. This can then be used in the MC simulations in Chap. 8 to understand the pore structure of the system.



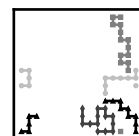
Chapter 7

Molecular Dynamics of P84

The importance of the solvent-polymer interactions in the manufacture of P84 membranes makes it essential to study these systems at high precision. MD simulations provide atomistic insight into the behaviour of polymeric systems whilst allowing access to micro-second time scales. Over the course of this project many avenues to study P84 in solution were explored, the first used the Dreiding forcefield[62] in DL_POLY to study isolated polymers in vacuum, the second considered the use of an *ab initio* forcefield to accurately represent the chemical environment surrounding the atoms, the third used the OPLS-AA forcefield in GROMACS to study the polymers in both dilute and concentrated solutions, the final simulations used the OPLS-AA forcefield in GROMACS to study polymers in concentrated solutions at the experimental temperatures.

7.1 P84 in the Dreiding Force Field

The Dreiding forcefield was selected for the initial study of P84 primarily due to its availability within the DL_POLY (v4.0) graphical user interface. The forcefield has a similar functional form to the OPLS-AA forcefield given in Eq. (2.5), differing only in that a single multiplicity of the dihedral angles (k_i) is used rather than a sum over several in OPLS-AA. Another key difference between the two forcefields is that Dreiding uses a very small set of parameters defined only by an atom's hybridisation rather than a combination of its atomic number, hy-



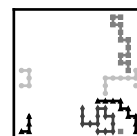
bridisation and the type and hybridisation of the atoms involved in the bond[62]. This radically reduces the number of forcefield parameters required for complex molecules and allows almost any molecule to be constructed. However, as the authors of the original parameter set acknowledge, it does not result in the same accuracy as forcefields which consider the chemical environment of the system in more detail[62] such as OPLS-AA.

The initial simulations in DL_POLY began by analysing the relative orientation of aromatic groups in their minimum energy configurations. Fragments of P84 consisting of pairs of aromatic groups were constructed by hand and their energies were then minimised using the built in function in DL_POLY. The structural relaxation procedure in DL_POLY is one of the best minimising algorithms in any MD code I have used. The user runs a canonical ensemble calculation, then at periodic intervals the code performs a structural relaxation, and the lowest energy configuration of these relaxed structures is stored. This allows the configuration to escape shallow local minima. It was clear from these simulations that the length and structures of P84 would be determined primarily by the relative orientation of the aromatic groups. The Dreiding forcefield’s use of very generic parameters, which do not include the effects of the chemical environment, would not reproduce the structure of P84 effectively.

7.2 Parametrising a Custom P84 Force Field

The behaviour of an atom in a molecule is inherently quantum[194]. Therefore, to build a classical forcefield which reflects the chemistry of the environment the parameters should be fitted to quantum mechanical data for the molecule. Recent developments in DFT and in linear-scaling methods[195, 196, 197], combined with modern high performance computing permit the generation of large data sets of quantum mechanical information for large organic molecules¹. Furthermore, it has been shown that parametrising existing forcefields fitted to match DFT data for organic molecules can improve the agreement with experimental observations[172]. Based on this it was decided to investigate the development of a custom forcefield to P84.

¹For quantum simulations large currently implies hundreds of atoms.



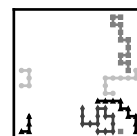
The approach investigated is outlined in Sec. 4.1.5. In summary, it involves generating large numbers of, statistically independent, configurations from a MD simulation, using them as input coordinates for a DFT simulation, then fitting the forcefield parameters to minimise the difference between forces from MD and DFT. New configurations are then generated by running MD simulations with the new forcefield and the process is iterated until the forcefield parameters converge.

The P84 molecule has comparatively large monomer units (43 or 53 atoms). Combining the possible BTDA-TDA-BTDA configurations, adding in an MDA unit and terminating at both ends with methyl groups resulted in a 177 atom oligomer¹. This molecule includes all components of the P84 molecule in their normal chemical environment, and terminating the molecule with methyl groups as opposed to hydrogen prevented the end groups becoming highly polar which introduces deep minima into the potential energy landscape making it difficult to sample. This molecule was therefore chosen to represent the polymer for the purposes of force fitting. The presence of multiple copies of the atoms in similar configurations and the methyl termination were felt to be sufficient mitigation against finite size effects. However, this assertion was to be verified by simulating a smaller number of configurations of a 359 atom molecule² using the fitted forcefield and comparing against DFT.

The authors of Ref. [172] kindly provided their modified version of the AMBER[183] v9 source code to enable the parametrisation of a P84 forcefield. However, the SANDER MD program from the AMBER v9 MD toolkit which had been modified to enable force fitting had several problems related to its data structure which made it less than ideal for force matching. Firstly, the code checks for similar parameters for the bonded interactions in the input file, and references all similar interactions to the same place in memory. Therefore, if two distinct interactions are started from the same initial guess the SANDER program will treat them identically and the force matching algorithm will also treat them as the same interaction. This procedure was originally used to save memory in the simulation. A work around for it is adding a small random term to the fifth significant figure of each term of every bonded interaction in the initial forcefield. The second

¹methyl-BTDA-TDA-BTDA-MDA-BTDA-TDA-BTDA-methyl

²Two 177 atom oligomers connected by a TDA group.

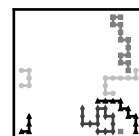


issue was more serious. Unlike for bonded interactions the code always stores the charge and Lennard-Jones parameters for every atom. Therefore, whilst it is clear that there are inherent symmetries in the molecule, which should result in there being only 30-45 chemically unique atoms in P84¹, the code required the non-bonded terms for each of the 177 atoms to be fitted individually. This feature was coded into the most basic data structures of the SANDER component of the AMBER v9 MD suite, and no practical workaround for it could be found.

Ref. [172] demonstrated that for their systems fitting the Lennard-Jones parameters was unnecessary. Therefore, only the bond, angle, dihedral angle, improper dihedral angle, and atomic charge terms in the forcefield needed to be fitted. For P84 using a GAFF style forcefield, and fitting in the modified SANDER code there, are 13 bond types, 23 angle types, 42 dihedral angle types, 13 improper dihedral angle types and 177 charges giving a total of 359 parameters. Enforcing charge neutrality reduces the number of free parameters to 358. This is a vast number of parameters which would require a very large amount of data to fit, and furthermore it contains a large number of parameters which should be either constrained to be identical or will have minimal impact on the final conformation of the molecule.

The key properties of interest in these simulations of P84 are the intermolecular energies and the length scales (radius of gyration, polymer solvent separation, and l_p) of the molecule. Consider for instance increasing the bond lengths of the system by 5%: this would have a far smaller impact on the radius of gyration of the polymer than reducing the energy barrier associated with reorienting the two aromatic groups in the BTDA unit relative to the central carbonyl group. Two procedures were developed to aid in the force fitting. The first was designed to restrict the degrees of freedom in a simulation, and hence the number of forcefield parameters, whilst maintaining numerical stability and atomic level resolution. This eventually became the CG procedure described in Chap. 4. Applied to P84 this had the potential to reduce the number of bonded parameters to be fitted in the forcefield from 182 to 70. However, the lack of symmetry in the aromatic groups of the BTDA unit made the procedure difficult to apply and inhibited parallelisation, due to the use of the SHAKE algorithm, which was

¹This depends on how far the charge is expected to de-localise over the system.

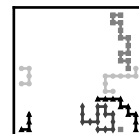


required for backbone aromatic groups, and resulted in this avenue eventually being stopped. The second part of the problem which was undertaken simultaneously, was to write a custom force fitting code which permitted assigning certain atoms to have the same charge, and restricting certain degrees of freedom in the system. The resulting python program is able to read GROMACS topology, and trajectory files and match against forces from a ONETEP[195, 196, 197] DFT simulation. However, without the reduced degrees of freedom it was clear that potentially more than a thousand DFT simulations, and many millions of core hours of computer time, would be required to fit a custom P84 forcefield. It was therefore decided, after a year of working on this, to focus on using MD simulations with existing forcefields, in particular OPLS-AA, to obtain results for the behaviour of P84.

7.3 Molecular Dynamics of P84 in the OPLS-AA forcefield

The OPLS-AA forcefield is specifically designed to capture the behaviour of liquid systems[38, 173]. It is therefore extremely useful for studying polymers in solution. The dihedral angle and improper dihedral angle components of OPLS-AA are four-body interactions. The number of dihedral parameters necessary scales as $O(N^4)$, where N is the number of atom types in the forcefield¹. Therefore, a complete parametrisation of all chemically possible atom combinations is an impractical task, and the forcefield developers focus primarily on molecules which are either widely used or which are of interest in their own research[38, 173]. It is therefore not surprising that not all of the dihedral angles in P84 are parametrised in OPLS-AA. The dihedral angles which do not have parameters are indicated in Fig. 7.1. In all cases the parameters were replaced with zeros for the purposes of conducting the simulations. For cases 1, 2, and 3 the effect of this is expected

¹There are 809 atom types in the 2001 release of OPLS-AA. However these are reduced to 109 types for covalent bonding as many types differ only in their non-bonded parameters. Not all combinations of bonding are possible. Therefore, the final number of dihedral angles necessary for a complete forcefield will be significantly less than the 10^8 one might initially expect[38].



to be minor as the planarity of this section of the system is primarily enforced by improper dihedral angles which are parametrised, albeit with very generic parameters. However, for case 4 the zeroing of the dihedral angle parameters may result in an increase in the flexibility of the BTDA group about the central ketone, as only the potential energy associated with the dihedral angle between the aromatic and ketone oxygen is restricting the rotation. All components of dioxane are parametrised in OPLS-AA and DMF is one of the molecules included in the parametrisation of the forcefield[38, 173].

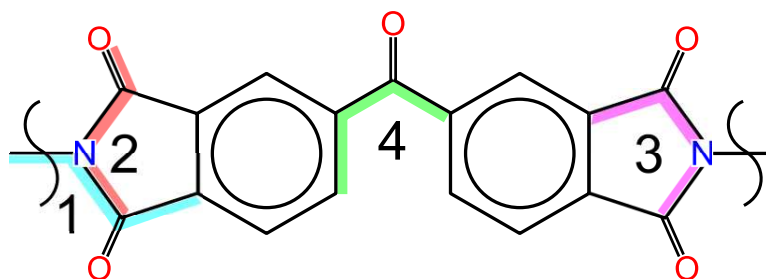
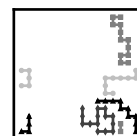


Figure 7.1: A schematic diagram indicating the un-parametrised dihedral angles in the BTDA group of the P84 molecule. Only a single dihedral angle of each type (1,2,3,4) is indicated. There are four of each type present in the BTDA. Furthermore, type 1 dihedral angles could connect to either an aromatic carbon or the carbon from a methyl group. During all simulations of P84 using the OPLS-AA forcefield presented in this work all four types of dihedral which do not have explicit parameters have their energy terms set to be zero.

Two sets of simulations were designed. The first studied the behaviour of short P84 oligomers in dilute solution. The second considered the dynamics of complete P84 molecules at experimental concentrations (24wt%). The simulations were conducted in GROMACS v4.5.5 in double precision unless otherwise stated. Covalent bonds involving hydrogen were fixed with constraints. The LJ parameters used a cut-off radius of 1.1 nm smoothly tapered¹ in the final 0.05 nm. The Coulomb interactions used a 1.1 nm cut-off shifted to 0 at the cut-off with periodic interactions, if present, accounted for using particle mesh Ewald summation (PME).

¹Smooth in this context means continuous in the first derivative of the potential, which is the force on the atom.



7.4 P84 Oligomers in Dilute Solution

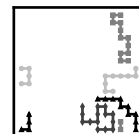
The final structure of P84 membranes is primarily determined by the solvent used in the initial polymer dope[6, 7, 8, 9, 10]. Therefore, studying the behaviour of P84 in various solvents is important in understanding the final structures of the polymer membrane. However, at experimental concentrations of 18–24wt% the dope is left to equilibrate for 24 hours to ensure complete mixing of the polymer with the solvents. A 24 hour atomic scale simulation of a polymeric system is beyond the reach of MD on current computers by 9 orders of magnitude. In contrast solvents such as DMF and dioxane equilibrate on a pico-second time scale, and short oligomers equilibrate far faster than long polymers. As a result of this it was decided to simulate single short oligomers, composed of 253–392 atoms, in a simulation cell large enough to permit the polymer to extend fully without interacting with its periodic image¹.

The purpose of these simulations was to understand the behaviour intrinsic to the polymers. In particular the persistence length of the polymer at its Θ temperature² was sought as this offers a direct point of comparison with the MC model described in Chap. 8. It was therefore, decided to simulate several P84 oligomers in a variety of solvents across a range of temperatures. Finding the Θ temperature for each polymer solvent pair then permits simulations of the full P84 molecule in that solvent to be conducted at the Θ temperature to determine the enthalpy of mixing for each solvent. It was decided to conduct simulations at 250 K, 300 K, 350 K, 400 K, and 450 K, in DMF, dioxane, and water.

Ten P84 oligomers were constructed, consisting of 4 oligomers containing 5 TDA groups and 6 BTDA groups, 2 oligomers containing 4 TDA groups, 1 MDA group and 6 BTDA groups, 2 oligomers containing 3 TDA groups, 2 MDA groups and 6 BTDA groups and 2 oligomers containing 7 TDA groups, 1 MDA groups and 9 BTDA groups. The polymers were generated as random co-polymers in Materials Studio 6.1, and the ratio of TDA and MDA groups was controlled by generating the groups with probabilities based on their abundance in industrial

¹Due to the nature of PME for periodic simulations there will be some component of the reciprocal Coulombic interaction. However, this is not expected to significantly affect the results.

²This is the temperature where the ideal, non self-avoiding scaling of a polymer is obtained.



P84. These initial coordinates were then exported for use in GROMACS using a program I wrote to convert P84 molecules generated to be used with the COMPASS forcefield in the Discover MD suite¹. The polymer conformations were then simulated for 2 ns in vacuum at 2000 K with the canonical ensemble enforced using a Langevin thermostat and a 2 fs time step. This time step was used for all simulations involving dilute solutions. This was followed by a further 2 ns of simulation in the micro-canonical ensemble. The resulting polymer conformations were then considered to be free from any bias imposed by the generating procedure and were placed in a rhombic dodecahedron cell with point to point separation of 11 nm. The rhombic dodecahedron cell was chosen as it represents the smallest volume for a given point to point separation of any space filling box.

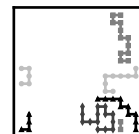
The solvent was prepared by generating small (64 nm³) cubic boxes of DMF, dioxane, and TIP3P water[198]². These boxes were initially packed at a density of 1 g ml⁻¹ by placing structurally relaxed individual molecules on a cubic lattice within the cubic cell. The resulting cell was then simulated for 500 ps with a Langevin thermostat at 300 K, followed by 1.5 ns in the micro-canonical ensemble. A 1:1 mass fraction mixture of DMF and dioxane was constructed by joining two half cells of DMF and dioxane together and, equilibrating the systems using two Langevin thermostats at 300 K, one for each subsystem, for 500 ps, after which the two systems were permitted to mix in the micro-canonical ensemble for 3 ns.

The P84 oligomers were solvated using the *genbox* utility from the GROMACS MD suite. This works by overlaying the solvent box with the oligomer box and adding all the molecules which do not overlap the oligomer. If the oligomer's box is larger than the solvent box periodic copies of the solvent box are added until it is larger. The *genbox* utility defines overlap based on a fixed radius for all atoms in the system. The default radius, which this work uses, is 0.12 nm. Therefore, the minimum separation between molecules is 0.24 nm, which is less than the 0.3-0.4 nm typically expected for aromatic molecules, meaning the solvent molecules are often slightly too close to the polymer. Furthermore, the resulting system is often below the target density³. Both these issues can be addressed by

¹Discover is the MD program within Materials Studio 6.

²Throughout this work any reference to water in an MD simulation implies TIP3P water unless explicitly stated otherwise.

³*genbox* defines the atomic masses based on their names in the input file. This process does



performing a structural relaxation of the system, and if necessary adding more solvent molecules using *genbox*.

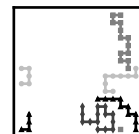
The simulation cells were constructed for the eight shorter P84 oligomers in DMF, dioxane, and water. These cells were simulated at 300 K using a Langevin thermostat for 5 ns, they were then allowed to equilibrate in a micro-canonical ensemble for a further 5 ns. The specified PME mesh for these simulations was $128 \times 128 \times 128$. It was decided that the significant increase in performance afforded by using the v4.6.2 (double precision) release of GROMACS, which increased the throughput from 16.3 ns day^{-1} to 23.5 ns day^{-1} , justified switching from the v4.5.5 version of the source code. However, this necessitated changing the cut-off settings as the v4.6.x series of GROMACS does not support switched non-bonded potentials. Therefore, a cut-off radius of 1.2 nm was used with a shift applied to both the potential and the force such that both were exactly zero at the cut-off radius for both the Coulomb and LJ interactions. The value of 1.2 nm was selected as it accurately reproduced the diffusivity of DMF observed over the central 1.6 ns of a 2 ns micro-canonical ensemble calculation using a periodic cubic box containing 4259 DMF molecules in v4.5.5. The total non-bonded energy for the simulation was also observed to differ only by a constant shift relative to the v4.5.5 simulation. The 4.6.x series of GROMACS adapts the Coulomb cut-off radius and PME mesh to balance the load during a simulation. It does this by varying the Fourier grid spacing in the PME mesh and the Coulomb cut-off radius whilst keeping their ratio constant. As a result of this the values specified in this section do not necessarily indicate the values used by GROMACS v4.6.2, as these will depend on computer architecture and load balance.

The oligomers were simulated in every solvent combination (DMF, dioxane and water) for an initial period of 300 ns. This choice was motivated by the convergence of the average radius of gyration for the oligomers containing 6 BTDA units simulated in dioxane at 250 K, see Fig. 7.2¹. However, it was realised that simulating at a rate of 23.5 ns day^{-1} on 144 cores² would require almost 3×10^5

not always result in the correct mass. Therefore, users are strongly encouraged to calculate the density independently to ensure the correct value is obtained.

¹This configuration was chosen as it was the lowest temperature, in the poorest solvent and, therefore, needed to overcome the worst energy barriers.

²This was tested on cx2 Imperial College High Performance Computing cluster, using the



core hours. This amount of computer time was not readily available. Therefore it was decided to reduce the number of oligomers being simulated from 10 to 5, by choosing only a single oligomer for each number of MDA groups from all of the short oligomers containing 6 BTDA groups. Furthermore, it was decided to simulate the two longer oligomers containing 9 BTDA groups only in DMF. This resulted in a total of 55 simulation boxes instead of 150.

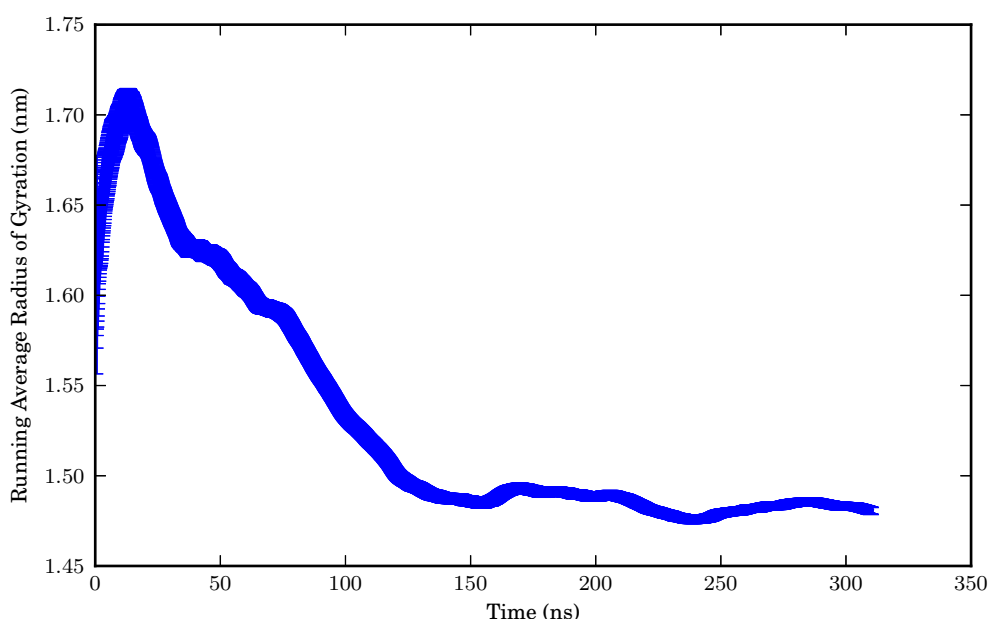
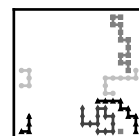


Figure 7.2: The running average of the mean radius of gyration, averaged across all oligomers containing 6 BTDA groups. The running average covers all times up to and including the specified time and the statistical error on the average is given by the size of the plotting symbol.

The simulation boxes were relaxed in the canonical ensemble for 5 ns, using a Langevin thermostat. They were then equilibrated for a further 5 ns in the micro-canonical ensemble. These equilibrated configurations were then used as the

Intel Xeon-E5 sub-system, machines such as HECToR and Shaheen achieved significantly lower peak performances of 14.4 ns day^{-1} on 288 cores and 8.2 ns day^{-1} on 512 cores respectively. As recent gcc or intel compilers are unavailable on Shaheen BlueGene the observed performance specified is for P84 in water simulated in gromacs v4.5.7 as v4.6.2 obtained less than 2 ns day^{-1} in initial testing and was, therefore, not tested further.



inputs for 300 ns micro-canonical simulations, this process took almost 6 months to complete.

Unfortunately it was discovered after these simulations had completed that the v4.6.x series of GROMACS has a bug, see Sec. 5.3 for details of how this was discovered. The nature of the bug meant that not all increments¹ of the simulations were affected, therefore, as only some of the simulations were checked as they were being run, the bug was not identified until the entire set of simulations was analysed. As the bug affected the energy conservation of the micro-canonical simulations, none of the data from these simulations can be considered reliable. Due to time constraints, and the large amount of computer resources which would be necessary, it was not possible to repeat these simulations using an MD code free from this bug. Therefore, it was decided to obtain both the persistence length data and the interaction parameter from simulations of complete P84 molecules at experimental concentrations.

7.5 P84 at Experimental Concentrations

A complete P84 molecule has a mass of around 62000 g mol^{-1} . As a random co-polymer the exact number is variable. However, its dispersity is 1.46[190], indicating that the mass of any given P84 molecule will not be far from this value. This work uses the distribution of monomers given in Fig. 6.1, which yields an average mass² of a P84 monomer of $423.59 \text{ g mol}^{-1}$, and an average of 6587 atoms in a complete P84 molecule. P84 phase inversion membranes are typically manufactured using a dope containing 24wt% of P84. The solvent used is a mixture of DMF (12 atoms, 73.1 g mol^{-1}) and dioxane (14 atoms, 88.1 g mol^{-1}). Since the solvents have similar masses a simulation cell at experimental density will consist of around 38000 atoms for each P84 molecule. Therefore, it was decided, based on the available computational resources, that two simulation cells each containing a single P84 molecule would be simulated for 300 ns at 300 K in the micro-canonical ensemble.

¹Increments refers to the period of the simulations between restarts.

²The atomic masses from the OPLS-AA forcefield have been used throughout this section { C= $12.01100 \text{ g mol}^{-1}$, H= $1.00800 \text{ g mol}^{-1}$, N= $14.00670 \text{ g mol}^{-1}$, O= $15.99940 \text{ g mol}^{-1}$ }.



Following the problems encountered with GROMACS v4.6.2 (see Sec. 5.3 and Sec. 7.4) it was decided to use GROMACS v4.5.5 (double precision) which had been tested extensively during the early stages of this project, and particularly whilst building the force fitting program (see Sec. 4.1.5 and Sec. 7.2). It was also decided to obtain both the FH coefficient and the persistence length of the polymer dope from these simulations, which introduces two issues. The first is that, without the Θ temperature determined from the dilute simulations, there is no direct point of comparison between the MC and MD, which will be discussed in Chap. 8. The second is that it is not possible to enable a P84 molecule to equilibrate fully in a MD simulation at this density and temperature, and certainly not in 300 ns. Therefore, when calculating the persistence length the assumption must be made that whilst the large scale structure of the full polymer is not relaxed, small sections of the polymer will have reached equilibrium. This assumption is discussed with the results of the persistence length calculations in Sec. 7.5.3.

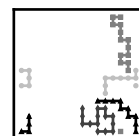
7.5.1 Theory

The behaviour of polymers occurs on many length and time scales. One length scale which is of particular interest for our model is the persistence length (l_p). There are several definitions of this length and the relationships used to define it have been shown to hold only under certain regimes[35]. However, the persistence length does provide a useful characterisation of the polymer's behaviour. Throughout this section we will define the persistence length by

$$\langle \hat{\tau}(l) \cdot \hat{\tau}(l+s) \rangle_{t,l} = e^{-\frac{s}{l_p}}, \quad (7.1)$$

where l and s are arc-length positions along the polymer, $\hat{\tau}(l)$ is the unit tangent vector at position l , and t is time. This definition is designed for ideal polymers, but an exponential decay is expected to be a reasonable approximation to the observed behaviour of P84.

The interaction between separate molecules in the OPLS-AA forcefield is mediated by the non-bonded component of the potential. Non-bonded interactions are interactions which depend only on the relative positions of the atoms, and



consist of electrostatic and LJ energies. Since these are known quantities it is possible to evaluate them directly. However, this does not directly tell us the amount of energy associated with mixing a polymer with the solvent. During a mixing process the conformation of the polymer and the structure of the solvent are altered, which changes all the terms in the forcefield. Calculating a value for this interaction energy requires performing separate simulations for each of the components of the mixture as well as a simulation of the mixture itself. These simulations allow the enthalpy of mixing to be calculated as

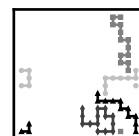
$$\Delta H_{\text{mix}} = H_{\text{mixture}} - H_{\text{Polymer}} - H_{\text{Solvent}}, \quad (7.2)$$

where H_A is the enthalpy extracted from a simulation of A [134]. This is the total amount of energy required to mix the solvent and polymer system.

The enthalpy of mixing is an important term when considering dopes. However, following the issues with thermostats discussed in Chap. 5 it was decided not to use isobaric or isothermal simulations. It was decided that micro-canonical simulations would be used as more testing of the GROMACS simulation package had been conducted in this ensemble. The resulting configurations will be subtly different from those of a canonical ensemble. However, the temperature drift during the simulations was minimal and there are not expected to be any phase transitions at 300 K at these densities. Despite this the pressure was found to vary significantly throughout these simulations; therefore, the calculation of the enthalpy of mixing remains a topic for future work. Therefore, the simulations should provide a reasonable value for the enthalpy of mixing. The need to extract the results from only a small number of simulations resulted in a multi-stage simulation procedure.

7.5.2 Simulation Details

All simulations were conducted using GROMACS v4.5.5, using the modified OPLS-AA forcefield, described in Sec. 7.3. A time step of 1 fs was used in all simulations. Unless otherwise stated covalent bonds involving hydrogen were modelled with rigid constraints enforced by the P-LINCS algorithm with LINCS-order 6, and 3 LINCS iterations. Both the electrostatic and vdW interactions



were treated with a 1.1 nm cut-off radius, with a switching function applied over the last 0.05 nm to smoothly taper both the forces and the potential energy to zero at the cut-off. The electrostatic interactions also used PME with a 0.07 nm grid spacing for periodic simulations¹.

Neighbour searching used a grid based algorithm with a 1.5 nm cut-off with a neighbour list lifetime of 10 steps. This set up was chosen to ensure accurate energy conservation and ensure exact cut-off radii were enforced by the group based cut-off scheme in GROMACS v4.5.5. The resulting simulations maintain excellent energy conservation albeit at a high computational cost.

When a canonical ensemble was used it was enforced by a Langevin thermostat with a temperature coupling constant $\tau_T = 5$ ps. The traditional Langevin coupling constant (λ) is given by

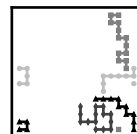
$$\lambda_i = \frac{m_i}{\tau_T}, \quad (7.3)$$

where m_i is the mass of the i^{th} particle. The use of a large value for τ_T (> 0.5 ps) results in weak temperature coupling which enables large conformational changes as the friction coefficient is reduced. However, this does increase the time necessary to add or remove kinetic energy to the system which can necessitate long equilibration simulations. To reduce the equilibration time an initial set of velocities for each canonical ensemble simulation were generated from the Maxwell-Boltzmann distribution.

The two solvated simulation boxes used the same P84 molecule. It was initially constructed in Materials Studio[®] v6.1 as a random co-polymer. The resulting molecule contained 147 BTDA groups, 126 TDA groups, 21 MDA groups² and was terminated with methyl groups, for a total of 6539 atoms. The coordinates were converted for use in GROMACS v4.5.5 and a molecular topology created. The molecule was initially simulated in a canonical ensemble, at 2000 K for 500 ps with harmonic potentials used for bonds involving hydrogen. This was used to remove the intrinsic biases applied during the construction of the initial

¹Individual atoms were treated as charge groups during these simulations.

²This is $14 \pm 3 \text{ wt}\%$ MDA which is within two standard errors of the ideal value and therefore, acceptable.



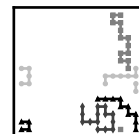
coordinates. Following this the P84 molecule was placed in a cubic cell with lattice parameter 7.53 nm representing a density of 0.24 g cm^{-3} . A structural relaxation was performed, using harmonic covalent bonds for hydrogen, to remove close approaches between atoms. The system was then simulated in a canonical ensemble at 2000 K for 500 ps. The resulting molecular conformation of P84 was used as the starting point for calculations in solvent.

The construction of the initial solvent boxes for DMF, dioxane and a mixture of the two solvents is described in Sec. 7.4. These simulation boxes were used with the *genbox* utility from GROMACS to generate the initial solvated polymer boxes. The pure DMF (75.6wt%) box was packed to a density of 0.98 g cm^{-3} , and the mixture of DMF (38.0wt%) and dioxane (38.0wt%) was packed to a density of 1.00 g cm^{-3} . These densities were achieved by filling the boxes with solvent and performing 500 ps canonical ensemble calculations at 2000 K followed by structural minimisations. The resulting structures contained small voids which were packed with additional solvent molecules. The process was repeated until the target number of solvent molecules was reached.

Once the two boxes had been constructed a 500 ps, 2000 K canonical ensemble calculation was conducted, using harmonic potentials for the covalent bonds involving hydrogen. This was followed by a further 500 ps in the micro-canonical ensemble, also using harmonic bonds. A structural relaxation was then performed to remove any highly non-physical atomic positions. The two boxes were then simulated at 300 K in the canonical ensemble for 1 ns, and equilibrated in the micro-canonical ensemble for 5 ns. The resulting states were then used as the inputs for 300 ns micro-canonical ensemble simulations.

The multi-stage process described above was designed to eliminate biases as much as possible from the initial construction of the polymer dopes. This is advantageous because it allows credible physical interactions between the polymer and solvent molecules to determine the structure of the solvated system.

For the pure P84 system the final structure from the DMF production simulation was used as the starting configuration for the simulation. In the case of the solvents the simulation cells were constructed by using *genbox* utility to build cubic boxes of side 4.02 nm and 3.98 nm for the DMF and mixed solvent boxes respectively. The resulting boxes had densities of 0.950 g mol^{-1} and 0.990 g mol^{-1}



respectively, which closely match the experimentally determined densities of the solvents[190].

These boxes were then simulated in the canonical ensemble for 500 ps at 2000 K using harmonic potentials for the covalent bonds involving hydrogen. They were then allowed to evolve in the micro-canonical ensemble for a further 500 ps, to remove any bias from the initial generation procedure. The systems were then simulated for a further 1 ns in the canonical ensemble at 300 K, followed by 1 ns in the micro-canonical ensemble. The resultant configurations were used as the inputs for 10 ns production micro-canonical ensemble simulations.

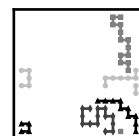
7.5.3 Results

The drift in total energy was monitored in all simulations, and is shown in Tab. 7.1. From these drifts it is clear that the energy was well conserved during the simulations. However, the drifts are somewhat smaller than is necessary for simulations of this length. Therefore, it might have been possible to increase the time step from 1 fs to 2 fs which is often used in biological simulations with constraints[150], although this was not verified directly. The temperatures of the simulations are shown in Tab. 7.2, and again show negligible drift over the course of the simulation. This indicates that the configuration was in a stable region of phase space and suggests that the simulations should be reasonable representations of the behaviour of the systems at 300 K.

	Energy Drift (meV ns ⁻¹ atom ⁻¹)	Total energy (meV atom ⁻¹)
P84 in DMF	$(28 \pm 1) \times 10^{-7}$	98.9
P84 in 1:1	$(27 \pm 4) \times 10^{-7}$	119.0
P84	$(9 \pm 9) \times 10^{-5}$	436.7
DMF	$(3 \pm 6) \times 10^{-5}$	32.3
1:1	$(7 \pm 6) \times 10^{-5}$	56.6

Table 7.1: The energy drifts and total starting energies per atom for each of the simulations. 1:1 is an equal mass mixture of DMF and dioxane.

The tangent vectors to the polymer were calculated as the unit vectors connecting the centres of mass of sequential groups along the backbone of the poly-



P84 at Experimental Concentrations

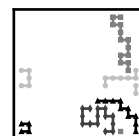
	Temperature Drift (K ns ⁻¹)	Average Temperature
P84 in DMF	$(22 \pm 6) \times 10^{-5}$	301.209 ± 0.004
P84 in 1:1	$(1 \pm 1) \times 10^{-4}$	300.12 ± 0.01
P84	$(14 \pm 3) \times 10^{-2}$	303.99 ± 0.07
DMF	$0 \pm 3 \times 10^{-2}$	300.98 ± 0.07
1:1	$0 \pm 3 \times 10^{-2}$	301.68 ± 0.08

Table 7.2: The temperature drifts for each of the simulations. 1:1 is an equal mass mixture of DMF and dioxane.

	Pressure (eV nm ⁻³)
P84 in DMF	-4.19 ± 0.01
P84 in 1:1	-7.25 ± 0.03
P84	-1.28 ± 0.05
DMF	9.4 ± 0.2
1:1	7.4 ± 0.2

Table 7.3: The Pressures for each of the simulations. 1:1 is an equal mass mixture of DMF and dioxane.

mer. The groups are defined as the TDA, MDA sub-units of the monomer, the terminal methyl groups and the two halves of the BTDA group with the central carbonyl group taken to belong exclusively to one half of the sub group. The vectors connecting these groups also define the arc-length parameter s . The correlation defined in Eq. (7.1) was calculated using frames separated by 5 ns. The results for the whole polymer are shown in Fig. 7.3. Neither solvated polymer has effectively sampled the complete configuration space available to it. However, the more closely spaced monomers, shown in Fig. 7.4, have sampled the configurational space available to them more effectively. The initial rise of the correlation function is caused by the chemistry of the P84 molecule which prevents the polymer from behaving as a freely jointed chain at this scale. Therefore, the persistence length of the two polymers was fitted to the tangent vectors separated by less than 12 nm, yielding 1.105 nm for P84 in DMF and 1.158 nm for P84 in the mixed solvent. By varying the distance cut-off used when calculating the persistence lengths, see Fig. 7.5, we can see that the persistence length is reasonably insensitive to the exact value of the cut-off used.



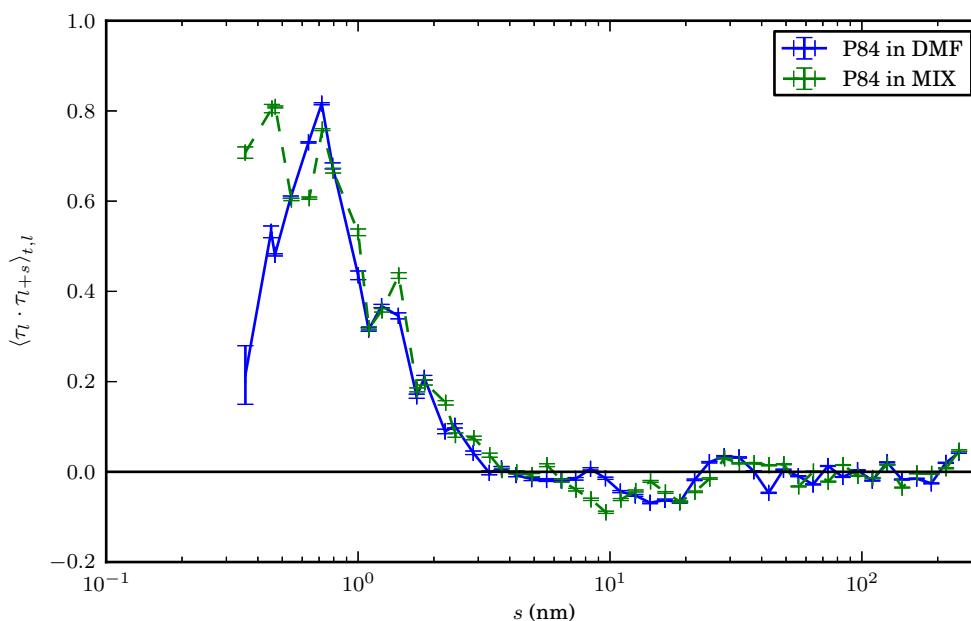
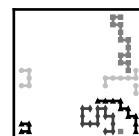


Figure 7.3: The correlation of tangent vectors separated by s along the back bone of 24 *wt%* P84 in pure DMF and a equal mass mixture of DMF and dioxane. The initial decay in correlation is inhibited because the P84 molecule is not a freely jointed chain. As the tangent vectors become further apart the decay behaviour asserts itself. However, at large separations the conformational changes necessary to sample the region have not taken place with the result that the correlation does not decay to zero. Each data point represents an average as there were in excess of 5 million individual evaluations of several values of separation s .



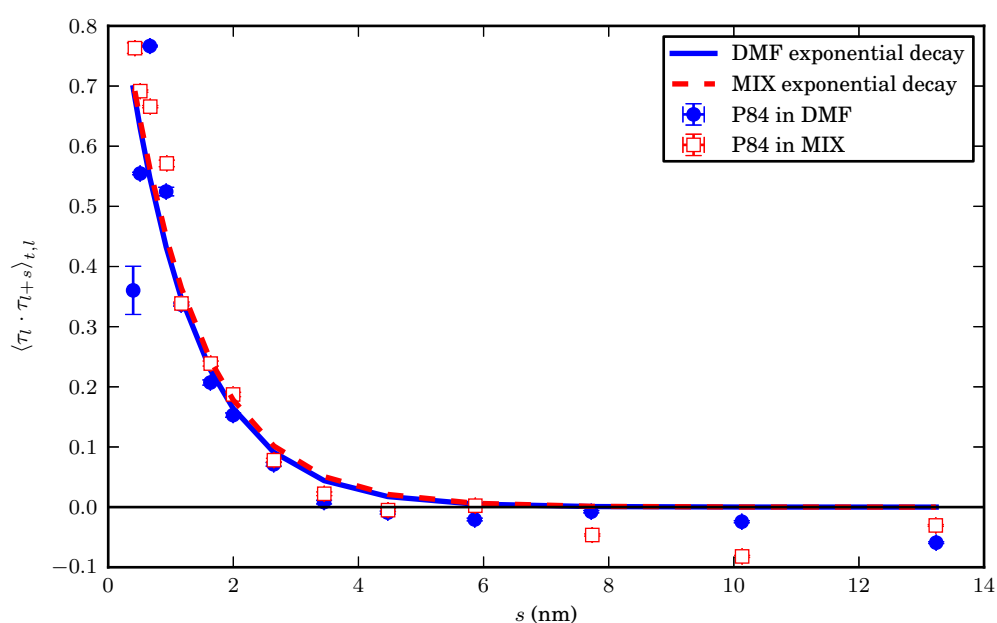


Figure 7.4: The correlation of tangent vectors separated by s along the back bone of 24 wt% P84 in pure DMF and a equal mass mixture of DMF and dioxane. This graph shows more detail of the behaviour of the correlation for separations of less than 15 nm. Also shown are the exponential decay functions fitted to the complete data sets.



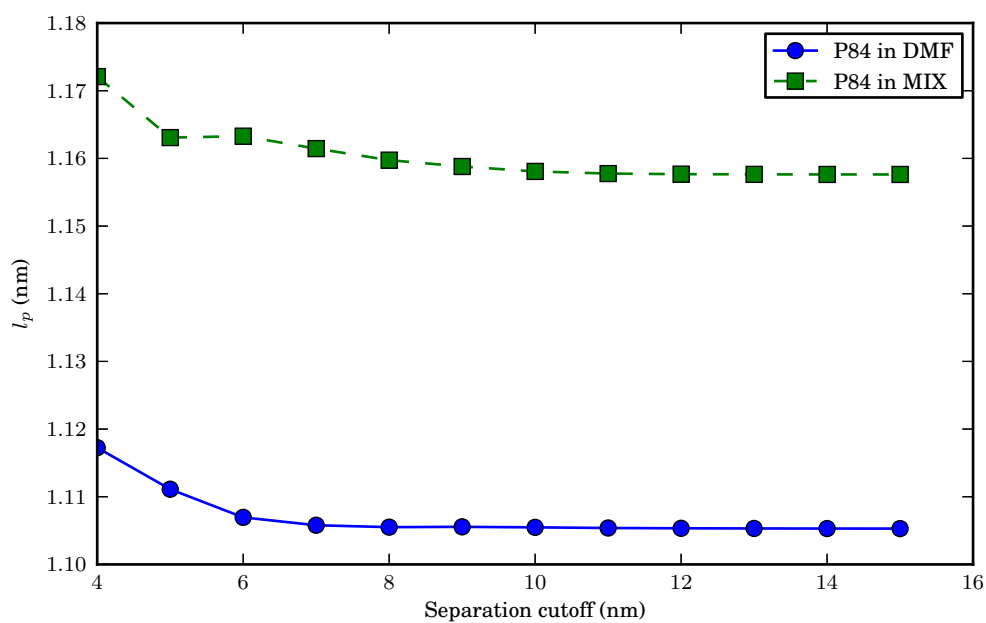


Figure 7.5: The persistence length against separation cutoff for 24 *wt%* P84 in pure DMF and a equal mass mixture of DMF and dioxane. Error bars are comparable to the line width.



The calculation of the enthalpy of mixing requires equal pressures between the simulations. However, the pressures in the simulations, shown in Tab. 7.3, were not similar, therefore, it is not possible to derive an enthalpy of mixing from these simulations.

7.5.4 Conclusion

The simulation of industrial polymers under experimental conditions is a highly complex process. The growth in computational capabilities in the late 20th and early 21st centuries has made atomistic modelling of these highly complex systems feasible. The simulations conducted in this chapter approach the limit of current simulation capabilities. It is, therefore, unsurprising that issues were encountered. These issues, particularly the loss of over 6 months worth of simulation time due to a software bug have impacted on the final results. The inability to study the variation in persistence length with respect to temperature makes it difficult to directly compare the behaviour of P84 observed in these MD simulations with the behaviour observed in the MC simulations in Chap. 8. Furthermore, whilst the enthalpies of mixing calculated in these simulations do show the trends we expect for P84, they do not cover all the solvent concentrations that would ideally be used to for comparison with MC, and experiments. Despite these setbacks the MD simulations described above have shown, that on intermediate length scales the persistence length model does reflect the behaviour of P84, and that the OPLS-AA forcefield succeeds in demonstrating the expected trends in enthalpy of mixing for P84 in DMF and dioxane. Therefore, MD simulations similar to those described in this chapter have the potential to be of use in the development of industrial polymer systems.



Chapter 8

Monte Carlo Simulations of P84 Membranes

The large length and time scales involved in equilibrating multiple P84 molecules in solution makes dynamically simulating the process impractical, and quite probably impossible. However, MC is ideally suited to studying the properties of systems with large numbers of degrees of freedom, such as polymers. Furthermore, it can provide huge amounts of information on the equilibrium properties of a system without the need to understand the physical mechanism by which the system moves between different possible micro-states. MC is, therefore, ideally suited to studying the behaviour of large collections of P84 molecules in solution.

As is often the case with MC models, choosing a model to use for P84 membranes is not simple. The process began with a consideration of polymers as chains of LJ beads. However, this model fails to incorporate the polymer solvent interaction. The simplest solution is to include explicit solvent particles in the system. However, the number of dimensions in a problem scales exponentially with the number of particles in the system. Therefore, this appealing approach rapidly becomes untenable and a different approach is required.

Lattice models have an advantage over continuum models: their phase space is vastly smaller. It was therefore decided to model the P84 solution on a lattice. The first lattice model considered was a one dimensional model consisting of layers of polymer and solvent, discussed in Sec. 8.1. However, the model which



was eventually decided upon was a widely used lattice polymer model[199, 200, 201], based on FH solution theory[128, 129]. This was initially tested on a two dimensional square lattice, see Sec. 8.2, before being upgraded to work in three dimensions, as discussed in Sec. 8.3.

8.1 Layers Model

An initial attempt to gain qualitative insight into the molecular structures within the P84 membranes tested the idea that the bunching up or spreading out of P84 molecules within a membrane could be captured by looking at the volume change associated with phase inversion. The model consisted of polymer chains arranged in flat uniform layers. This is amongst the most basic approximations which can be made for a polymer and was not expected to be accurate but rather to be the first stage in understanding the behaviour of the polymer solvent mixture. The polymer layers have a thickness of 0.5 nm, determined by the maximum width of the P84 monomer, and were separated from adjacent polymer layers by the equilibrium spacing of a pair of benzene dimers, see Fig. 8.1. The solvent layers were constructed by looking at the typical density of DMF and dioxane at standard conditions;¹, then using this to define a sphere with the same volume as a single molecule, the diameter of 0.6 nm was found to apply to both solvents. Layers adjacent to solvent had no separation associated with them, see Fig. 8.2

A polymer layer can be in one of three states:

1. Fully solvated, where a single mono-layer of solvent covered each side of the polymer layer. In this case they have a thickness given by twice the solvent thickness plus the thickness of a P84 molecule. This thickness (D), is shown in Fig. 8.2 (a).
2. Partially solvated whereby only one mono-layer of solvent was added to one side of the polymer layer and half the equilibrium separation of two benzene dimers (0.2 nm) added to the other to approximate the vdW separation between two P84 molecules Fig. 8.2 (b).

¹300K and 10^5 Pa



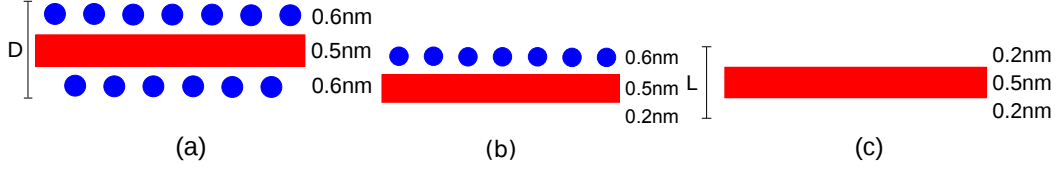


Figure 8.1: Schematic diagrams showing a solvated (a), partially solvated (b), and un-solvated (c) layers of polymer. Red rectangles represent the layers of polymer, and blue circles indicate layers of solvent. The thickness of a solvated layer (D) and an unsolvated layer (L) are also shown.

3. Un-solvated where a mono layer of polymer was created by adding half the equilibrium separation of two benzene dimers above a layer of P84 and the same below as shown in Fig. 8.2 (c). The thickness labelled L is then the width of a P84 layer plus the equilibrium separation of two benzene dimers.

From Fig. 8.1 it is clear that two partially solvated layers are equivalent in thickness to a solvated layer and an unsolvated layer. The thickness of the solvated system, (T_i), is therefore

$$T_i = N(cL + (1 - c)D), \quad (8.1)$$

where N is the total number of polymer layers and c is the number fraction of unsolvated layers. Assuming that during phase inversion the solvent is completely removed the final thickness of the membrane is simply

$$T_f = NL, \quad (8.2)$$

which yields the fractional volume change

$$\Delta = 1 - \frac{1}{c\left(1 - \frac{D}{L}\right) + \frac{D}{L}}. \quad (8.3)$$

This is shown in Fig. 8.3, as can be clearly seen the thickness change decreases as c increases. c is expected to decrease as the quality of the solvent increases.

Experiments demonstrated that the thickness change of the membrane increases as solvent quality decreases[190], the reverse of the trend shown by this model. The reason for this is likely the fact that the model doesn't take account



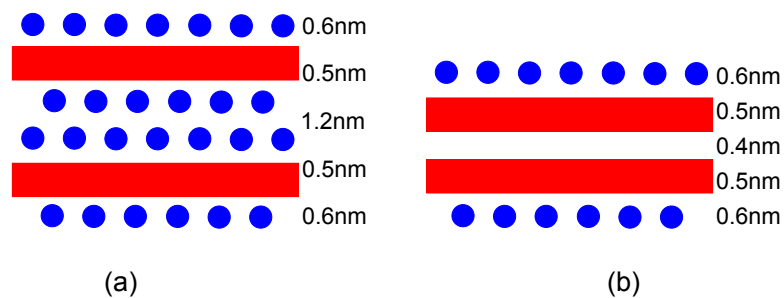


Figure 8.2: Schematic diagrams showing, two solvated layers (a), a bundle of two layers (b). Red rectangles represent the layers of polymer, and blue circles indicate layers of solvent. The thickness of the solvent layers comes from the density of DMF and dioxane. The polymer separation is taken to be the typical separation of two benzene dimers. The polymer layer has the thickness expected from adding together the bond lengths within a P84 molecule.

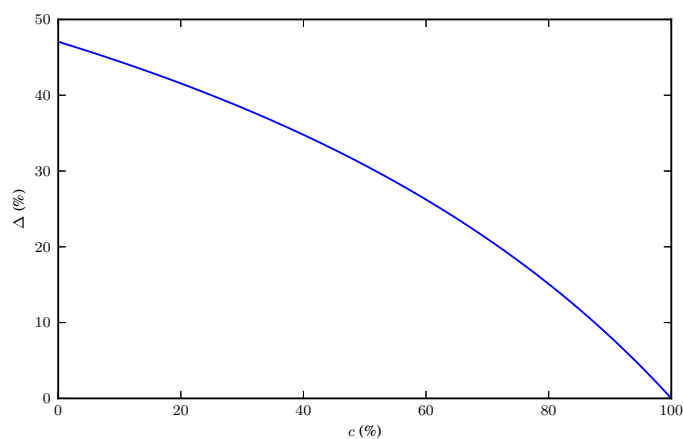


Figure 8.3: Fractional thickness change against concentration of unsolvated polymer layers within the polymer layers model.



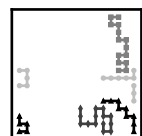
of the formation of finger like macro-voids which can be seen in the SEM images of P84 membranes[6, 7, 8, 9], and shown in Fig. 3.1 for cellulose acetate membranes. Based on the experimental observations available at the time it was decided that a more complex model was required which included both the solvent-polymer interactions and the polymer's geometry. It is worth noting that 2 years after this divergence between experimental observation and this model was developed, high resolution TEM experiments showed that the trend shown in this model does match that observed in the thickness of the surface layer of P84 membranes[6]. Whilst this does not change the need for the more complex MC approaches outlined in the remainder of this chapter it is interesting to see that simple models can sometimes match experimental trends.

8.2 Two Dimensional Lattice Model

Moving to two dimensions allows some of the geometrical freedom of the polymer to be included in the model. The model chosen derives from FH solution theory[128, 129], as discussed in Sec. 3.3.2. The model consists of a lattice filled with two types of sites: polymer (p) and solvent (s). The energy associated with a site is based only on its neighbours with energies given by

$$E_i = \frac{1}{2} \sum_{j=1}^z E_{t_i t_j}, \quad (8.4)$$

where j indexes all the neighbouring sites to site i , z is the lattice coordination number, and $t_i \in \{p, s\}$ is the type of site in position i , and $E_{t_i t_j} \in \{E_{pp}, E_{ss}, E_{ps}\}$, is the energy associated with the pairing. However, unlike in FH theory, the polymer is restricted to be a line of connected neighbouring sites. This restriction means that the model cannot be solved analytically and must, therefore, be modelled computationally. When considering a fixed number of polymers of a fixed length in a fixed volume of solvent, the only condition considered in this work, sites can only be converted from one type to another in pairs e.g., if you change one site from s to p a different site must be changed from a p to an s to maintain the fixed numbers of sites of each type. This results in the useful



property that all energy changes in the system can be considered in terms of

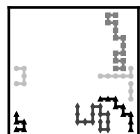
$$\Delta\varepsilon = \frac{1}{2} (E_{pp} + E_{ss}) - E_{ps}. \quad (8.5)$$

This single parameter incorporates all the detail of the polymer solvent energetics in the model, $\Delta\varepsilon < 0$ indicates a *poor* solvent, whilst $\Delta\varepsilon > 0$ indicates a *good* solvent and $\Delta\varepsilon = 0$ is an intermediate solvent or equivalently a SAW. A square lattice with periodic boundary conditions was chosen for the model because this permits simple handling of sites near the edge of the simulation box and permits the simulation of bulk behaviour. A MH procedure was used to simulate the model, which has the advantage that only the energy difference between two states is required, meaning that only $\Delta\varepsilon$ from Eq. (8.5) need be defined. Furthermore, since $\Delta\varepsilon$ is only used in the MH procedure it can be defined in terms of $k_B T$ reducing the number of simulation parameters. The model naturally accommodates multiple polymers and energy evaluations are computationally simple which makes it ideal for simulating large groups of polymers.

A reptation algorithm[46, 47] was used to change the system's configuration. A reptation move involves removing a polymer site from one end of a polymer and placing a new polymer site at the other end. This makes the assumption that the solvent would always be able to enter the lattice site left by a polymer moving. In a physical system this is not necessarily true. However, since this simulation is not kinetic but rather a means to sample configurations, any valid¹ arrangement of polymer and solvent sites is suitable for inclusion in the MH integration. An attempt to place a polymer site onto another polymer site was treated as a rejected move. This process is inherently more wasteful and samples fewer allowable configurations for a given number of iterations than a biased sampling scheme such as RR sampling. However, the computational cost of the moves never proved to be limiting factor in the MC simulations. Therefore, whilst algorithms were written for RR biased sampling in the model they were never extensively tested and no data from those schemes is presented in this work.

An initial state for the system was generated by growing a polymer site by site

¹In this context valid implies no overlapping molecules, no gaps in the polymer, and the correct total number of sites of each type.

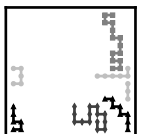


considering only if the trial site was free and trying a different site if it was not free. A polymer was considered to have failed to grow if there were no free sites to place the next site or if there was not at least one free site neighbouring an end of the polymer. This second case prevents the unmovable caged configurations described in Sec. 2.1.2.2. In the event of a growth failure the polymer was deleted and a new polymer grown. Once the polymer was generated the next was grown until all the requested polymers had been grown. The resulting configuration suffers from the issue of configurational bias, and does not factor the energy of the configuration. The reptation algorithm was used for a sufficient number of steps to permit each polymer to move half the space diagonal of the simulation box, in order to remove any inherent bias in the initial configuration. The simulation then progressed for a fixed number of steps and various properties of the configurations were recorded. An example showing every tenth configuration from a simulation of four polymers of length 14 on a 16×16 lattice, with $\Delta\varepsilon = 0$, can be seen in the lower right hand corner of the pages of this thesis. However, it was found that it was extremely computationally expensive to generate dense two dimensional configurations as polymers often became densely packed and failed the criterion of having at least one free site to move onto. Therefore, it was decided to move to a three dimensional lattice where the increased degrees of freedom would reduce the likelihood of the polymer becoming caged and be more physically realistic by allowing polymers to move past one another.

8.3 Three Dimensional Lattice Model

Moving the MC model onto a three dimensional lattice greatly improved its ability to generate high density polymer configurations. The process was accomplished by creating an alternative code path which included additional neighbouring sites. This permits a cubic lattice to be used instead of a square lattice with no other changes to the model described in Sec. 8.2. After some minor changes to the implementation to improve the performance of the algorithm it was clear that the algorithm could be used even on large systems¹ in less than 72 hours on a

¹Large here implies a $512 \times 512 \times 512$ cubic lattice packed to 15% volume concentration with polymers of length 470 sites.



standard desktop. The model was initially tested using a single polymer, this was then extended to testing of the behaviour of the polymer at different densities. High density simulations were then used to generate representative density maps to establish whether the behaviour seen in experimental images of P84 phase inversion membranes could be predicted in this model.

8.3.1 Single Polymers

The first tests of the three dimensional lattice MC program modelled a single polymer chain in the simulation cell. The radius of gyration,

$$R_g^2 = \frac{1}{N} \sum_{i=1}^N (\mathbf{r}_i - \langle \mathbf{r}_j \rangle)^2, \quad (8.6)$$

where \mathbf{r}_i is the position of the i^{th} site, and N is the number of sites, is a good measure of the size of a polymer as it includes all sites in the polymer, making it slow to equilibrate. It was therefore selected as the measure to check the behaviour of the polymer chains was consistent with the expected behaviour of the model. A series of simulations of single polymer chains with lengths $N \in \{25, 50, \dots, 225, 250\}$ were conducted for values of $\Delta\epsilon \in [-1, 1]$. Each simulation contained 10^7 MC steps¹ and 30 repeats of each set of simulation parameters were made with different random number seeds to establish consistency. The results for the radius of gyration in Fig. 8.4 clearly show that, as expected, when the polymer solvent interaction energy (E_{ps}) is high ($\Delta\epsilon < 0$) the polymer contracts to minimise its contact area with the solvent, and similarly when the E_{ps} is low ($\Delta\epsilon > 0$) the polymer expands to maximise its contact area. The larger error at $\Delta\epsilon \ll 0$ is due to the increasing difficulty of making reptation moves as the number of unfavourable polymer-solvent neighbour pairings increases. The Θ point for this model was estimated, from the intersection of linear interpolations

¹Here and throughout this chapter the number of MC steps refers to the number of production steps after equilibration. The equilibration period varies based on the number of polymers, and is equal to half the length of the space diagonal of the box times the number of polymers rounded upwards. A test was conducted using 30 simulations with an equilibration period 50 times longer, on a chain of length 200 the resulting radii of gyration were in perfect agreement with $\chi_{\text{red}}^2 = 0.07$.



between the data points, to be $\Delta\varepsilon = -0.15$. The $N = 25$ polymer does not hit the intercept value as the concept of the Θ point comes from a scaling argument which does not hold for short polymers. As expected this is lower than the FH value:

$$\Delta\varepsilon_{\Theta}^{\text{FH}} = -\frac{\chi_{\Theta}^{\text{FH}}}{z} = -\frac{1}{12}, \quad (8.7)$$

where the FH coefficient $\chi_{\Theta}^{\text{FH}} = \frac{1}{2}$ at the theta point and the lattice coordination number $z = 6$ for a square lattice, because the FH model does not take into account the restriction in the positioning of the sites due to the connected nature of polymer segments[52]. The result is also inline with the results of Panagiotopoulos *et. al.* Ref. [199]. The behaviour exhibited by these simulations provides an excellent match against the expectations of the model and is highly consistent between simulations. Therefore, it was decided to test simulations containing multiple polymers.

8.3.2 Multiple Polymers

Simulations of multiple polymer chains are inherently more complicated than simulations of a single polymer. The number of possible configurations scales exponentially with the number of polymers. Care must, therefore, be taken to ensure that the simulation is well equilibrated before statistics are collected. The advantage of simulations involving multiple polymers is that it is possible to extract properties from all the molecules and confirm that they are behaving in a consistent manner. This provides a good check that the system is well equilibrated.

The calculations in the previous section were repeated using a pair of identical polymers with the number of MC steps increased to 10^8 . Each set of simulation parameters was repeated 30 times with different seeds to ensure consistent results. The calculations were conducted on a $256 \times 256 \times 256$ cubic lattice, making the density¹ for the pair of 250 site polymers $3.0 \times 10^{-3}\%$. Therefore, minimal interaction between the polymers is expected as long range interaction is not included in the model. The radii of gyration of each polymer showed perfect agreement, ($\chi_{\text{red}} = 0.035$) between the pairs, indicating that the simulation treated the two

¹Density here refers to the fraction of the simulation cell occupied by the polymer.



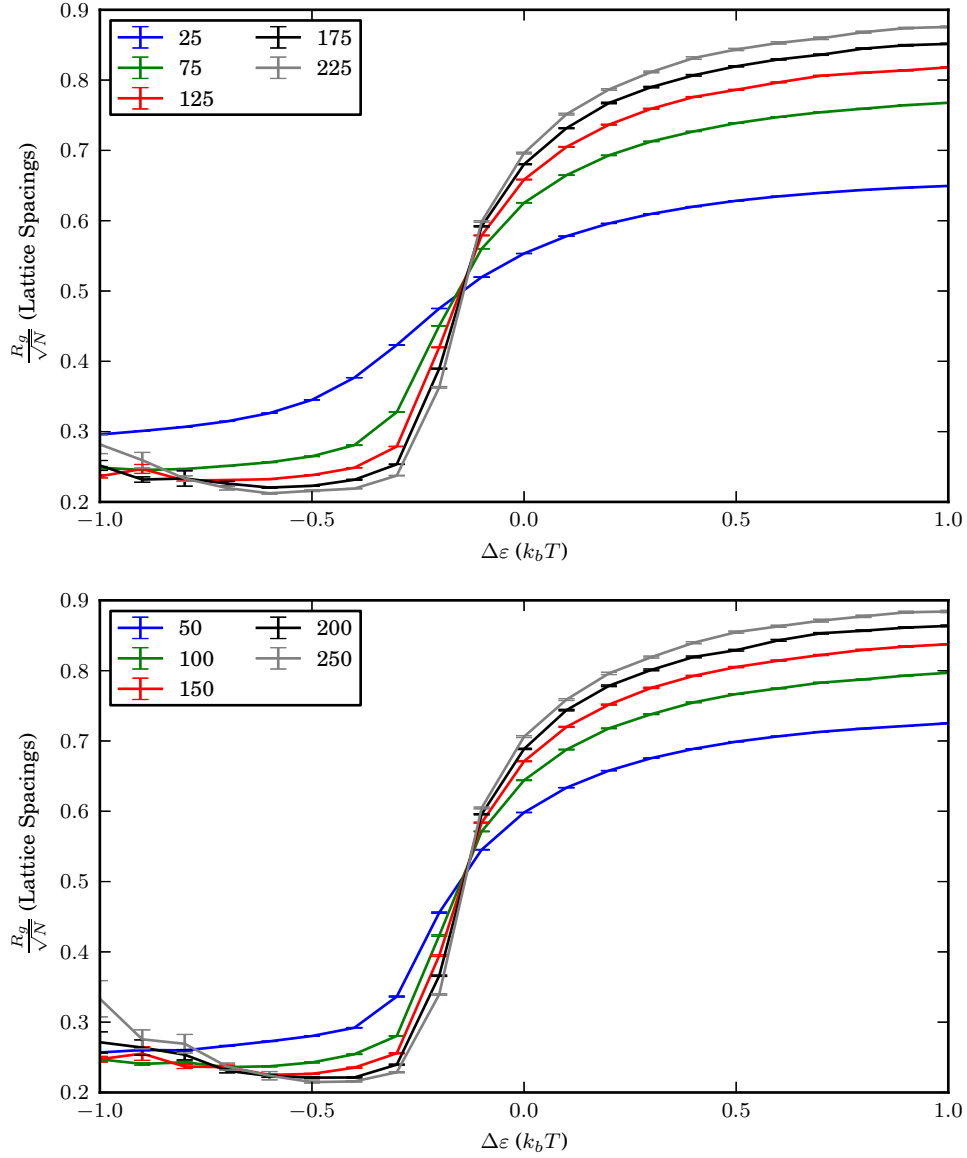
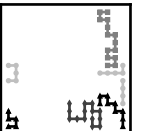


Figure 8.4: The behaviour of the radius of gyration with respect to solvent polymer interaction parameter $\Delta\epsilon$ for a range of polymer lengths. The data is split over two graphs for clarity. Each data point is an average over 30 simulations. We can estimate the Θ solvent interaction parameter for this mode from the cross over point.



polymers identically. Furthermore, the agreement between the simulations of single polymers and the pairs, shown in Fig. 8.5, is excellent ($\chi_{\text{red}} = 0.17$).

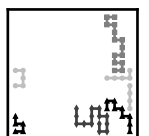
Following the successful tests on pairs of polymers, an investigation was conducted into the effect of density on the behaviour of the polymer chains. For this a polymer length of 100 was selected, a box of size $125 \times 125 \times 125$ was used and 2.5×10^8 MC iterations. Each simulation was repeated 25 times to ensure a consistent result. Simulations were conducted at concentrations $\rho \in \{10\%, 15\%, 20\%, 25\%, 30\%\}$ for $\Delta\epsilon \in [-0.5, 0.5]$. The radius of gyration as a function of $\Delta\epsilon$ is shown in Fig. 8.6. From this it is clear that the radius of gyration is strongly influenced by the density of the system, and that in *good* solvent conditions the presence of more polymers inhibits the ability of the polymers to spread out, whilst in *poor* solvent the polymers are able to avoid bunching up by grouping together, thus causing a larger radius of gyration. The absence of any discernible change in the cross over point for the four densities is in line with the predictions of Refs [199, 200], and its position to the left of the 0% density case is also expected. However, it should be noted that the 10–30% density cases were not directly sampled in their research due to the negligible variation in transition behaviour in this region.

The persistence length (l_p) of a polymer is defined in Eq. (7.1). For a polymer in this model the unit tangent vector is defined by

$$\hat{\tau}(i) = \mathbf{r}_i - \mathbf{r}_{i-1} \quad (8.8)$$

where \mathbf{r}_i is the position of the i^{th} site after the periodic boundary conditions have been unwrapped. An investigation was undertaken into the effect of density on the persistence length of the polymer. These simulations were identical to the calculations above except that only a single simulation at each density and $\Delta\epsilon$ value was made¹. This limitation does not significantly impact the results as the scatter in radius of gyration, shown by the error bars in Fig. 8.6, is typically small. Note that even the lowest density simulation contains 1953 polymers, allowing phase space to be efficiently explored. As can be seen in Fig. 8.7, the

¹The need to restrict the number of repeats occurs as the calculation takes 50 times longer to run and requires more than 5000 times the data storage space with this calculation included.



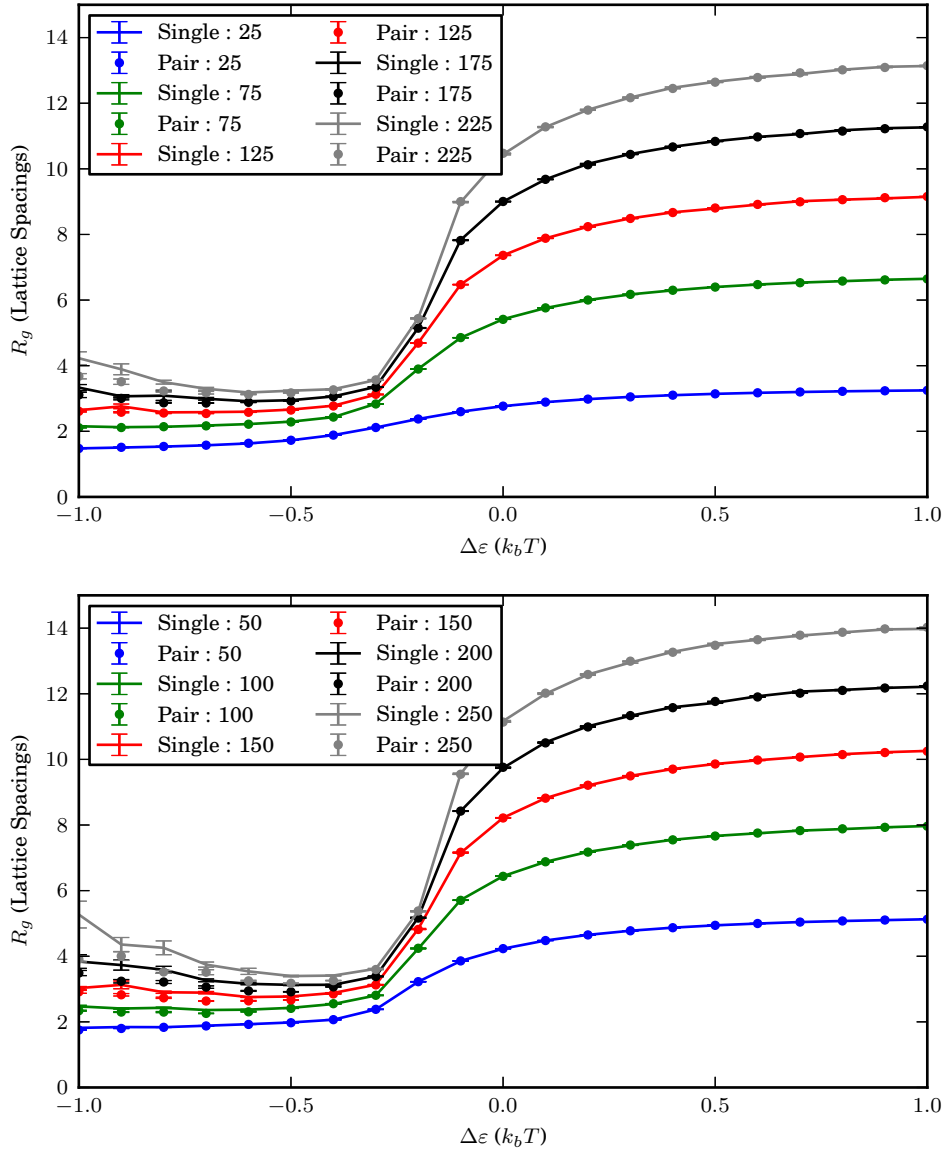
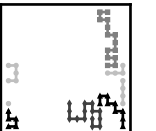


Figure 8.5: A comparison between single polymer chains (lines) and pairs of polymers (circles). As can be seen the behaviour of the radius of gyration with respect to solvent polymer interaction parameter $\Delta\epsilon$ for a range of polymer lengths is highly consistent between the two types of simulation. The data is split over two graphs for clarity. Each data point is an average over 30 simulations.



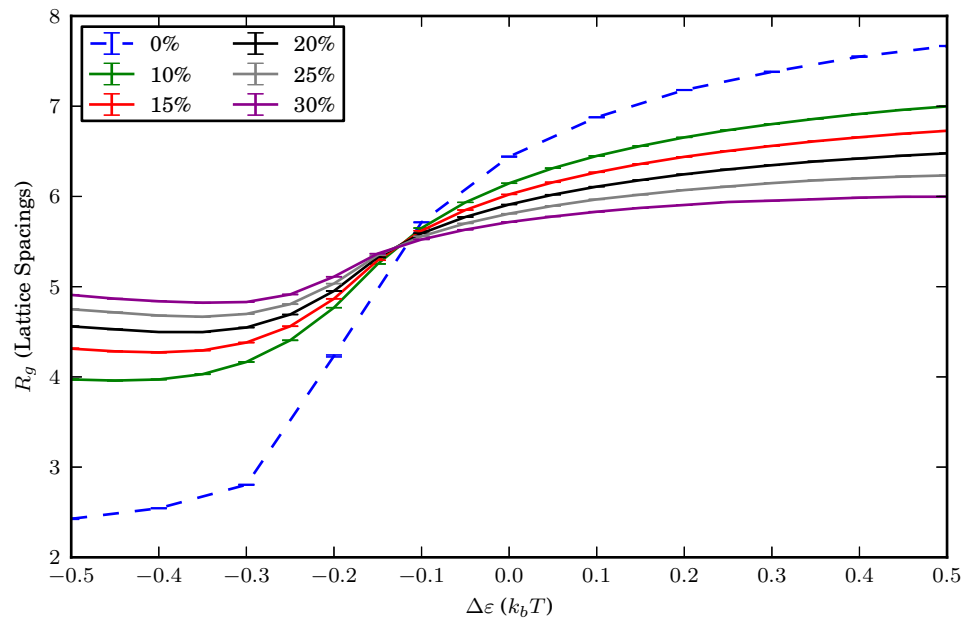


Figure 8.6: The behaviour of polymer chains of length 100 at different densities is compared. The data from previous simulations of single chains (Fig. 8.4) is used for the 0% density case. Each data point is an average over 25 simulations except for the 0% density case which used 30 simulations.



persistence length varies with the density of the polymer in a similar manner to radius of gyration. However, it is also apparent the persistence length of the polymer is seen to be somewhat under converged in the region of strongly negative $\Delta\epsilon$ (< -0.3). As running multiple repeats of each condition was not practical¹ two conditions were instead tested more thoroughly. A density of 30% with $\Delta\epsilon = -0.45$ and $\Delta\epsilon = -0.30$ were chosen as the highest density corresponds to the smallest number of moves per site and these values of $\Delta\epsilon$ represent interactions where the simulation appears poorly and well converged respectively. The scatter in values from 10 repeats at each of these conditions was used to generate the error bars shown in Fig. 8.7, these indicate that the behaviour above $\Delta\epsilon = -0.30$ is well captured by these simulations.

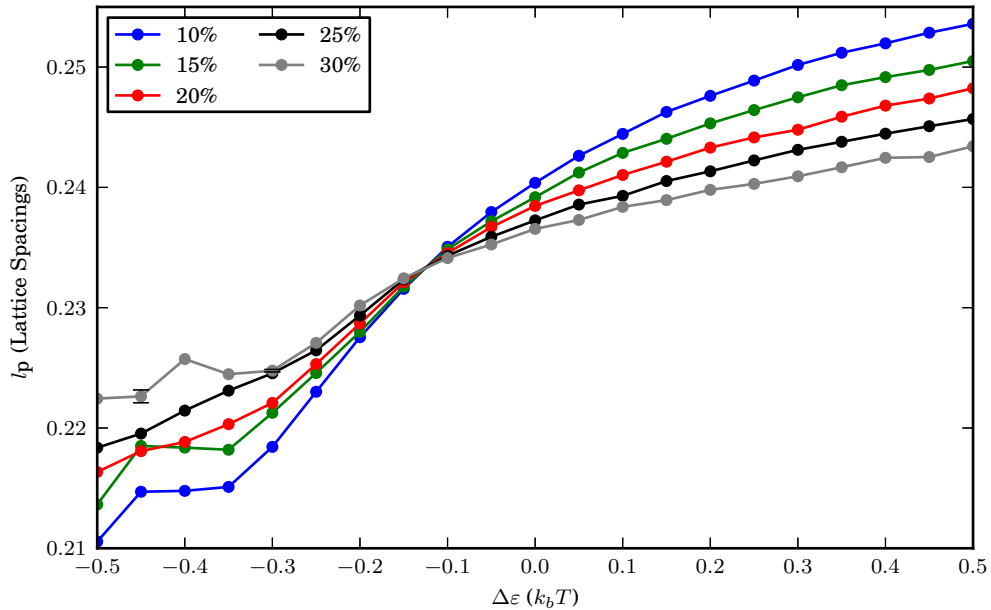


Figure 8.7: The persistence lengths of polymers at different densities for a range of solvent polymer interaction energies. The increase in the scatter for strongly negative values of $\Delta\epsilon$ is due to the difficulty of sampling this region of phase space with a reptation algorithm. The points at $\Delta\epsilon = -0.45$ and $\Delta\epsilon = -0.30$ on the 30% density curves have error bars created by averaging over 10 simulations for each data point

¹Even using file compression the raw data for Fig. 8.7 exceeds 50 GB



These tests clearly show that the model is performing as expected. However, they do not show whether the model demonstrates the behaviour seen in P84 membranes. Therefore, a series of simulations of long molecules were conducted to compare against the experiments of Stawikowska *et. al.* [6].

8.3.3 Dense Polymer Systems

In P84 membranes the ability of a membrane to separate organic molecules of different masses is strongly correlated with the pore size[6]. This is easily understood: if the molecule is larger than the pore it will be difficult for that molecule to pass through the pore. Therefore, we would like this model to show the same behaviour as the pores in a membrane, specifically a *good* solvent should have larger pores than a *poor* solvent. A series of simulations were conducted using simulation cells of dimension $512 \times 512 \times 512$, with polymers of length 470, at a density of 15% (42835 polymers), $\Delta\epsilon \in \{-0.5, 0.25, 0.0, 0.25, 0.5\}$ for 10^{10} MC steps¹. These were then used to study the behaviour of pore size with respect to $\Delta\epsilon$.

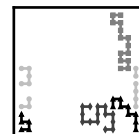
8.3.3.1 Estimation of Pore Size

The definition of a pore might appear intuitively obvious, namely an enclosed region without polymer. This definition is not suitable for use as a computer algorithm. Consider a sponge, such as you might use for household cleaning, which contains many thousands of small pores. Intuitively we can see the individual pores and could estimate their size; however, the pores themselves are interconnected, forming a complex network of connected pores that are not fully enclosed. Therefore, a simple algorithm which looks for a completely enclosed shape would consider all the connected pores as a single large pore.

A more sophisticated algorithm involves considering the largest sphere or ellipsoid which can fit into a region[202]. Another would consider narrow gaps² to be closed. However, at the molecular level these can have difficulty with situa-

¹This represents almost 497 moves per polymer site, or almost 233454 moves per polymer

²A general definition of narrow is difficult but it could imply a fixed number of lattice points wide.



tions where an isolated chain protrudes into the pore or the gap. A solution to this is to impose a larger grid on the lattice then identify whether the cells in the larger grid are occupied and use this to identify the pore size[203]. However, as the authors of Ref. [203] identify the method is sensitive to the chosen grid size, and the authors identify two methods for resolving this issue. The first approach, their “V-method,” assumes a constant pressure ensemble rather than the constant volume used in these simulations. The second approach described in Ref. [203], known as the “M-method,” is potentially suitable as it would correctly identify narrow passages as closed since since a grid cell over the passage would most likely contain polymer. However, this method for determining pore sizes was developed for identify nucleating bubbles of gas, whilst we are interested in the volume of a pore which would be seen by a molecule moving through a membrane. In a physical system, if the chain is free to move, then it could move out of the way of an incoming particle or molecule. This would mean that the pore size identified by a method such as that described by Ref. [203] might be too small. A simple solution might be to prune isolated chains from the system, however, if the surface has many of these or the chain has a large balled up structure at the end this would not be representative either. It is naturally possible to build a sophisticated model which can account for these effects. Instead this work considers how the STEM images of P84 membranes[6] were made and uses an analogous algorithm to allow for a fair comparison.

STEM, at a basic level, maps the electron density of a sample of the polymer membrane. The accessible pores are filled with OsO₂ nano-particles, which have a substantially higher electron density than P84. The resultant STEM images essentially provide a density map of the polymer membrane. Similar density maps can be obtained for the MC simulations using the following procedure. Configurations from the simulation were stored at fixed intervals. Sections of these configurations were created by considering all occupied lattice sites in a given region. This is necessary as using a complete simulation cell averages out too much detail. By picking a direction to act as the vertical and summing all the occupied lattice sites in each vertical column, a two-dimensional density map can be created. These density maps are then used to compare against the STEM images of P84 membranes[6].



8.3.3.2 Comparison with STEM Images

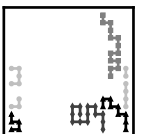
From the 5 MC simulations the initial (post equilibration), middle, and final state was stored. Subsets of these configurations were made of size $160 \times 160 \times 160$. There are many possible subsets which can be chosen, therefore, it was decided that for consistency to take the origin of the subset to be the origin of the lattice. It was further decided to compare only the final frames from each simulation as these frames have had the largest number of MC moves applied and should therefore be least susceptible to any bias in the initial structures.

As can be seen in Fig. 8.8 increasing $\Delta\epsilon$ causes the system to move towards a uniform low density system. These images can then be analysed by standard methods such as those used in ImageJ[204], to produce the pore size and distribution. Fig. 8.8 is reminiscent of the more porous open structures seen in the STEM images of P84 membranes[6], included for reference in Fig. 8.9. The similarity with the STEM images is particularly apparent in the $\Delta\epsilon = -0.50$ and $\Delta\epsilon = -0.25$ simulations, which match well against the 2:1, and 4:1 DMF:dioxane mass ratio membranes of Ref. [6]. However, the scale of these features, as indicated by the scale in Fig. 8.8, is not representative of P84 membranes: Fig. 8.7 shows that a lattice spacing is over $4l_p$, and based on the results in Sec. 7.5.3¹ this is around 4.4 nm. As a result an analysis of the pore sizes using ImageJ[204] was not conducted. Therefore, to capture the features of interest in this model the persistence length of the polymers in the MC model needs to be increased, relative to the lattice spacing.

8.3.4 Directional Biasing

There are two primary length scales involved in this MC model: the length scale of the polymer, which we have characterised by l_p , and the inter-molecular separation, which is governed by the lattice spacing. For a given density and $\Delta\epsilon$ the persistence length of the model as described above is a fixed number of lattice spacings. Therefore, another parameter is required to increase the model polymers stiffness and hence its l_p relative to the lattice spacing.

¹It should be stressed that the values reported in Sec. 7.5.3 do not include the effects of multiple polymers which clearly influences l_p .



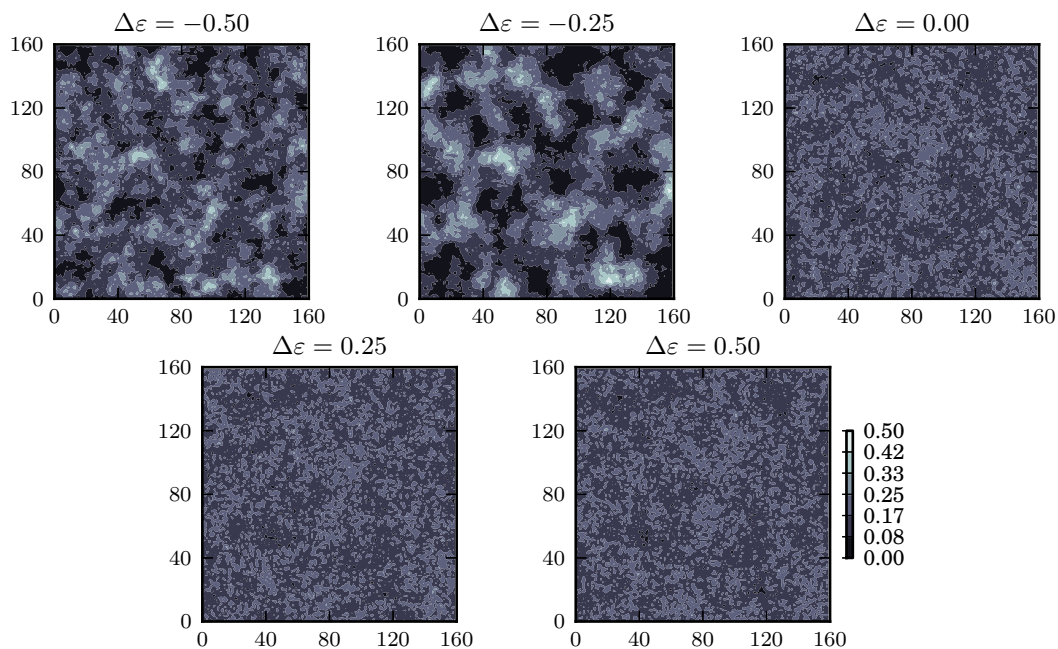


Figure 8.8: Projections through $160 \times 160 \times 160$ subsets of MC simulations. The value of $\Delta\epsilon$ is specified above every image. There is a clear trend that as $\Delta\epsilon$ increases the system tends towards a uniform relatively low density structure

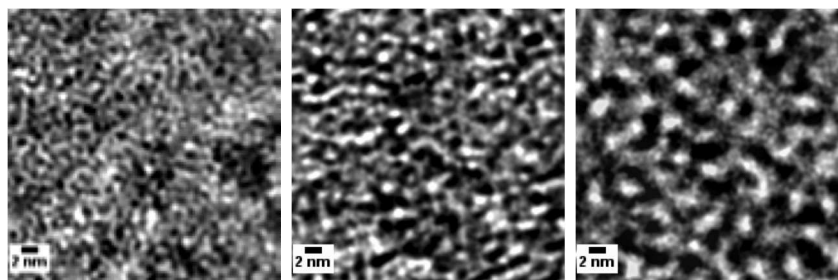


Figure 8.9: STEM images P84 membranes from Ref. [6]. The images are ordered with increasing solvent quality. The membranes were prepared using respectively a 1:2, 2:1, and 4:1 DMF:dioxane mass ratio. For full details of the experiments see Ref. [6]. Reprinted from Journal of Membrane Science, 413–414, Joanna Stawikowska and Andrew G. Livingston, Nanoprobe imaging molecular scale pores in polymeric membrane, 1–6, 2012, with permission from Elsevier, license number 3393131159500.



Increasing l_p can be accomplished by increasing the number of segments which are in line with one another. To accomplish this increase in linearity, a directional biasing scheme was developed. The directional biasing scheme modifies the probability of selecting a move which is in line with the previous segment. In the unbiased scheme each of the possible move sites has probability $\frac{1}{z-1}$ of being selected, for the three dimensional cubic lattice, the lattice coordination number $z = 6$, and the -1 accounts for the fact that moves which fold back on themselves are forbidden. In contrast, the bias scheme assigns an in line direction the probability w with the remaining, out of line, move sites having probability $\frac{1-w}{z-2}$. By selecting $w > \frac{1}{z-1} = 0.2$ we can increase the number of segments which lie in line with the previous segments, thereby increasing the polymer's persistence length. This scheme was tested for $w \in \{0.1, 0.2, \dots, 0.6\}$, with $\Delta\varepsilon \in [-0.5, 0.5]$, all simulations used $\rho = 15\%$, $N = 100$ and 2.5×10^8 MC steps. The resulting persistence lengths for these simulations are shown in Fig. 8.10, from this it is clear that persistence length increases with w . However, it is also clear that increasing w from 0.2 to 0.6 only causes an increase of $< 20\%^1$. Taking the inter-molecular spacing of benzene (0.34 nm), as a rough approximation for the intermolecular separation of P84 and hence the lattice parameter, $l_p \approx 3$ lattice spacings is required to match the MD simulations. Obtaining $l_p \approx 3$ requires increasing l_p by a factor of 11–15, which is clearly not practical with this form of directional biasing.

This directional biasing scheme fails to achieve the desired increase in persistence length as a result of two factors: the underlying nature of the lattice itself, which results in very few possible in line moves, and the logarithmic sensitivity of the persistence length to changes in the correlation between segments. To explain why these two effects make schemes such as this unfeasible consider a non-self avoiding random walk on a cubic lattice where the only restriction is that the next segment of the polymer cannot directly overlie the previous segment. If the probability of placing a segment in line with the previous segment is given by w as in the above model, then the correlation between one segment and the next

¹ The largest observed increase was 17%, at $\Delta\varepsilon = -0.1$.



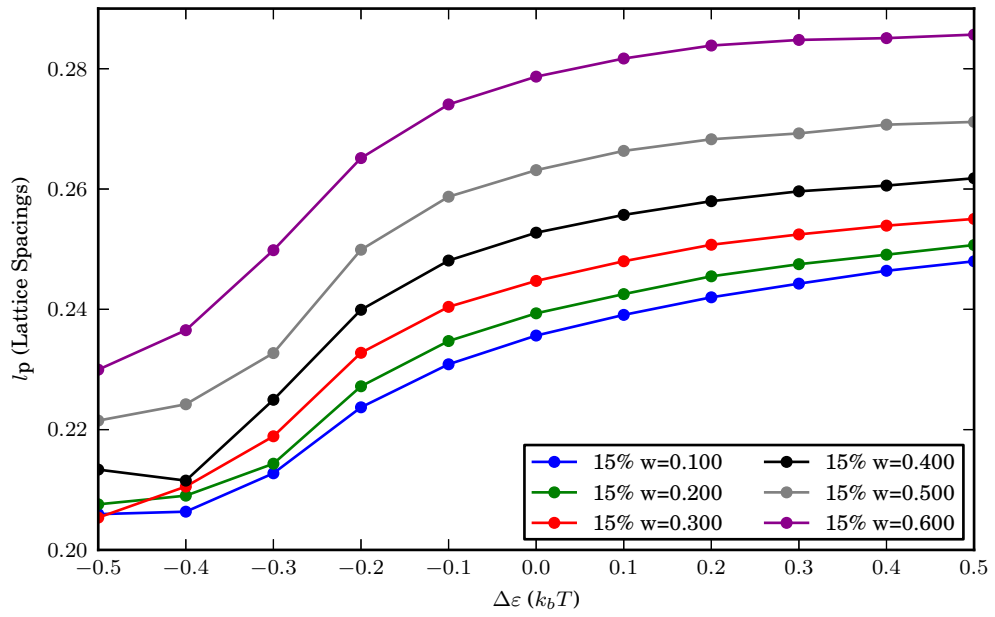


Figure 8.10: Persistence length against $\Delta\epsilon$ for a range of directional bias parameters w . The lattice coordination number for this system is 6 therefore the number of possible move sites is 5 making $w = 0.2$ unbiased. It is clear from these simulations that increasing the bias in favour of making the polymer straight increases l_p . However, it is also clear that an extremely large bias would be required to obtain large values of the l_p .



segment will be given by w . Let us define

$$w_0 = \frac{1}{z-1} \stackrel{z=6}{=} 0.2, \quad (8.9)$$

$$l_{p_0} = -\frac{1}{\ln(w_0)}, \quad (8.10)$$

$$l_p = al_{p_0}, \quad (8.11)$$

for some target value of a . Using

$$w = e^{-\frac{1}{l_p}}, \quad (8.12)$$

it is clear that,

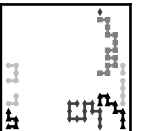
$$w = w_0^{\frac{1}{a}}. \quad (8.13)$$

Since $l_{p_0} \approx 0.2$ lattice spacings, then if we want $l_p = 4$ lattice spacings, we require $a = 20$. Therefore, $w \approx 0.9$ which is not practical to sample. Whilst this analysis does not include the effects of avoiding other polymer sites, the effect of more distant polymer segments, or the effect of $\Delta\varepsilon$, amongst many other effects, it does demonstrate the difficulty in this sort of biasing scheme. It can be shown in a similar manner that changing the lattice type yields:

$$\alpha = \alpha_0^{\frac{1}{a}}. \quad (8.14)$$

where $\alpha = bw$ and b is a lattice dependent constant which does not significantly influence the final result. This type of simple biasing is, therefore, insufficient to allow the lattice model to capture both the polymers intrinsic length scale and the inter polymer separations for a rigid molecule such as P84.

Other groups have had more success by introducing the bias into the energy landscape and using more advanced sampling techniques. Hsu *et. al.* Ref. [205] imposed a substantial energy penalty on turning the polymer, $0.9\text{--}5.3 k_bT$, and used a far longer polymer, $N=50000$. This resulted in significant changes in the persistence length of the polymers up to an order of magnitude. Unfortunately these levels of bias towards straight chains are beyond the capabilities of the reptation based sampling scheme used in this work, and the tests conducted by



Hsu *et. al.* used only a single chain rather than a dense collection of chains which makes the system easier to sample[201].

Future work in this area should be focused on developing a direct link between the P84 molecule and the MC simulation, some possible approaches which have been considered are presented here. Using more advanced sampling techniques such as PERM have the potential to allow the necessary levels of directional biasing to be achieved[205]. Having a number of segments defined to lie in a straight line can, depending on the exact implementation, result in polymers moving on separate sub-lattices. Another scheme would involve assessing the straightness of several polymer segments together and using them to define the bias on the next segment's placement, or even using the average direction of several segments together to assess the polymers tangent direction thus smoothing out the sharp corners of the lattice. However, the most likely scheme to succeed, is to use the large scale structures obtained from this model as starting points for more detailed studies of the polymers at an atomistic or near atomistic scale. This would allow unfavourable configurations to be screened out efficiently in the MC model and the most likely structures to be sampled in detail to understand the role of chemical interactions as well as geometrical constraints in the filtration process. Even without these improvements MC simulations have matched the trends seen in experiments with remarkable accuracy. From these simulations, particularly those of Sec. 8.3.3, it is clear that the trends in the structure of a polymer in solution with respect to changing solvent quality matches the variation in structure of the surface layers of P84 membranes. This shows that the structure and porosity of a P84 membrane is determined by its behaviour in the initial solution.



Chapter 9

Conclusions

This research has resulted in a molecular model for the formation of polymer membranes by phase inversion. By considering the molecular interactions between P84 and its solvents it has been possible to replicate the trends seen in experiments in particular that pore size increases with solvent quality. Furthermore, whilst developing tools to couple DFT and MD a new approach to the coarse graining of aromatic polymers has been developed. However, this does not represent an end to this research but rather a beginning.

The collaborations started with this project have already led to further research into desalination membranes produced by interfacial polymerization, as well as elastomer seals operating in extreme environments, that will doubtless yield much valuable research. There are additional avenues available to directly follow on from this research. I have selected a few directions which I believe will result in interesting and useful contributions to the field. The remainder of this chapter is structured as follows. In Sec. 9.1 I summarise the coarse graining work in Chap. 4. The key aspects of the multi-scale modelling of polymer membranes from Chap. 6, Chap. 7, and Chap. 8 are summarised in Sec. 9.2. Suggestions for future research are discussed in Sec. 9.3.



9.1 Coarse Graining

The new coarse graining scheme presented in Chap. 4, shows excellent consistency with traditional all atom approaches. By coupling chemical insight with knowledge of geometry and mechanics, it ensures that the dynamic and structural properties of aromatic polymers are preserved. The removal of fast degrees of freedom allows accurate integration at large time steps, and the reduction in the number of constraints improves the stability of the solution.

Coarse graining PEEK demonstrates that this procedure can be applied to industrially relevant molecules. The combination of this technique with force fitting has the potential to allow the generation of custom forcefields where high accuracy is required. Furthermore, the fact that it can be used with an unmodified version of GROMACS removes the need for code development by groups wishing to use the technique making it more accessible. The primary limitations of this coarse graining technique are that the SHAKE algorithm, which is used to implement the constraints, is not currently parallelised within GROMACS. Despite this the technique has great potential as it increases the time scales accessible with atomistic precision which will be of use to researchers both in the field of membranes and in the study of polymers more generally.

9.2 Modeling P84 Membranes

The simulations developed to model P84 membranes demonstrate that the initial structure of P84 in solution follows the trends seen in P84 OSN membranes. By showing that processes such as evaporation are not necessary to produce a phase inversion membrane, and that the interaction between the solvents and the polymer, which can be modelled atomistically, are important in the development of pores in the membrane, this model has improved our understanding of the phase inversion mechanism. The MC simulations provide a means of rapidly generating structures to allow further study, while the MD simulations provide the length scales associated with P84 in solution. There remains work to do to allow MD to be used to refine the structures developed in MC. However, it has been demonstrated that the solvent polymer interaction in the initial *dope* obeys



the same trends as the filtration layer of a P84 phase inversion membrane.

9.3 Further Work

The primary limitations of the coarse graining strategy developed in Chap. 4 are those of the SHAKE algorithm; Sec. 4.1.3 describes the problem and proposes a new hybrid methodology for addressing these issues. The approach is to combine several existing methods to create a more stable and potentially faster constraint algorithm. This approach could potentially be used in a variety of problems as highly coupled constraints arise in many contexts both in polymers and in fields such as robotics.

Coupling DFT to MD through the iterative matching of observables is not common practice. Whilst codes do exist that will take the data from DFT and MD simulations and combine them, these programs often use custom MD codes[175] or use a modified version of the source code of a commercial program[172, 206]. However, programs such as VOTKA[207] have demonstrated the usefulness of external programs which can interface with the latest versions of commercial codes. Developing an external tool kit to enable force fitting using existing codes would be a useful task. Accomplishing this would complete both directions of the coupling between DFT and MD. The code I developed during my work on coarse graining, see Sec. 4.1.5, could serve as a useful starting point for this work. However, tools such as VOTKA provide interfaces to more MD programs, are more fully developed and documented than my code, providing a sensible approach to merge the ideas from my code into a more developed project. Whilst some level of coding knowledge for this project is essential, the more important aspect is a good understanding of DFT and MD as fitting a forcefield is a very challenging undertaking [54, 172, 175, 176, 206].

There are many topics and ideas about membranes that have come out of this three year project; selecting and outlining self-contained projects is difficult. Perhaps the most useful area to follow directly from this research would be the development of a process for coupling MD to MC and back. Removing information by moving from atomistic simulation to CG MC representations is often far easier than restoring the atomistic information from the CG MC. Once a means



of doing this has been developed the initial structures can be generated efficiently using MC then refined in MD. The resulting refined structures can then be used to address many other problems.

Given a process for generating refined membrane structures such as the one described above, the most useful area to target for research is probably adhesion and fouling. Cleaning membrane systems often accounts for a high proportion of energy use in filtration systems[2]. Therefore, understanding how and why membranes become fouled may lead to better overall efficiency than focussing primarily on topics such as flux, rejection, or longevity. Researching this will require the use of atomistic and quantum techniques, and most probably fluid mechanics as well as an understanding of chemical interactions at both inter- and intra-molecular levels. Research into the flow through interfacially polymerised desalination membranes at Imperial College London has already begun and this will certainly lead to progress in this field.

Many other possible research topics have come out of this work, and those outlined above I believe will produce useful contributions to this field and beyond. The study of industrial problems can introduce issues that idealised models do not present; however, by addressing these issues we can gain new insights which would otherwise have gone unnoticed. An example of this is the coarse graining strategy discussed in Chap. 4 which would not have come about were it not for the need to reduce the degrees of freedom in the simulation of industrial polymers such as PEEK and P84 to enable longer simulations and the efficient development of custom forcefields.



References

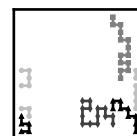
- [1] S. Adler, E. Beaver, P. Bryan, S. Robinson, and J. Watson. *Vision 2020: 2000 Separations Roadmap*. The Center for Waste Reduction Technologies of the AIChE in cooperation with the US Dept. of Energy, Office of Industrial Technologies, 2000.
- [2] Menachem Elimelech and William A. Phillip. The future of seawater desalination: Energy, technology, and the environment. *Science*, 333(6043): 712–717, 2011.
- [3] S. Loeb. *The Loeb-Sourirajan membrane: How it came about*, chapter 2, pages 1–9. ACS Publications, 1981.
- [4] Vladimir Neburchilov, Jonathan Martin, Haijiang Wang, and Jiujun Zhang. A review of polymer electrolyte membranes for direct methanol fuel cells. *J. Power Sources*, 169(2):221–238, 2007.
- [5] Mohammad Soltanieh and William N. Gill. Review of reverse osmosis membranes and transport models. *Chem. Eng. Commun.*, 12(4–6):279–363, 1981.
- [6] Joanna Stawikowska and Andrew G. Livingston. Nanoprobe imaging molecular scale pores in polymeric membranes. *J. membrane Sci.*, 413–414(0): 1–16, 2012.
- [7] Iwona Soroko, Mafalda Pessoa Lopes, and Andrew Livingston. The effect of membrane formation parameters on performance of polyimide membranes for organic solvent nanofiltration (OSN): Part A: Effect of



- polymer/solvent/non-solvent system choice. *J. membrane Sci.*, 381(1–2): 152–162, 2011.
- [8] Iwona Soroko, Marcin Makowski, Fabian Spill, and Andrew Livingston. The effect of membrane formation parameters on performance of polyimide membranes for organic solvent nanofiltration (OSN): Part B: Analysis of evaporation step and the role of a co-solvent. *J. membrane Sci.*, 381(1–2): 163–171, 2011.
- [9] Iwona Soroko, Malladi Sairam, and Andrew Livingston. The effect of membrane formation parameters on performance of polyimide membranes for organic solvent nanofiltration (OSN): Part C. effect of polyimide characteristics. *J. membrane Sci.*, 381(1–2):172–182, 2011.
- [10] Joanna Stawikowska and Andrew G. Livingston. Assessment of atomic force microscopy for characterisation of nanofiltration membranes. *J. membrane Sci.*, 425–426(0):58–70, 2013.
- [11] Detlev Fritsch, Petra Merten, Kathleen Heinrich, Marina Lazar, and Markus Priske. High performance organic solvent nanofiltration membranes: Development and thorough testing of thin film composite membranes made of polymers of intrinsic microporosity (PIMs). *J. membrane Sci.*, 401–402(0):222–231, 2012.
- [12] Maria Fernanda Jimenez Solomon, Yogesh Bhole, and Andrew Guy Livingston. High flux membranes for organic solvent nanofiltration (OSN)-interfacial polymerization with solvent activation. *J. membrane Sci.*, 423–424(0):371–382, 2012.
- [13] Peter D. Chapman, Teresa Oliveira, Andrew G. Livingston, and K. Li. Membranes for the dehydration of solvents by pervaporation. *J. membrane Sci.*, 318(1–2):5–37, 2008.
- [14] Samhun Yun and S. Ted Oyama. Correlations in palladium membranes for hydrogen separation: A review. *J. membrane Sci.*, 375(1–2):28–45, 2011.



- [15] Maryam Takht Ravanchi, Tahereh Kaghazchi, and Ali Kargari. Application of membrane separation processes in petrochemical industry: A review. *Desalination*, 235(1–3):199–244, 2009.
- [16] Richard J. Broadbent, James S. Spencer, Arash A. Mostofi, and Adrian P. Sutton. Accelerated simulations of aromatic polymers: application to polyether ether ketone (PEEK). *Mol. Phys.*, 112(20):2633–2645, 2014.
- [17] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of state calculations by fast computing machines. *J. Chem. Phys.*, 21(6):1087–1092, 1953.
- [18] Lloyd L. Schulman and Christopher G. DesAutels. *Computational fluid dynamics simulations to predict wind-induced damage to a steel building during hurricane Katrina*, chapter 84, pages 793–800. American Society of Civil Engineers, 2012.
- [19] Sunil K. Sinha. Rotordynamic analysis of asymmetric turbofan rotor due to fan blade-loss event with contact-impact rub loads. *J. Sound. Vib.*, 332(9):2253–2283, 2013.
- [20] Jiří Klimeš and Angelos Michaelides. Perspective: Advances and challenges in treating van der Waals dispersion forces in density functional theory. *J. Chem. Phys.*, 137:120901, 2012.
- [21] R. Kilburn, A. Gregory, and A. G. Murray. Using a markov-chain monte-carlo modelling approach to identify the relative risk to farmed scottish rainbow trout (*oncorhynchus mykiss*) in a multi-sector industry of viral haemorrhagic septicaemia viruses from introduction and emergent sources. *Ecol. Model.*, 237–238(0):34–42, 2012.
- [22] Patrick J. Golden, Harry R. Millwater, and Xiaobin Yang. Probabilistic fretting fatigue life prediction of Ti-6Al-4V. *Int. J. Fatigue*, 32(12):1937–1947, 2010.



-
- [23] J. S. Spencer, N. S. Blunt, and W. M. C. Foulkes. The sign problem and population dynamics in the full configuration interaction quantum Monte Carlo method. *J. Chem. Phys.*, 136(5):054110, 2012.
- [24] W. K. Hastings. Monte Carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.
- [25] Daan Frenkel and Berend Smit. *Understanding Mol. Simulat.* Academic Press, Inc., Orlando, FL, U.S.A., 2nd edition, 2001. ISBN 0122673514.
- [26] M. P. Allen and D. J. Tildesley. *Computer Simulation of Liquids*. Oxford Science Publications, 1987.
- [27] Fugao Wang and D. P. Landau. Efficient, multiple-range random walk algorithm to calculate the density of states. *Phys. Rev. Lett.*, 86(10):2050–2053, Mar 2001.
- [28] John Skilling. Nested sampling. *AIP Conference Proceedings*, 735(1), 2004.
- [29] Makoto Matsumoto and Takuji Nishimura. Mersenne twister: A 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM T. Model. Comput. S.*, 8(1):3–30, 1998.
- [30] François Panneton, Pierre L’ecuyer, and Makoto Matsumoto. Improved long-period generators based on linear recurrences modulo 2. *ACM T. Math. Software*, 32(1):1–16, 2006.
- [31] Mutsuo Saito and Makoto Matsumoto. A PRNG specialized in double precision floating point numbers using an affine transition. In *Monte Carlo and Quasi-Monte Carlo Methods 2008*, pages 589–602. Springer, 2009.
- [32] George Marsaglia. Random numbers fall mainly in the planes. *P. Natl. Acad. Sci. U.S.A.*, 61(1):25, 1968.
- [33] Marshall N. Rosenbluth and Arianna W. Rosenbluth. Monte Carlo calculation of the average extension of molecular chains. *J. Chem. Phys.*, 23(2):356–359, 1955.



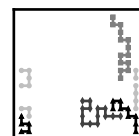
- [34] Jonathan Harris and Stuart A. Rice. A lattice model of a supported monolayer of amphiphile molecules: Monte Carlo simulations. *J. Chem. Phys.*, 88(2):1298–1306, 1988.
- [35] Hsiao-Ping Hsu, Wolfgang Paul, and Kurt Binder. Standard definitions of persistence length do not describe the local “intrinsic” stiffness of real polymer chains. *Macromolecules*, 43(6):3094–3102, March 2010.
- [36] G. Rychlewski and S. G. Whittington. Self-avoiding walks and polymer adsorption: Low temperature behaviour. *J. Stat. Phys.*, 145(3):661–668, 2011.
- [37] Jiörn Ilja Siepmann and Daan Frenkel. Configurational bias Monte Carlo: A new sampling scheme for flexible chains. *Mol. Phys.*, 75(1):59–70, 1992.
- [38] George A. Kaminski, Richard A. Friesner, Julian Tirado-Rives, and William L. Jorgensen. Evaluation and reparametrization of the OPLS-AA force field for proteins via comparison with accurate quantum chemical calculations on peptides. *J. Phys. Chem. B*, 105(28):6474–6487, 2001.
- [39] Jannis Batoulis and Kurt Kremer. Statistical properties of biased sampling methods for long polymer chains. *J. Phys. A-Math. Gen.*, 21(1):127–146, 1988.
- [40] D. Frenkel, G. C. A. M. Mooij, and B. Smit. Novel scheme to study structural and thermal properties of continuously deformable molecules. *J. Phys.-Condens. Mat.*, 4(12):3053, 1992.
- [41] D. Györfy, P. Zavodsky, and A. Szilágyi. “pull moves” for rectangular lattice polymer models are not fully reversible. *Comput. Biol. Bioinform., IEEE/ACM Trans.*, 9(6):1847–1849, 2012.
- [42] Peter H. Verdier and W. H. Stockmayer. Monte Carlo calculations on the dynamics of polymers in dilute solution. *J. Chem. Phys.*, 36(1):227–235, 1962.
- [43] Peter H. Verdier. Monte Carlo studies of lattice-model polymer chains. II. end-to-end length. *J. Chem. Phys.*, 45(6):2122–2128, 1966.



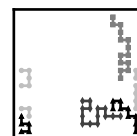
-
- [44] H. J. Hilhorst and J. M. Deutch. Analysis of Monte Carlo results on the kinetics of lattice polymer chains with excluded volume. *J. Chem. Phys.*, 63(12):5153–5161, 1975.
- [45] Neal Madras and Alan D. Sokal. The pivot algorithm: A highly efficient Monte Carlo method for the self-avoiding walk. *J. Stat. Phys.*, 50(1–2):109–186, 1988.
- [46] A. K. Kron. The Monte Carlo method in statistical calculations of macromolecules. *Polym. Sci. U.S.S.R.*, 7(7):1361–1367, 1965.
- [47] Frederick T. Wall and Frederic Mandel. Macromolecular dimensions obtained by an efficient Monte Carlo method without sample attrition. *J. Chem. Phys.*, 63(11):4592–4595, 1975.
- [48] Sylvie Neyertz. Tutorial: Molecular dynamics simulations of microstructure and transport phenomena in glassy polymers. *Soft. Mater.*, 4(1):15–83, 2007.
- [49] Xinchao Zhao. Advances on protein folding simulations based on the lattice HP models with natural computing. *Appl. Soft Comput.*, 8(2):1029–1040, 2008.
- [50] A.A. Albrecht, A. Skaliotis, and K. Steinhöfel. Stochastic protein folding simulation in the three-dimensional HP-model. *Comput. Biol. Chem.*, 32(4):248–255, 2008.
- [51] Xiangqian Hu, David N. Beratan, and Weitao Yang. A gradient-directed Monte Carlo method for global optimization in a discrete space: Application to protein sequence design and folding. *J. Chem. Phys.*, 131(15):154117, 2009.
- [52] Masao Doi. *Introduction to polymer physics*. Oxford Science Publications. Clarendon Press, Oxford, 2001.
- [53] Jindal K. Shah, Joan F. Brennecke, and Edward J. Maginn. Thermodynamic properties of the ionic liquid 1-n-butyl-3-methylimidazolium hex-



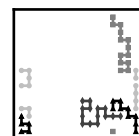
- affluorophosphate from Monte Carlo simulations. *Green Chem.*, 4:112–118, 2002.
- [54] Paul Tangney and Sandro Scandolo. Melting slope of MgO from molecular dynamics and density functional theory. *J. Chem. Phys.*, 131(12):124510, 2009.
- [55] John L. Klepeis, Kresten Lindorff-Larsen, Ron O. Dror, and David E. Shaw. Long-timescale molecular dynamics simulations of protein structure and function. *Curr. Opin. Struc. Biol.*, 19(2):120–127, 2009.
- [56] Hans C. Andersen. Molecular dynamics simulations at constant pressure and/or temperature. *J. Chem. Phys.*, 72(4):2384–2393, 1980.
- [57] H. J. C. Berendsen, J. P. M. Postma, W. F. van Gunsteren, A. DiNola, and J. R. Haak. Molecular dynamics with coupling to an external bath. *J. Chem. Phys.*, 81(8):3684–3690, 1984.
- [58] Giovanni Bussi, Davide Donadio, and Michele Parrinello. Canonical sampling through velocity rescaling. *J. Chem. Phys.*, 126(1):014101, 2007.
- [59] Shūichi Nosé. A molecular dynamics method for simulations in the canonical ensemble. *Mol. Phys.*, 52(2):255–268, 1984.
- [60] William G. Hoover. Canonical dynamics: Equilibrium phase-space distributions. *Phys. Rev. A*, 31:1695–1697, Mar 1985.
- [61] W. F. Van Gunsteren and H. J. C. Berendsen. A leap-frog algorithm for stochastic dynamics. *Mol. Simulat.*, 1(3):173–185, 1988.
- [62] Stephen L. Mayo, Barry D. Olafson, and William A. Goddard. DREIDING: A generic force field for molecular simulations. *J. Phys. Chem.*, 94(26):8897–8909, 1990.
- [63] Junmei Wang, Romain M. Wolf, James W. Caldwell, Peter A. Kollman, and David A. Case. Development and testing of a general AMBER force field. *J. Comput. Chem.*, 25(9):1157–1174, July 2004.



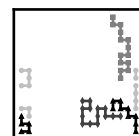
- [64] Alexander D. Mackerell, Michael Feig, and Charles L. Brooks. Extending the treatment of backbone energetics in protein force fields: Limitations of gas-phase quantum mechanics in reproducing protein conformational distributions in molecular dynamics simulations. *J. Comput. Chem.*, 25(11):1400–1415, 2004.
- [65] Lukas D. Schuler, Xavier Daura, and Wilfred F. van Gunsteren. An improved gromos96 force field for aliphatic hydrocarbons in the condensed phase. *J. Comput. Chem.*, 22(11):1205–1218, 2001.
- [66] H. Sun. Compass: An ab initio force-field optimized for condensed-phase applications overview with details on alkane and benzene compounds. *J. Phys. Chem. B*, 102(38):7338–7364, 1998.
- [67] J. S. Bell. On the Einstein Podolsky Rosen paradox. *Physics*, 1(3):195–200, 1964.
- [68] Richard H. Boyd, Richard H. Gee, Jie Han, and Yong Jin. Conformational dynamics in bulk polyethylene: A molecular dynamics simulation study. *J. Chem. Phys.*, 101(1):788–797, 1994.
- [69] Jie Han and Richard H. Boyd. Molecular packing and small-penetrant diffusion in polystyrene: A molecular dynamics simulation study. *Polymer*, 37(10):1797–1804, 1996.
- [70] Aaron M. Ring, Jian-Xin Lin, Dan Feng, Suman Mitra, Mathias Rickert, Gregory R. Bowman, Vijay S. Pande, Peng Li, Ignacio Moraga, Rosanne Spolski, Engin Özkan, Warren J Leonard, and K Christopher Garcia. Mechanistic and structural insight into the functional dichotomy between IL-2 and IL-15. *Nat. Immunol.*, 13:1187–1195, 2012.
- [71] Berthold Heymann and Helmut Grubmüller. Elastic properties of poly(ethylene-glycol) studied by molecular dynamics stretching simulations. *Chem. Phys. Lett.*, 307(5–6):425–432, 1999.



- [72] Peter L. Freddolino, Feng Liu, Martin Gruebele, and Klaus Schulten. Ten-microsecond molecular dynamics simulation of a fast-folding WW domain. *Biophys. J.*, 94(10):L75–L77, May 2008.
- [73] Li Xi-Han, Li Chun-Yan, Deng Fu-Guo, Zhou Ping, Liang Yu-Jie, and Zhou Hong-Yu. Quantum secure direct communication with quantum encryption based on pure entangled states. *Chinese Phys.*, 16(8):2149, 2007.
- [74] Maya Mizuno and Kaori Fukunaga. Analysis of tissue condition based on interaction between inorganic and organic matter in cuttlefish bone. *J. Biol. Phys.*, 39(1):123–130, 2013.
- [75] F. Körmann, A. Dick, B. Grabowski, B. Hallstedt, T. Hickel, and J. Neugebauer. Free energy of bcc iron: Integrated *ab initio* derivation of vibrational, electronic, and magnetic contributions. *Phys. Rev. B*, 78:033102, Jul 2008.
- [76] Fabiano Corsetti and Arash A Mostofi. System-size convergence of point defect properties: The case of the silicon vacancy. *Phys. Rev. B*, 84(3):035209, 2011.
- [77] Richard M. Martin. *Electronic Structure Basic Theory and Practical Methods*. Cambridge University Press, Cambridge, UK, 2004.
- [78] Adrian P. Sutton. *Electronic Structure of Materials*. Oxford University Press, Oxford, UK, 1993.
- [79] Max Born and Robert Oppenheimer. Zur quantentheorie der molekeln. *Ann. Phys.-Berlin*, 389(20):457–484, 1927.
- [80] Xin-Zheng Li, Brent Walker, and Angelos Michaelides. Quantum nature of the hydrogen bond. *P. Natl. A. Sci. U.S.A.*, 108(16):6369–6373, 2011.
- [81] Rex Godby. The GW approach to spectral, ground-state and transport properties of electrons in matter. TYC Soiree on the GW Approach, 2011. URL <http://www.thomasyoungcentre.org/events/a-soiree-on-the-gw-approach/>.



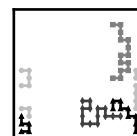
-
- [82] Brian M. Austin, Dmitry Yu Zubarev, and William A. Lester Jr. Quantum Monte Carlo and related approaches. *Chem. Rev.*, 112(1):263–288, 2011.
- [83] P. Hohenberg and W. Kohn. Inhomogeneous electron gas. *Phys. Rev.*, 136(3B):B864–B871, 1964.
- [84] Mel Levy. Universal variational functionals of electron densities, first-order density matrices, and natural spin-orbitals and solution of the v -representability problem. *P. Natl. A. Sci. U.S.A.*, 76(12):6062–6065, 1979.
- [85] W. Kohn and L. J. Sham. Self-consistent equations including exchange and correlation effects. *Phys. Rev.*, 140(4A):A1133–A1138, 1965.
- [86] Philipp Haas, Fabien Tran, and Peter Blaha. Calculation of the lattice constant of solids with semilocal functionals. *Phys. Rev. B*, 79(8):085104, 2009.
- [87] D. M. Ceperley and B. J. Alder. Ground state of the electron gas by a stochastic method. *Phys. Rev. Lett.*, 45:566–569, Aug 1980.
- [88] John P. Perdew, Kieron Burke, and Matthias Ernzerhof. Generalized gradient approximation made simple. *Phys. Rev. Lett.*, 77:3865–3868, Oct 1996.
- [89] A. D. Becke. Density-functional exchange-energy approximation with correct asymptotic behavior. *Phys. Rev. A*, 38:3098–3100, Sep 1988.
- [90] John P. Perdew, J. A. Chevary, S. H. Vosko, Koblar A. Jackson, Mark R. Pederson, D. J. Singh, and Carlos Fiolhais. Atoms, molecules, solids, and surfaces: Applications of the generalized gradient approximation for exchange and correlation. *Phys. Rev. B*, 46:6671–6687, Sep 1992.
- [91] K. Kim and K. D. Jordan. Comparison of density functional and MP2 calculations on the water monomer and dimer. *J. Phys. Chem.*, 98(40):10089–10094, 1994.
- [92] M. Dion, H. Rydberg, E. Schröder, D. C. Langreth, and B. I. Lundqvist. van der Waals density functional for general geometries. *Phys. Rev. Lett.*, 92:246401, Jun 2004.



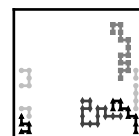
- [93] Alexandre Tkatchenko and Matthias Scheffler. Accurate molecular van der Waals interactions from ground-state electron density and free-atom reference data. *Phys. Rev. Lett.*, 102:073005, Feb 2009.
- [94] M. C. Payne, M. P. Teter, D.C. Allan, T.A. Arias, and J.D. Joannopoulos. Iterative minimization techniques for ab initio total-energy calculations: Molecular dynamics and conjugate gradients. *Rev. Mod. Phys.*, 64(4):1045–1097, 1992.
- [95] R. Ditchfield, W. J. Hehre, and J. A. Pople. Self-consistent molecular-orbital methods. IX. An extended gaussian-type basis for molecular-orbital studies of organic molecules. *J. Chem. Phys.*, 54(2):724–728, 1971.
- [96] J. S. Spencer, N. S. Blunt, and W. M. C. Foulkes. The sign problem and population dynamics in the full configuration interaction quantum Monte Carlo method. *J. Chem. Phys.*, 136(5):054110, 2012.
- [97] Greg Lever, Daniel J. Cole, Nicholas D. M. Hine, Peter D. Haynes, and Mike C. Payne. Electrostatic considerations affecting the calculated HOMO-LUMO gap in protein molecules. *J. Phys.-Condens. Mat.*, 25(15):152101, 2013.
- [98] Kieron Burke. Perspective on density functional theory. *J. Chem. Phys.*, 136:150901, 2012.
- [99] Peter G. Kerr and Louis Huang. Review: Membranes for haemodialysis. *Nephrology*, 15(4):381–385, 2010.
- [100] Kah Peng Lee, Tom C. Arnot, and Davide Mattia. A review of reverse osmosis membrane materials for desalination-development to date and future potential. *J. membrane Sci.*, 370(1–2):1–22, 2011.
- [101] Lloyd S. White. Development of large-scale applications in organic solvent nanofiltration and pervaporation for chemical and refining processes. *J. membrane Sci.*, 286(1–2):26–35, 2006.



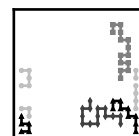
- [102] Geoffrey M. Geise, Hae-Seung Lee, Daniel J. Miller, Benny D. Freeman, James E. McGrath, and Donald R. Paul. Water purification by membranes: The role of polymer science. *J. Polym. Sci. Pol. Phys.*, 48(15):1685–1718, 2010.
- [103] Laura A. Hoover, Jessica D. Schiffman, and Menachem Elimelech. Nanofibers in thin-film composite membrane support layers: Enabling expanded application of forward and pressure retarded osmosis. *Desalination*, 308(0):73–81, 2013.
- [104] Wilfredo Yave, Anja Car, Jan Wind, and Klaus-Viktor Peinemann. Nano-metric thin film membranes manufactured on square meter scale: Ultra-thin films for CO₂ capture. *Nanotechnology*, 21(39):395301, 2010.
- [105] H. Strathmann and K. Kock. The formation mechanism of phase inversion membranes. *Desalination*, 21(3):241–255, September 1977.
- [106] Pieter Vandezande, Xianfeng Li, Lieven E.M. Gevers, and Ivo F. J. Vankelecom. High throughput study of phase inversion parameters for polyimide-based SRNF membranes. *J. membrane Sci.*, 330(1–2):307–318, 2009.
- [107] Pieter Vandezande, Lieven E.M. Gevers, Pierre A. Jacobs, and Ivo F. J. Vankelecom. Preparation parameters influencing the performance of SRNF membranes cast from polyimide solutions via SEPPI. *Sep. Purif. Technol.*, 66(1):104–110, 2009.
- [108] Katrien Vanherck, Pieter Vandezande, Steliana O. Aldea, and Ivo F. J. Vankelecom. Cross-linked polyimide membranes for solvent resistant nanofiltration in aprotic solvents. *J. membrane Sci.*, 320(1–2):468–476, 2008.
- [109] Agnieszka K. Holda, Ben Aernouts, Wouter Saeys, and Ivo F. J. Vankelecom. Study of polymer concentration and evaporation time as phase inversion parameters for polysulfone-based SRNF membranes. *J. membrane Sci.*, 442(0):196–205, 2013.



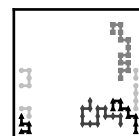
-
- [110] Allan F. M. Barton. *CRC handbook of solubility parameters and other cohesion parameters*. CRC Press, London UK, 1983.
- [111] Y. H. See Toh, X. X. Loh, K. Li, A. Bismarck, and A. G. Livingston. In search of a standard method for the characterisation of organic solvent nanofiltration membranes. *J. membrane Sci.*, 291(1):120–125, 2007.
- [112] W. Richard Bowen and Julian S. Welfoot. Modelling the performance of membrane nanofiltration-critical assessment and model development. *Chem. Eng. Sci.*, 57(7):1121–1137, 2002.
- [113] W. Richard Bowen and Julian S. Welfoot. Modelling of membrane nanofiltration-pore size distribution effects. *Chem. Eng. Sci.*, 57(8):1393–1407, 2002.
- [114] Julius Glater. The early history of reverse osmosis membrane development. *Desalination*, 117(1):297–309, 1998.
- [115] Richard J. Blaikie and David O. S. Melville. Imaging through planar silver lenses in the optical near field. *J. Opt. A-Pure Appl. Op.*, 7(2):S176, 2005.
- [116] Lüthi R., Meyer, Bammerlin M., Baratoff A., Lü J., Guggisberg M., and Güntherodt H.-J. *Resolution Limits of Force Microscopy*, chapter 19, pages 300–311. American Chemical Society, 1998.
- [117] G. H. Michler. *Electron Microscopy of Polymers*. Springer, 2008.
- [118] Jr. Prince E. Rouse. A theory of the linear viscoelastic properties of dilute solutions of coiling polymers. *J. Chem. Phys.*, 21(7):1272–1280, 1953.
- [119] Bruno H. Zimm. Dynamics of polymer molecules in dilute solution: Viscoelasticity, flow birefringence and dielectric loss. *J. Chem. Phys.*, 24(2):269–278, 1956.
- [120] Kurt Kremer, Gary S. Grest, and I. Carmesin. Crossover from rouse to reptation dynamics: A molecular-dynamics simulation. *Phys. Rev. Lett.*, 61:566–569, Aug 1988.



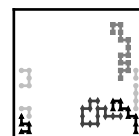
-
- [121] Hsiao-Ping Hsu, Wolfgang Paul, and Kurt Binder. Understanding the multiple length scales describing the structure of bottle-brush polymers by Monte Carlo simulation methods. *Macromol. Theor. Simul.*, 20(7):510–525, 2011.
- [122] K. Binder and W. Paul. Recent developments in Monte Carlo simulations of lattice models for polymer systems. *Macromolecules*, 41(13):4537–4550, 2008.
- [123] Peter Grassberger. Pruned-enriched rosenbluth method: Simulations of θ polymers of chain length up to 1 000 000. *Phys. Rev. E*, 56:3682–3693, Sep 1997.
- [124] Beste Bayramoglu and Roland Faller. Structural properties of polystyrene oligomers in different environments: A molecular dynamics study. *Phys. Chem. Chem. Phys.*, 13:18107–18114, 2011.
- [125] Hiroo Fukunaga, Jun ichi Takimoto, and Masao Doi. A coarse-graining procedure for flexible polymer chains with bonded and nonbonded interactions. *J. Chem. Phys.*, 116(18):8183–8190, 2002.
- [126] Scott J. Weiner, Peter A. Kollman, David A. Case, U. Chandra Singh, Caterina Ghio, Giuliano Alagona, Salvatore Profeta, and Paul Weinerl. A New Force Field for Molecular Mechanical Simulation of Nucleic Acids and Proteins. *J. Am. Chem. Soc.*, 106(17):765–784, 1984.
- [127] Christine Peter and Kurt Kremer. Multiscale simulation of soft matter systems — from the atomistic to the coarse-grained level and back. *Soft Matter*, 5:4357–4366, 2009.
- [128] Maurice L. Huggins. Solutions of long chain compounds. *J. Chem. Phys.*, 9(5):440–440, 1941.
- [129] Paul J. Flory. Thermodynamics of high polymer solutions. *J. Chem. Phys.*, 9(8):660–661, 1941.



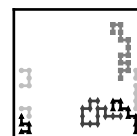
-
- [130] P. Traskelin, T. L. Kuhl, and R. Faller. Molecular dynamics simulations of polystyrene brushes in dry conditions and in toluene solution. *Phys. Chem. Chem. Phys.*, 11:11324–11332, 2009.
- [131] P. G. de Gennes. Reptation of a polymer chain in the presence of fixed obstacles. *J. Chem. Phys.*, 55(2):572–579, 1971.
- [132] J Klein. The onset of entangled behavior in semidilute and concentrated polymer solutions. *Macromolecules*, 11(5):852–858, 1978.
- [133] Paul D. Grossman and David S. Soane. Experimental and theoretical studies of DNA separations by capillary electrophoresis in entangled polymer solutions. *Biopolymers*, 31(10):1221–1228, 1991.
- [134] Herbert B. Callen. *Thermodynamics and an Introduction to Thermostatistics*. John Wiley & Sons, New York, U.S.A., 2 edition, 1985.
- [135] Zhonglin Luo and Jianwen Jiang. Molecular dynamics and dissipative particle dynamics simulations for the miscibility of poly(ethylene oxide)/poly(vinyl chloride) blends. *Polymer*, 51(1):291–299, 2010.
- [136] Inger Martinez de Arenaza, Emilio Meaurio, Borja Coto, and Jose-Ramon Sarasua. Molecular dynamics modelling for the analysis and prediction of miscibility in polylactide/polyvinylphenol blends. *Polymer*, 51(19):4431–4438, 2010.
- [137] Abolfazl Noorjahan and Phillip Choi. Thermodynamic properties of poly(vinyl alcohol) with different tacticities estimated from molecular dynamics simulation. *Polymer*, 54(16):4212–4219, 2013.
- [138] Ruhong Zhou and Bruce J. Berne. Can a continuum solvent model reproduce the free energy landscape of a β -hairpin folding in water? *P. Natl. A. Sci. U.S.A.*, 99(20):12777–12782, 2002.
- [139] Di Qiu, Peter S. Shenkin, Frank P. Hollinger, and W. Clark Still. The GB/SA continuum model for solvation. A fast analytical method for the calculation of approximate born radii. *J. Phys. Chem. A*, 101(16):3005–3014, 1997.



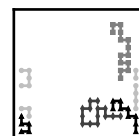
- [140] E. Tocci, D. Hofmann, D. Paul, N. Russo, and E. Drioli. A molecular simulation study on gas diffusion in a dense poly(ether-ether-ketone) membrane. *Polymer*, 42(2):521–533, 2001.
- [141] Hendrik Frentrup, Carlos Avendaño, Martin Horsch, Alaaeldin Salih, and Erich A Müller. Transport diffusivities of fluids in nanopores by non-equilibrium molecular dynamics simulation. *Mol. Simulat.*, 38(7):540–553, 2012.
- [142] Hailu Liu, Xuejia Ding, Jun Yi, Liqun Zhang, and Sizhu Wu. Study on the gas permeabilities in styrene-butadiene rubber by molecular dynamics simulation. *Frontiers of Chemical Engineering in China*, 4(3):257–262, 2010.
- [143] Sylvie Neyertz and David Brown. Oxygen sorption in glassy polymers studied at the molecular level. *Macromolecules*, 42(21):8521–8533, 2009.
- [144] S. Neyertz, D. Brown, S. Pandiyan, and N. F. A. van der Vegt. Carbon dioxide diffusion and plasticization in fluorinated polyimides. *Macromolecules*, 43(18):7813–7827, 2010.
- [145] S. Neyertz and D. Brown. A trajectory-extending kinetic Monte Carlo (TEKMC) method for estimating penetrant diffusion coefficients in molecular dynamics simulations of glassy polymers. *Macromolecules*, 43(21):9210–9214, 2010.
- [146] Grégory Marque, Sylvie Neyertz, Jacques Verdu, Valéry Prunier, and David Brown. Molecular dynamics simulation study of water in amorphous kapton. *Macromolecules*, 41(9):3349–3362, 2008.
- [147] Sudharsan Pandiyan, David Brown, Sylvie Neyertz, and NFA Van der Vegt. Carbon dioxide solubility in three fluorinated polyimides studied by molecular dynamics simulations. *Macromolecules*, 43(5):2605–2621, 2010.
- [148] S. Pandiyan, D. Brown, NFA van der Vegt, and S. Neyertz. Atomistic models of three fluorinated polyimides in the amorphous state. *J. Polym. Sci. Pol. Phys.*, 47(12):1166–1180, 2009.



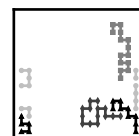
- [149] Sylvie Neyertz, Anthony Douanne, and David Brown. A molecular dynamics simulation study of surface effects on gas permeation in free-standing polyimide membranes. *J. membrane Sci.*, 280(1):517–529, 2006.
- [150] K. Anton Feenstra, Berk Hess, and Herman J. C. Berendsen. Improving efficiency of large time-scale molecular dynamics simulations of hydrogen-rich systems. *J. Comput. Chem.*, 20(8):786–798, 1999.
- [151] Jean-Paul Ryckaert, Giovanni Ciccotti, and Herman J.C. Berendsen. Numerical integration of the cartesian equations of motion of a system with constraints: Molecular dynamics of n-alkanes. *J. Comput. Phys.*, 23(3):327–341, 1977.
- [152] Hans C Andersen. RATTLE: A “velocity” version of the SHAKE algorithm for molecular dynamics calculations. *J. Comput. Phys.*, 52(1):24–34, 1983.
- [153] Berk Hess, Henk Bekker, Herman J. C. Berendsen, and Johannes G. E. M. Fraaije. LINCS: A linear constraint solver for molecular simulations. *J. Comput. Chem.*, 18(12):1463–1472, 1997.
- [154] Berk Hess. P-LINCS: A parallel linear constraint solver for mol. simulat. *J. Chem. Theory Comput.*, 4(1):116–122, 2008.
- [155] Berk Hess, Carsten Kutzner, David van der Spoel, and Erik Lindahl. GRO-MACS 4: Algorithms for Highly Efficient, Load-Balanced, and Scalable Mol. Simulat. *J. Chem. Theory Comput.*, 4(3):435–447, March 2008.
- [156] Kurt Anderson, Rudranarayan Mukherjee, James Critchley, John Ziegler, and Scott Lipton. POEMS: Parallelizable open-source efficient multibody software. *Eng. Comput.*, 23(1):11–23, 2007.
- [157] Giovanni Ciccotti and Jean-Paul Ryckaert. Molecular dynamics simulation of rigid molecules. *Comput. Phys. Rep.*, 4(6):346–392, 1986.
- [158] Shiguang Li, Dennis J. Rocha, S. James Zhou, Howard S. Meyer, Benjamin Bikson, and Yong Ding. Post-combustion CO₂ capture using superhydrophobic, polyether ether ketone, hollow fiber membrane contactors. *J. Membrane Sci.*, 430(0):79–86, 2013.



- [159] H. Koike, K. Kida, E. C. Santos, J. Rozwadowska, Y. Kashima, and K. Kanemasu. Self-lubrication of PEEK polymer bearings in rolling contact fatigue under radial loads. *Tribol. Int.*, 49(0):30–38, 2012.
- [160] Mohammed Naffakh, Ana M. Diez-Pascual, and Marian A. Gomez-Fatou. New hybrid nanocomposites containing carbon nanotubes, inorganic fullerene-like WS₂ nanoparticles and poly(ether ether ketone) (PEEK). *J. Mater. Chem.*, 21:7425–7433, 2011.
- [161] Shuichi Miyamoto and Peter A Kollman. SETTLE: An analytical version of the SHAKE and RATTLE algorithm for rigid water models. *J. Comput. Chem.*, 13(8):952–962, 1992.
- [162] A. G. Bailey, C. P. Lowe, and A. P. Sutton. Efficient constraint dynamics using MILC SHAKE. *J. Comput. Phys.*, 227(20):8949–8959, 2008.
- [163] A. G. Bailey, C. P. Lowe, and A. P. Sutton. REVLD: A coarse-grained model for polymers. *Comput. Phys. Commun.*, 180(4):594–599, 2009.
- [164] Sang-Ho Lee, Kim Palmo, and Samuel Krimm. WIGGLE: A new constrained molecular dynamics algorithm in cartesian coordinates. *J. Comput. Phys.*, 210(1):171–182, 2005.
- [165] Carlo Pierleoni and Jean-Paul Ryckaert. On the use of the Nosé-Hoover thermostat in the simulation of dynamic properties of a single chain molecule in solvent. *Mol. Phys.*, 75(3):731–737, 1992.
- [166] Gerald R. Kneller and T. Mülders. Nosé-Andersen dynamics of partially rigid molecules: Coupling all degrees of freedom to heat and pressure baths. *Phys. Rev. E*, 54(6):6825, 1996.
- [167] H. Kamberaj, R. J. Low, and M. P. Neal. Time reversible and symplectic integrators for molecular dynamics simulations of rigid molecules. *J. Chem. Phys.*, 122:224114, 2005.
- [168] H. Berendsen. GROMACS: A message-passing parallel molecular dynamics implementation. *Comput. Phys. Commun.*, 91(1–3):43–56, September 1995.



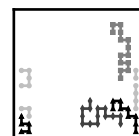
- [169] Erik Lindahl, Berk Hess, and D. van der Spoel. GROMACS 3.0: A package for Mol. Simulat. and trajectory analysis. *J. Mol. Model.*, 7(8):306–317, 2001.
- [170] David van der Spoel, Erik Lindahl, Berk Hess, Gerrit Groenhof, Alan E. Mark, and Herman J. C. Berendsen. GROMACS: Fast, flexible, and free. *J. Comput. Chem.*, 26(16):1701–1718, December 2005.
- [171] H. J. C. Berendsen and van Gunsteren. *W. F. Molecular dynamics simulations: Techniques and Approaches. in: Molecular Liquids-Dynamics and Interactions.* Reidel Dordrecht, The Netherlands, 1984.
- [172] Mark Robinson and Peter D Haynes. Dynamical effects in ab initio NMR calculations: Classical force fields fitted to quantum forces. *J. Chem. Phys.*, 133:084109, 2010.
- [173] William L. Jorgensen, David S. Maxwell, and Julian Tirado-Rives. Development and testing of the OPLS all-atom force field on conformational energetics and properties of organic liquids. *J. Am. Chem. Soc.*, 118(45):11225–11236, 1996.
- [174] Chr. Møller and M. S. Plesset. Note on an approximation treatment for many-electron systems. *Phys. Rev.*, 46:618–622, Oct 1934.
- [175] Paul Tangney and Sandro Scandolo. An ab initio parametrized interatomic force field for silica. *J. Chem. Phys.*, 117:8898, 2002.
- [176] P. Tangney and S. Scandolo. A many-body interatomic potential for ionic systems: Application to MgO. *J. Chem. Phys.*, 119(18):9673–9685, 2003.
- [177] Peter Debye. Näherungsformeln für die zylinderfunktionen für große werte des arguments und unbeschränkt veränderliche werte des index. *Math. Ann.*, 67(4):535–558, 1909.
- [178] Magnus R. Hestenes and Eduard Stiefel. Methods of conjugate gradients for solving linear systems. *J. Res. Nat. Bur. Stand.*, 49(6):409–436, 1952.



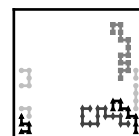
- [179] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical recipes 3rd edition: The art of scientific computing*. Cambridge university press, 2007.
- [180] Lee-Ping Wang and Troy Van Voorhis. Communication: Hybrid ensembles for improved force matching. *J. Chem. Phys.*, 133(23):231101, 2010.
- [181] Victor Rühle, Christoph Junghans, Alexander Lukyanov, Kurt Kremer, and Denis Andrienko. Versatile object-oriented toolkit for coarse-graining applications. *J. Chem. Theory Comput.*, 5(12):3211–3223, 2009.
- [182] Naveen Michaud-Agrawal, Elizabeth J. Denning, Thomas B. Woolf, and Oliver Beckstein. MDAnalysis: A toolkit for the analysis of molecular dynamics simulations. *J. Comput. Chem.*, 32(10):2319–2327, 2011.
- [183] D. A. Case, T. A. Darden, T. E. Cheatham, C. L. Simmerling, J. Wang, R. E. Duke, R. Luo, R. C. Walker, W. Zhang, K. M. Merz, B. Roberts, B. Wang, S. Hayik, A. Roitberg, G. Seabra, I. Kolossvai, K. F. Wong, F. Paesani, J. Vanicek, J. Liu, X. Wu, S. R. Brozell, T. Steinbrecher, H. Gohlke, Q. Cai, X. Ye, J. Wang, M. J. Hsieh, G. Cui, D. R. Roe, D. H. Mathews, M. G. Seetin, C. Sagui, V. Babin, T. Luchko, S. Gusarov, A. Kovalenko, and P. A. Kollman. AMBER 9, University of California, San Francisco, 2006.
- [184] B. R. Brooks, C. L. Brooks, A. D. Mackerell, L. Nilsson, R. J. Petrella, B. Roux, Y. Won, G. Archontis, C. Bartels, S. Boresch, A. Caffisch, L. Caves, Q. Cui, A. R. Dinner, M. Feig, S. Fischer, J. Gao, M. Hodoscek, W. Im, K. Kuczera, T. Lazaridis, J. Ma, V. Ovchinnikov, E. Paci, R. W. Pastor, C. B. Post, J. Z. Pu, M. Schaefer, B. Tidor, R. M. Venable, H. L. Woodcock, X. Wu, W. Yang, D. M. York, and M. Karplus. CHARMM: The biomolecular simulation program. *J. Comput. Chem.*, 30(10):1545–1614, 2009.
- [185] I. T. Todorow, W. Smith, K. Trachenko, and M. T. Dove. DL_POLY_3: New dimensions in molecular dynamics simulations via massive parallelism. *J. Mater. Chem.*, 16:1911–1918, 2006.



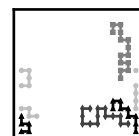
- [186] Steve Plimpton. Fast parallel algorithms for short-range molecular dynamics. *J. Comput. Phys.*, 117(1):1–19, 1995.
- [187] James C. Phillips, Rosemary Braun, Wei Wang, James Gumbart, Emad Tajkhorshid, Elizabeth Villa, Christophe Chipot, Robert D. Skeel, Laxmikant Kalé, and Klaus Schulten. Scalable molecular dynamics with NAMD. *J. Comput. Chem.*, 26(16):1781–1802, 2005.
- [188] Alessandro Laio and Michele Parrinello. Escaping free-energy minima. *P. Natl. A. Sci. U.S.A.*, 99(20):12562–12566, 2002.
- [189] Robert T. F. Stepto. Dispersity in polymer science (IUPAC recommendations 2009). *Pure Appl. Chem.*, 81(2):351–353, 2009.
- [190] Iwona Soroko. *Polyimide Organic Solvent Nanofiltration Membranes-Formation and Function*. PhD thesis, Imperial College London, 2011.
- [191] Allan J. Easteal and Lawrence A. Woolf. Self-diffusion and volumetric measurements for N-methylformamide and N,N-dimethylformamide at temperatures from 240 to 313 K and pressures up to 300 MPa. *J. Chem. Soc., Faraday Trans. 1*, 81:2821–2833, 1985.
- [192] Manfred Holz, Stefan R. Heil, and Antonio Sacco. Temperature-dependent self-diffusion coefficients of water and six selected molecular liquids for calibration in accurate ^1H NMR PFG measurements. *Phys. Chem. Chem. Phys.*, 2:4740–4742, 2000.
- [193] R. Mills. Self-diffusion in normal and heavy water in the range 1-45.deg. *J. Phys. Chem.-U.S.*, 77(5):685–688, 1973.
- [194] P. A. M. Dirac. Quantum mechanics of many-electron systems. *P. R. soc. Lond. A-Conta.*, 123(792):714–733, 1929.
- [195] Chris-Kriton Skylaris, Peter D. Haynes, Arash A. Mostofi, and Mike C. Payne. Introducing ONETEP: Linear-scaling density functional simulations on parallel computers. *J. Chem. Phys.*, 122(8):84119, March 2005.



- [196] Nicholas Hine, Mark Robinson, Peter Haynes, Chris-Kriton Skylaris, Mike Payne, and Arash Mostofi. Accurate ionic forces and geometry optimization in linear-scaling density-functional theory with local orbitals. *Phys. Rev. B*, 83(19):1–10, May 2011.
- [197] A.A. Mostofi, P.D. Haynes, C.-K. Skylaris, and M. C. Payne. ONETEP: Linear-scaling density-functional theory with plane-waves. *Mol. Simulat.*, 33(7):551–555, June 2007.
- [198] William L. Jorgensen, Jayaraman Chandrasekhar, Jeffry D. Madura, Roger W. Impey, and Michael L. Klein. Comparison of simple potential functions for simulating liquid water. *J. Chem. Phys.*, 79(2):926–935, 1983.
- [199] Athanassios Z. Panagiotopoulos, Vicky Wong, and M. Antonio Floriano. Phase equilibria of lattice polymers from histogram reweighting monte carlo simulations. *Macromolecules*, 31(3):912–918, 1998.
- [200] Qiliang Yan and Juan J. de Pablo. Hyperparallel tempering monte carlo simulation of polymeric systems. *J. Chem. Phys.*, 113(3):1276–1282, 2000.
- [201] Hsiao-Ping Hsu and Peter Grassberger. A review of monte carlo simulations of polymers with PERM. *J. Stat. Phys.*, 144(3):597–637, 2011.
- [202] Supriyo Bhattacharya and Keith E. Gubbins. Fast method for computing pore size distributions of model materials. *Langmuir*, 22(18):7726–7731, 2006.
- [203] Miguel A. González, Georg Menzl, Juan L. Aragones, Philipp Geiger, Frederic Caupin, Jose L. F. Abascal, Christoph Dellago, and Chantal Valeriani. Detecting vapour bubbles in simulations of metastable water. *J. Chem. Phys.*, 141(18):511, 2014.
- [204] C.A. Schneider, W.S. Rasband, and K.W. Eliceiri. NIH Image to ImageJ: 25 years of image analysis. *Nat. Methods*, 9:671–675, 2012.
- [205] Hsiao-Ping Hsu, Wolfgang Paul, and Kurt Binder. Polymer chain stiffness vs. excluded volume: A monte carlo study of the crossover towards the worm-like chain model. *Europhys. Lett.*, 92(2):28003, 2010.



- [206] Mark Robinson. *Accessing Large Length and Time Scales with Density Functional Theory*. PhD thesis, University of Cambridge, 2010.
- [207] Andrzej J. Rzepiela, Martti Louhivuori, Christine Peter, and Siewert J. Marrink. Hybrid simulations: combining atomistic and coarse-grained force fields using virtual sites. *Phys. Chem. Chem. Phys.*, 13:10437–10448, 2011.
- [208] R. K. Pathria. *Statistical Mechanics*. Butterworth Heineman, Oxford, UK, second edition, 1996.



Appendix A

Monte Carlo

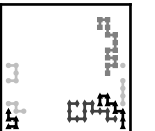
MC is often used to evaluate integrals, therefore, to understand the technique it is necessary to consider the process of numerical integration or quadrature. To begin with consider an integral which needs to be evaluated numerically such as

$$I = \int_a^b f(x)dx \quad (\text{A.1})$$

where a , and b are finite, $x \in \mathbb{R}$, and $f(x)$ is finite in the range $[a, b]$. It is possible to approximate this integral by evaluating the function at N points x_i in the interval $[a, b]$ then summing their values

$$I \approx \sum_{i=1}^N w_i f(x_i). \quad (\text{A.2})$$

The weight (w_i) of each interval is generally proportional to the width of the sampled interval, for instance in the case of uniform sampling it is $\frac{b-a}{N}$. If the x_i are chosen in systematic way (e.g., uniformly or a cosine distribution) the error in the integration is generally classified by the value of M in the leading term of the error $O(N^{-M})$, where M is an integer that depends on the algorithm used. For one dimensional integrals such as Eq. (A.1) a variety of techniques are available which can easily achieve $M \geq 4$ (e.g., extended Simpson's rule). However, when a higher dimensional integral is sampled using a regular grid the error behaves as $O(N^{-\frac{M}{D}})$ where D is the dimensionality of the system. Higher



dimensional integrals often occur in molecular systems for instance the partition function has six dimensions (three position, three momentum) for every particle in the system¹. Clearly for higher dimensional integrals an extremely large value of N or M would be required to ensure reasonable sampling using systematic grid.

An alternative approach to sampling an integral using a grid is to sample it randomly. This approach is generally slow to converge, however, its error is always $O(N^{-\frac{1}{2}})$ regardless of the number of dimensions being integrated over. As a result MC integration is invaluable in numerical evaluation of integrals. However, the true power of MC integration is that it can be used to bias the sampling such that certain areas of phase space are sampled more often. This is useful when the value of the integral is dominated by a subspace of the total integration region. To understand the use of biased MC sampling consider calculating $\langle A \rangle$, for some observable A , in statistical mechanics this is given by,

$$\langle A \rangle = \frac{1}{\mathcal{Z}} \int A(\{\mathbf{q}_i\}, \{\mathbf{p}_i\}) \exp(-\beta H(\{\mathbf{q}_i\}, \{\mathbf{p}_i\})) d\Omega. \quad (\text{A.3})$$

Here $\{\mathbf{q}_i\}$, $\{\mathbf{p}_i\}$ are respectively the sets of all position and momenta vectors in the system, together they span the phase space Ω , H is the Hamiltonian of the system, $\beta = 1/(k_B T)$ where k_B is Boltzmann's constant, T is temperature, and the partition function $\mathcal{Z} = \int \exp(-\beta H(\{\mathbf{q}_i\}, \{\mathbf{p}_i\})) \prod_i d\mathbf{q}_i d\mathbf{p}_i$. For further details consult e.g., [208]. If the integral is sampled randomly it will eventually converge to the correct value, and all values H will occur with a frequency proportional to the volume of phase space over which they occur. Examining Eq. (A.3), it is clear that large values of H , relative to β will be exponentially suppressed in the integral. Therefore, regions of phase space of interest represents a small subspace of the total phase space, resulting in random sampling expending a large number of evaluations calculating data points which will have minimal contribution to the final answer. To see that the subspace is small, consider the kinetic energy part of the Hamiltonian ($\sum_i \mathbf{p}_i^2 / (2m_i)$ where m_i is the mass of the i^{th} particle), this will always have a larger proportion of its possible phase space above a given

¹The partition function is also evaluated over all space, rather than a finite interval. This adds a layer of complexity which will be addressed only in the context of MC sampling.



energy than below it.

Metropolis *et. al.* demonstrated a Markov chain method for sampling an integral in an efficient manner[17]. Their insight was that if $(\{\mathbf{q}_i\}_i, \{\mathbf{p}_i\}_i)$ were picked based on the distribution $\exp(-\beta H(\{\mathbf{q}_i\}_i, \{\mathbf{p}_i\}_i))$, the integral could be approximated by:

$$\langle A \rangle \approx \sum_{i=1}^N \frac{A(\{\mathbf{q}_i\}_i, \{\mathbf{p}_i\}_i)}{N}. \quad (\text{A.4})$$

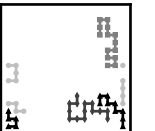
This ensures that the values of A with the largest contribution to the integral are sampled most often. This process was generalised in 1970 by Hastings[24] resulting in MH.

The MH method samples according to a Markov chain, a chain in this context is a set of sequential values for the state variables, in this case $(\{\mathbf{q}_i\}, \{\mathbf{p}_i\})$, of the system. Sequential values in the chain are obtained by modifying the values of the state variables in some manner, then either accepting the new values or rejecting them. The modification of the variables is called a move, the choice of move is highly system dependent and is discussed in the context of polymers in Sec. 2.1.2. Any move satisfying detailed balance, will given a sufficient number of MH moves, produce the desired result. Detailed balance requires that the probability of making a move is equal to the probability of making its reverse move, this can be expressed mathematically as

$$P(i)P_A(j|i) = P(j)P_A(i|j), \quad (\text{A.5})$$

where $P(i)$ is the probability of being in state i , and $P_A(j|i)$ is the probability of accepting a move to j given that the system is in state i . Detailed balance is not required for a MH move to be valid, the weaker balance condition can be sufficient, however, all detailed balance moves are valid moves whilst some balance moves are not therefore it is generally advisable to use a detailed balance move[25].

The MH acceptance criteria is obtained by assigning a probability of occupancy to each state of the system given by its Boltzmann weight, $P(i) = \exp(-\beta E_i)$ where $E_i = H(\{\mathbf{q}_i\}_i, \{\mathbf{p}_i\}_i)$ is the energy of the state i . Rearranging



detailed balance relationship, Eq. (A.5) yields:

$$\frac{P_A(j|i)}{P_A(i|j)} = \frac{P(j)}{P(i)}, \quad (\text{A.6})$$

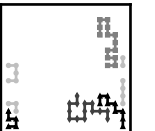
which after substitution becomes,

$$\frac{P_A(j|i)}{P_A(i|j)} = \exp(-\beta \Delta E_{ij}), \quad (\text{A.7})$$

where $\Delta E_{ij} = E_j - E_i$. In principle any acceptance probability satisfying this equation is valid, the MH scheme uses,

$$P_A(j|i) = \min[1, \exp(-\beta \Delta E_{ij})], \quad (\text{A.8})$$

which clearly satisfies Eq. (A.7). Using this to generate a chain of configurations allows the evaluation of Eq. (A.3), with a high degree of precision by simply adding the value of the observable in every configuration in the chain and dividing by the chain length.



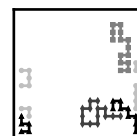
Appendix B

Supplemental Energy Drift Graphs

This appendix provides the plots of energy drift against time for the 40 simulations of short P84 oligomers in dilute solution. The captions specify the oligomer and solvent used. The oligomers are labelled as Short 1, 2, 3 and Long 1, 2, which are described in Tab. B.1. The figures on the subsequent pages are grouped by solvent polymer with all temperatures for a given pair shown as a single figure.

Oligomer	BTDA groups	TDA groups	MDA groups
Short 1	6	4	1
Short 2	6	5	0
Short 3	6	3	2
Long 1	9	7	1
Long 2	9	7	1

Table B.1: The labels and number of various groups for each of the five P84 oligomers simulated in dilute solution.



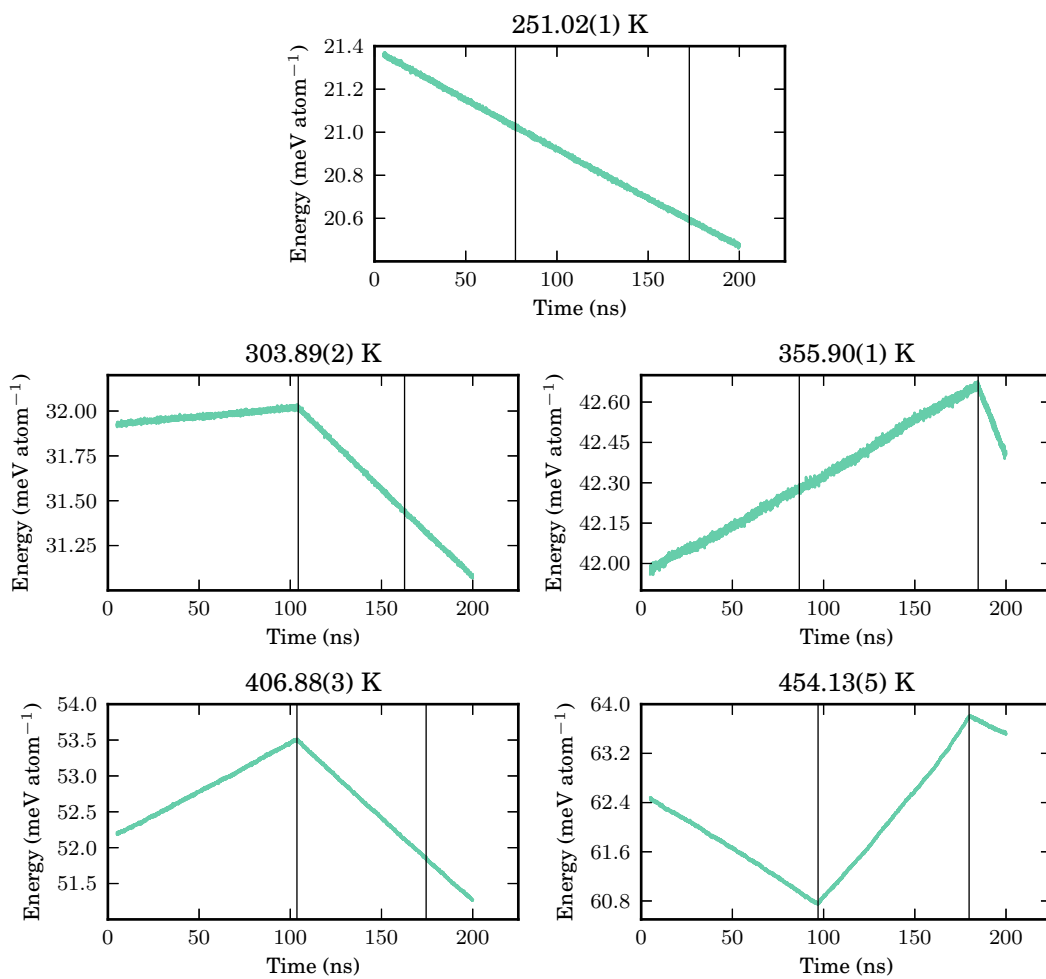
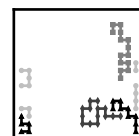


Figure B.1: Short oligomer 1 in DMF, 93287 atoms. The simulations average temperature is shown immediately above the plot, with the error on the last digit indicated by the number enclosed in brackets. Vertical lines indicate simulation restarts.



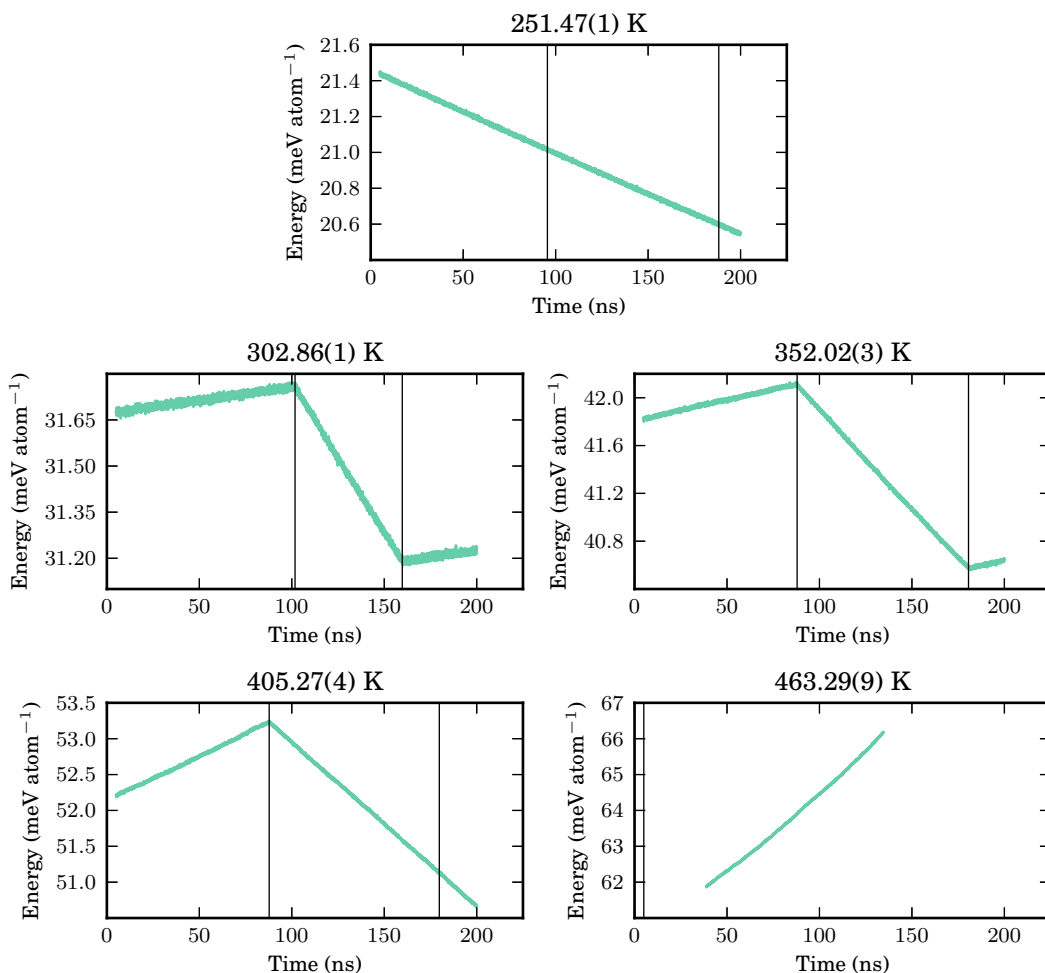
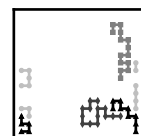


Figure B.2: Short oligomer 2 in DMF, 93325 atoms. The simulations average temperature is shown immediately above the plot, with the error on the last digit indicated by the number enclosed in brackets. Vertical lines indicate simulation restarts. The 463.29(9) K simulation was continued from a simulation which had crashed due to a power outage on the cx2 machine at imperial. This power outage had corrupted several files that were being written to at the time. Therefore it was continued from a backup checkpoint file which was not affected, however, all data from the first 38.8 ns was lost. The simulation was not continued after 134.5 ns as by that point the issues discussed in Sec. 5.3 had been identified.



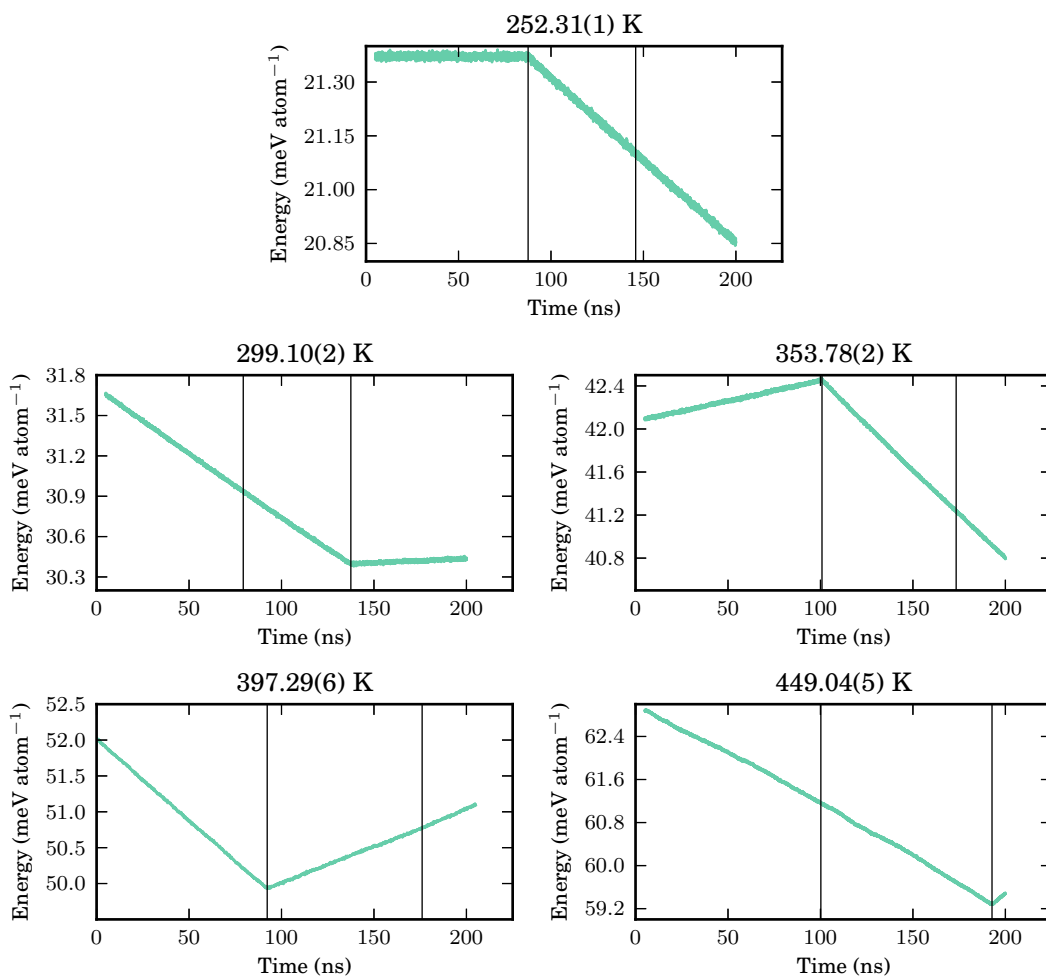
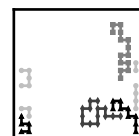


Figure B.3: Short oligomer 3 in DMF, 93189 atoms. The simulations average temperature is shown immediately above the plot, with the error on the last digit indicated by the number enclosed in brackets. Vertical lines indicate simulation restarts.



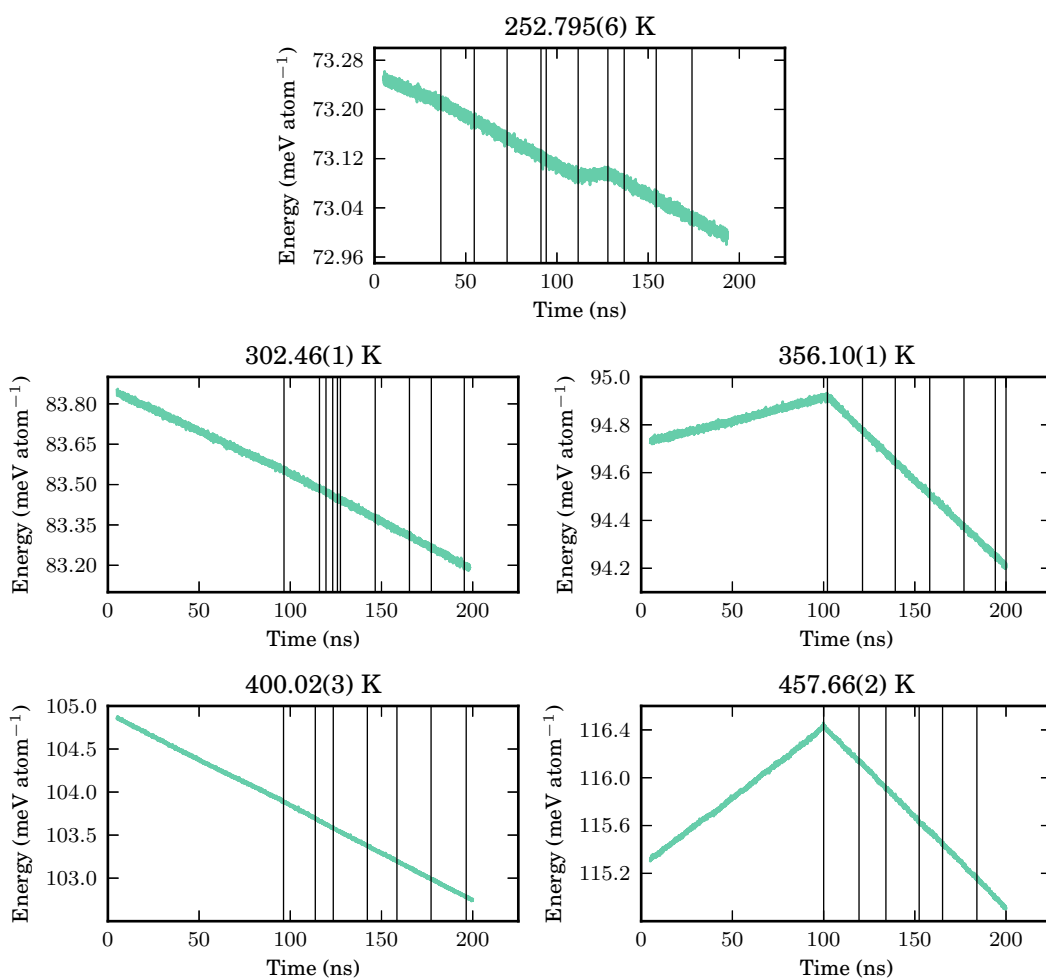
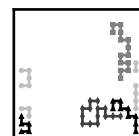


Figure B.4: Short oligomer 1 in dioxane, 91893 atoms. The simulations average temperature is shown immediately above the plot, with the error on the last digit indicated by the number enclosed in brackets. Vertical lines indicate simulation restarts. The large number of restarts in the 300 K simulation was caused by a series of issues relating to the Lustre file system on HECToR which caused the simulation to stop. As the simulations were grouped together this affected all 300 K simulations in dioxane.



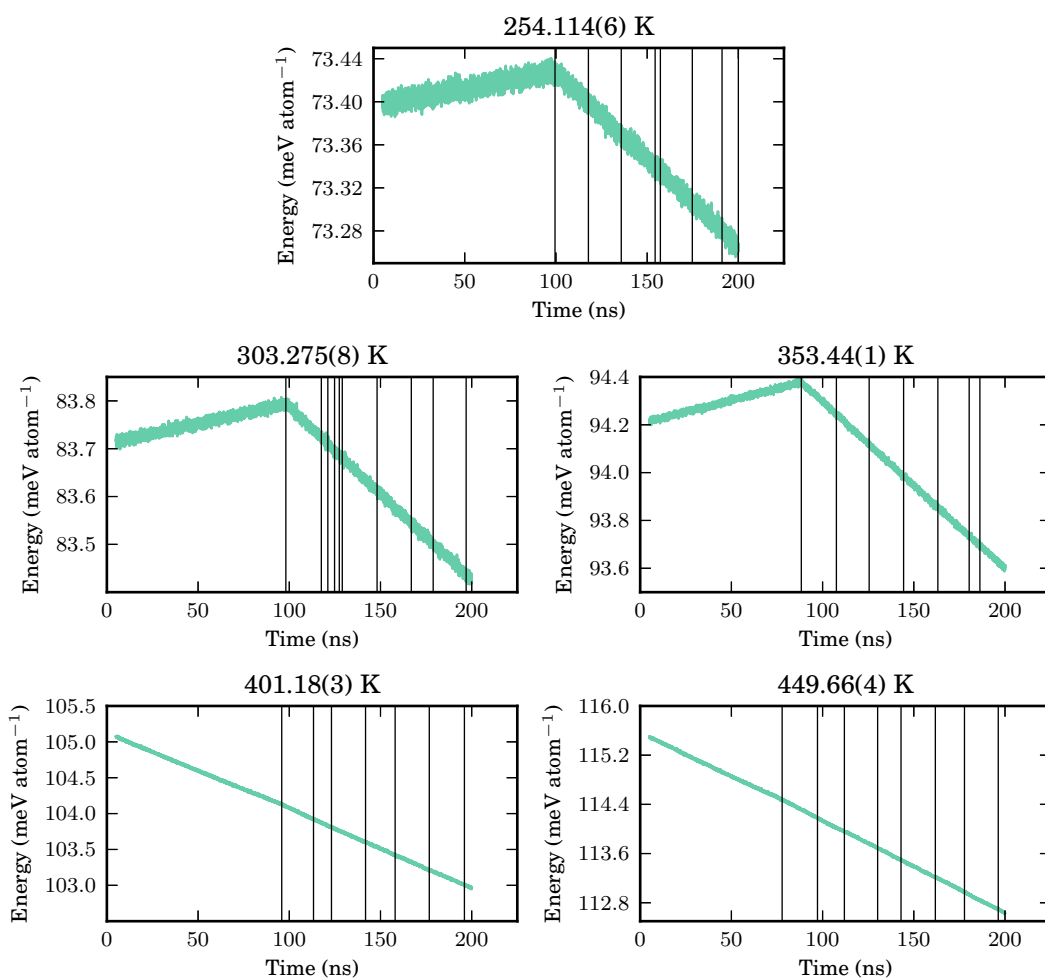
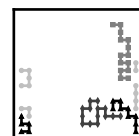


Figure B.5: Short oligomer 2 in dioxane, 91953 atoms. The simulations average temperature is shown immediately above the plot, with the error on the last digit indicated by the number enclosed in brackets. Vertical lines indicate simulation restarts. The large number of restarts in the 300 K simulation was caused by a series of issues relating to the Lustre file system on HECToR which caused the simulation to stop. As the simulations were grouped together this affected all 300 K simulations in dioxane.



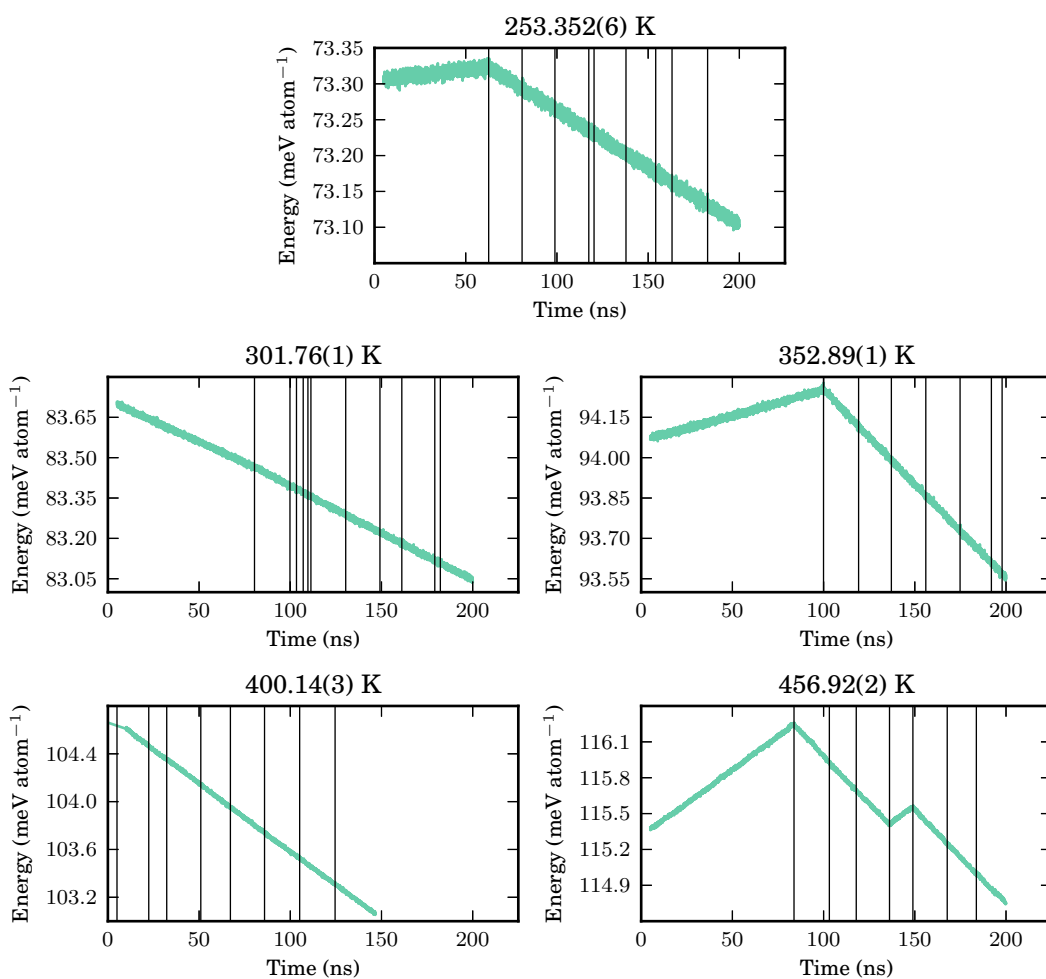
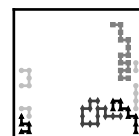


Figure B.6: Short oligomer 3 dioxane, 91875 atoms. The simulations average temperature is shown immediately above the plot, with the error on the last digit indicated by the number enclosed in brackets. Vertical lines indicate simulation restarts. The large number of restarts in the 300 K simulation was caused by a series of issues relating to the Lustre file system on HECToR which caused the simulation to stop. As the simulations were grouped together this affected all 300 K simulations in dioxane.



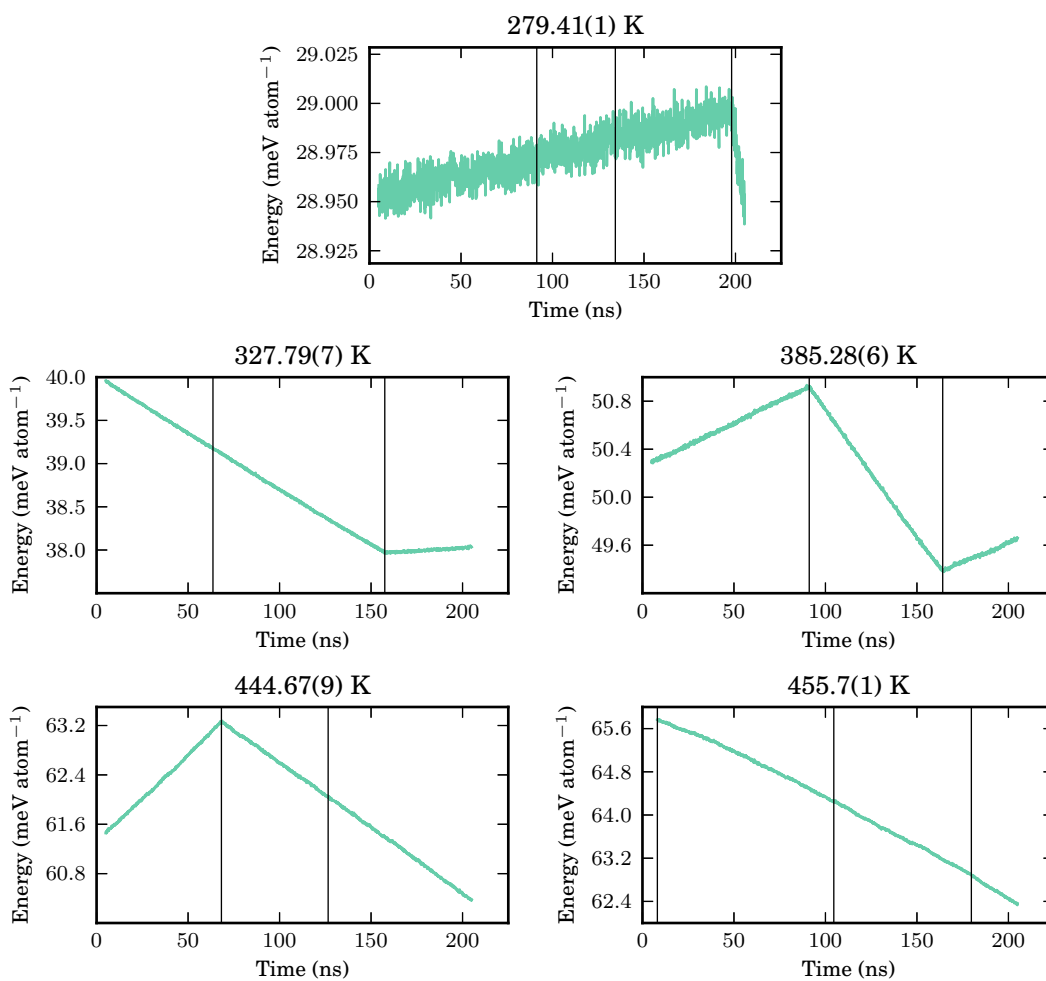
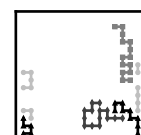


Figure B.7: Long oligomer 1 in DMF, 89060 atoms. The simulations average temperature is shown immediately above the plot, with the error on the last digit indicated by the number enclosed in brackets. Vertical lines indicate simulation restarts.



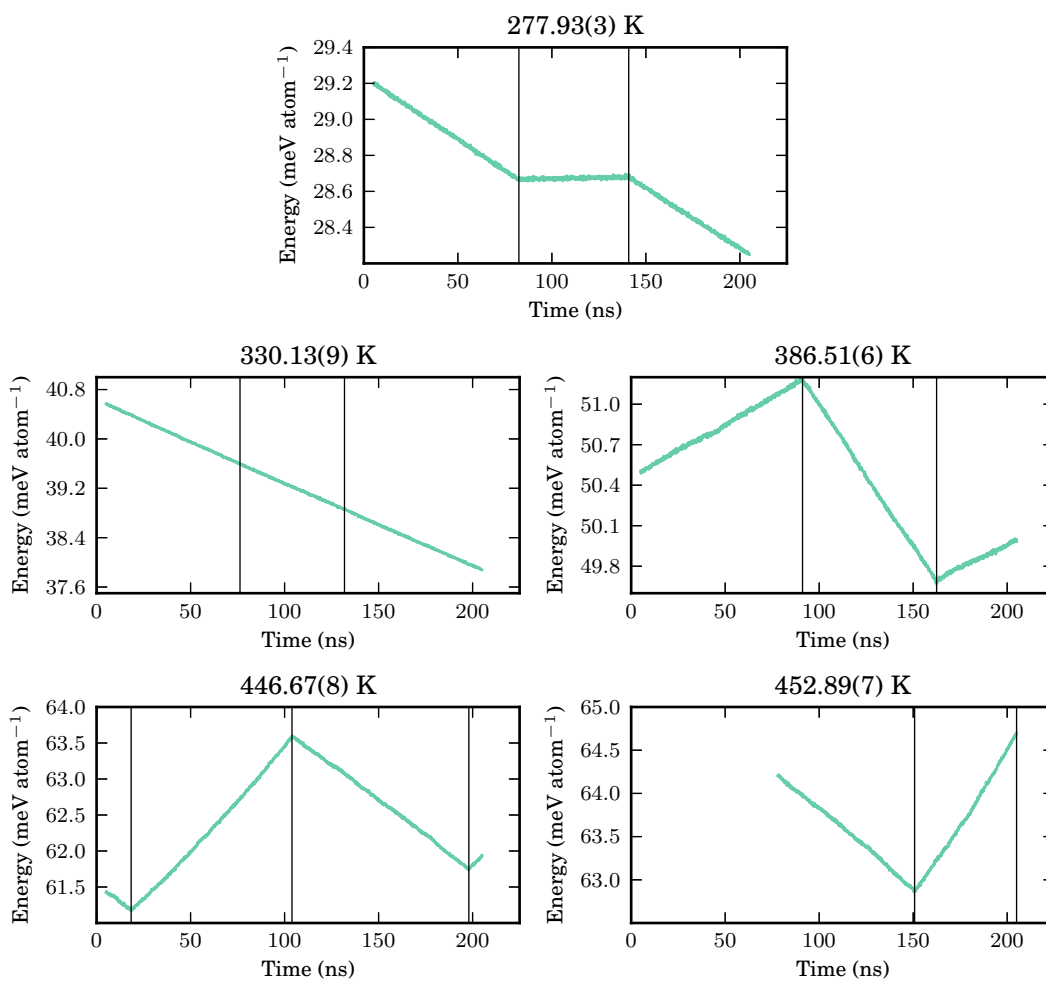


Figure B.8: Long oligomer 2 in DMF, 89000 atoms. The simulations average temperature is shown immediately above the plot, with the error on the last digit indicated by the number enclosed in brackets. Vertical lines indicate simulation restarts.

