Coarse–Grained Molecular Dynamics

PhD Thesis

David Edmunds
Department of Physics
Imperial College London

Abstract

In this work, we investigate the application of coarse–graining (CG) methods to molecular dynamics (MD) simulations. These methods provide access to length and time scales previously inaccessible to traditional materials simulation techniques. However, care must be taken when applying any coarse–graining strategy to ensure that we preserve the material properties of the system we are interested in.

We discuss common CG strategies, including their strengths, weaknesses and ease of application. The theory of coarse–graining is discussed within the framework of statistical mechanics, together with an analytic derivation of the CG partition function for a harmonic potential. We then apply this theory to a simple system of two interacting dimers, obtaining expressions for the CG free and internal energy. This example serves as a motivation for how to coarse–grain more realistic systems numerically. We introduce five different approaches to generating a CG potential, which we have termed the rigid and relaxed approximation, the constrained pair approach, the unconstrained box approach and the entropic approach. We apply each of these techniques to a system of C_{60} molecules, comparing our results against reference fully atomistic MD simulations of the same system. We find that the constrained pair approach provides an optimal balance between ease of generation and accuracy when compared to the reference model.

Declaration of Originality

The work in this thesis is entirely that of the author, except where otherwise referenced.

Copyright Declaration

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

Acknowledgements

I would like to thank my supervisors, Prof WMC Foulkes, Prof DD Vvedensky and Dr P Tangney, for their experience, advice and ideas. I appreciate their patience in countless hours of meetings, talking through challenging problems with me. I would also like to thank my friends and colleagues in the Centre for Doctoral Training in Theory and Simulation of Materials for many helpful discussions. Lastly, I would like to thank my family for their encouragement and support over the last four years, especially my wife Katie. I would not have finished this thesis without her.

List of Abbreviations

CG	coarse–grained
DFT	density functional theory
IBI	iterative Boltzmann inversion
IMC	inverse Monte Carlo
MD	molecular dynamics
MSCG	multiscale coarse–graining

Contents

Abstract **1**

Acknowledgements **3**

List of Abbreviations **4**

1 Introduction **9**

2 An Overview Of Coarse–Graining Methods **15**

2.1 Force Matching 16
2.2 Iterative Boltzmann Inversion 17
2.3 Inverse Monte Carlo 19
2.4 Relative Entropy Minimisation 21
2.5 Ad–Hoc Methods 22
2.6 Summary 23

3 Theory **25**

3.1 Recap of Basic Statistical Mechanics 25
3.1.1 The Hamiltonian 25
3.1.2 The Microcanonical Ensemble 26
3.1.3 Entropy 27
3.1.4 The Canonical Ensemble 27
3.1.5 The Partition Function 30
3.1.6 Partition Function of a Simple Harmonic Oscillator 31
3.1.7 The Pair Correlation Function 33
3.1.8 Potential of Mean Force 35
3.2 Coarse–Graining Theory 35

3.2.1 The CG Partition Function 35
3.2.2 CG Internal and Free Energies 36
3.2.3 Evaluation of CG Partition Function for a Harmonic Potential 38
3.2.4 Ensuring Conjugacy of Positions and Momenta 44
3.2.5 The Selective Integration Method 47

4 A Technical Description of Coarse–Graining Methods 55
4.1 Force Matching 56
4.2 Relative Entropy Minimization 57

5 Molecular Dynamics 63
5.1 Introduction 63
5.2 Integration Schemes 64
5.2.1 The Verlet Algorithm 64
5.2.2 The Velocity Verlet Algorithm 64
5.2.3 The Leapfrog Algorithm 66
5.3 Temperature and Thermostat Algorithms 66
5.3.1 The Berendsen Thermostat 67
5.3.2 The Langevin Thermostat 67
5.3.3 The Nosé–Hoover Thermostat 68
5.4 Periodic Boundary Conditions 69
5.5 The Force Field 70
5.5.1 Nonbonded Terms 71
5.5.2 Bonded Terms 71
5.6 Energy Minimisation Algorithms 76
5.6.1 Steepest Descent 76
5.6.2 Conjugate Gradients 76
5.7 Constraint Algorithms 77
5.7.1 The SHAKE Algorithm 77
5.7.2 The RATTLE Algoritm 80
5.8 Implementation Details 82
5.8.1 Initialising a Simulation 82
5.8.2 Neighbour Lists 82
5.8.3 Cutoff Radii 83
5.9 Calculating Quantities from an MD Simulation 83

5.9.1 Energies 84
5.9.2 Diffusion Constant 84
5.9.3 Pair Correlation Function 85
5.9.4 Pressure 86
5.10 Normal Mode Analysis 88

6 A System of Two Dimers **91**
6.1 The Microscopic System 91
6.2 Coarse Graining at Zero Temperature 94
6.2.1 The Rigid Approach 94
6.2.2 The Relaxed Dimer Approach 95
6.3 Coarse–Graining at Finite Temperature 100
6.4 Kinetic Contribution to Partition Function 105

7 Atomistic Simulation Results **109**
7.1 Buckyballs 109
7.2 Atomistic Simulation Results 111

8 Generating Coarse–Grained Potentials **119**
8.1 The Girifalco Potential 119
8.2 The Rigid Approximation 121
8.3 The Relaxed Approximation 123
8.4 Constrained Pair Dynamical Simulations 124
8.4.1 The Brute Force Approach 125
8.4.2 Generating Free Energy Potentials from Forces 128
8.4.3 Generating Internal Energy Potentials from Free Energy Potentials 131
8.5 Unconstrained Dynamical Simulations 136
8.6 The Entropic Approach 141
8.7 Using Coarse–Grained Potentials in Simulations 146

9 Coarse–Grained Simulation Results **152**
9.1 Inadequacy of Rigid and Relaxed Potentials at High T 152
9.2 Inconsistency Between Free and Internal Energy Potentials . . . 157
9.3 Transferability of CG Potentials 160
9.4 Importance of Many–Body Effects 163
9.5 Entropy of Buckyball Pairs 166

9.6 Computational Efficiency 168

10 Discussion and Conclusion **169**

10.1 Summary 169

10.2 Future Work 173

10.3 Conclusion 174

Bibliography **176**

Appendix A Derivation of the Girifalco Potential **184**

Appendix B The Conjugate Gradients Algorithm **188**

Appendix C Copyright Permissions **193**

Chapter 1

Introduction

For centuries, scientists have been trying to make sense of nature by formulating theories which describe the world we observe. Many of these theories share a striking property: they describe the behaviour of nature in terms of just a small number of collective variables.

Take, for example, a small glass of water. There are on the order of Avogadro's number ($N_{\mathrm{A}} \approx 6.0 \times 10^{23}$) of water molecules inside the glass, each in a constant state of motion. The theory of thermodynamics describes the behaviour of this astronomically large system in terms of just a few collective variables, such as the pressure P, temperature T and volume V of the glass. Similarly, if we want to describe the mechanical properties of the glass of water, we need only to know a few quantities such as its position, velocity, and moment of inertia. This is enough to apply the theory of classical mechanics to the glass: it is not necessary to keep track of the individual state of every single water molecule.

In its most general sense, this kind of reduction from a highly complex model to a simpler one is called *coarse–graining*. Care must be taken to ensure that the essential behaviour of the complex model is preserved and reproducible in the coarse–grained (CG) model.

In this work, we will consider the application of CG methods to one particular area: the theory and simulation of materials using molecular dynamics (MD) [1, 2]. Molecular dynamics is a simulation technique for tracking the motion of particles in a system on the atomic scale. The forces between particles are

defined by the user through a *force field*, usually based on experimental results or some physical intuition about the system. The forces can also come from a quantum mechanical simulation technique such as density functional theory (DFT) . The motion of each particle is then calculated by numerically solving Newton's equations of classical mechanics.

The most common approach used in MD is the *all–atom* description. In this approach, each individual atom in the system is tracked explicitly [3]. Recent increases in computing power and the advent of highly parallel MD codes have allowed the simulation of very large systems over long time scales. For example, simulations of up to 8 billion atoms for over 1 microsecond have recently been performed [4]. Using customised supercomputer architectures, simulations of time periods approaching 1 millisecond have also been performed [5], representing the current state–of–the–art. However, certain length and time scales will always remain inaccessible to all–atom MD simulations. A cubic millimetre of lead alone contains approximately 10^{19} atoms, making investigation of macroscopic length scales completely out of bounds for even the most powerful modern MD simulations.

Clearly, a further level of simplification is needed, without losing all of the atomistic detail by moving to a completely continuum model of the system. This makes MD a perfect target for CG methods. The hope is to develop fast simulation techniques which allow us access to increasingly large length and time scales, without relying solely on the brute force increase in computing power set out by Moore's law [6].

Coarse-graining techniques have already been used to study a diverse set of interesting phenomena, including the propagation of cracks in body-centred cubic crystals [7], the interaction of water molecules with biological implants [8], and the interaction between defects in a solid [9]. They have enabled the study of systems which were previously too large to examine with existing MD techniques, such as the shape and stability of an entire HIV viral capsid [10].

At this point, it would be fair to ask why we would ever need to resort to fully atomistic MD simulations if CG methods are so fast and effective. Unfortunately, any CG method intrinsically involves some loss of information about the original system. A great deal of care must be employed to ensure that the CG model is capable of reproducing the desired properties from the fully atomistic model. In general, it will be impossible to maintain accuracy in every

respect. Instead, specific properties must be chosen for the CG model to match accurately.

The most computationally costly part of an MD simulation is the evaluation of the inter–particle forces, which account for the vast majority of the total simulation time [5]. A useful CG method will therefore focus on speeding up the force evaluation part of the simulation. There are two main ways to achieve this: coarse–graining in time and coarse–graining in space. When coarse–graining in time, the time step used for solving Newton's equations of motion is artificially increased. This means fewer iterations of the costly force evaluation loop in the MD code are needed to simulate the same total time, resulting in computational savings. Algorithms have even been developed which allow different time steps to be used for different components of the simulation cell, allowing fast processes to be captured with a small time step and slower, diffusion-like processes to be captured with a larger time step [11]. Of course, these methods are limited by requiring a small enough time step to ensure that the dynamics of interest are captured accurately by the numerical integration scheme. Typically, atomic vibrational frequencies can range from 10^{12} to 10^{14} Hz, so time steps on the order of 1 femtosecond are traditionally used in order to capture this motion. An alternative method of coarse–graining in time involves accelerating the simulation to focus on points of interest, for example using the techniques of metadynamics [12]. Here, sampling the free energy landscape of the system is accelerated by progressively adding a sequence of Gaussian functions to the potential, discouraging the system from returning to a previously visited state point.

In this work, we will focus on coarse–graining in space. The computational savings here come from a reduction in the number of particles which need to be simulated, a simpler functional form for the inter–particle force field, or a combination of both. The first step in spatial coarse–graining is to group the particles into larger CG units, treating each CG unit as its own discrete object. This reduces the number of degrees of freedom we need to keep track of in the simulation. It is not necessarily obvious how the mapping between atomistic and CG units should be performed: often some physical intuition must be used (see Fig. 1.1). This is a complicated and interesting issue in its own right, but in this work we will adopt the common approach of using the centre of mass of each group of atoms to define the CG mapping. We are then free to concentrate

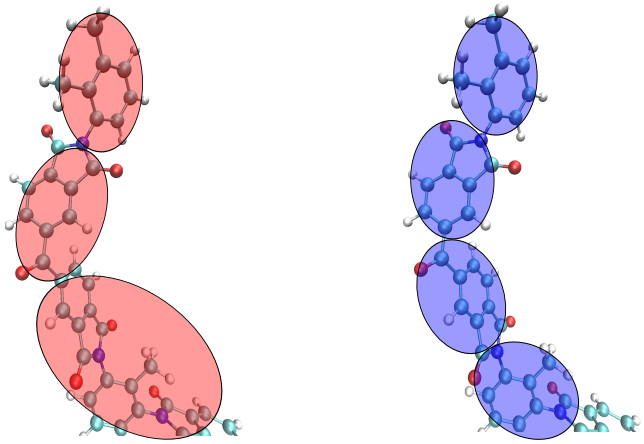


Figure 1.1: A section from a P84 polymer molecule. On the left, a schematic CG representation of the molecule is shown in red, with three CG units. On the right, a different CG representation is shown in blue, with four CG units. A priori, it is not clear which representation is "best". Careful study is required to find out which most accurately reproduces a given property of the underlying polymer.

on the second step in spatial coarse–graining: how to define the effective forces between CG units.

The aim of this thesis is to define and investigate several different methods of generating these forces. We will apply these methods to a system of simple molecules, namely buckminsterfullerene (C_{60}). Each method will be benchmarked against a reference fully atomistic MD simulation of a box of C_{60}. C_{60} is an ideal molecule for testing CG methods, for several reasons. Firstly, its spherical symmetry makes the centre of mass the obvious location of the CG unit. Secondly, there are only 60 atoms in a C_{60} molecule, which makes it easy to perform the reference fully atomistic MD simulations. C_{60} also has a high degree of symmetry, which simplifies issues related to orientation which we would encounter in more complex molecules. We will begin with some very simple CG models, and build up increasing layers of complexity by gradually incorporating more information about the system. The aim is to discover which CG methods produce the most accurate values of a given property, taking the value from the reference fully atomistic MD simulation as "correct". Hopefully, an optimal trade-off between model complexity, performance and accuracy can be found.

The structure of this thesis is as follows. In Chapter 2, we will introduce some of the most widely used coarse–graining methods in the field of molecular dynamics. We will discuss some common applications of these methods, and summarise the strengths and weaknesses of each one.

In Chapter 3, we will review the parts of statistical mechanics which are necessary for understanding the theory of coarse–graining. We then describe the modifications to this theory that are used in CG methods, such as the CG partition function. We will demonstrate how to evaluate the CG partition function analytically for a system which can be described by a harmonic interaction potential. We present a new derivation of this result, which considerably simplifies the mathematics and removes some challenging technical obstacles. We also include a new method for ensuring the positions and momenta in a CG simulation remain properly conjugate to each other. This subtlety has not been analysed closely before.

Once the necessary theory has been introduced, in Chapter 4 we revisit the methods described in Chapter 2 and discuss them in more mathematical detail.

In Chapter 5, we describe the algorithms necessary to run a MD simulation. We discuss practical implementation details, and how to calculate various statistical quantities from a successful simulation. We also discuss related topics which are not directly part of MD, such as energy minimisation algorithms and how to calculate the normal modes of a molecule.

In Chapter 6, we consider a system of two interacting dimers in one dimension. This system is simple enough that we can solve the equations of coarse–graining analytically. This gives us formulae for quantities such as the CG internal energy, free energy and entropy. It provides an informative example of common behaviour that is seen when applying CG methods to more complex and realistic systems.

In Chapter 7, we present the results of fully atomistic MD simulations performed on a system of C_{60} molecules. We will use the results of these calculations as a benchmark to compare our CG simulations against.

In Chapter 8, we present a description of several new techniques for generating CG potentials. We apply these techniques to C_{60}, but the chapter serves as an outline for how to extend the techniques to more complex systems. We begin with the conceptually simplest CG methods, which also happen to be the easiest to implement practically. These are the rigid approximation, where

intramolecular motion is restricted, and the relaxed approximation, where we use information from energy minimised structures to generate the potential. We then move on to methods which are more difficult and time consuming to implement, but provide much better quantitative agreement with benchmark simulations. The first of these is the constrained pair approach, which uses dynamical simulations of molecules constrained at a fixed separation. The next is the unconstrained approach, which takes information from a large MD simulation of many unconstrained molecules. Finally, we discuss a method which we have called the entropic approach, which can generate CG potentials in a fraction of the time of some other methods.

In Chapter 9, we compare these different CG techniques by performing simulations of C_{60} using the various CG potentials. We generate quantities including phase diagrams, diffusion constants and correlation functions for comparison against the reference fully atomistic MD simulations. We discuss the relative merits and drawbacks of each method. We then conclude with some discussion and final remarks.

Chapter 2

An Overview Of Coarse–Graining Methods

In this chapter, we will give a general overview of several of the most popular coarse–graining techniques used in MD simulations, including their benefits, drawbacks and areas of application. This chapter will be quite descriptive in nature. Once the necessary technical concepts have been introduced later in this thesis, we will return to a more technical description of these CG methods in Chapter 4.

The existing body of literature on coarse–graining techniques in materials simulation is very large. By necessity, we will restrict our discussion to techniques which are directly applicable to MD simulations. Many CG techniques can be grouped together into categories which follow a similar philosophy. We will attempt to provide an overview of these categories, together with some representative examples of each.

The field of biomolecular modelling provides a particularly well–developed set of CG methodologies. CG methods are often the only practical way of studying large biomolecules. Careful attempts have been made to put these methods on a rigorous theoretical footing, since we must have confidence in the CG models we are using without necessarily being able to perform the equivalent fully atomistic simulation as a comparison. Most of the examples we will mention in this chapter come from this area, and there are already several review articles on coarse–graining in biomolecular simulation [13–16].

2.1 Force Matching

A large class of coarse–graining techniques fall under the title of *force matching*. In these methods, the underlying atomic forces are used to derive the forces acting on CG particles. These CG forces are then used to fit parameters in a CG force field.

The most prominent of these techniques is the so–called multiscale coarse–graining (MS–CG) method [17–22]. In this method, a fully atomistic MD simulation of a system is performed, and snapshots of the forces and positions of each atom are stored at regular time intervals. The atoms in the simulation are divided into groups, and the centre of mass of each group is taken to be the location of a CG particle. The force on each CG particle f_{CG} is then simply the sum of the forces acting on the individual atoms that constitute that CG particle. These are known as the reference forces. The problem now is to find a functional form for the CG force field which agrees as closely as possible with the reference forces.

In [17] and [18], the authors take the simple approach of representing the CG force field as a set of pair potentials made up of piecewise cubic spline functions. To do this, the distance between CG particles is represented on a grid of points r_k, with f_k and f_k'' the unknown force and second derivative of force at each grid point. The CG force can then be written as

$$f_{\mathrm{CG}}(r, r_k, f_k, f_k'') = A(r, r_k)f_i + B(r, r_k)f_{i+1} + C(r, r_k)f_i'' + D(r, r_k)f_{i+1}'' \quad (2.1)$$

for $r \in [r_i, r_{i+1}]$, where A, B, C and D are known cubic spline functions. The key is then to find the values of f_k and f_k'' which minimise the difference between the CG force field and the reference forces, in a least squares sense. Providing that enough snapshots of the atomistic simulation are used, this forms an overdetermined system of linear equations which can be solved by a method such as QR decomposition to find the optimal CG force field parameters.

In [19–22], the authors take a slightly different approach by choosing a set of basis functions $u_d(x)$ to represent the CG force, so that it can be written as a basis expansion,

$$f_{\mathrm{CG}}(r) = \sum_d \phi_d u_d(x). \quad (2.2)$$

The problem is then to find the set of basis coefficients ϕ_d which again minimise the difference between the CG force field and the reference forces.

There are several problems with this force matching approach. The first is that the system of linear equations to be solved can be extremely large. This leads the authors to use various approximations to find the optimal solution, such as taking small subsets of atomistic simulation snapshots and performing a block averaging procedure.

The second problem is that because molecules repel each other at short distances, these distances will be very poorly sampled unless an extremely large number of atomistic snapshots are used in the least squares fit. One solution to this problem is to calculate the potential at distances as close as is practically possible, and then fit a sensible functional form to the repulsive wall. Linear, quadratic or exponential functions are often used for this purpose.

Force matching methods such as MS–CG have been used to simulate a multitude of different molecular systems. The earliest applications include one–site CG models of water [17] and methane [18]. In these models, the molecules are represented by a single CG point located at the centre of mass. In both cases, structural properties such as the pair correlation function are reproduced very well by the CG force field. Other properties tested include the diffusion constant, specific heat capacity and thermal conductivity, all of which are in poor agreement with the reference MD simulation. This is a common situation with CG simulations, which we will see repeated later.

The method has also been applied to more complex systems such as a lipid bilayer surrounded by water [23], peptides [24] and carbonaceous nanoparticles [25].

2.2 Iterative Boltzmann Inversion

The force matching method makes use of the underlying atomistic forces to generate a CG potential. This is often described as a "bottom up" approach to coarse–graining. There is another class of techniques often referred to as "top down", which takes material properties measured from experiment or calculated from simulations, and tries to find a CG potential which reproduces these properties as closely as possible. Often, distribution functions such as the pair correlation function $g(r)$ (see Section 3.1.7) are used for this purpose, but in principle any other distribution function such as angular or dihedral can be used.

A popular example of the top down approach is known as iterative Boltzmann inversion (IBI) [26, 27]. IBI is guaranteed to generate a CG potential which exactly reproduces the input distribution function when it is used to perform a CG simulation.

The method works as follows. As a first approximation to $V(r)$, the potential of mean force (see Section 3.1.8) is used as a trial function,

$$V_0(r) = -k_{\mathrm{B}}T \ln g(r). \tag{2.3}$$

A simulation is run with $V_0(r)$ as the potential. This simulation will yield a modified pair correlation function, $g_0(r)$. The potential can then be iterated according to the formula

$$V_{i+1}(r) = V_i(r) - k_{\mathrm{B}}T \ln \frac{g_i(r)}{g(r)}. \tag{2.4}$$

Clearly, the potential has converged once $g_i(r) = g(r)$, so that the final potential is guaranteed to reproduce the correct pair correlation function.

There are some limitations with this method. Firstly, the pair correlation functions generated at each iteration will not be smooth because of statistical noise. This will create a potential which is also not smooth when Eqn. (2.4) is applied, leading to energy conservation problems when it is used in a simulation. This can be avoided by applying a Gaussian smoothing function or fitting a smooth cubic spline, but will also lead to modifications in the potential function. A second issue is that the pair potentials generated will have more than one minimum, which is necessary to reproduce the structure in $g(r)$ correctly. These potentials are unphysical, and may only be suitable for reproducing $g(r)$. There is no guarantee that IBI potentials will predict any material property other than the input distribution function correctly, which greatly limits their applicability and usefulness.

Nevertheless, the IBI method has been used to simulate various different phenomena, including protein folding [28] and the behaviour of cellulose fibres in water [29]. The authors of the original paper test their methodology on a polyisoprene polymer molecule [26]. They find that distribution functions generated with the IBI potential are in very good agreement with the atomistic simulations, as is to be expected. They do not test any other properties with their model, but it is unlikely that their model would give quantitative predictions of these properties.

2.3 Inverse Monte Carlo

The inverse Monte Carlo (IMC) technique, sometimes also known as reverse Monte Carlo, is a top down approach similar to IBI. It again attempts to reconstruct effective interaction potentials for coarse-grained systems from a known radial distribution function which is calculated from the full system [30–35].

The method is most easily applied to a system of particles interacting via a pair potential only, so that the total potential energy of the system can be written as a sum of pair terms,

$$V_{\text{total}} = \sum_{j>i} V(r_{ij}), \tag{2.5}$$

where r_{ij} is the distance between particle i and j. Let us assume the pair potential V is defined from 0 up to a cut–off radius r_{cut}, and is 0 otherwise. The potential is then discretised as a grid of M points with positions given by

$$r_\alpha = \alpha \frac{r_{\text{cut}}}{M} \text{ for } \alpha = 1, \ldots, M. \tag{2.6}$$

The true potential at position r is then approximated by $V(r_\alpha)$, for the value of α such that

$$|r - r_\alpha| < \frac{1}{2} \frac{r_{\text{cut}}}{M}. \tag{2.7}$$

The total energy in Eqn. (2.5) can then be estimated by

$$V_{\text{total}} = \sum_\alpha V_\alpha S_\alpha, \tag{2.8}$$

where S_α is the number of particle pairs with distances found inside the αth slice. By comparison with Eqn. (5.66), it can be seen that the average of S_α over a simulation run forms an estimator for the pair correlation function,

$$\langle S_\alpha \rangle = \frac{4\pi r^2 g(r) N^2}{2V}, \tag{2.9}$$

where the factor of 2 accounts for the fact that each pair consists of two particles. Now the average number of pairs in each slice is clearly a function of the value of the potential at each discrete point, $S_\alpha = S_\alpha(\{V_\alpha\}_{\alpha=1,M})$, so the total derivative of $\langle S_\alpha \rangle$ can be written as

$$d\langle S_\alpha \rangle = \sum_\gamma \frac{\partial \langle S_\alpha \rangle}{\partial V_\gamma} dV_\gamma + O(dV^2). \tag{2.10}$$

The partial derivatives here can be calculated exactly using the formula for a thermodynamic expectation value,

$$\frac{\partial \langle S_\alpha \rangle}{\partial V_\gamma} = \frac{\partial}{\partial V_\gamma} \left\{ \frac{\int dq S_\alpha(q) \exp\left[-\beta \sum_\alpha V_\alpha S_\alpha(q)\right]}{\int dq \exp\left[-\beta \sum_\alpha V_\alpha S_\alpha(q)\right]} \right\} = \frac{\partial}{\partial V_\gamma} \left\{ \frac{g}{h} \right\} \tag{2.11}$$

Using the quotient rule,

$$\begin{aligned} g' = \frac{\partial g}{\partial V_\gamma} &= -\beta \int dq S_\alpha S_\gamma \exp\left[-\beta \sum_\alpha V_\alpha S_\alpha(q)\right] = -\beta Z \langle S_\alpha S_\gamma \rangle, \\ h' = \frac{\partial h}{\partial V_\gamma} &= -\beta \int dq S_\gamma \exp\left[-\beta \sum_\alpha V_\alpha S_\alpha(q)\right] = -\beta Z \langle S_\gamma \rangle, \end{aligned} \tag{2.12}$$

where Z is the partition function (see Section 3.1.5). Therefore,

$$\frac{\partial \langle S_\alpha \rangle}{\partial V_\gamma} = \frac{g'h - gh'}{h^2} = -\beta \left(\langle S_\alpha S_\gamma \rangle - \langle S_\alpha \rangle \langle S_\gamma \rangle \right). \tag{2.13}$$

Now we have everything necessary to iteratively find a potential which will reproduce the pair correlation exactly. Starting from an initial trial potential $V_\alpha^{(0)}$, which could for example be the potential of mean force

$$V_\alpha^{(0)} = -k_\mathrm{B} T \ln g(r_\alpha), \tag{2.14}$$

we carry out an MD simulation with this trial potential and calculate the average pair numbers $\langle S_\alpha \rangle^{(0)}$ in each slice. We know the "exact" values of S_α from the pair correlation function, and we use these as reference values S_α^*. We can then work out the deviations of our trial values from the reference values using

$$d\langle S_\alpha \rangle = \langle S_\alpha \rangle^{(0)} - S_\alpha^*. \tag{2.15}$$

Neglecting $O(\Delta V^2)$ terms from Eqn. (2.10), and making use of Eqn. (2.13), we are left with a system of linear equations which we can solve for the dV_γ. Using these, we can generate a next approximation to the potential,

$$V_\alpha^{(1)} = V_\alpha^{(0)} + dV_\alpha^{(0)}. \tag{2.16}$$

This procedure can be iterated, and clearly once the trial values of $\langle S_\alpha \rangle$ match the reference values, the potential has converged and will reproduce the correct pair correlation function.

The IMC method is therefore very similar to IBI, albeit slightly more difficult to implement. IMC has also been applied to a wide variety of systems. In [35], a one–site CG model of water is developed, with the CG site located at the position of the oxygen atom in each molecule. This model provides very good agreement with atomistic simulations in the case of pair correlation functions, but badly overestimates the pressure.

2.4 Relative Entropy Minimisation

A considerably different approach to coarse–graining from the top down and bottom up approaches described above is to take ideas from information theory, and apply them to generate a CG potential. It is possible to define a quantity called the *relative entropy*, which measures the amount of information about the system which is lost when we perform coarse–graining. We can attempt to find a CG potential which retains as much information as possible about the system, thus minimising the relative entropy. This type of technique is therefore known as relative entropy minimisation [36, 37].

The relative entropy can loosely be defined as

$$S_{\text{rel}} = \sum_i p(i) \ln \frac{p(i)}{p_{\text{CG}}(i)}, \tag{2.17}$$

where the probability of being in state i is given by $p(i)$ and $p_{\text{CG}}(i)$ in the atomistic and CG system respectively. Assuming the CG potential depends on a set of parameters λ_k, the authors minimise S_{rel} with respect to each λ_i using a method such as Newton–Raphson iteration.

In [36], the method is applied to find the parameters ϵ and σ in a Lennard–Jones potential which best match the properties of extended simple point charge (SPC/E) water. The optimal parameters depend on the temperature and density of the original water simulation. The authors find that the quantity S_{rel} is a good indicator of the deviation of water's behaviour from the simple Lennard

–Jones fluid. In [38], optimal parameters for a soft sphere potential of liquids are found using a relative entropy approach. The approach has also been used to study proteins [39] and peptides [40]. Relative entropy minimisation is a relatively new CG technique which has only been applied in a few areas, but shows great promise in the near future.

2.5 Ad–Hoc Methods

Many coarse-graining models are based entirely around an empirical description of a system, and make no attempt to justify the theoretical validity of the method. Nevertheless, they can produce fast simulations from which useful qualitative, if not quantitative data can be extracted.

One such example is the Adaptive Resolution (AdResS) method due to Praprotnik et al. [41] (see Fig. 2.1). This method applies to systems of interacting molecules. Some molecules are treated fully atomistically, with both inter- and intra-molecular interactions, and some molecules are treated as a single coarse-grained unit. At the interface between fully atomistic and coarse-grained regions, there is a hybrid region where molecules are represented by a mixture of the atomistic and coarse-grained descriptions. The molecular resolution is allowed to vary spatially, hence "adaptive" resolution.

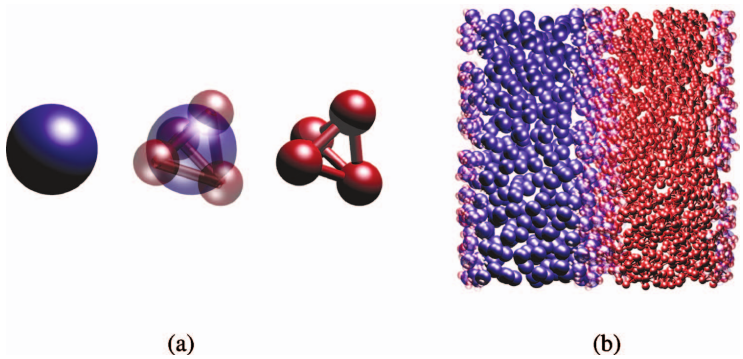


Figure 2.1: (a) CG, hybrid and atomistic representations of a tetrahedral molecule in the AdResS method. (b) Snapshot of the hybrid simulation. Molecules with fully atomistic degrees of freedom are shown in red, and CG molecules are shown in blue. The hybrid region can be seen at the interface between red and blue. Reproduced from [41] with permission.

As an example, the authors consider a system of n tetrahedral molecules, each composed of $N = 4$ atoms of mass m. The forces between molecules are calculated from a Lennard-Jones potential, and the forces between atoms in a molecule use a finitely extensible nonlinear elastic (FENE) potential.

Each coarse-grained molecule has a mass $M = 4m$. An effective pair potential for the coarse-grained molecules is obtained using a potential of mean force (see Section 3.1.8) derived from the radial distribution function of the system, $U(r) \approx PMF(r) = -k_B T \log \rho(r)$.

In the hybrid region, each hybrid molecule is treated as a tetrahedral molecule with an additional massless centre-of-mass particle. As a coarse-grained particle passes into the hybrid region, 4 tetrahedral atoms are spawned with random orientations relative to the centre-of-mass particle. Each of the tetrahedral atoms is given the same velocity, to preserve linear momentum when moving from the CG to hybrid region.

As a hybrid molecule passes into the atomistic region, the positions and velocities of the tetrahedral atoms are preserved. Once the molecule is sufficiently far away from the hybrid region, the centre-of-mass coordinates may be dropped. The procedure works in reverse for molecules passing through atomistic $\rightarrow$ hybrid $\rightarrow$ CG regions.

The authors go on to define an effective force for the hybrid region, which is an interpolation of the CG and fully atomistic forces. This force depends on a weighting function which is forced to be differentiable, continuous and between 0 and 1.

2.6 Summary

There are many other CG techniques, although those listed above represent the most commonly used methods. Software toolkits exist which implement some of the techniques above [42], although they remain difficult to use and have not been extensively validated. We include a summary table of the methods outlined in this chapter in Table 2.1 for reference.

CG method	Strengths	Weaknesses
Force matching	Not restricted to matching distribution functions. Can predict several quantities.	Not transferable to different temperatures and densities. Linear equations produced are difficult to solve. Sampling issues at small atomic separations.
Iterative Boltzmann inversion	Accurately reproduces reference distribution functions and related structural properties.	Not guaranteed to reproduce properties other than the pair-correlation function. Numerical smoothing required which may alter dynamics. Not transferable.
Inverse Monte Carlo	See IBI.	See IBI.
Relative entropy minimisation	Provides natural metric for quality of CG model.	Calculating S_{rel} is computationally intensive. Recent method, further development needed.
Ad–hoc Methods	Can be made to fit experiment closely. Makes use of physical insight about system.	No underlying theory so no general methodology. Each new system must be considered separately.

Table 2.1: Summary of CG techniques referred to in this chapter.

Chapter 3

Theory

3.1 Recap of Basic Statistical Mechanics

Before we dive into a discussion of coarse–graining theory, it is useful to revisit some of the concepts of statistical mechanics. Many of these concepts, when generalised slightly, will lead to the equations of coarse-graining. Much of the material in this section is covered in the excellent books [43] and [44].

3.1.1 The Hamiltonian

Consider a system of n particles with positions $\mathbf{q} = (\mathbf{q}_1, \ldots, \mathbf{q}_n)$ and corresponding momenta $\mathbf{p} = (\mathbf{p}_1, \ldots, \mathbf{p}_n)$. We can define a function $H(\mathbf{q}, \mathbf{p}, t)$, and we call it a *Hamiltonian* if it satisfies *Hamilton's equations*,

$$\frac{d\mathbf{p}}{dt} = -\frac{\partial H}{\partial \mathbf{q}}, \tag{3.1}$$

$$\frac{d\mathbf{q}}{dt} = +\frac{\partial H}{\partial \mathbf{p}}. \tag{3.2}$$

The total derivative of H with respect to t is given by

$$\frac{dH}{dt} = \frac{\partial H}{\partial \mathbf{p}}\frac{d\mathbf{p}}{dt} + \frac{\partial H}{\partial \mathbf{q}}\frac{d\mathbf{q}}{dt} + \frac{\partial H}{\partial t}. \tag{3.3}$$

Assuming the Hamiltonian has no explicit time dependence, that is $H(\mathbf{p}, \mathbf{q}, t) = H(\mathbf{p}, \mathbf{q})$, we find $\frac{\partial H}{\partial t} = 0$. Inserting Hamilton's equations, the total derivative becomes

$$\frac{dH}{dt} = -\frac{\partial H}{\partial \mathbf{p}}\frac{\partial H}{\partial \mathbf{q}} + \frac{\partial H}{\partial \mathbf{q}}\frac{\partial H}{\partial \mathbf{p}} + 0 = 0, \tag{3.4}$$

which implies H is a constant of motion,

$$H(\mathbf{q}, \mathbf{p}) = E. \tag{3.5}$$

E is then defined to be the *total energy* of the system. For most physical systems, the Hamiltonian is separable, that is

$$H(\mathbf{q}, \mathbf{p}) = T(\mathbf{p}) + V(\mathbf{q}) = E, \tag{3.6}$$

where $T(\mathbf{p})$ is the *kinetic energy*, a function of the particle momenta only, and $V(\mathbf{q})$ is the *potential energy*, a function of the particle positions only.

3.1.2 The Microcanonical Ensemble

Returning to our system of n particles with positions $\mathbf{q} = (\mathbf{q}_1, \ldots, \mathbf{q}_n)$ and momenta $\mathbf{p} = (\mathbf{p}_1, \ldots, \mathbf{p}_n)$, we can see that $6n$ numbers ($3n$ position vector components and $3n$ momentum vector components) are needed to specify the state of the system completely. We say that the system has $6n$ *degrees of freedom*. The $6n$–dimensional space of possible positions and momenta is called *phase space*, and a particular point in phase space, corresponding to a particular configuration of the system, is known as a *microstate* of the system. In a system with discrete microstates which are countable, the number of microstates is typically denoted by Ω. In real systems, the number of particles n is typically of order Avogadro's number $N_A \approx 10^{23}$ particles, making the phase space exceedingly large. Because of this, it is impractical to expect to specify the state of a system exactly, and instead we must turn to probabilistic arguments to draw useful conclusions about the system.

Imagine that our system of n particles is confined within a fixed volume V. We call it an *isolated system* if it is unable to exchange matter or energy with its surroundings, in which case the total energy of the system is constant. An infinite number of copies of this system, each with fixed n, V and E but with differing microscopic configurations of the particles is then known as the *microcanonical ensemble* (often also called the *NVE ensemble*). We say the system is in *equilibrium* if the probability of finding the system in any one state

is independent of time. We then introduce the following *postulate of equal a priori probabilities*:

An isolated system in equilibrium is equally likely to be in any of its microstates.

This implies that if our system is in equilibrium, any microstate picked from the NVE ensemble is equally likely to occur.

3.1.3 Entropy

Before proceeding further, let us define a quantity called *entropy*,

$$S = k_B \log \Omega, \tag{3.7}$$

where k_B is known as *Boltzmann's constant*, which is 1.3807×10^{-23}J/K in SI units. The entropy is of fundamental importance in statistical mechanics, and can be thought of as a measure of the information content of a system. One important property of the entropy is that it is additive (or *extensive*). If we imagine that two isolated systems A and B, with number of microstates Ω_A and Ω_B respectively, are brought into contact, the total number of microstates is clearly multiplicative, $\Omega = \Omega_A \Omega_B$. Our choice of the logarithm function for S then implies the entropy is properly additive,

$$S = k_B \log \Omega = k_b B \log(\Omega_A \Omega_B) = k_B \log \Omega_A + k_B \log \Omega_B = S_A + S_B. \tag{3.8}$$

The prefactor k_B is necessary to make our definition of entropy consistent with the Kelvin scale of temperature. If temperatures were instead expressed as thermal energies, we could drop the prefactor to make entropy dimensionless, recovering the definition of entropy from information theory.

3.1.4 The Canonical Ensemble

In practice, it is hard to make measurements from the NVE ensemble since it is difficult to keep the total energy in an experiment fixed. It is often much easier to keep the temperature fixed, in which case we say we are in the *canonical ensemble* (often also called the *NVT ensemble*). To derive the probability of being in a particular microstate in the NVT ensemble, consider a large isolated system A in the NVE ensemble. Imagine a small subsystem A_{sub} embedded

inside A. The remainder of A which surrounds A_{sub} acts as a heat bath A_{bath}. The probability P_j of the subsystem being in state j is equal to the fraction of the number of states of the entire system in which the subsystem is in state j,

$$P_j = \frac{\Omega_{\text{bath}}(E - E_j)}{\Omega(E)}. \tag{3.9}$$

Using our definition of entropy, this can be rewritten as

$$P_j = \frac{\exp\left[k_B^{-1} S_{\text{bath}}(E - E_j)\right]}{\exp\left[k_B^{-1} S(E)\right]}. \tag{3.10}$$

Let U be the average energy of the subsystem. Then additivity of entropy implies

$$S_{\text{total}}(E) = S(U) + S_{\text{bath}}(E - U). \tag{3.11}$$

Expanding $S_{\text{bath}}(E - E_j)$ as a Taylor series, we find

$$\begin{aligned} S_{\text{bath}}(E - E_j) &= S_{\text{bath}}(E - U + U - E_j) \\ &= S_{\text{bath}}(E - U) + \frac{\partial S_{\text{bath}}}{\partial E_{\text{bath}}}(U - E_j). \end{aligned} \tag{3.12}$$

Since the energy of the subsystem is tiny compared to the total energy (this being the definition of a resevoir), higher order terms can safely be neglected. From basic thermodynamics (see e.g. [44]), the above partial derivative can be identified with the inverse temperature of the heat bath, $1/T_{\text{bath}}$. So we have

$$S_{\text{bath}}(E - E_j) = S_{\text{bath}}(E - U) + \frac{1}{T_{\text{bath}}}(U - E_j). \tag{3.13}$$

Inserting Eqn. (3.11) and Eqn. (3.13) into Eqn. (3.10), we find

$$P_j = \exp\left[\beta(U - TS(U))\right] \exp(-\beta E_j), \tag{3.14}$$

where $\beta = (k_B T)^{-1}$. Since the probabilities must sum to unity, we find

$$\sum_j P_j = \exp\left[\beta(U - TS(U))\right] \sum_j \exp(-\beta E_j) = 1, \tag{3.15}$$

which implies that

$$\exp\left[-\beta(U - TS(U))\right] = \sum_j \exp(-\beta E_j). \tag{3.16}$$

The right-hand side of this equation is known as the *partition function* Z,

$$Z = \sum_j \exp(-\beta E_j). \tag{3.17}$$

The symbol Z is from the German *zustandssumme*, which means "sum over states", for obvious reasons. If we make the definition

$$F = U - TS(U), \tag{3.18}$$

where F is known as the *Helmholtz free energy*, Eqn. (3.16) becomes

$$\exp(-\beta F) = Z, \tag{3.19}$$

or

$$F = -k_B T \ln Z. \tag{3.20}$$

From Eqn. (3.14), we now have a useful equation for the probability of being in a given microstate:

$$P_j = \frac{\exp(-\beta E_j)}{Z}. \tag{3.21}$$

From this formula, it is easy to see that Z acts as a normalising factor on the probability. If we want to know the average value of some observable quantity O, denoted $\langle O \rangle$, assuming it takes a value O_j for microstate j, this is now easy to compute from the formula

$$\langle O \rangle = \sum_j O_j P_j = \frac{\sum_j O_j \exp(-\beta E_j)}{Z}. \tag{3.22}$$

In the particular case where $O = E$, the internal energy of the system, we find

$$\langle E \rangle = \frac{\sum_j E_j \exp(-\beta E_j)}{\sum_j \exp(-\beta E_j)} = -\frac{\partial \ln Z}{\partial \beta}. \tag{3.23}$$

The average energy is usually given the symbol U, so that

$$U = \langle E \rangle = -\frac{\partial \ln Z}{\partial \beta}. \tag{3.24}$$

3.1.5 The Partition Function

In the above derivation of the canonical ensemble, it was assumed throughout that each microstate j can be assigned to a particular discrete energy level E_j. This is completely justified in a quantum view of the world, and hence the above partition function is often known as the *quantum partition function*. However, in the world of classical mechanics which we are concerned with in this work, the energy of the system is continuous and the above discrete picture makes no sense. We can, however, make some modifications to the quantum partition function which will allow progress to be made.

Recall our system of n particles with positions $\mathbf{q} = (\mathbf{q}_1, \ldots, \mathbf{q}_n)$ and momenta $\mathbf{p} = (\mathbf{p}_1, \ldots, \mathbf{p}_n)$. We can then define a volume element in phase space by

$$d\mathbf{p}d\mathbf{q} = d\mathbf{p}_1 \ldots d\mathbf{p}_n d\mathbf{q}_1 \ldots d\mathbf{q}_n, \tag{3.25}$$

a $6n$–dimensional hypercube in phase space. We can then say that a microstate of the system with a coordinate in the range $(\mathbf{p}+\delta\mathbf{p}, \mathbf{q}+\delta\mathbf{q})$ is located within a particular cell in phase space. Let us assume that a particular two–dimensional "face" of this hypercube $dp_i^{(x)}dq_i^{(x)}$ has an "area" of h_0, where h_0 is some tiny number with units of angular momentum. Then the volume element has a total volume of h_0^{3n}. This leads us to the definition of the classical partition function,

$$Z = \frac{1}{h_0^{3n}} \int e^{-\beta H(\mathbf{p},\mathbf{q})} d\mathbf{p}d\mathbf{q}, \tag{3.26}$$

where the discrete energy levels E_j have been replaced with the continuous Hamiltonian $H(\mathbf{p}, \mathbf{q})$ and the factor of $1/h_0^{3n}$ makes Z a dimensionless quantity. Recall the Heisenberg uncertainty principle tells us that $\Delta q_j \Delta p_j \geq h$, where h is Planck's constant. This means that we can never know the product of position and momentum to an accuracy less than h, and so it is meaningless to divide phase space into units of size less than h^{3n}. We can therefore equate h_0 with h in the above equation. It is quite common to drop the prefactor of $1/h^{3n}$, leaving the partition function as a dimensioned quantity with units of angular momentum.

Assuming the Hamiltonian is separable, the partition function may be factorised into

$$
\begin{aligned}
Z &= \int e^{-\beta T(\mathbf{p})} d\mathbf{p} \times \int e^{-\beta V(\mathbf{q})} d\mathbf{q} \\
&= Z_{\mathbf{p}} \times Z_{\mathbf{q}}.
\end{aligned} \tag{3.27}
$$

Since $Z_{\mathbf{q}}$ depends on the particle positions only, it is known as the *configurational integral*. The kinetic energy $T(\mathbf{p})$ can be written as

$$
T(\mathbf{p}) = \sum_{i=1}^{n} \frac{\mathbf{p}_i^2}{2m_i}, \tag{3.28}
$$

where m_i is the mass of the ith particle. Then we can write

$$
\begin{aligned}
Z_{\mathbf{p}} &= \int \exp\left(-\beta \sum_{i=1}^{n} \frac{\mathbf{p}_i^2}{2m_i}\right) d\mathbf{p} \\
&= \prod_{i=1}^{n} \int \exp\left(-\frac{\beta \mathbf{p}_i^2}{2m_i}\right) d\mathbf{p}_i \\
&= \prod_{i=1}^{n} \left[\int \exp\left(-\frac{\beta x^2}{2m_i}\right) dx\right]^3.
\end{aligned} \tag{3.29}
$$

Using the substitution $u = (\beta/2m_i)^{1/2} x$, this integral can be reduced to a simple one-dimensional Gaussian integral of the form

$$
\int e^{-x^2} dx = \sqrt{\pi}. \tag{3.30}
$$

The final result becomes

$$
Z_{\mathbf{p}} = \prod_{i=1}^{n} \left(\frac{2\pi m_i}{\beta}\right)^{3/2}. \tag{3.31}
$$

This result is the same for any system of point particles, and hence is easily evaluated. However, the configurational integral $Z_{\mathbf{q}}$ is in general much more difficult to evaluate analytically, because the potential function $V(\mathbf{q})$ is often a complicated function of the particle coordinates which is not separable.

3.1.6 Partition Function of a Simple Harmonic Oscillator

It will be useful later on to consider the evaluation of the classical partition function for a very simple system, a single harmonic oscillator in one dimension.

If the oscillator has a mass m, momentum p, displacement q and spring constant k, the Hamiltonian is

$$H(p,q) = \frac{p^2}{2m} + \frac{1}{2}kq^2. \tag{3.32}$$

The classical partition function is

$$\begin{aligned} Z(p,q) &= \frac{1}{h}\int e^{-\beta H(p,q)}dpdq \\ &= \frac{1}{h}\int \exp\left(-\frac{\beta p^2}{2m}\right)dp \int \exp\left(-\frac{\beta}{2}kq^2\right)dq. \end{aligned} \tag{3.33}$$

Both of these integrals are Gaussian integrals of the form

$$\int e^{-ax^2}dx = \sqrt{\frac{\pi}{a}}, \tag{3.34}$$

so we find

$$\begin{aligned} Z(p,q) &= \frac{1}{h}\sqrt{\frac{2m\pi}{\beta}}\sqrt{\frac{2\pi}{\beta k}} \\ &= \frac{2\pi}{h}\frac{1}{\beta}\sqrt{\frac{m}{k}}. \end{aligned} \tag{3.35}$$

Recall the standard definition $\hbar = h/2\pi$, and the natural frequency $\omega = \sqrt{k/m}$. We arrive at the result

$$Z(p,q) = \frac{1}{\beta\hbar\omega} = \frac{k_{\mathrm{B}}T}{\hbar\omega}. \tag{3.36}$$

Then

$$U = -\frac{\partial \ln Z}{\partial \beta} = \frac{1}{\beta}, \tag{3.37}$$

and

$$F = -\frac{1}{\beta}\ln Z = \frac{\ln(\beta\hbar\omega)}{\beta}. \tag{3.38}$$

From the formula $F = U - TS$, we find

$$
\begin{aligned}
S &= k_{\mathrm{B}}\beta(U-F) \\
&= k_{\mathrm{B}}[1-\ln(\beta\hbar\omega)] \\
&= k_{\mathrm{B}}\left[1+\ln\left(\frac{k_{\mathrm{B}}T}{\hbar\omega}\right)\right].
\end{aligned} \tag{3.39}
$$

So in the high temperature limit,

$$
S \approx k_{\mathrm{B}}\ln\left(\frac{k_{\mathrm{B}}T}{\hbar\omega}\right). \tag{3.40}
$$

3.1.7 The Pair Correlation Function

In a system of N atoms, the potential depends on the position of each atom, $V = V(\mathbf{r}_1,\ldots,\mathbf{r}_N)$, and the configurational partiton function of the system is given by

$$
Z = \int e^{-\beta V(\mathbf{r}_1,\ldots,\mathbf{r}_N)} d\mathbf{r}_1 \ldots d\mathbf{r}_N. \tag{3.41}
$$

Then the probability of finding particle 1 in $d\mathbf{r}_1$, particle 2 in $d\mathbf{r}_2$ and so on is given by

$$
P^{(N)}(\mathbf{r}_1,\ldots,\mathbf{r}_N)d\mathbf{r}_1\ldots d\mathbf{r}_N = Z^{-1}e^{-\beta V(\mathbf{r}_1,\ldots,\mathbf{r}_N)}d\mathbf{r}_1\ldots d\mathbf{r}_N. \tag{3.42}
$$

We would prefer to know the probability of finding n particles fixed at $\mathbf{r}_1,\ldots,\mathbf{r}_n$ with the other $N-n$ particles free to move anywhere in configuration space. This is done simply by integrating Eqn. (3.42) over the degrees of freedom we do not keep fixed, to arrive at

$$
P^{(n)}(\mathbf{r}_1,\ldots,\mathbf{r}_n) = Z^{-1}\int e^{-\beta V(\mathbf{r}_1,\ldots,\mathbf{r}_N)} d\mathbf{r}_{n+1}\ldots d\mathbf{r}_N. \tag{3.43}
$$

The particles are indistinguishable, so it does not matter which is at $\mathbf{r}_1$, which is at $\mathbf{r}_2$ and so on. We therefore define the n–particle probability density as

$$
\rho^{(n)}(\mathbf{r}_1,\ldots,\mathbf{r}_n) = \frac{N!}{(N-n)!}P^{(n)}(\mathbf{r}_1,\ldots,\mathbf{r}_n). \tag{3.44}
$$

For $n=1$, this formula reduces to

$$
\rho^{(1)}(\mathbf{r}_1) = NP^{(1)}(\mathbf{r}_1). \tag{3.45}
$$

For a homogeneous system, $\rho^{(1)}(\mathbf{r}_1)$ is clearly independent of $\mathbf{r}_1$, so that the integral of the left–hand side becomes

$$
\int \rho^{(1)}(\mathbf{r}_1)d\mathbf{r}_1 = \rho^{(1)}\int d\mathbf{r}_1 = \rho^{(1)}V. \tag{3.46}
$$

The integral of the right–hand side becomes

$$N \int P^{(1)}(\mathbf{r}_1) d\mathbf{r}_1 = N. \tag{3.47}$$

Together, these imply that

$$\rho^{(1)} = \frac{N}{V} = \rho, \tag{3.48}$$

which is the familiar formula for the average density. We may now define an n–particle correlation function $g^{(n)}$ as

$$g^{(n)}(\mathbf{r}_1, \ldots, \mathbf{r}_n) = \frac{1}{\rho^n} \rho^{(n)}(\mathbf{r}_1, \ldots, \mathbf{r}_n). \tag{3.49}$$

This is the ratio between the n–particle probability density of the system and the probability density of an entirely homogeneous system. It therefore provides a measure of the correlations between atoms, hence the name correlation function. The correlation function becomes unity when $\rho^{(n)} = \rho^n$, when the atoms are decorrelated.

Substituting Eqn. (3.49) into Eqn. (3.44), we obtain the general formula for the n–particle correlation function,

$$g^{(n)}(\mathbf{r}_1, \ldots, \mathbf{r}_n) = \frac{V^n N!}{N^n (N-n)!} Z^{-1} \int e^{-\beta V(\mathbf{r}_1, \ldots, \mathbf{r}_N)} d\mathbf{r}_{n+1} \ldots d\mathbf{r}_N. \tag{3.50}$$

We are particularly interested in the special case where $n = 2$, which provides us with a measure of the chance of finding a particle at $\mathbf{r}_2$ given that there is a particle at $\mathbf{r}_1$. From Eqn. (3.50), this is given by

$$g^{(2)}(\mathbf{r}_1, \mathbf{r}_2) = \frac{V^2 (N-1)}{N} Z^{-1} \int e^{-\beta V(\mathbf{r}_1, \ldots, \mathbf{r}_N)} d\mathbf{r}_3 \ldots d\mathbf{r}_N. \tag{3.51}$$

This two–particle correlation function is so useful that it is given a special name, the *pair correlation function* (sometimes also known as a radial distribution function). Since $\mathbf{r}_1$ and $\mathbf{r}_2$ appear only in the potential V above, $g(\mathbf{r}_1, \mathbf{r}_2)$ can only depend on the interparticle spacing $r = |\mathbf{r}_2 - \mathbf{r}_1|$, so that $g(\mathbf{r}_1, \mathbf{r}_2) = g(r)$. If we assume that one particle is at the origin, the quantity $\rho g(r)$ is the conditional probability of finding another particle a distance r away. This leads us to the formula

$$\int_0^\infty \rho g(r) 4\pi r^2 dr = N - 1. \tag{3.52}$$

3.1.8 Potential of Mean Force

Let us consider the mean force on particle 1, which is fixed at $\mathbf{r}_1$, when particle 2 is fixed at $\mathbf{r}_2$, but all of the remaining $N-2$ particles are free to explore the thermal equilibrium. The mean force is given by

$$\begin{aligned}
\langle \mathbf{F}_1 \rangle &= -\left\langle \frac{dV}{d\mathbf{r}_1} \right\rangle \\
&= -\left(\int \frac{dV}{d\mathbf{r}_1} e^{-\beta V} d\mathbf{r}_3 \dots d\mathbf{r}_N \right) \Big/ \left(\int e^{-\beta V} d\mathbf{r}_3 \dots d\mathbf{r}_N \right) \\
&= +\frac{1}{\beta} \left(\frac{d}{d\mathbf{r}_1} \int e^{-\beta V} d\mathbf{r}_3 \dots d\mathbf{r}_N \right) \Big/ \left(\int e^{-\beta V} d\mathbf{r}_3 \dots d\mathbf{r}_N \right) \\
&= \frac{1}{\beta} \frac{d}{d\mathbf{r}_1} \ln \left(\int e^{-\beta V} d\mathbf{r}_3 \dots d\mathbf{r}_N \right) \\
&= \frac{1}{\beta} \frac{d}{d\mathbf{r}_1} \ln \left[N(N-1) \left(\int e^{-\beta V} d\mathbf{r}_3 \dots d\mathbf{r}_N \right) \Big/ \left(\int e^{-\beta V} d\mathbf{r} \right) \right] \\
&= \frac{1}{\beta} \frac{d}{d\mathbf{r}_1} \ln g(\mathbf{r}_1, \mathbf{r}_2). \qquad (3.53)
\end{aligned}$$

This shows that the quantity

$$w(r) = -k_{\mathrm{B}} T \ln g(r) \qquad (3.54)$$

is the integral of the mean force. It is therefore known as the *potential of mean force*.

3.2 Coarse–Graining Theory

Let us consider a new set of coarse–grained (CG) coordinates $(\mathbf{Q}, \mathbf{P})$, where $\mathbf{Q} = (\mathbf{Q}_1, \dots, \mathbf{Q}_N)$ and $\mathbf{P} = (\mathbf{P}_1, \dots, \mathbf{P}_N)$. In general, $N < n$ to obtain the reduction in degrees of freedom that we desire in a CG system. We will typically use capital letters to refer to CG quantities and lower case letters to refer to atomistic quantities from now on.

3.2.1 The CG Partition Function

Recall the classical partition function is given by

$$Z = \int e^{-\beta H(\mathbf{p},\mathbf{q})} d\mathbf{p} d\mathbf{q}. \qquad (3.55)$$

We can convert this into a CG partition function using the formula

$$Z_{\mathrm{CG}}(\mathbf{P},\mathbf{Q}) = \int e^{-\beta H(\mathbf{p},\mathbf{q})}\delta(\mathbf{P}-\boldsymbol{\alpha}(\mathbf{p}))\delta(\mathbf{Q}-\boldsymbol{\gamma}(\mathbf{q}))d\mathbf{p}d\mathbf{q}. \tag{3.56}$$

The delta function notation here is shorthand for a product of Dirac delta functions,

$$\begin{aligned}\delta(\mathbf{P}-\boldsymbol{\alpha}(\mathbf{p})) &= \prod_{i=1}^{N}\delta(\mathbf{P_i}-\boldsymbol{\alpha}_i(\mathbf{p})),\\ \delta(\mathbf{Q}-\boldsymbol{\gamma}(\mathbf{q})) &= \prod_{i=1}^{N}\delta(\mathbf{Q_i}-\boldsymbol{\gamma}_i(\mathbf{q})).\end{aligned} \tag{3.57}$$

The functions $\boldsymbol{\alpha}(\mathbf{p}) = (\boldsymbol{\alpha}_1(\mathbf{p}),\ldots,\boldsymbol{\alpha}_N(\mathbf{p}))$ and $\boldsymbol{\gamma}(\mathbf{q}) = (\boldsymbol{\gamma}_1(\mathbf{q}),\ldots,\boldsymbol{\gamma}_N(\mathbf{q}))$ are known as *mapping functions* which map atomistic coordinates onto CG coordinates.

To understand the effect of including these delta functions in the partition function, consider a particular point in CG phase space given by $(\mathbf{Q},\mathbf{P})$, and imagine evaluating the CG partition function at this point. For a given atomistic configuration specified by $(\mathbf{p},\mathbf{q})$, if $\mathbf{P}_i \neq \boldsymbol{\alpha}_i(\mathbf{p})$ or $\mathbf{Q}_i \neq \boldsymbol{\gamma}_i(\mathbf{q})$ for at least one CG coordinate i, the product of delta functions will be zero and so the point $(\mathbf{p},\mathbf{q})$ contributes nothing to the integral. In this way, only regions of atomistic phase space which satisfy our defined mappings will contribute to the CG partition function. We can therefore think of the CG partition function as a constrained integral over atomistic phase space, counting only "sensible" configurations of the particles.

Note that if the Hamiltonian is separable, Eqn. (3.56) becomes

$$\begin{aligned}Z_{\mathrm{CG}}(\mathbf{P},\mathbf{Q}) &= \left[\int e^{-\beta T(\mathbf{p})}\delta(\mathbf{P}-\boldsymbol{\alpha}(\mathbf{p}))d\mathbf{p}\right]\left[\int e^{-\beta V(\mathbf{q})}\delta(\mathbf{Q}-\boldsymbol{\gamma}(\mathbf{q}))d\mathbf{q}\right]\\ &= Z^{\mathbf{P}}_{\mathrm{CG}}(\mathbf{P}) \times Z^{\mathbf{Q}}_{\mathrm{CG}}(\mathbf{Q}),\end{aligned} \tag{3.58}$$

and the CG partition function is also separable.

3.2.2 CG Internal and Free Energies

Several important quantities may now be derived from the CG partition function. Firstly, following Eqn. (3.37), let us define

$$
\begin{aligned}
U_{\mathrm{CG}}(\mathbf{P},\mathbf{Q}) &= -\frac{\partial}{\partial\beta}\left[\ln Z_{\mathrm{CG}}(\mathbf{P},\mathbf{Q})\right] \\
&= -\frac{1}{Z_{\mathrm{CG}}}\frac{\partial Z_{\mathrm{CG}}}{\partial\beta} \\
&= \frac{1}{Z_{\mathrm{CG}}}\int H(\mathbf{p},\mathbf{q})e^{-\beta H(\mathbf{p},\mathbf{q})}\delta(\mathbf{P}-\boldsymbol{\alpha}(\mathbf{p}))\delta(\mathbf{Q}-\boldsymbol{\gamma}(\mathbf{q}))d\mathbf{p}d\mathbf{q}. \quad (3.59)
\end{aligned}
$$

This is the average of $H(\mathbf{p},\mathbf{q})$ over the region of configuration space consistent with the CG coordinates P and Q. Therefore, this definition of U_{CG} corresponds to the familiar notion of an average internal energy. Secondly, following Eqn. (3.38), let us define

$$
F_{\mathrm{CG}}(\mathbf{P},\mathbf{Q}) = -\frac{1}{\beta}\ln Z_{\mathrm{CG}}(\mathbf{P},\mathbf{Q}) \quad (3.60)
$$

as the CG free energy. This function has an important property. Consider an observable $A(\mathbf{P},\mathbf{Q})$ which is a function of the CG coordinates only. If the Hamiltonian describing the CG system is given by $H_{\mathrm{CG}}(\mathbf{P},\mathbf{Q})$, the expectation value of A is given by

$$
\langle A(\mathbf{P},\mathbf{Q})\rangle_{\mathrm{CG}} = \frac{\int A(\mathbf{P},\mathbf{Q})e^{-\beta H_{\mathrm{CG}}(\mathbf{P},\mathbf{Q})}d\mathbf{P}d\mathbf{Q}}{\int e^{-\beta H_{\mathrm{CG}}(\mathbf{P},\mathbf{Q})}d\mathbf{P}d\mathbf{Q}}. \quad (3.61)
$$

If we consider the special case where $H_{\mathrm{CG}} = F_{\mathrm{CG}}$, this becomes

$$
\langle A(\mathbf{P},\mathbf{Q})\rangle_{\mathrm{CG}} = \frac{\int A(\mathbf{P},\mathbf{Q})Z_{\mathrm{CG}}(\mathbf{P},\mathbf{Q})d\mathbf{P}d\mathbf{Q}}{\int Z_{\mathrm{CG}}(\mathbf{P},\mathbf{Q})d\mathbf{P}d\mathbf{Q}}. \quad (3.62)
$$

Considering the numerator,

$$
\begin{aligned}
&\int A(\mathbf{P},\mathbf{Q})Z_{\mathrm{CG}}(\mathbf{P},\mathbf{Q})d\mathbf{P}d\mathbf{Q} \\
&= \int A(\mathbf{P},\mathbf{Q})\left[\int e^{-\beta H(\mathbf{p},\mathbf{q})}\delta(\mathbf{P}-\boldsymbol{\alpha}(\mathbf{p}))\delta(\mathbf{Q}-\boldsymbol{\gamma}(\mathbf{q}))d\mathbf{p}d\mathbf{q}\right]d\mathbf{P}d\mathbf{Q} \\
&= \int\left[\int A(\mathbf{P},\mathbf{Q})\delta(\mathbf{P}-\boldsymbol{\alpha}(\mathbf{p}))\delta(\mathbf{Q}-\boldsymbol{\gamma}(\mathbf{q}))d\mathbf{P}d\mathbf{Q}\right]e^{-\beta H(\mathbf{p},\mathbf{q})}d\mathbf{p}d\mathbf{q} \\
&= \int A(\boldsymbol{\alpha}(\mathbf{p}),\boldsymbol{\gamma}(\mathbf{q}))e^{-\beta H(\mathbf{p},\mathbf{q})}d\mathbf{p}d\mathbf{q} \\
&= \int A'(\mathbf{p},\mathbf{q})e^{-\beta H(\mathbf{p},\mathbf{q})}d\mathbf{p}d\mathbf{q}, \quad (3.63)
\end{aligned}
$$

by swapping the order of integration and making use of the nature of the delta function. Similarly for the denominator,

$$\int Z_{\mathrm{CG}}(\mathbf{P},\mathbf{Q})d\mathbf{P}d\mathbf{Q} = \int e^{-\beta H(\mathbf{p},\mathbf{q})}d\mathbf{p}d\mathbf{q}. \tag{3.64}$$

Taken together, these imply

$$\langle A(\mathbf{P},\mathbf{Q})\rangle_{\mathrm{CG}} = \frac{\int A'(\mathbf{p},\mathbf{q})e^{-\beta H(\mathbf{p},\mathbf{q})}d\mathbf{p}d\mathbf{q}}{\int e^{-\beta H(\mathbf{p},\mathbf{q})}d\mathbf{p}d\mathbf{q}} = \langle A'(\mathbf{p},\mathbf{q})\rangle_{\mathrm{atomistic}}. \tag{3.65}$$

What we have shown is that, if we take the CG Hamiltonian to be the CG free energy in Eqn. (3.60), the expectation value of an observable A is the same whether we calculate it from the CG coordinates alone or the atomistic coordinates alone. This is good news for us, since calculating the expectation value from a CG simulation with many fewer degrees of freedom than an atomistic simulation is likely to be much faster, but is guaranteed to yield the same answer.

The CG free energy in Eqn. (3.60) thus turns out to be of fundamental importance when developing CG potentials. We will discuss this in more detail in Chapter 4.

3.2.3 Evaluation of CG Partition Function for a Harmonic Potential

In general, we cannot evaluate the CG partition function analytically, since the potential term in the Hamiltonian may have a complicated many-body form. However, in the special case that the potential is harmonic, it is possible to obtain an analytic formula for Z_{CG}. This derivation is based on the work of Rudd et al. in [45] and [46]. Let us consider the kinetic part of the CG partition function first, since it is easier to evaluate,

$$Z^{\mathbf{P}}_{\mathrm{CG}}(\mathbf{P}) = \int e^{-\beta T(\mathbf{p})}\delta(\mathbf{P} - \boldsymbol{\alpha}(\mathbf{p}))d\mathbf{p}. \tag{3.66}$$

To make further progress, we must assume that the mapping function $\boldsymbol{\alpha}(\mathbf{p})$ is in fact a linear mapping, so that

$$\boldsymbol{\alpha}(\mathbf{p}) = \boldsymbol{\alpha}\mathbf{p}, \tag{3.67}$$

where $\boldsymbol{\alpha}$ is a $(3N \times 3n)$–dimensional rectangular matrix. This is commonly the case in practice.

For notational convenience, let us begin to consider our CG and atomistic coordinates $\mathbf{P}$ and $\mathbf{p}$ as $3N$– and $3n$–dimensional column vectors, respectively. We will use Roman characters to label components of CG vectors, and Greek characters to label components of atomistic vectors from now on. Let us define a $(3n \times 3n)$–dimensional diagonal mass matrix which has the vector $\mathbf{m} = (m_1, m_1, m_1, \ldots, m_n, m_n, m_n)$ as the diagonal, and an "inverse" mass matrix $\mathbf{m}^{-1}$ which has the vector $(m_1^{-1}, m_1^{-1}, m_1^{-1}, \ldots, m_n^{-1}, m_n^{-1}, m_n^{-1})$ along its diagonal. Then the atomistic kinetic energy can be written as

$$T(\mathbf{p}) = \frac{1}{2}\mathbf{p}^\intercal \mathbf{m}^{-1}\mathbf{p}. \tag{3.68}$$

Using the Fourier representation of the delta function, we can write

$$\begin{aligned}
\delta(\mathbf{P} - \boldsymbol{\alpha}\mathbf{p}) &= \prod_{j=1}^{3N} \delta\left(P_j - \sum_{\mu=1}^{3n} \alpha_{j\mu} p_\mu\right) \\
&= \left(\frac{1}{2\pi}\right)^{3N} \prod_{j=1}^{3N} \int \exp\left[i\lambda_j \left(P_j - \sum_{\mu=1}^{3n} \alpha_{j\mu} p_\mu\right)\right] d\lambda_j \\
&= \left(\frac{1}{2\pi}\right)^{3N} \int d\boldsymbol{\lambda} \exp\left[i\boldsymbol{\lambda}^\intercal(\mathbf{P} - \boldsymbol{\alpha}\mathbf{p})\right]
\end{aligned} \tag{3.69}$$

if $\boldsymbol{\lambda}$ is the column vector $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_{3N})$ and $d\boldsymbol{\lambda} = d\lambda_1 \ldots d\lambda_{3N}$. Combining Eqn. (3.68) and Eqn. (3.69), the CG partition function becomes

$$Z_{\mathrm{CG}}^{\mathbf{P}}(\mathbf{P}) = \left(\frac{1}{2\pi}\right)^{3N} \iint \exp\left(-\frac{\beta}{2}\mathbf{p}^\intercal \mathbf{m}^{-1}\mathbf{p} + i\boldsymbol{\lambda}^\intercal(\mathbf{P} - \boldsymbol{\alpha}\mathbf{p})\right) d\boldsymbol{\lambda} d\mathbf{p}. \tag{3.70}$$

We will perform the integral over $\mathbf{p}$ first, and the integral over $\boldsymbol{\lambda}$ second. We need to get the integral into the form of a Gaussian integral, so let us complete the square on the exponent:

$$\begin{aligned}
&-\frac{\beta}{2}\mathbf{p}^\intercal \mathbf{m}^{-1}\mathbf{p} + i\boldsymbol{\lambda}^\intercal(\mathbf{P} - \boldsymbol{\alpha}\mathbf{p}) \\
&= -\frac{\beta}{2}\left(\mathbf{p} + i\beta^{-1}\mathbf{m}\boldsymbol{\alpha}^\intercal\boldsymbol{\lambda}\right)^\intercal \mathbf{m}^{-1}\left(\mathbf{p} + i\beta^{-1}\mathbf{m}\boldsymbol{\alpha}^\intercal\boldsymbol{\lambda}\right) \\
&- \frac{1}{2\beta}\boldsymbol{\lambda}^\intercal\boldsymbol{\alpha}\mathbf{m}\boldsymbol{\alpha}^\intercal\boldsymbol{\lambda} + i\boldsymbol{\lambda}^\intercal\mathbf{P}.
\end{aligned} \tag{3.71}$$

Let us use the substitution $\mathbf{u} = \beta^{1/2}(\mathbf{p} + i\beta^{-1}\mathbf{m}\boldsymbol{\alpha}^\intercal\boldsymbol{\lambda})$. Then $d\mathbf{u} = \beta^{3n/2} d\mathbf{p}$, and the CG partition function becomes

$$Z^{\mathbf{P}}_{\mathrm{CG}}(\mathbf{P}) = \left(\frac{1}{2\pi}\right)^{3N} \beta^{-3n/2} \int \exp\left(-\frac{1}{2}\mathbf{u}^{\intercal}\mathbf{m}^{-1}\mathbf{u}\right) d\mathbf{u}$$
$$\times \int \exp\left(-\frac{1}{2\beta}\boldsymbol{\lambda}^{\intercal}\boldsymbol{\alpha}\mathbf{m}\boldsymbol{\alpha}^{\intercal}\boldsymbol{\lambda} + i\boldsymbol{\lambda}^{\intercal}\mathbf{P}\right) d\boldsymbol{\lambda}. \tag{3.72}$$

At this point, we make use of the identity

$$\int \exp\left(-\frac{1}{2}\mathbf{x}^{\intercal}\mathbf{A}\mathbf{x}\right) d^n\mathbf{x} = \frac{(2\pi)^{n/2}}{(\det \mathbf{A})^{1/2}} \tag{3.73}$$

for an invertible matrix $\mathbf{A}$ to evaluate the integral over $\mathbf{u}$. The CG partition function becomes

$$Z^{\mathbf{P}}_{\mathrm{CG}}(\mathbf{P}) = \left(\frac{1}{2\pi}\right)^{3N} \frac{(2\pi)^{3n/2}}{(\det \mathbf{m}^{-1})^{1/2}} \beta^{-3n/2} \int \exp\left(-\frac{1}{2\beta}\boldsymbol{\lambda}^{\intercal}\boldsymbol{\alpha}\mathbf{m}\boldsymbol{\alpha}^{\intercal}\boldsymbol{\lambda} + i\boldsymbol{\lambda}^{\intercal}\mathbf{P}\right) d\boldsymbol{\lambda}. \tag{3.74}$$

All that remains is to compute the integral over $\boldsymbol{\lambda}$. Let us introduce a new matrix, called the *CG mass matrix*,

$$\mathbf{M} = \boldsymbol{\alpha}\mathbf{m}\boldsymbol{\alpha}^{\intercal}. \tag{3.75}$$

$\mathbf{M}$ is symmetric, since

$$\mathbf{M}^{\intercal} = (\boldsymbol{\alpha}\mathbf{m}\boldsymbol{\alpha}^{\intercal})^{\intercal} = (\boldsymbol{\alpha}\mathbf{m}^{\intercal}\boldsymbol{\alpha}^{\intercal})^{\intercal} = \boldsymbol{\alpha}\mathbf{m}\boldsymbol{\alpha}^{\intercal} = \mathbf{M}. \tag{3.76}$$

The exponent above can then be rewritten as

$$-\frac{1}{2\beta}\left(\boldsymbol{\lambda}^{\intercal}\mathbf{M}\boldsymbol{\lambda} - 2i\beta\boldsymbol{\lambda}^{\intercal}\mathbf{P}\right)$$
$$= -\frac{1}{2\beta}\left(\boldsymbol{\lambda} - i\beta\mathbf{M}^{-1}\mathbf{P}\right)^{\intercal}\mathbf{M}\left(\boldsymbol{\lambda} - i\beta\mathbf{M}^{-1}\mathbf{P}\right) - \frac{\beta}{2}\mathbf{P}^{\intercal}\mathbf{M}^{-1}\mathbf{P} \tag{3.77}$$

by completing the square. If we let $\mathbf{u} = \beta^{-1/2}(\boldsymbol{\lambda} - i\beta\mathbf{M}^{-1}\mathbf{P})$, the integral reduces to

$$\beta^{3N/2} \exp\left(-\frac{\beta}{2}\mathbf{P}^{\intercal}\mathbf{M}^{-1}\mathbf{P}\right) \int \exp\left(-\mathbf{u}^{\intercal}\mathbf{M}\mathbf{u}\right) d\mathbf{u}$$
$$= \beta^{3N/2} \frac{(2\pi)^{3N/2}}{(\det \mathbf{M}^{-1})^{1/2}} \exp\left(-\frac{\beta}{2}\mathbf{P}^{\intercal}\mathbf{M}^{-1}\mathbf{P}\right). \tag{3.78}$$

The CG partition function is then

$$Z^{\mathbf{P}}_{\mathrm{CG}}(\mathbf{P}) = C_1 \beta^{\frac{3}{2}(N-n)} \exp\left(-\frac{\beta}{2}\mathbf{P}^{\intercal}\mathbf{M}^{-1}\mathbf{P}\right). \tag{3.79}$$

where C_1 is a constant independent of β.

So far, we have only evaluated the kinetic part of the CG partition function. The potential part is more difficult to evaluate, and we must first make some assumptions about the form of the potential to make it possible. Let us assume each atom i with position $\mathbf{x}_i$ has an equilibrium position $\mathbf{x}_i^{\text{eq}}$, so that its position at an arbitrary time can be written as $\mathbf{x}_i = \mathbf{x}_i^{\text{eq}} + \mathbf{q}_i$. This situation corresponds to, for example, a crystal lattice where thermal vibrations cause each atom to be displaced slightly from its lowest energy position. Then, by performing a multidimensional Taylor expansion, the atomistic potential can be written as

$$\begin{aligned} V(x_1, \ldots, x_{3n}) &= V(x_1^{\text{eq}} + q_1, \ldots, x_{3n}^{\text{eq}} + q_{3n}) \\ &= V(x_1^{\text{eq}}, \ldots, x_{3n}^{\text{eq}}) \\ &+ \sum_{i=1}^{3n} \frac{\partial V(x_1^{\text{eq}}, \ldots, x_{3n}^{\text{eq}})}{\partial x_i} q_i \\ &+ \sum_{i=1}^{3n} \sum_{j=1}^{3n} \frac{1}{2} \frac{\partial^2 V(x_1^{\text{eq}}, \ldots, x_{3n}^{\text{eq}})}{\partial x_i \partial x_j} q_i q_j \\ &+ \ldots . \end{aligned} \tag{3.80}$$

If we assume the atomic displacements are small, we may neglect $O(\Delta x^3)$ terms and higher. This is known as the *harmonic approximation*. The first term here, $V(x_1^{\text{eq}}, \ldots, x_{3n}^{\text{eq}})$, is the value of the potential when every atom is at its equilibrium position, a constant which we will denote as V^{eq}. The second term vanishes, since the derivative of the potential (the force) is zero when evaluated at the equilibrium positions. The third term can be written as the matrix product $\frac{1}{2}\mathbf{q}^\intercal \mathbf{k} \mathbf{q}$, where the matrix elements k_{ij} are given by

$$k_{ij} = \frac{\partial^2 V(x_1^{\text{eq}}, \ldots, x_{3n}^{\text{eq}})}{\partial x_i \partial x_j}. \tag{3.81}$$

$\mathbf{k}$ is often called the *dynamical matrix*. The atomistic potential can then be written as

$$V(\mathbf{q}) = V^{\text{eq}} + \frac{1}{2}\mathbf{q}^\intercal \mathbf{k} \mathbf{q}, \tag{3.82}$$

which is now a function of the displacements only. V_{eq} is a constant which has no effect on the atomic forces, so without loss of generality, let us set it to zero.

Recall the potential part of the CG partition function is given by

$$Z_{\mathrm{CG}}^{\mathbf{Q}}(\mathbf{Q}) = \int e^{-\beta V(\mathbf{q})} \delta(\mathbf{Q} - \boldsymbol{\gamma}(\mathbf{q})) d\mathbf{q}. \tag{3.83}$$

Again, the delta function can be written as

$$\delta(\mathbf{Q} - \boldsymbol{\gamma}\mathbf{q}) = \left(\frac{1}{2\pi}\right)^{3N} \int d\boldsymbol{\lambda} \exp\left[i\boldsymbol{\lambda}^{\intercal}(\mathbf{Q} - \boldsymbol{\gamma}\mathbf{q})\right] \tag{3.84}$$

if we assume $\boldsymbol{\gamma}$ is linear. $Z_{\mathrm{CG}}^{\mathbf{Q}}(\mathbf{Q})$ becomes

$$Z_{\mathrm{CG}}^{\mathbf{Q}}(\mathbf{Q}) = \left(\frac{1}{2\pi}\right)^{3N} \int \exp\left(-\frac{\beta}{2}\mathbf{q}^{\intercal}\mathbf{k}\mathbf{q} - i\boldsymbol{\lambda}^{\intercal}\boldsymbol{\gamma}\mathbf{q} + i\boldsymbol{\lambda}^{\intercal}\mathbf{Q}\right) d\mathbf{q} d\boldsymbol{\lambda}. \tag{3.85}$$

By completing the square, the exponent becomes

$$\begin{aligned} &-\frac{\beta}{2}\left[\left(\mathbf{q} + \frac{i}{\beta}\mathbf{k}^{-1}\boldsymbol{\gamma}^{\intercal}\boldsymbol{\lambda}^{\intercal}\right)^{\intercal}\mathbf{k}\left(\mathbf{q} + \frac{i}{\beta}\mathbf{k}^{-1}\boldsymbol{\gamma}^{\intercal}\boldsymbol{\lambda}\right) + \frac{1}{\beta^2}\boldsymbol{\lambda}^{\intercal}\boldsymbol{\gamma}\mathbf{k}^{-1}\boldsymbol{\gamma}^{\intercal}\boldsymbol{\lambda}\right] + i\boldsymbol{\lambda}^{\intercal}\mathbf{Q} \\ &= -\frac{1}{2}\mathbf{u}^{\intercal}\mathbf{k}\mathbf{u} - \frac{1}{2\beta}\boldsymbol{\lambda}^{\intercal}\boldsymbol{\gamma}\mathbf{k}^{-1}\boldsymbol{\gamma}^{\intercal}\boldsymbol{\lambda} + i\boldsymbol{\lambda}^{\intercal}\mathbf{Q} \end{aligned} \tag{3.86}$$

if we use the substitution

$$\begin{aligned} \mathbf{u} &= \beta^{1/2}\left(\mathbf{q} + \frac{i}{\beta}\mathbf{k}^{-1}\boldsymbol{\gamma}^{\intercal}\boldsymbol{\lambda}^{\intercal}\right), \\ d\mathbf{u} &= \beta^{3n/2} d\mathbf{q}. \end{aligned} \tag{3.87}$$

It should be noted that, as written above, the matrix $\mathbf{k}$ is actually singular. This is because three of the eigenvectors of $\mathbf{k}$, corresponding to uniform translations of the atoms in a given direction, have eigenvalues of zero. Thus the inverse $\mathbf{k}^{-1}$ is not well defined. For the moment, we can sidestep this problem by using the regularisation

$$\mathbf{k}^{-1} = \lim_{\epsilon \to 0} (\mathbf{k} + \epsilon \mathbf{I})^{-1}, \tag{3.88}$$

but we will return to this in more detail later. Computing the integral over $\mathbf{u}$, we have

$$Z_{\mathrm{CG}}^{\mathbf{Q}}(\mathbf{Q}) = C_2 \beta^{-3n/2} \int \exp\left(-\frac{1}{2\beta}\boldsymbol{\lambda}^{\intercal}\boldsymbol{\gamma}\mathbf{k}^{-1}\boldsymbol{\gamma}^{\intercal}\boldsymbol{\lambda} + i\boldsymbol{\lambda}^{\intercal}\mathbf{Q}\right) d\boldsymbol{\lambda}, \tag{3.89}$$

where C_2 is another constant independent of β. Let us define a new *CG dynamical matrix*,

$$\mathbf{K}^{-1} = \boldsymbol{\gamma}\mathbf{k}^{-1}\boldsymbol{\gamma}^{\intercal}. \tag{3.90}$$

We are now in exactly the situation of Eqn. (3.74). We can therefore write down the solution of Eqn. (3.89) as

$$Z_{\mathrm{CG}}^{\mathbf{Q}}(\mathbf{Q}) = C_3 \beta^{\frac{3}{2}(N-n)} \exp\left(-\frac{\beta}{2}\mathbf{Q}^{\intercal}\mathbf{K}\mathbf{Q}\right). \tag{3.91}$$

The complete CG partition function is now given by the product of the kinetic and potential parts, Eqn. (3.79) and Eqn. (3.91):

$$\begin{aligned} Z_{\mathrm{CG}}(\mathbf{P},\mathbf{Q}) &= Z_{\mathrm{CG}}^{\mathbf{P}}(\mathbf{P}) \times Z_{\mathrm{CG}}^{\mathbf{Q}}(\mathbf{Q}) \\ &= C\beta^{3(N-n)} \exp\left[-\frac{\beta}{2}\left(\mathbf{P}^{\intercal}\mathbf{M}^{-1}\mathbf{P} + \mathbf{Q}^{\intercal}\mathbf{K}\mathbf{Q}\right)\right]. \end{aligned} \tag{3.92}$$

From Eqn. (3.37),

$$U_{\mathrm{CG}}(\mathbf{P},\mathbf{Q}) = \frac{3}{\beta}(n-N) + \frac{1}{2}\mathbf{P}^{\intercal}\mathbf{M}^{-1}\mathbf{P} + \frac{1}{2}\mathbf{Q}^{\intercal}\mathbf{K}\mathbf{Q}. \tag{3.93}$$

From Eqn. (3.38),

$$F_{\mathrm{CG}}(\mathbf{P},\mathbf{Q}) = -\beta^{-1}\ln C - 3(N-n)\beta^{-1}\ln\beta + \frac{1}{2}\mathbf{P}^{\intercal}\mathbf{M}^{-1}\mathbf{P} + \frac{1}{2}\mathbf{Q}^{\intercal}\mathbf{K}\mathbf{Q}. \tag{3.94}$$

How should we interpret the CG internal energy? The first term contributes an energy of $\frac{1}{2}k_{\mathrm{B}}T$ for each of the $6(n-N)$ degrees of freedom which have been integrated out of the system. The effect of this term is then to compensate for the equipartition energy lost by reducing the number of degrees of freedom. The remainder of U_{CG} appears very similar to the original atomistic Hamiltonian,

$$H(\mathbf{p},\mathbf{q}) = \frac{1}{2}\mathbf{p}^{\intercal}\mathbf{m}^{-1}\mathbf{p} + \frac{1}{2}\mathbf{q}^{\intercal}\mathbf{k}^{-1}\mathbf{q}. \tag{3.95}$$

Because of this, we can interpret the CG mass and stiffness matrices as "renormalised" versions of the atomistic masses and spring stiffnesses, which act on the CG coordinates. Treating U_{CG} as a Hamiltonian for the CG coordinates, let us assume the CG coordinates obey Hamilton's equations:

$$\begin{aligned} \frac{\partial U_{\mathrm{CG}}}{\partial \mathbf{Q}} &= \mathbf{K}\mathbf{Q} = -\dot{\mathbf{P}}, \\ \frac{\partial U_{\mathrm{CG}}}{\partial \mathbf{P}} &= \mathbf{M}^{-1}\mathbf{P} = \dot{\mathbf{Q}}, \end{aligned} \tag{3.96}$$

where the dots indicate derivatives with respect to time. Differentiating one equation with respect to time and inserting into the other, we obtain

$$\begin{aligned} \ddot{\mathbf{P}} &= -\mathbf{K}\dot{\mathbf{Q}} = -\mathbf{K}\mathbf{M}^{-1}\mathbf{P}, \\ \ddot{\mathbf{Q}} &= \mathbf{M}^{-1}\dot{\mathbf{P}} = -\mathbf{M}^{-1}\mathbf{K}\mathbf{Q}. \end{aligned} \tag{3.97}$$

These are the equations of motion of the CG particles, which can be integrated in time using, for example, the Verlet integrator to find the position and momentum of a CG particle at a given time t.

3.2.4 Ensuring Conjugacy of Positions and Momenta

In the previous section, in Eqn. (3.96) we assumed that the CG positions and momenta obeyed Hamilton's equations, taking U_{CG} to be the CG Hamiltonian. This is not actually the case unless we make some restrictions on the form of the CG mapping functions α and γ. This issue is not considered in [45], which the above derivation is based on, since the authors coarse–grain in terms of $(\mathbf{q}, \dot{\mathbf{q}})$ which are not conjugate variables to start with. In this section, we propose a modification to the above derivation which maintains conjugacy by careful selection of the mapping functions.

Firstly, let us make a simple coordinate transformation from $(\mathbf{q}, \mathbf{p})$ coordinates to $(\mathbf{q}', \mathbf{p}')$ coordinates. There are as many primed coordinates as unprimed coordinates, so nothing has been coarse-grained away yet. At this time, the Hamiltonian expressed in terms of primed or unprimed coordinates is equal,

$$H(\mathbf{p}, \mathbf{q}) = H'(\mathbf{p}', \mathbf{q}'). \tag{3.98}$$

Let us go back to the general, nonlinear case where the mapping function between primed and unprimed coordinates is given by

$$q_i' = \gamma_i(\mathbf{q}) \tag{3.99}$$

for $i \in 1, \ldots, 3n$. Let us assume that we can find a set of inverse functions $\bar{\gamma}_i$ for $i \in 1, \ldots, 3n$ such that

$$q_i = \bar{\gamma}_i(\mathbf{q}'). \tag{3.100}$$

The Hamiltonian $H(\mathbf{q}, \mathbf{p})$ can be related to the Lagrangian $L(\mathbf{q}, \dot{\mathbf{q}})$ via the equation

$$H(\mathbf{q}, \mathbf{p}) = \mathbf{p} \cdot \dot{\mathbf{q}} - L(\mathbf{q}, \dot{\mathbf{q}}). \tag{3.101}$$

It is also useful to remember that the conjugate momenta can be found from the Lagrangian using the formula

$$\mathbf{p}_i = \frac{\partial L}{\partial \dot{\mathbf{q}}_i}. \tag{3.102}$$

Since we know the relation between $\mathbf{q}$ and $\mathbf{q}'$, we can write this Lagrangian as a function of the primed coordinates, $L(\mathbf{q}', \dot{\mathbf{q}}')$. If we want the primed momenta $\mathbf{p}'$ to be properly conjugate to the primed positions $\mathbf{q}'$, they must obey the equations

$$p'_i = \frac{\partial L}{\partial \dot{q}_i} = \sum_{j=1}^{3n} \left(\frac{\partial L}{\partial q_j} \frac{\partial q_j}{\partial \dot{q}'_i} + \frac{\partial L}{\partial \dot{q}_j} \frac{\partial \dot{q}_j}{\partial \dot{q}'_i} \right) \tag{3.103}$$

by the chain rule. The first term here is zero, since the q_j do not depend on the $\dot{q}'_i$. Using Eqn. (3.102), we can then write p'_i as

$$p'_i = \sum_{j=1}^{3n} p_j \frac{\partial \dot{q}_j}{\partial \dot{q}'_i}. \tag{3.104}$$

Since $q_j = \bar{\gamma_j}(\mathbf{q}')$,

$$\dot{q_j} = \sum_{k=1}^{3n} \frac{\partial \bar{\gamma}_j}{\partial q'_k} \dot{q}'_k. \tag{3.105}$$

Then

$$\frac{\partial \dot{q_j}}{\partial \dot{q}'_i} = \frac{\partial \bar{\gamma}_j}{\partial q'_i}. \tag{3.106}$$

Hence, p'_i can be written as

$$p'_i = \sum_j p_j \frac{\partial \bar{\gamma}_j}{\partial q'_i}. \tag{3.107}$$

In matrix form, this is

$$\mathbf{p}' = \mathbf{A}\mathbf{p}, \tag{3.108}$$

where

$$a_{ij} = \frac{\partial q_j}{\partial q'_i}. \tag{3.109}$$

Assuming $\mathbf{A}$ can be inverted, we can write

$$\mathbf{p} = \mathbf{A}^{-1}\mathbf{p}' = \mathbf{B}\mathbf{p}', \tag{3.110}$$

or

$$p_i = \sum_j b_{ij} p'_j. \tag{3.111}$$

In summary, the transformations from unprimed to primed coordinates are given by

$$q'_i = \gamma_i(\mathbf{q}), \tag{3.112}$$

$$p'_i = \sum_j p_j \frac{\partial \bar{\gamma}_j}{\partial q'_i}. \tag{3.113}$$

and the inverse transformations from primed to unprimed coordinates are given by

$$\begin{aligned} q_i &= \bar{\gamma}_i(\mathbf{q}'), \\ p_i &= \sum_j b_{ij} p'_j. \end{aligned} \tag{3.114}$$

If we insist that the mapping functions γ_i are linear, Eqn. (3.112) simplifies to

$$q'_i = \sum_{j=1}^{3n} \gamma_{ij} q_j, \tag{3.115}$$

or

$$\mathbf{q}' = \boldsymbol{\gamma} \mathbf{q}. \tag{3.116}$$

In Eqn. (3.100), we defined the inverse mapping function as

$$q_i = \bar{\gamma_i}(\mathbf{q}'). \tag{3.117}$$

In this case, the inverse mapping becomes

$$\mathbf{q} = \boldsymbol{\gamma}^{-1} \mathbf{q}', \tag{3.118}$$

or

$$q_i = \bar{\gamma}_i(\mathbf{q}') = \sum_{j=1}^{3n} \gamma_{ij}^{-1} q'_j. \tag{3.119}$$

This implies that

$$\frac{\partial \bar{\gamma}_j}{\partial q'_i} = \gamma^{-1}_{ji}. \tag{3.120}$$

The formula for the conjugate momenta,

$$p'_i = \sum_j \frac{\partial \bar{\gamma}_j}{\partial q'_i} p_j, \tag{3.121}$$

becomes

$$p'_i = \sum_j \gamma^{-1}_{ji} p_j, \tag{3.122}$$

or

$$\mathbf{p}' = \boldsymbol{\gamma}^{-\intercal}\mathbf{p}. \tag{3.123}$$

Then the inverse mapping becomes

$$\mathbf{p} = \boldsymbol{\gamma}^{\intercal}\mathbf{p}'. \tag{3.124}$$

So in summary, the linear mapping formulae which preserve conjugacy of positions and momenta are

$$\begin{aligned} \mathbf{q}' &= \boldsymbol{\gamma}\mathbf{q}, \\ \mathbf{p}' &= \boldsymbol{\gamma}^{-\intercal}\mathbf{p}, \\ \mathbf{q} &= \boldsymbol{\gamma}^{-1}\mathbf{q}', \\ \mathbf{p} &= \boldsymbol{\gamma}^{\intercal}\mathbf{p}'. \end{aligned} \tag{3.125}$$

Therefore, we are no longer free to choose the mapping functions $\boldsymbol{\alpha}$ and $\boldsymbol{\gamma}$ independently if we want to maintain conjugacy, but must instead choose

$$\boldsymbol{\alpha} = \boldsymbol{\gamma}^{-\intercal}. \tag{3.126}$$

3.2.5 The Selective Integration Method

The derivation in Section 3.2.3 relies on the use of cumbersome delta function notation. In this section, we propose an alternative derivation. Our method

works via careful selection of the integration elements in the CG partition function, so we shall refer to it as the selective integration method. This method has two principal advantages: it makes the derivation considerably simpler, and it allows us to sidestep the problem of the ill–defined inverse of the dynamical matrix which we encountered previously.

Firstly, we must consider the effect of the coordinate transformations given in Section 3.2.4. What is the Jacobian corresponding to these transformations? Let us consider each transformation independently. For the transformation from $\mathbf{q}$ to $\mathbf{q}'$, the elements of the Jacobian matrix $\mathbf{J}^{q'}$ are given by

$$J_{ij}^{q'} = \frac{\partial q_i'}{\partial q_j} = \frac{\partial \gamma_i}{\partial q_j} = \sum_{k=1}^{3n} \frac{\partial \gamma_i}{\partial q_k} \frac{\partial q_k}{\partial q_j}, \tag{3.127}$$

using the fact that $\gamma_i(\mathbf{q}) = q_i'$. For the transformation from $\mathbf{p}$ to $\mathbf{p}'$, the elements of the Jacobian matrix $\mathbf{J}^{p'}$ are given by

$$J_{ij}^{p'} = \frac{\partial p_i'}{\partial p_j} = \frac{\partial \bar{\gamma}_j}{\partial q_i'} = \frac{\partial q_j}{\partial q_i'} \tag{3.128}$$

by using the fact that $\bar{\gamma}_j(\mathbf{q}') = q_j$. The combined Jacobian determinant of these two transformations is then given by

$$J = \det(\mathbf{J}^{q'}) \det(\mathbf{J}^{p'}) = \det(\mathbf{J}^{q'} \mathbf{J}^{p'}). \tag{3.129}$$

Now,

$$(\mathbf{J}^{q'} \mathbf{J}^{p'})_{ij} = \sum_k J_{ik}^{q'} J_{kj}^{p'} = \sum_k \frac{\partial q_i'}{\partial q_k} \frac{\partial q_j}{\partial q_k'}. \tag{3.130}$$

This is close to, but not quite equal to $\frac{\partial q_i'}{\partial q_j'}$. To make it so, let us use the trick that

$$J = \det(\mathbf{J}^{q'}) \det(\mathbf{J}^{p'}) = \det(\mathbf{J}^{q'}) \det(\left(\mathbf{J}^{p'}\right)^{\mathsf{T}}) = \det(\mathbf{J}^{q'} \left(\mathbf{J}^{p'}\right)^{\mathsf{T}}). \tag{3.131}$$

Then

$$(\mathbf{J}^{q'} \left(\mathbf{J}^{p'}\right)^{\mathsf{T}})_{ij} = \sum_k J_{ik}^{q'} J_{jk}^{p'} = \sum_k \frac{\partial q_i'}{\partial q_k} \frac{\partial q_k}{\partial q_j'} = \frac{\partial q_i'}{\partial q_j'} = \delta_{ij}. \tag{3.132}$$

Hence,

$$J = \det(\mathbf{I}) = 1, \tag{3.133}$$

and the integration element remains invariant under this coordinate transformation:

$$d\mathbf{q}d\mathbf{p} = d\mathbf{q}'d\mathbf{p}'. \tag{3.134}$$

Now the full atomistic partition function is

$$Z = \int e^{-\beta H(\mathbf{p},\mathbf{q})} d\mathbf{p}d\mathbf{q}. \tag{3.135}$$

Because $J = 1$, this transforms to

$$Z = \int e^{-\beta H'(\mathbf{p}',\mathbf{q}')} d\mathbf{p}'d\mathbf{q}'. \tag{3.136}$$

At the moment, there are still as many primed coordinates as unprimed coordinates: we have not thrown anything away yet. Let us now pick $N < n$ CG coordinates to keep, we will denote them $(\mathbf{P}, \mathbf{Q})$. This leaves $(n-N)$ coordinates to integrate out and throw away, we will denote them $(\tilde{\mathbf{P}}, \tilde{\mathbf{Q}})$. The partition function can then be written as

$$Z = \int e^{-\beta H'(\mathbf{P},\mathbf{Q},\tilde{\mathbf{P}},\tilde{\mathbf{Q}})} d\mathbf{P}d\mathbf{Q}d\tilde{\mathbf{P}}d\tilde{\mathbf{Q}}. \tag{3.137}$$

This is still the full, unconstrained partition function. If we now consider *only* integrating over the $(n - N)$ coordinates we want to remove, we arrive at the constrained CG partition function,

$$Z_{\mathrm{CG}}(\mathbf{P}, \mathbf{Q}) = \int e^{-\beta H'(\mathbf{P},\mathbf{Q},\tilde{\mathbf{P}},\tilde{\mathbf{Q}})} d\tilde{\mathbf{P}}d\tilde{\mathbf{Q}}. \tag{3.138}$$

We can recover the familiar delta function formula for the CG partition function by writing this as

$$Z_{\mathrm{CG}}(\mathbf{P}, \mathbf{Q}) = \int e^{-\beta H'(\hat{\mathbf{P}},\hat{\mathbf{Q}},\tilde{\mathbf{P}},\tilde{\mathbf{Q}})} \delta(\mathbf{P} - \hat{\mathbf{P}})\delta(\mathbf{Q} - \hat{\mathbf{Q}}) d\hat{\mathbf{P}}d\hat{\mathbf{Q}}d\tilde{\mathbf{P}}d\tilde{\mathbf{Q}}. \tag{3.139}$$

Recall from Section 3.2.3 that the atomistic Hamiltonian can be written as

$$\begin{aligned} H(\mathbf{p}, \mathbf{q}) &= T(\mathbf{p}) + V(\mathbf{q}) \\ &= \frac{1}{2}\mathbf{p}^\intercal \mathbf{m}^{-1}\mathbf{p} + \frac{1}{2}\mathbf{q}^\intercal \mathbf{k}\mathbf{q}. \end{aligned} \tag{3.140}$$

Transforming to primed coordinates, this becomes

$$H'(\mathbf{p}', \mathbf{q}') = \frac{1}{2}\mathbf{p}'^{\intercal}\boldsymbol{\gamma}\mathbf{m}^{-1}\boldsymbol{\gamma}^{\intercal}\mathbf{p}' + \frac{1}{2}\mathbf{q}'^{\intercal}\boldsymbol{\gamma}^{-\intercal}\mathbf{k}\boldsymbol{\gamma}^{-1}\mathbf{q}', \tag{3.141}$$

using the linear transformation defined by $\boldsymbol{\gamma}$. If we define a CG mass matrix

$$\mathbf{M}^{-1} = \boldsymbol{\gamma}\mathbf{m}^{-1}\boldsymbol{\gamma}^{\intercal}, \tag{3.142}$$

and CG dynamical matrix

$$\mathbf{K} = \boldsymbol{\gamma}^{-\intercal}\mathbf{K}\gamma^{-1}, \tag{3.143}$$

this becomes

$$H'(\mathbf{p}', \mathbf{q}') = \frac{1}{2}\mathbf{p}'^{\intercal}\mathbf{M}^{-1}\mathbf{p}' + \frac{1}{2}\mathbf{q}'^{\intercal}\mathbf{K}\mathbf{q}'. \tag{3.144}$$

From Eqn. (3.136), the full partition function is given by

$$\begin{aligned} Z &= \int e^{-\beta H'(\mathbf{p}', \mathbf{q}')} d\mathbf{p}' d\mathbf{q}' \\ &= \int \exp\left[-\beta\left(\frac{1}{2}\mathbf{p}'^{\intercal}\mathbf{M}^{-1}\mathbf{p}' + \frac{1}{2}\mathbf{q}'^{\intercal}\mathbf{K}\mathbf{q}'\right)\right] d\mathbf{p}' d\mathbf{q}'. \end{aligned} \tag{3.145}$$

For now, let us consider only the kinetic energy contribution to the partition function,

$$Z_{\text{kin}} = \int \exp\left[-\frac{\beta}{2}\mathbf{p}'^{\intercal}\mathbf{M}^{-1}\mathbf{p}'\right] d\mathbf{p}'. \tag{3.146}$$

Without loss of generality, let us assume that the first 3N components of $\mathbf{p}'$ are the coarse-grained coordinates we want to keep, and call them $\mathbf{P}$. That leaves the last $3(n-N)$ coordinates of $\mathbf{p}'$ that we want to integrate out, let us call them $\tilde{\mathbf{P}}$. We can then write $\mathbf{M}^{-1}$ as a matrix composed of four submatrices,

$$\mathbf{M}^{-1} = \begin{pmatrix} (\mathbf{M}^{-1})_{11} & (\mathbf{M}^{-1})_{12} \\ (\mathbf{M}^{-1})_{21} & (\mathbf{M}^{-1})_{22} \end{pmatrix}, \tag{3.147}$$

where $(\mathbf{M}^{-1})_{11}$ is a $3n \times 3n$–dimensional square matrix, $(\mathbf{M}^{-1})_{12}$ is a $3N \times 3(n-N)$–dimensional rectangular matrix, $(\mathbf{M}^{-1})_{21}$ is a $3(n-N) \times 3N$–dimensional rectangular matrix and $(\mathbf{M}^{-1})_{22}$ is a $3(n-N) \times 3(n-N)$–dimensional square matrix. Now $\mathbf{M}^{-1}$ is symmetric, since

$$(\mathbf{M}^{-1})^\intercal = (\boldsymbol{\gamma}\mathbf{m}^{-1}\boldsymbol{\gamma}^\intercal)^\intercal = \boldsymbol{\gamma}\mathbf{m}^{-\intercal}\boldsymbol{\gamma}^\intercal = \boldsymbol{\gamma}\mathbf{m}^{-1}\boldsymbol{\gamma}^\intercal = \mathbf{M}^{-1}. \tag{3.148}$$

This implies some useful properties about the submatrices, namely that

$$\begin{aligned}((\mathbf{M}^{-1})_{11})^\intercal &= (\mathbf{M}^{-1})_{11},\\ ((\mathbf{M}^{-1})_{21})^\intercal &= (\mathbf{M}^{-1})_{12},\\ ((\mathbf{M}^{-1})_{22})^\intercal &= (\mathbf{M}^{-1})_{22}.\end{aligned} \tag{3.149}$$

Now,

$$\begin{aligned}\mathbf{p}'^\intercal\mathbf{M}^{-1}\mathbf{p}' &= \begin{pmatrix}\mathbf{P}^\intercal & \tilde{\mathbf{P}}^\intercal\end{pmatrix}\begin{pmatrix}(\mathbf{M}^{-1})_{11} & (\mathbf{M}^{-1})_{12}\\ (\mathbf{M}^{-1})_{21} & (\mathbf{M}^{-1})_{22}\end{pmatrix}\begin{pmatrix}\mathbf{P}\\ \tilde{\mathbf{P}}\end{pmatrix}\\ &= \mathbf{P}^\intercal(\mathbf{M}^{-1})_{11}\mathbf{P} + 2\mathbf{P}^\intercal(\mathbf{M}^{-1})_{12}\tilde{\mathbf{P}} + \tilde{\mathbf{P}}^\intercal(\mathbf{M}^{-1})_{22}\tilde{\mathbf{P}},\end{aligned} \tag{3.150}$$

since $(\mathbf{M}^{-1})_{12}$ is symmetric. In order to integrate out the $\tilde{\mathbf{P}}$ coordinates, we must complete the square on $\tilde{\mathbf{P}}$ to get the integrals into Gaussian form. We find that

$$\begin{aligned}\mathbf{p}'^\intercal\mathbf{M}^{-1}\mathbf{p}' = &\left(\tilde{\mathbf{P}} + [(\mathbf{M}^{-1})_{22}]^{-1}(\mathbf{M}^{-1})_{21}\mathbf{P}\right)^\intercal(\mathbf{M}^{-1})_{22}\left(\tilde{\mathbf{P}} + [(\mathbf{M}^{-1})_{22}]^{-1}(\mathbf{M}^{-1})_{21}\mathbf{P}\right)\\ &+ \mathbf{P}^\intercal(\mathbf{M}^{-1})_{11}\mathbf{P} - \mathbf{P}^\intercal(\mathbf{M}^{-1})_{12}[(\mathbf{M}^{-1})_{22}]^{-1}(\mathbf{M}^{-1})_{21}\mathbf{P}.\end{aligned} \tag{3.151}$$

We can then use the substitution

$$\tilde{\mathbf{P}}' = \tilde{\mathbf{P}} + [(\mathbf{M}^{-1})_{22}]^{-1}(\mathbf{M}^{-1})_{21}\mathbf{P} \tag{3.152}$$

Since this is just a constant coordinate shift, the integration element $d\tilde{\mathbf{P}}' = d\tilde{\mathbf{P}}$ is invariant. Then Eqn. (3.146) becomes

$$\begin{aligned}Z_{\text{kin}} = &\int \exp\left[-\frac{\beta}{2}\tilde{\mathbf{P}}'^\intercal(\mathbf{M}^{-1})_{22}\tilde{\mathbf{P}}'\right]d\tilde{\mathbf{P}}'\\ &\times\int\exp\left[-\frac{\beta}{2}\left(\mathbf{P}^\intercal(\mathbf{M}^{-1})_{11}\mathbf{P} - \mathbf{P}^\intercal(\mathbf{M}^{-1})_{12}[(\mathbf{M}^{-1})_{22}]^{-1}(\mathbf{M}^{-1})_{21}\mathbf{P}\right)\right]d\mathbf{P}.\end{aligned} \tag{3.153}$$

The integral over $\tilde{\mathbf{P}}'$ is now in Gaussian form, so

$$\int \exp\left[-\frac{\beta}{2}\tilde{\mathbf{P}}'^{\intercal}(\mathbf{M}^{-1})_{22}\tilde{\mathbf{P}}'\right] d\tilde{\mathbf{P}}' = \sqrt{\frac{(2\pi)^{3(n-N)}}{\det((\mathbf{M}^{-1})_{22})}}\beta^{-\frac{3}{2}(n-N)}$$
$$= C_1\beta^{-\frac{3}{2}(n-N)}. \tag{3.154}$$

Z_{kin} becomes

$$Z_{\text{kin}} = C_1\beta^{-\frac{3(n-N)}{2}} \int \exp\left[-\frac{\beta}{2}\left(\mathbf{P}^{\intercal}(\mathbf{M}^{-1})_{11}\mathbf{P} - \mathbf{P}^{\intercal}(\mathbf{M}^{-1})_{12}[(\mathbf{M}^{-1})_{22}]^{-1}(\mathbf{M}^{-1})_{21}\mathbf{P}\right)\right] d\mathbf{P}. \tag{3.155}$$

Now to complete our coarse-graining, we simply do not perform the integrals over the $\mathbf{P}$ coordinates. The partition function then becomes an explicit function of the $\mathbf{P}$,

$$Z_{\text{kin}}^{\text{CG}}(\mathbf{P}) = C_1\beta^{-\frac{3(n-N)}{2}} \exp\left[-\frac{\beta}{2}\left(\mathbf{P}^{\intercal}(\mathbf{M}^{-1})_{11}\mathbf{P} - \mathbf{P}^{\intercal}(\mathbf{M}^{-1})_{12}[(\mathbf{M}^{-1})_{22}]^{-1}(\mathbf{M}^{-1})_{21}\mathbf{P}\right)\right]. \tag{3.156}$$

We can simplify this slightly by writing

$$\mathbf{M}_{\text{CG}}^{-1} = (\mathbf{M}^{-1})_{11} - (\mathbf{M}^{-1})_{12}[(\mathbf{M}^{-1})_{22}]^{-1}(\mathbf{M}^{-1})_{21}. \tag{3.157}$$

Then

$$Z_{\text{kin}}^{\text{CG}}(\mathbf{P}) = C_1\beta^{-\frac{3(n-N)}{2}} \exp\left[-\frac{\beta}{2}\mathbf{P}^{\intercal}\mathbf{M}_{\text{CG}}^{-1}\mathbf{P}\right]. \tag{3.158}$$

We have now evaluated the kinetic part of the partition function. Since the potential part is of essentially the same form, we can easily follow the above procedure to evaluate

$$Z_{\text{pot}} = \int \exp\left[-\frac{\beta}{2}\mathbf{q}'^{\intercal}\mathbf{K}\mathbf{q}'\right] d\mathbf{q}'. \tag{3.159}$$

By doing so, we find

$$Z_{\text{pot}}^{\text{CG}}(\mathbf{Q}) = C_2\beta^{-\frac{3(n-N)}{2}} \exp\left[-\frac{\beta}{2}\mathbf{Q}^{\intercal}\mathbf{K}_{\text{CG}}\mathbf{Q}\right]. \tag{3.160}$$

where

$$\mathbf{K}_{\mathrm{CG}} = \mathbf{K}_{11} - \mathbf{K}_{12}\mathbf{K}_{22}^{-1}\mathbf{K}_{21}. \tag{3.161}$$

An important difference here to the delta function method of coarse–graining is that we need only the inverse of the bottom–right block of the dynamical matrix, $\mathbf{K}_{22}^{-1}$, to exist rather than the inverse of the entire dynamical matrix, which is ill–defined. By a careful choice of our initial CG mapping function, we can ensure that the inverse of this block is well-defined.

The complete CG partition function, derived from the selective integration method, is then given by

$$\begin{aligned} Z_{\mathrm{CG}}(\mathbf{P},\mathbf{Q}) &= Z_{\mathrm{kin}}^{\mathrm{CG}} \times Z_{\mathrm{pot}}^{\mathrm{CG}} \\ &= C_3\beta^{-3(n-N)} \exp\left[-\frac{\beta}{2}\left(\mathbf{P}^{\intercal}\mathbf{M}_{\mathrm{CG}}^{-1}\mathbf{P} + \mathbf{Q}^{\intercal}\mathbf{K}_{\mathrm{CG}}\mathbf{Q}\right)\right], \end{aligned} \tag{3.162}$$

where $C_3 = C_1C_2$ is a constant independent of β. The CG internal energy is then given by

$$\begin{aligned} U_{\mathrm{CG}}(\mathbf{P},\mathbf{Q}) &= -\frac{\partial \ln Z_{\mathrm{CG}}}{\partial\beta} \\ &= \frac{3(n-N)}{\beta} + \frac{1}{2}\mathbf{P}^{\intercal}\mathbf{M}_{\mathrm{CG}}^{-1}\mathbf{P} + \frac{1}{2}\mathbf{Q}^{\intercal}\mathbf{K}_{\mathrm{CG}}\mathbf{Q}. \end{aligned} \tag{3.163}$$

This expression is remarkably similar to the atomistic Hamiltonian Eqn. (3.140) that we started from, except with "renormalised" CG mass and stiffness matrices. The first term here is an additional temperature– dependent term which is not present in the atomistic Hamiltonian. To physically interpret this term, recall from Eqn. (3.37) that the internal energy of a simple harmonic oscillator is β^{-1}. Therefore, this term represents the mean energy of a simple harmonic oscillator for each of the $3(n-N)$ degrees of freedom we have removed from the system.

Also, the coarse-grained free energy is given by

$$\begin{aligned} F_{\mathrm{CG}}(\mathbf{P},\mathbf{Q}) &= -\frac{1}{\beta}\ln Z_{\mathrm{CG}} \\ &= -\frac{1}{\beta}\ln C_3 + 3(n-N)\frac{\ln\beta}{\beta} + \frac{1}{2}\mathbf{P}^{\intercal}\mathbf{M}_{\mathrm{CG}}^{-1}\mathbf{P} + \frac{1}{2}\mathbf{Q}^{\intercal}\mathbf{K}_{\mathrm{CG}}\mathbf{Q} \\ &= U_{\mathrm{CG}}(\mathbf{P},\mathbf{Q}) + C(\beta), \end{aligned} \tag{3.164}$$

or in other words, the CG free energy is equal to the CG internal energy plus a temperature-dependent constant. It should be noted that this constant is independent of position and momentum only because of the original assumption that the Hamiltonian is harmonic; it is not true in general. Again, this temperature–dependent term can be related to the free energy of a simple harmonic oscillator for each of the $3(n-N)$ missing degrees of freedom, by using Eqn. (3.38).

Since the CG momenta are now properly conjugate to the CG positions, they must obey Hamilton's equations of motion,

$$\begin{aligned} \frac{d\mathbf{P}}{dt} &= -\frac{\partial U_{\mathrm{CG}}}{\partial \mathbf{Q}}, \\ \frac{d\mathbf{Q}}{dt} &= +\frac{\partial U_{\mathrm{CG}}}{\partial \mathbf{P}}. \end{aligned} \tag{3.165}$$

In matrix form, these are

$$\begin{aligned} \dot{\mathbf{P}} &= -\mathbf{K}_{\mathrm{CG}}\mathbf{Q}, \\ \dot{\mathbf{Q}} &= +\mathbf{M}_{\mathrm{CG}}\mathbf{P}. \end{aligned} \tag{3.166}$$

Differentiating the second equation here and substituting in the first, we arrive at

$$\ddot{\mathbf{Q}} = -\mathbf{M}_{\mathrm{CG}}\mathbf{K}_{\mathrm{CG}}\mathbf{Q}. \tag{3.167}$$

This allows us to calculate the accelerations on the CG particles, which can be used in the Verlet integrator to evolve the CG positions in time. Similarly, differentiating the first equation and substituting in the second, we arrive at

$$\ddot{\mathbf{P}} = -\mathbf{K}_{\mathrm{CG}}\mathbf{M}_{\mathrm{CG}}\mathbf{P}. \tag{3.168}$$

This can also be used in the Verlet integrator to evolve the CG momenta in time.

In summary, we have shown in this chapter that the approach to coarse–graining outlined in [45] and [46] does not integrate the correct equations of motion.

Chapter 4

A Technical Description of Coarse–Graining Methods

In Section 3.2.2, it was noted that Eqn. (3.60) for the CG free energy is of fundamental importance to CG techniques. This is because thermodynamic expectation values calculated in the CG ensemble using the CG free energy as a potential are guaranteed to produce the same results as the true atomistic potential in the atomistic ensemble, provided the expectation value can be expressed *only* as a function of the CG coordinates. Since the CG partition function is separable, we could equally well consider the *configurational CG free energy*, which is a function of the CG positions only:

$$F^{\mathbf{Q}}_{\mathrm{CG}}(\mathbf{Q}) = -k_{\mathrm{B}}T \ln Z^{\mathbf{Q}}_{\mathrm{CG}}(\mathbf{Q}). \tag{4.1}$$

If we knew this function exactly, we could in principle calculate expectation values from CG simulations with no loss of accuracy. In Section 3.2.3, we showed how this is possible for the special case of a harmonic potential. Unfortunately, in general this potential is impossible to obtain analytically, since we cannot perform the necessary integrations over configurational space for a real anharmonic potential. Even if it was possible to evaluate Eqn. (4.1) exactly, it would be a complicated many–body function of the N CG coordinates $\mathbf{Q}$, which would be difficult to use in practice. In particular, the CG potential would be so slow to evaluate that it might defeat the purpose of coarse–graining in the first place.

Therefore, the goal of any useful CG methodology is to approximate Eqn.

(4.1) as closely as possible, whilst also producing a CG potential which is simple enough to use in practice. The following methods all approach this in different ways.

4.1 Force Matching

Following the notation in [20], let there be N CG particles, with positions labelled by $\mathbf{R}_I$ for $I \in 1, \ldots, N$. The authors consider four possible types of interaction to be modelled in the CG force field:

1. A bond vibration interaction between two CG sites on the same molecule, a function of the scalar distance between sites only
2. An angular interaction formed by three CG sites, a function of the angle only
3. A dihedral interaction formed by four CG sites, a function of the dihedral angle only
4. A pairwise nonbonded interaction between two CG sites on different molecules, a function of the scalar distance between sites only.

The CG potential can then be written as

$$U(\mathbf{R}^N) = \sum_{\zeta i \gamma} U_{\zeta i}(x_\zeta(\mathbf{R}_\gamma)). \tag{4.2}$$

This is a sum over all interactions, where ζ denotes which of the four interaction types we are considering from the list above, i denotes the functional form of an interaction of type ζ, and γ denotes one of the many sets of sites that interact via interaction ζ using functional form γ. x_ζ denotes the single scalar variable that interaction ζ depends on (for example, the scalar distance between sites in 1.) above). $\mathbf{R}_\gamma$ denotes the positions of the set of sites γ which the interaction depends on. The force on each CG site can be derived from Eqn. (4.2) as

$$\mathbf{F}_I(\mathbf{R}^N) = -\frac{\partial U}{\partial \mathbf{R}_I} = -\sum_{\zeta i \gamma} \frac{dU_{\zeta i}}{dx_\zeta} \frac{\partial x_\zeta}{\partial \mathbf{R}_I}. \tag{4.3}$$

For each $U_{\zeta i}$, we choose a set of basis functions $u_{\zeta i d}(x_\zeta)$ which are functions of the appropriate scalar variable only. $U_{\zeta i}$ can then be written as a basis expansion,

$$U_{\zeta i}(x_\zeta) = \sum_d \phi_{\zeta id} u_{\zeta id}(x_\zeta). \tag{4.4}$$

Let us define

$$f_{\zeta id}(x) = -\frac{du_{\zeta id}(x)}{dx}. \tag{4.5}$$

Then we have

$$\mathbf{F}_I(\mathbf{R}^N) = \sum_{\zeta id} \phi_{\zeta id} \left[\sum_\gamma f_{\zeta id}(x_\zeta) \frac{\partial x_\zeta}{\partial \mathbf{R}_I} \right] = \sum_{\zeta id} \phi_{\zeta id} \mathbf{G}_I(\mathbf{R}^N), \tag{4.6}$$

where

$$\mathbf{G}_I(\mathbf{R}^N) = \left[\sum_\gamma f_{\zeta id}(x_\zeta) \frac{\partial x_\zeta}{\partial \mathbf{R}_I} \right]. \tag{4.7}$$

The notation is becoming complicated, so let us replace the sum in Eqn. (4.6) with a sum over each of the N_D basis coefficients, where the index D corresponds to a particular set of (ζid). Eqn. (4.6) becomes

$$\mathbf{F}_I(\mathbf{R}^N) = \sum_{D=1}^{N_D} \phi_D \mathbf{G}_I(\mathbf{R}^N). \tag{4.8}$$

Briefly returning to the n_t snapshots of atomic positions and forces referred to earlier, let us define the force acting on CG site I in snapshot t as $\mathbf{f}_I(\mathbf{r}_t)$. Then finding the optimal set of CG force field parameters ϕ_D becomes a least squares optimisation problem, where we want to minimise the function

$$\chi^2(\phi_D) = \frac{1}{3n_t N} \sum_{t=1}^{n_t} \sum_{I=1}^{N} \left| \mathbf{f}_I - \sum_{D=1}^{N_D} \phi_D \mathbf{G}_I \right|^2. \tag{4.9}$$

This equation can be recast in matrix form and solved by QR decomposition or singular value decomposition, as outlined in [20].

4.2 Relative Entropy Minimization

When deriving a CG model from an atomistic system, it would be very useful to be able to quantify the amount of information about the system that is lost.

From the principles of information theory, Shell et al. define a quantity called the *relative entropy* which makes this possible [36, 37].

The authors refer to the underlying atomistic system as the *target system*, denoted by T, as this is the system whose properties we would like to reproduce. They refer to the CG model of the atomistic system as the *model system*, denoted by M. To simplify the initial discussion, let us assume that there are the same number of degrees of freedom in the target and model system. This is the case when no actual coarse–graining is being performed, and we are simply performing a transformation of coordinates. Let us also assume that the system can only be in a discrete set of configurations, which are labelled by the index i.

Let the probability of being in state i be given by $p_M(i)$ and $p_T(i)$ in the model and target system respectively. In the model system, if we make n measurements, the expected number of times we will find the system in state i is therefore

$$\bar{n}(i) = n \times p_M(i). \tag{4.10}$$

From basic combinatorics, the probability of being in state 1 $\bar{n}(1)$ times, state 2 $\bar{n}(2)$ times and so on out of n total observations is given by

$$\begin{aligned} &\frac{n!}{\bar{n}(1)!\bar{n}(2)!\ldots} p_M(1)^{\bar{n}(1)} p_M(2)^{\bar{n}(2)} \ldots \\ &= n! \prod_i \frac{1}{\bar{n}(i)!} p_M(i)^{\bar{n}(i)}. \end{aligned} \tag{4.11}$$

We would like to know the likelihood that this test of the model system gives us the same expected frequency for each state i as in the target system, i.e. the chance that the model is an ideal representation of the target. We denote this likelihood as $L(T|M)$, and it can be found simply by replacing the expected frequencies in Eqn. (4.11) with the target frequencies

$$\bar{n}(i) = n \times p_T(i), \tag{4.12}$$

so that

$$L(T|M) = n! \prod_i \frac{1}{[np_T(i)]!} p_M(i)^{np_T(i)}. \tag{4.13}$$

Taking the logarithm of this equation,

$$\ln L(T|M) = \ln n! + \sum_i \left[-\ln[np_T(i)]! + np_T(i) \ln p_M(i) \right]. \tag{4.14}$$

Recall Stirling's approximation,

$$\ln(n!) \approx n \ln n - n \tag{4.15}$$

for large n. Then the above equation becomes

$$\begin{aligned}
\ln L(T|M) &\approx n \ln n - n - n \sum_i p_T(i) \left\{ \ln[np_T(i)] - 1 - \ln p_M(i) \right\} \\
&= n \ln n - n - n \sum_i p_T(i) \left\{ \ln n + \ln \frac{p_T(i)}{p_M(i)} - 1 \right\} \\
&= -n \sum_i p_T(i) \ln \frac{p_T(i)}{p_M(i)} \\
&= -n S_{\text{rel}},
\end{aligned} \tag{4.16}$$

making use of the fact that $\sum_i p_T(i) = 1$, where

$$S_{\text{rel}} = \sum_i p_T(i) \ln \frac{p_T(i)}{p_M(i)} \tag{4.17}$$

is the relative entropy. Setting $n = 1$ above, we can then physically interpret S_{rel} as the log likelihood that one test configuration of the model system is representative of the target.

We can now lift the restriction that there are the same number of degrees of freedom in the target and model system, to actually perform coarse–graining. Let us denote the set of coordinates by $\mathbf{r}_T$ and $\mathbf{r}_M$ in the target and model system respectively, with the coordinates related to each other through the mapping function M, such that

$$\mathbf{r}_M = M(\mathbf{r}_T). \tag{4.18}$$

We will use the shorthand notation $j = M(i)$ for the model configuration j generated from a target configuration i. It is important to note that in general, many atomic configurations in the target system will generate the same configuration in the model system. We denote the number of target configurations which give rise to a model configuration k by

$$\Omega_{\rm map}(k) = \sum_i \delta_{k,M(i)}. \tag{4.19}$$

This is the discrete equivalent to the continuous delta function description of Eqn. (3.60).

We must now modify the model probabilities $p_M(i)$ to account for this degeneracy. The probability that a specific target configuration i is produced in the model ensemble is given by

$$p'_M(i) = \frac{p_M(M(i))}{\Omega_{\rm map}(M(i))}. \tag{4.20}$$

Now the sum over all target configurations i of $p'_M(i)$ is unity, whereas the sum over all model configurations $M(j)$ of $p_M(M(j))$ is unity. Replacing p_M with p'_M in Eqn. (4.17), we find

$$\begin{aligned} S'_{\rm rel} &= \sum_i p_T(i) \ln \frac{p_T(i)}{p'_M(i)} \\ &= \sum_i p_T(i) \ln \frac{p_T(i)}{p_M(M(i))} + \sum_i p_T(i) \Omega_{\rm map}(M(i)) \\ &= \sum_i p_T(i) \ln \frac{p_T(i)}{p_M(M(i))} + \langle S_{\rm map} \rangle_T, \end{aligned} \tag{4.21}$$

where $\langle S_{\rm map} \rangle_T$ is known as the *mapping entropy*.

The relative entropy has some useful properties, which show it behaves in a way consistent with the familiar thermodynamic picture of entropy. Firstly, from Eqn. (4.16), since L is a probability, $\ln L$ is negative, which implies that $S_{\rm rel}$ is always positive. We can write $L = e^{-nS_{\rm rel}}$, so it is clear that as $S_{\rm rel}$ decreases, L increases. When $S_{\rm rel}$ is zero, that is, when the model and target systems have the same set of states and $p_M(i) = p_T(i)$ for all i, then $L = 1$. This is the case where the model system perfectly replicates the target system.

The formulae above were derived without reference to any particular thermodynamic ensemble. To apply the method usefully, we now specialise to the case of the canonical ensemble. From Eqn. (3.14), the probability in this ensemble is given by

$$p_T(i) = Z^{-1} e^{-\beta(U_T(i) - F_T)}, \tag{4.22}$$

where $U_T(i)$ is the internal energy of the target system in configuration i and F_T is the free energy in the target system, a constant in the NVT ensemble, and Z is the canonical partition function. Substituting this into Eqn. (4.21), we find

$$\begin{aligned} S'_{\text{rel}} &= \beta \sum_i Z^{-1} e^{-\beta(U_T(i)-F_T)} \left[U_M(M(i)) - U_T(i) + F_T - F_M\right] + \langle S_{\text{map}} \rangle_T \\ &= \beta \sum_i Z^{-1} e^{-\beta(U_T(i)-F_T)} \left[U_M(M(i)) - U_T(i)\right] - \beta(F_M - F_T) + \langle S_{\text{map}} \rangle_T, \end{aligned} \tag{4.23}$$

since $\sum_i p_T(i) = 1$. This simplifies to

$$S'_{\text{rel}} = \beta \langle U_M - U_T \rangle_T - \beta(F_M - F_T) + \langle S_{\text{map}} \rangle_T \tag{4.24}$$

if we define

$$\langle U_M - U_T \rangle_T = \sum_i p_T \left[U_M(M(i)) - U_T(i)\right]. \tag{4.25}$$

We can now use Eqn. (4.24) as the basis for finding an optimal CG potential. Let us assume the model potential U_M depends on a set of variable parameters $\{\lambda_1, \lambda_2, \ldots\}$. We wish to minimise S'_{rel} with respect to one of these parameters, λ_k. The only terms in Eqn. (4.24) which depend on λ_k are the model internal energy U_M and (implicitly) the model free energy F_M. The free energy can be written as

$$F_M = -\beta^{-1} \ln Z = -\beta^{-1} \sum_i e^{-\beta U_M(M(i))}, \tag{4.26}$$

so that

$$\frac{\partial F_M}{\partial \lambda_k} = Z^{-1} \sum_i \frac{\partial U_M}{\partial \lambda_k} e^{-\beta U_M} = \left\langle \frac{\partial U_M}{\partial \lambda_k} \right\rangle_M. \tag{4.27}$$

Therefore, minimising with respect to λ_k gives us

$$
\begin{aligned}
0 = \frac{\partial S'_{\rm rel}}{\partial \lambda_k} &= \beta \frac{\partial}{\partial \lambda_k} \langle U_M - U_T \rangle_T - \beta \frac{\partial F_M}{\partial \lambda_k} \\
&= \beta \sum_i p_T \frac{\partial U_M(M(i))}{\partial \lambda_k} - \beta \frac{\partial F_M}{\partial \lambda_k} \\
&= \beta \left\langle \frac{\partial U_M}{\partial \lambda_k} \right\rangle_T - \beta \left\langle \frac{\partial U_M}{\partial \lambda_k} \right\rangle_M . \qquad (4.28)
\end{aligned}
$$

This implies that

$$
\left\langle \frac{\partial U_M}{\partial \lambda_k} \right\rangle_T = \left\langle \frac{\partial U_M}{\partial \lambda_k} \right\rangle_M . \qquad (4.29)
$$

Eqn. (4.29) shows that when the relative entropy is minimised, the average derivatives of the λ parameters are the same in the model and target systems. This suggests a useful method of optimising the parameters in the model potential. Recall for a function $f(x)$, a root $f(x^*) = 0$ can be found by the Newton–Raphson iterative method,

$$
x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}, \qquad (4.30)
$$

repeating until a value of x sufficiently close to a root is found. Applying this formula to find a value of one of the parameters λ such that

$$
\frac{\partial S'_{\rm rel}}{\partial \lambda} = 0, \qquad (4.31)
$$

we obtain

$$
\begin{aligned}
\lambda_{i+1} &= \lambda_i - \frac{\partial S'_{\rm rel}/\partial \lambda}{\partial^2 S'_{\rm rel}/\partial \lambda^2} \\
&= \lambda_i - \left[\left\langle \frac{\partial U_M}{\partial \lambda_k} \right\rangle_T - \left\langle \frac{\partial U_M}{\partial \lambda_k} \right\rangle_M \right] \times \\
&\left[\left\langle \frac{\partial^2 U_M}{\partial \lambda^2} \right\rangle_T - \left\langle \frac{\partial^2 U_M}{\partial \lambda^2} \right\rangle_M + \beta \left\langle \left(\frac{\partial U_M}{\partial \lambda} \right)^2 \right\rangle - \beta \left\langle \frac{\partial U_M}{\partial \lambda} \right\rangle_M^2 \right]^{-1} . \qquad (4.32)
\end{aligned}
$$

Chapter 5

Molecular Dynamics

5.1 Introduction

Molecular dynamics (MD) is a method of tracking the trajectories of a set of N particles. The particles interact with each other via a given interaction potential $V(\mathbf{x}_1, \ldots, \mathbf{x}_N)$, where $\mathbf{x}_i(t)$ is the position of particle i at time t. The force acting on particle i is given by

$$\mathbf{f}_i = -\frac{\partial V}{\partial \mathbf{x}_i}. \tag{5.1}$$

Newton's second law, $\mathbf{f}_i = m_i \mathbf{a}_i$, can then be used to find the acceleration of particle i if its mass is m_i.

In MD, we start with a system of N particles, and assume that their positions, velocities and accelerations are known at a given time ($t = 0$ is the most common choice). The problem is then to find the particle positions, velocities and accelerations at a later time $t + \Delta t$, where Δt is known as the *time step* of the MD simulation. By discretising time using a small enough time step, we can use a numerical integration algorithm to accomplish this. Typical atomic vibrational frequencies range between 10^{12} and 10^{14} Hz, correponding to a minimum vibrational period of 10^{-14} seconds. A time step of 10^{-15} seconds (one femtosecond) would then achieve 10 samples per period. The choice of time step can greatly effect the energy conservation of a simulation, so it is important to choose carefully on a system by system basis. A typical rule of thumb is to use at least 20 samples per shortest vibrational period in the system.

We now discuss some of the most popular integration algorithms used for MD.

5.2 Integration Schemes

5.2.1 The Verlet Algorithm

One of the most commonly used integration schemes in MD is the Verlet algorithm, due to its simplicity and ease of implementation. Let $\mathbf{x}(t)$ be the position of a particle at time t. By using a Taylor expansion, we can write the particle's position at a later time, $t + \Delta t$, as

$$\mathbf{x}(t + \Delta t) = \mathbf{x}(t) + \Delta t\dot{\mathbf{x}}(t) + \frac{\Delta t^2}{2}\ddot{\mathbf{x}}(t) + \frac{\Delta t^3}{6}\dddot{\mathbf{x}}(t) + O(\Delta t^4). \tag{5.2}$$

Performing the same expansion at an earlier time $t - \Delta t$, we find

$$\mathbf{x}(t - \Delta t) = \mathbf{x}(t) - \Delta t\dot{\mathbf{x}}(t) + \frac{\Delta t^2}{2}\ddot{\mathbf{x}}(t) - \frac{\Delta t^3}{6}\dddot{\mathbf{x}}(t) + O(\Delta t^4). \tag{5.3}$$

Adding these two expressions, identifying $\dot{\mathbf{x}}(t)$ as the particle's velocity $\mathbf{v}(t)$ and $\ddot{\mathbf{x}}(t)$ as the particle's acceleration $\mathbf{a}(t)$, and ignoring $O(\Delta t^4)$ terms and higher, we arrive at

$$\mathbf{x}(t + \Delta t) = 2\mathbf{x}(t) - \mathbf{x}(t - \Delta t) + \Delta t^2\mathbf{a}(t). \tag{5.4}$$

This is the Verlet algorithm. At each time step, the particle accelerations are calculated from Eqn. 5.1 and used in Eqn. 5.4 to update the particle positions. The error in this equation is $O(\Delta t^4)$, making it accurate and simple to implement. The particle velocities can be estimated using the expression

$$\mathbf{v}(t) = \frac{\mathbf{x}(t + \Delta t) - \mathbf{x}(t - \Delta t)}{2\Delta t}. \tag{5.5}$$

The error in this expression is $O(\Delta t^2)$ instead of $O(\Delta t^4)$. There are some potential problems with this algorithm, discussed in the next section.

5.2.2 The Velocity Verlet Algorithm

One problem with the standard Verlet algorithm is that the particle positions at two consecutive initial time steps are needed. This means the second time step positions must be generated from the first in some crude way. A second problem

is that the particle velocities do not appear explicitly in the Verlet algorithm, so a different algorithm must be used if we want to have direct access to the particle velocities in an MD simulation. Both of these problems can be solved by using a modification of the Verlet algorithm, known as Velocity Verlet. This variant was first derived by Swope et al. in [47], but this explanation is based on Chapter 4.3 of [2].

By Taylor expanding the position $\mathbf{x}(t + \Delta t)$ to second order, we obtain

$$\mathbf{x}_i(t + \Delta t) = \mathbf{x}_i(t) + \mathbf{v}_i(t)\Delta t + \mathbf{a}_i(t)\frac{\Delta t^2}{2} \tag{5.6}$$

$$= \mathbf{x}_i(t) + \mathbf{v}_i(t)\Delta t + \frac{\mathbf{f}_i(t)}{2m_i}\Delta t^2. \tag{5.7}$$

Now suppose we define the velocity at time step $t + \Delta t$ as

$$\mathbf{v}_i(t + \Delta t) = \mathbf{v}_i(t) + \frac{\mathbf{f}_i(t + \Delta t) + \mathbf{f}_i(t)}{2m_i}\Delta t. \tag{5.8}$$

Eqns. 5.7 and 5.8 together consitute the velocity Verlet algorithm. It is not immediately obvious that this definition is equivalent to the standard Verlet algorithm, but we can show this as follows. Use Eqn. 5.7 to advance $\mathbf{x}_i$ by one time step:

$$\mathbf{x}_i(t + 2\Delta t) = \mathbf{x}_i(t + \Delta t) + \mathbf{v}_i(t + \Delta t)\Delta t + \frac{\mathbf{f}_i(t + \Delta t)}{2m_i}\Delta t^2. \tag{5.9}$$

We can also rewrite Eqn. 5.7 as

$$\mathbf{x}_i(t) = \mathbf{x}_i(t + \Delta t) - \mathbf{v}_i(t)\Delta t - \frac{\mathbf{f}_i(t)}{2m_i}\Delta t^2. \tag{5.10}$$

Add together Eqns. 5.9 and 5.10:

$$\begin{aligned} \mathbf{x}_i(t + 2\Delta t) + \mathbf{x}_i(t) = 2\mathbf{x}_i(t + \Delta t) & \\ + \left[\mathbf{v}_i(t + \Delta t) - \mathbf{v}_i(t)\right]\Delta t & \\ + \frac{\mathbf{f}_i(t + \Delta t) - \mathbf{f}_i(t)}{2m_i}\Delta t^2. & \end{aligned} \tag{5.11}$$

Substitute in Eqn. 5.8 to get

$$\mathbf{x}_i(t + 2\Delta t) = 2\mathbf{x}_i(t + \Delta t) - \mathbf{x}_i(t) + \frac{\mathbf{f}_i(t + \Delta t)}{m_i}\Delta t^2, \tag{5.12}$$

or going back one time step,

$$\mathbf{x}_i(t+\Delta t) = 2\mathbf{x}_i(t) - \mathbf{x}_i(t-\Delta t) + \frac{\mathbf{f}_i(t)}{m_i}\Delta t^2, \tag{5.13}$$

which is indeed the standard Verlet algorithm we have been using to integrate the equations of motion up until now.

The Velocity Verlet algorithm must be implemented slightly differently to the standard Verlet algorithm. Before the simulation starts, we must first calculate the interatomic forces. Then, at each time step, we update the atomic positions, update the component of velocity corresponding to the force at that time step, update the forces, and then finish updating the velocity using the force at the next time step, in that order.

5.2.3 The Leapfrog Algorithm

A second alternative to the Verlet algorithm is the leapfrog algorithm. Here, the velocities are advanced half a time step first using the equation

$$\mathbf{v}\left(t+\frac{1}{2}\Delta t\right) = \mathbf{v}\left(t-\frac{1}{2}\Delta t\right) + \Delta t\mathbf{a}(t). \tag{5.14}$$

An estimate for the current velocity can then be obtained from

$$\mathbf{v}(t) = \frac{1}{2}\left(\mathbf{v}\left(t+\frac{1}{2}\Delta t\right) + \mathbf{v}\left(t-\frac{1}{2}\Delta t\right)\right). \tag{5.15}$$

The positions are then advanced a full time step using the equation

$$\mathbf{x}(t+\Delta t) = \mathbf{x}(t) + \Delta t\mathbf{v}\left(t+\frac{1}{2}\Delta t\right), \tag{5.16}$$

at which point the new accelerations can be calculated.

This algorithm is so-named because calculations of velocity and position alternate in a "leap-frogging" fashion. It is sometimes used instead of the regular Verlet algorithm, because there is no need to subtract two similar large numbers, reducing the risk of floating point arithmetic errors.

5.3 Temperature and Thermostat Algorithms

In order to study the properties of systems when they are heated or cooled, it is necessary to define a sensible measure of temperature in an MD simulation. From the equipartition formula, there is an average kinetic energy of $\frac{1}{2}k_BT$ for

each of the N_{deg} kinetic degrees of freedom in the system. In a system of N particles free from constraints, $N_{\text{deg}} = 3N$. Evaluating the total kinetic energy K at time t,

$$K(t) = \frac{1}{2}\sum_i m_i |\mathbf{v}_i(t)|^2, \tag{5.17}$$

we can then estimate the temperature at time t by

$$T = \frac{2K}{N_{\text{deg}} k_B}. \tag{5.18}$$

In order to study a system in the canonical ensemble (constant particle number N, volume V and temperature T) in MD, some method of controlling the temperature in the simulation is needed. This is done using thermostat algorithms, some examples of which are discussed below. For a comprehensive review and comparison of thermostat algorithms, see [48].

5.3.1 The Berendsen Thermostat

The Berendsen thermostat [49] is an example of the simplest form of thermostat algorithm, velocity rescaling. We pick a target temperature T_{target} that we would like our system to reach, calculate the actual temperature at time t from Eqn. 5.18, and at each time step rescale the velocities $\mathbf{v}_i \mapsto \lambda \mathbf{v}_i$, where

$$\lambda = \sqrt{1 + \frac{\Delta t}{\tau}\left(\frac{T_{\text{target}}}{T} - 1\right)}. \tag{5.19}$$

Here, τ is an arbitrary constant which determines how rapidly the velocity rescaling takes place.

This thermostat will allow the system to reach its target temperature quickly, and is computationally efficient. Unfortunately, it also restricts temperature fluctuations to an artificially small range, meaning it does not correctly sample from the canonical ensemble. In practice, the Berendsen thermostat is often used to quickly bring a system to its target temperature, and a more sophisticated thermostat is then used to bring the system closer to the canonical ensemble.

5.3.2 The Langevin Thermostat

The Langevin thermostat [1] modifies the equation of motion for each particle to mimic the presence of a stochastic heat bath, and adds a term for viscous

drag. The modified equation of motion becomes

$$m_i \mathbf{a}_i(t) = \mathbf{f}_i(t) - m_i \gamma_i \mathbf{v}_i(t) + \mathbf{R}_i(t), \tag{5.20}$$

where $\mathbf{f}_i(t)$ is the normal force on particle i in the absence of any thermostat, γ_i is the friction constant which determines the magnitude of the viscous drag, and $\mathbf{R}_i(t)$ is a stochastic force which obeys the following two properties:

$$\langle \mathbf{R}_i(t) \rangle = 0, \tag{5.21}$$

$$\langle \mathbf{R}_i(t) \mathbf{R}_j(t') \rangle = 2 m_i \gamma_i k_\mathrm{B} T \delta(t - t') \delta_{ij}. \tag{5.22}$$

The effect of the stochastic force is to "heat up" particles below the target temperature, and the effect of the drag term is to "cool down" particles above the target temperature.

5.3.3 The Nosé–Hoover Thermostat

So far, we have discussed examples of two common classes of thermostat algorithms. These are velocity rescaling (the Berendsen thermostat) and stochastic dynamics (the Langevin thermostat). There is a third common class of thermostat algorithms called extended Lagrangian approaches. The most popular example of this is the Nosé–Hoover Thermostat [50, 51]. The Nosé–Hoover Hamiltonian is given by

$$H_\mathrm{Nose} = \sum_{i=1}^{N} \frac{\mathbf{p}_i^2}{2 m_i} + V(\mathbf{r}) + \frac{\xi^2 Q}{2} + 3N \frac{\ln s}{\beta}, \tag{5.23}$$

where ξ is a friction constant, and we have introduced an extra degree of freedom s with mass Q that couples to the rest of the system. The coordinates $\mathbf{r}$ and $\mathbf{p}$ above correspond to "virtual" positions and momenta, to convert to real positions and momenta we must use the transformations

$$\begin{aligned} \mathbf{r}_i' &= \mathbf{r}_i, \\ \mathbf{p}_i' &= \mathbf{p}_i / s, \\ t' &= \int_0^t 1/s(\tau) d\tau. \end{aligned} \tag{5.24}$$

See Section 6.1.2 of [2] for a clear derivation of these equations, and a proof that the Nosé–Hoover thermostat correctly samples from the canonical ensemble.

The extra degree of freedom s represents a heat bath which is used to control the temperature of the system, and it has been explicitly included in H_{Nose}.

5.4 Periodic Boundary Conditions

A typical system contains Avogadro's number ($\approx 6.02 \times 10^{23}$) of particles. This system size is far beyond the reach of MD simulations on even the world's most powerful supercomputers. We must instead perform simulations on a much smaller subsystem, from which we can carefully infer properties of the bulk system.

Ideally, we would like to perform a simulation in vacuum, but in general the ratio between the volume of the system and its surface area will be heavily skewed such that surface effects will dominate. To get around this, we can approximate a material in bulk by using periodic boundary conditions.

The most common choice is to use cubic boundary conditions. Here, we define the *simulation cell* to be a cubic box with side length L, containing all the particles we wish to simulate. This choice is appropriate for simulating gases and liquids, but more complicated cells must be used for non-cubic solid systems. We then imagine infinite copies of the simulation cell, tiled in the x, y and z directions to fill space. We only keep track of the particles in the simulation cell, not in the periodic copies. This approach introduces two additional steps of complexity.

Firstly, we need to take account of what happens when a particle crosses the boundary of the simulation cell. Let us assume that the cell is centred on the origin, so that the boundaries are at $\left[-\frac{L}{2}, +\frac{L}{2}\right]$ in the x, y and z directions. Then for each particle i, after each MD time step, we perform the following adjustment:

$$\begin{aligned} x_i &\to x_i + L \text{ if } x_i < -\frac{L}{2}, \\ x_i &\to x_i - L \text{ if } x_i > +\frac{L}{2}, \end{aligned} \tag{5.25}$$

and similarly for the y and z directions. This ensures that the particles always remain within the central cell. If the particles are moving fast enough that they could move more than a single box length away, the above algorithm needs to be modified slightly, but in practice we are almost never interested in such high velocities.

Secondly, we need to take account of the effect of periodic boundary conditions on the distance between two distinct particles i and j. This distance is needed to compute many quantities, for example the interaction energy between particles i and j. Whilst it may appear that the particles are more than half a box length away from each other in the x direction in the central cell, particle i will in fact be closer to a periodic image of particle j than this. We must therefore apply the *minimum image convention* when calculating all particle-particle distances [1]. Since we have already insisted that the particles are kept in the central cell, this is relatively simple. Let

$$\Delta x = x_j - x_i, \tag{5.26}$$

that is, the relative rather than absolute distance. Then, the algorithm

$$\begin{aligned} \Delta x &\to \Delta x + L \text{ if } \Delta x < -\frac{L}{2}, \\ \Delta x &\to \Delta x - L \text{ if } \Delta x > +\frac{L}{2}, \end{aligned} \tag{5.27}$$

enforces the minimum image convention. A similar correction must be applied for the y and z directions.

Of course, the use of periodic boundary conditions can lead to unwanted artefacts in MD simulations. For example, if we imagine a large macromolecule such as a long polymer chain in a periodic box, it is possible that atoms at the head of the chain will "see" the periodic images of atoms in the tail, causing the polymer to interact with itself in an unphysical manner. This can be mitigated by using as large a box as possible.

5.5 The Force Field

In a first principles simulation technique such as DFT, the forces between atoms come from the theory of quantum mechanics and are not parameters of the model. By contrast, in classical MD the forces are empirical, and we must choose a sensible functional form for these forces. The force field is often chosen to accurately reproduce certain quantities known from experiment for a particular molecular system. Generally, a trade–off between accuracy and computational efficiency must be made when defining the force field. Typically, the force field is split into two parts: *nonbonded* and *bonded* terms:

$$V_{\text{total}} = V_{\text{nonbonded}} + V_{\text{bonded}}. \tag{5.28}$$

Let us discuss each of these in turn.

5.5.1 Nonbonded Terms

The nonbonded terms describe long–range interactions between pairs of atoms. They are typically further subdivided, into

$$V_{\text{nonbonded}} = V_{\text{VDW}} + V_{\text{coulomb}}. \tag{5.29}$$

A popular choice for V_{VDW} is the Lennard-Jones potential [52],

$$V_{\text{LJ}}(r) = 4\epsilon \left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6} \right] \tag{5.30}$$

Here, r is the pair separation of two atoms, ϵ is the depth of the potential well and σ is the distance at which the potential first becomes zero (see Fig. 5.1). The r^{-12} term represents the Pauli repulsion of the atoms at short distances, and the r^{-6} term represents the long–range attraction due to Van der Waal's forces. The power of 12 on the repulsive term is used purely for historical reasons, as exponentiation used to be a computationally expensive operation, and $12 = 6 \times 2$ was an efficient exponent to calculate.

The second nonbonded term is given by

$$V_{\text{coulomb}}(r_{ij}) = \frac{q_i q_j}{4\pi\epsilon_0\epsilon_r r_{ij}}, \tag{5.31}$$

where q_i and q_j are the charges on atoms i and j respectively, ϵ_0 is the permittivity of free space and ϵ_r is the relative permittivity. This is the standard formula describing the Coulomb interaction between two atoms due to electrostatics. The long–range nature of the Coulomb interaction means that careful attention has to be paid when periodic boundary conditions are applied. The technique of Ewald's summation [53] can be used for this purpose. In our work, the atoms are taken to be neutral and so V_{coulomb} can be safely neglected.

5.5.2 Bonded Terms

The bonded terms define short-range interactions between atoms, and are used to represent the complex behaviour of chemical bonding. There are two main

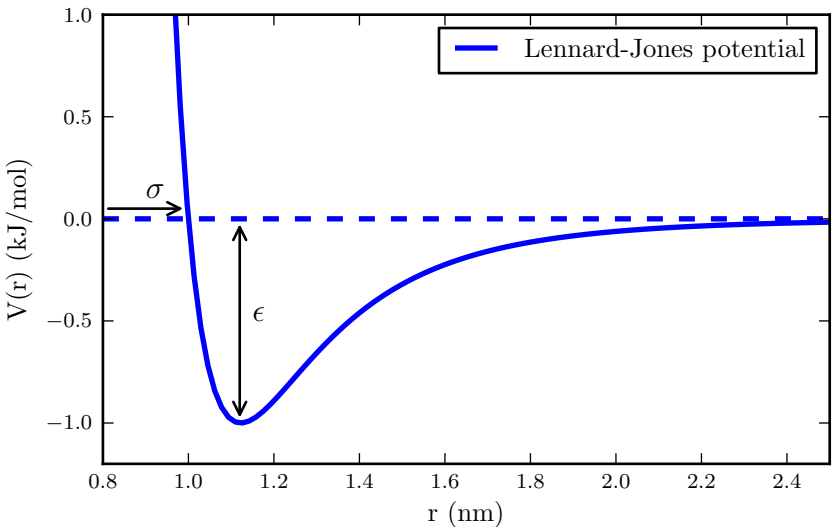


Figure 5.1: The Lennard–Jones potential, with $\epsilon = 1$ kJ/mol and $\sigma = 1$ nm marked on for clarity.

approaches for defining these interactions.

The first is to assume that the force on each atom is a function of its surrounding environment, as in the embedded atom method [54]. This is conceptually related to quantum mechanical methods such as DFT, and allows for the making and breaking of chemical bonds. Unfortunately, the force calculation in these methods can still be very time consuming, so a further level of simplification is often needed.

In the second approach, known as molecular mechanics, the list of bonded atoms and bond types is defined at the beginning of a simulation as a molecular topology, which cannot change during the course of the simulation. This means that chemical bonds cannot be made or broken. As compensation, force evaluation becomes computationally trivial because the bonded terms are often chosen to have a very simple functional form. The bonded part of the force field is commonly defined as

$$V_{\text{bonded}} = V_{\text{bond}} + V_{\text{angle}} + V_{\text{dihedral}} + V_{\text{misc}}. \tag{5.32}$$

Qualitatively, V_{bond} represents the energy cost of a pair of atoms moving away

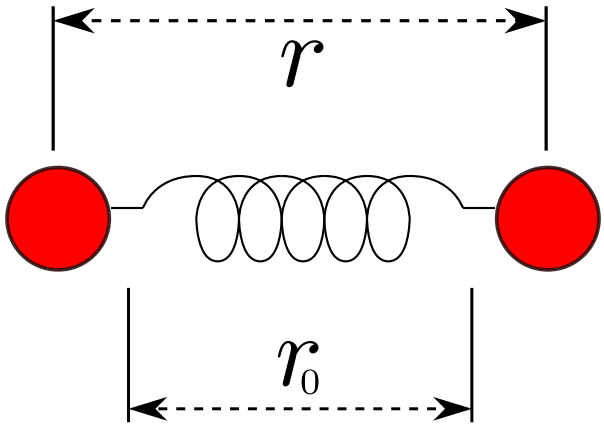


Figure 5.2: Diagram of bond between two atoms, represented by V_{bond} in the force field.

from their equilibrium separation. This is often represented by a harmonic term. V_{angle} represents the energy cost of two atoms bending away from each other with respect to a third atom. V_{dihedral} involves four atoms, and represents the energy cost of atoms moving out of plane. V_{misc} represents anything not included in the previous three terms, and can include complicated many–body interactions or other terms, depending on the force field chosen.

To make things more concrete, let us consider the particular force field used in this work, the Optimised Potentials for Liquid Simulation all–atom (OPLS–AA) force field [3, 55]. Here,

$$V_{\text{bond}} = \sum_{\text{bonds}} K_r (r - r_0)^2, \tag{5.33}$$

where K_r is the harmonic spring constant and r_0 is the equilibrium atomic separation (see Fig. 5.2).

$$V_{\text{angle}} = \sum_{\text{angles}} K_\theta (\theta - \theta_0)^2, \tag{5.34}$$

where K_θ is the angular harmonic spring constant, θ is the angle formed between two atoms by a third, and θ_0 is the equilibrium separation angle (see Fig. 5.3).

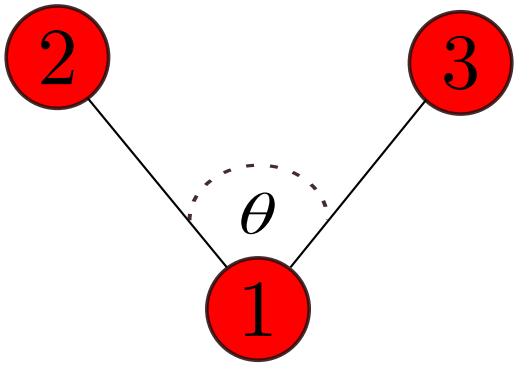


Figure 5.3: Diagram of angular force field term V_{angle}. θ is the angle made with atom 1 by atoms 2 and 3, as shown, with θ_0 being the equilibrium angle.

$$V_{\text{dihedral}} = \frac{V_1}{2}[1 + \cos(\phi - \phi_0)] + \frac{V_2}{2}[1 + 2\cos(\phi - \phi_0)] + \frac{V_3}{2}[1 + 3\cos(\phi - \phi_0)] + \frac{V_4}{2}[1 + 4\cos(\phi - \phi_0)], \tag{5.35}$$

where V_{1-4} are constants depending on the atom type, ϕ is the dihedral angle formed between two planes of atoms and ϕ_0 is the equilibrium dihedral angle (see Fig. 5.4). V_{misc} is zero in the OPLS–AA force field, but many more complicated force fields exist which incorporate features such as polarization and other many–body terms.

There is an additional feature in the OPLS–AA force field. Nonbonded interactions are ignored between atom pairs separated by less than three bonds. The interaction between atoms 1 and 4 in a chain is reduced by a factor of two, and interactions with all other atoms are not scaled in any way. This is known as a "fudge factor", and is used to stop the nonbonded energy terms from dominating at short distances. The rationale behind this approach is that the quadratic bonded terms are from a Taylor expansion of the energy between two atoms, which already includes components due to Van der Waal's forces, Coulomb attraction and so on.

For any particular MD force field, a parameter set is provided which defines the values of any necessary constants. In the above, for example, these include the Lennard–Jones ϵ and σ, equilibrium bond distances, spring stiffnesses and

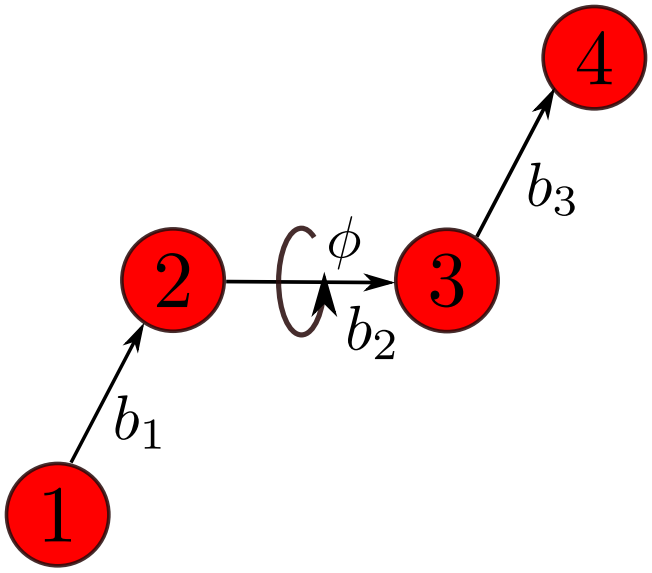


Figure 5.4: Diagram of dihedral force field term V_{dihedral}. Imagine two planes, one formed by the vectors b_1 and b_2, the other by vectors b_2 and b_3. The dihedral angle ϕ is the angle formed between these two planes.

so on. In general, these values are listed for every atom type that the force field was intended to simulate. The parameters are often fitted from experiment, but it is also becoming common to fit them from first principles simulations. In this way, MD simulations using these force fields are able to reproduce certain properties of the fitted model, at greatly reduced computational cost.

5.6 Energy Minimisation Algorithms

It is often necessary to find the minimum energy configuration of atoms according to the particular force field we have chosen. There are several algorithms available for this purpose.

5.6.1 Steepest Descent

In the steepest descent algorithm, we first calculate the force $\mathbf{F}_i$ on each of the N atoms using our MD force field. We define the force unit vector for each atom as

$$\hat{\mathbf{F}}_i = \frac{\mathbf{F}_i}{|\mathbf{F}_i|}. \tag{5.36}$$

We then move each atom i a small increment κ in the direction of this vector,

$$\mathbf{r}_i^t = \mathbf{r}_i^{t-1} + \kappa \hat{\mathbf{F}}_i^{t-1}, \tag{5.37}$$

repeating until the forces on every atom are below a given tolerance. κ can be small at the beginning of the process, to avoid problems due to very high initial forces. It can then be made bigger to speed up the process of moving towards the minimum energy configuration. Despite the name, this method actually converges to a minimum quite slowly, although it does have the advantage of numerical stability. The next method can be considerably faster.

5.6.2 Conjugate Gradients

The method of conjugate gradients is considerably more sophisticated than steepest descent, and often converges to the correct solution in far fewer iterations. It does this by ensuring that each successive search direction is conjugate to all previous search directions, so that searches are never performed in the

same direction twice. For a detailed technical explanation of the method, see Appendix B.

5.7 Constraint Algorithms

It is sometimes necessary to constrain the distances between atoms in an MD simulation, in order to force the system into a configuration we are interested in studying. This must be done carefully, to avoid problems with energy conservation and other artefacts. In this section, we discuss two common techniques for enforcing such constraints.

5.7.1 The SHAKE Algorithm

One of the first methods developed for applying constraints in MD simulations is the SHAKE algorithm [56]. This explanation is based on [57].

For simplicity, we will only consider rigid constraints between atom pairs, where atom i at position $\mathbf{r}_i$ and atom j at position $\mathbf{r}_j$ are kept a fixed distance l_{ij} apart. The vector joining atom i to atom j is given by $\mathbf{r}_{ij} = \mathbf{r}_j - \mathbf{r}_i$. Let us write the constraint equations as

$$\sigma_{ij} = \mathbf{r}_{ij}^2 - l_{ij}^2 = 0. \tag{5.38}$$

Let us also imagine there are n atoms in total, interacting via the potential $V(\mathbf{r})$. Using the method of Lagrange multipliers, the constrained equations of motion now take the form

$$m_i \frac{d^2\mathbf{r}_i(t)}{dt^2} = -\frac{\partial}{\partial \mathbf{r}_i}\left[V(\mathbf{r}) + \sum_{i\to p} \lambda_{ip}\sigma_{ip}\right], \tag{5.39}$$

where the first term on the right is the normal, unconstrained force $\mathbf{F}_i$ acting on atom i, and the second term on the right is the force due to the constraints being imposed on atom i. The sum here is indexed by p, and runs over all constraints which are affecting atom i. λ_{ip} is the undetermined Lagrange multiplier for the constraint between atoms i and p.

Let us first consider the constraint force $\mathbf{G}_i$ in isolation:

$$\begin{aligned}\mathbf{G}_i &= -\sum_{i\to p} \lambda_{ip} \frac{\partial}{\partial \mathbf{r}_i} \sigma_{ip}(\mathbf{r}) \\ &= -\sum_{i\to p} \lambda_{ip} \frac{\partial}{\partial \mathbf{r}_i} \left[(\mathbf{r}_i - \mathbf{r}_p)^2 - l_{ip}^2 \right] \\ &= -2 \sum_{i\to p} \lambda_{ip} \mathbf{r}_{ip}(t). \end{aligned} \tag{5.40}$$

Now let us consider what the position of atom i will be after one time step Δt, under the action of the normal, unconstrained force $\mathbf{F}_i$ *and* the constraint force $\mathbf{G}_i$. From the traditional Verlet algorithm,

$$\begin{aligned}\mathbf{r}_i(t+\Delta t) &= \mathbf{r}_i(t) + \Delta t \mathbf{v}_i(t) + \frac{1}{2}\Delta t^2 \mathbf{a}_i(t) \\ &= \mathbf{r}_i(t) + \Delta t \mathbf{v}_i(t) + \frac{\Delta t^2}{2m_i} \left(\mathbf{F}_i + \mathbf{G}_i \right) \\ &= \left[\mathbf{r}_i(t) + \Delta t \mathbf{v}_i(t) + \frac{\Delta t^2}{2m_i} \mathbf{F}_i \right] + \left[\frac{\Delta t^2}{2m_i} \mathbf{G}_i \right] \\ &= \tilde{\mathbf{r}}_i(t+\Delta t) + \frac{\Delta t^2}{2m_i} \mathbf{G}_i, \end{aligned} \tag{5.41}$$

where $\tilde{\mathbf{r}}_i(t+\Delta t)$ is the position of atom i after one time step if it was moving subject to no constraints. Substituting Eqn. 5.40 into Eqn. 5.41, we find

$$\mathbf{r}_i(t+\Delta t) = \tilde{\mathbf{r}}_i(t+\Delta t) - \frac{\Delta t^2}{m_i} \sum_{i\to p} \lambda_{ip} \mathbf{r}_{ip}(t). \tag{5.42}$$

We can absorb the factor of $\left(-\Delta t^2\right)$ into the Lagrange multipliers to simplify things, yielding

$$\mathbf{r}_i(t+\Delta t) = \tilde{\mathbf{r}}_i(t+\Delta t) + \frac{1}{m_i} \sum_{i\to p} \lambda_{ip} \mathbf{r}_{ip}(t). \tag{5.43}$$

Now, assuming the constraint is satisfied exactly at time $(t+\Delta t)$, the constraint equation for the link between atom i and atom j is given by

$$
\begin{aligned}
l_{ij}^2 &= \mathbf{r}_{ij}^2(t+\Delta t) \\
&= \left[\mathbf{r}_i(t+\Delta t) - \mathbf{r}_j(t+\Delta t)\right]^2 \\
&= \left[\tilde{\mathbf{r}}_i(t+\Delta t) + \frac{1}{m_i}\sum_{i\to p}\lambda_{ip}\mathbf{r}_{ip}(t) - \tilde{\mathbf{r}}_j(t+\Delta t) - \frac{1}{m_j}\sum_{j\to q}\lambda_{jq}\mathbf{r}_{jq}(t)\right]^2 \\
&= \left[\tilde{\mathbf{r}}_{ij}(t+\Delta t) + \frac{1}{m_i}\sum_{i\to p}\lambda_{ip}\mathbf{r}_{ip}(t) - \frac{1}{m_j}\sum_{j\to q}\lambda_{jq}\mathbf{r}_{jq}(t)\right]^2 . \qquad (5.44)
\end{aligned}
$$

Let us now make the first approximation: that the constraint between atoms i and j is independent of all the other constraints. Then Eqn. 5.44 reduces to

$$
\begin{aligned}
l_{ij}^2 &= \left[\tilde{\mathbf{r}}_{ij}(t+\Delta t) + \frac{1}{m_i}\lambda_{ij}\mathbf{r}_{ij}(t) - \frac{1}{m_j}\lambda_{ji}\mathbf{r}_{ji}(t)\right]^2 \\
&= \left[\tilde{\mathbf{r}}_{ij}(t+\Delta t) + \frac{1}{m_i}\lambda_{ij}\mathbf{r}_{ij}(t) + \frac{1}{m_j}\lambda_{ij}\mathbf{r}_{ij}(t)\right]^2 \\
&= \tilde{\mathbf{r}}_{ij}^2 + 2(m_i^{-1}+m_j^{-1})\lambda_{ij}\mathbf{r}_{ij}\cdot\tilde{\mathbf{r}}_{ij} + O(\lambda^2). \qquad (5.45)
\end{aligned}
$$

If we linearise this by discarding $O(\lambda^2)$ terms, we get

$$
l_{ij}^2 = \tilde{\mathbf{r}}_{ij}^2 + 2(m_i^{-1}+m_j^{-1})\lambda_{ij}\mathbf{r}_{ij}\cdot\tilde{\mathbf{r}}_{ij}. \qquad (5.46)
$$

We can invert this to get an approximation to λ_{ij},

$$
\lambda_{ij} = \frac{\tilde{\mathbf{r}}_{ij}^2 - l_{ij}^2}{2(m_i^{-1}+m_j^{-1})\mathbf{r}_{ij}\cdot\tilde{\mathbf{r}}_{ij}}. \qquad (5.47)
$$

Plugging this into Eqn. 5.43, we find the equations of motion for atoms i and j:

$$
\begin{aligned}
\mathbf{r}_i(t+\Delta t) &= \tilde{\mathbf{r}}_i(t+\Delta t) + \frac{1}{m_i}\lambda_{ij}\mathbf{r}_{ij}(t), \\
\mathbf{r}_j(t+\Delta t) &= \tilde{\mathbf{r}}_j(t+\Delta t) - \frac{1}{m_j}\lambda_{ij}\mathbf{r}_{ij}(t). \qquad (5.48)
\end{aligned}
$$

The positions that result from this equation are then taken into account when the next constrained link is adjusted. This may or may not violate the original constraint between i and j, but we carry on with this process iteratively until all constraints are satisfied to within a given tolerance.

5.7.2 The RATTLE Algoritm

The SHAKE algorithm only adjusts the particle positions to satisfy the constraints, and makes no attempt to modify the particle velocities. It was also developed for use with the Verlet integration algorithm, which does not make explicit use of the velocities. In order to apply constraints to the velocity Verlet integration algorithm, as used in DL_POLY, a modified procedure is needed. This is called the RATTLE algorithm [58].

The constraints on the particle velocities are found by taking the time derivative of the constraint equation, Eqn. 5.38:

$$\begin{aligned} \frac{d\sigma_{ij}}{dt} &= \frac{d}{dt}\left(\mathbf{r}_{ij}^2 - l_{ij}^2\right) = 0 \\ \implies & 2[\mathbf{r}_i(t) - \mathbf{r}_j(t)] \cdot [\dot{\mathbf{r}}_i(t) - \dot{\mathbf{r}}_j(t)] = 0 \\ \implies & [\mathbf{r}_i(t) - \mathbf{r}_j(t)] \cdot [\dot{\mathbf{r}}_i(t) - \dot{\mathbf{r}}_j(t)] = 0. \end{aligned} \tag{5.49}$$

The velocity Verlet algorithm is given by

$$\mathbf{r}_i(t + \Delta t) = \mathbf{r}_i(t) + \Delta t \mathbf{v}_i(t) + \frac{1}{2}\Delta t^2 \mathbf{a}_i(t), \tag{5.50}$$

$$\mathbf{v}_i(t + \Delta t) = \mathbf{v}_i(t) + \frac{1}{2}\Delta t \left[\mathbf{a}_i(t) + \mathbf{a}_i(t + \Delta t)\right]. \tag{5.51}$$

In the notation introduced in the SHAKE section, Eqn. 5.50 becomes

$$\mathbf{r}_i(t + \Delta t) = \mathbf{r}_i(t) + \Delta t \mathbf{v}_i(t) + \frac{\Delta t^2}{2m_i}\left[\mathbf{F}_i - 2\sum_{i \to p} \lambda_{ip}^{\mathrm{pos}}(t)\mathbf{r}_{ip}(t)\right]. \tag{5.52}$$

Here we have given λ_{ip} the superscript "pos" to denote that it is the Lagrange multiplier used to satisfy the position constraints. We solve this by finding the $\lambda_{ip}^{\mathrm{pos}}$ iteratively by the method described in the SHAKE section. In an exactly analagous way, Eqn. 5.51 becomes

$$\begin{aligned} \mathbf{v}_i(t + \Delta t) = \mathbf{v}_i(t) + \frac{\Delta t}{2m_i}\Big[&\mathbf{F}_i(t) - 2\sum_{i \to p} \lambda_{ip}^{\mathrm{pos}}(t)\mathbf{r}_{ip}(t) \\ &+ \mathbf{F}_i(t + \Delta t) - 2\sum_{i \to k} \lambda_{ik}^{\mathrm{vel}}(t + \Delta t)\mathbf{r}_{ik}(t + \Delta t)\Big]. \end{aligned} \tag{5.53}$$

Here we have introduced a new set of Lagrange multipliers $\lambda_{ij}^{\mathrm{vel}}$ which will be used to ensure the particle velocities satisfy the velocity constraints. We can rewrite Eqn. 5.53 as

$$\mathbf{v}_i(t+\Delta t) = \tilde{\mathbf{v}}_i(t+\Delta t) - \frac{1}{m_i}\sum_{i\to k}\lambda_{ik}^{\mathrm{vel}}(t+\Delta t)\mathbf{r}_{ik}(t+\Delta t), \tag{5.54}$$

where $\tilde{\mathbf{v}}_i$ are the unconstrained velocities. Again, Δt has been absorbed into the Lagrange multiplier. If we again treat the constraints as independent and consider constraint (ij), this becomes

$$\mathbf{v}_i(t+\Delta t) = \tilde{\mathbf{v}}_i(t+\Delta t) - \frac{1}{m_i}\lambda_{ij}^{\mathrm{vel}}(t+\Delta t)\mathbf{r}_{ij}(t+\Delta t). \tag{5.55}$$

We want to choose $\lambda_{ij}^{\mathrm{vel}}$ so that

$$[\mathbf{v}_i(t+\Delta t) - \mathbf{v}_j(t+\Delta t)]\cdot\mathbf{r}_{ij}(t+\Delta t) = 0. \tag{5.56}$$

This becomes

$$0 = [\tilde{\mathbf{v}}_i(t+\Delta t) - \tilde{\mathbf{v}}_j(t+\Delta t) - \lambda_{ij}^{\mathrm{vel}}(m_i^{-1}+m_j^{-1})\mathbf{r}_{ij}(t+\Delta t)]\cdot\mathbf{r}_{ij}(t+\Delta t). \tag{5.57}$$

This implies that

$$\lambda_{ij}^{\mathrm{vel}} = \frac{\mathbf{r}_{ij}(t+\Delta t)\cdot[\tilde{\mathbf{v}}_i(t+\Delta t) - \tilde{\mathbf{v}}_j(t+\Delta t)]}{l_{ij}^2(m_i^{-1}+m_j^{-1})}. \tag{5.58}$$

These Lagrange multipliers are found self-consistently by iterating over each constraint repeatedly until all constraints are satisfied to within a given tolerance, exactly as in the SHAKE algorithm.

We now have a practical way of constraining both the particle positions and velocities within an MD simulation. The above discussion explained how constraints are applied between individual atoms. The only difference when applying constraints between molecules is that the "atoms" are now two virtual sites representing the centre of masses of two distinct molecules. The RATTLE algorithm is applied exactly as before to constrain these two sites at a fixed separation.

5.8 Implementation Details

5.8.1 Initialising a Simulation

When starting an MD simulation of a liquid or gas, care must be taken when choosing the initial positions of the molecules. If any two molecules are placed too close together, the interaction energy between them can diverge, causing the simulation to crash. A common way to avoid this is to place the molecules on a simple cubic lattice with a sufficiently large lattice constant to avoid any infinities in the potential energy. An energy minimisation algorithm can then be used to relax the system into a more physical initial configuration.

When choosing the initial particle velocities, it is common to sample these from the Maxwell distribution at the temperature we would like to simulate,

$$p(v_i) = \sqrt{\frac{2}{\pi}} \frac{v_i^2 \exp\left(-v_i^2/2a^2\right)}{a^3,} \tag{5.59}$$

where $a = \sqrt{k_\mathrm{B}T/m_i}$.

5.8.2 Neighbour Lists

In a system of N particles, there are ${}^N C_2$ distinct pairs of particles, assuming each particle only interacts with the nearest periodic image of the other $N-1$ particles. Recall that

$${}^N C_2 = \frac{N!}{2!(N-2)!} = \frac{N(N-1)}{2}. \tag{5.60}$$

This formula is $O(N^2)$, so the number of distinct pairs grows rapidly with increasing system size. Calculating the distance vector between each of these pairs can therefore be a time-consuming part of an MD simulation. To get around this, the concept of a *neighbour list* is introduced.

Consider the ith atom in the simulation box. For each of the $N-1$ other atoms j, we check if $|\mathbf{x}_i - \mathbf{x}_j| < x_\mathrm{list}$, where x_list is our neighbour list cutoff length. If so, atom j is added to the neighbour list for atom i. We generate N such neighbour lists, one for each atom. In subsequent MD steps, we only check the distance between atom i and the atoms in its neighbour list, significantly reducing the total number of distance calculations required.

Of course, the use of a neighbour list can cause significant problems for energy conservation in a simulation. If an atom far away from atom i, which

was initially excluded from atom i's neighbour list, drifts to within x_{list} of atom i, its interaction with atom i will not be calculated even though it could be very close and should contribute a large amount to the potential energy. This problem can be partially resolved by periodically regenerating the neighbour lists for every atom, for example every ten MD time steps.

5.8.3 Cutoff Radii

Another way to enhance the performance of an MD simulation is to use *cutoff radii*. The nonbonded interactions can be truncated at a given distance r_{cut}, and interactions beyond this distance are not calculated. This leads to a discontinuity in the potential energy, which will lead to poor energy conservation in the simulation. There are two ways to rectify this.

The first is to apply a constant shift to the entire potential, removing the discontinuity. This has no effect on the forces, since

$$\begin{aligned} V'(r) &= V(r) - V(r_{\text{cut}}) \\ \implies F'(r) &= -\frac{dV'}{dr} = -\frac{dV}{dr} = F(r). \end{aligned} \tag{5.61}$$

Of course, if a small cutoff radius is used, this could lead to a significant modification in the potential energy function. The second method is to apply a switch function over a given range, which smoothly switches the potential energy to zero at the cutoff radius. This leads to a change in both the potential and force, but both remain continuous. We discuss this method in more detail in Section 8.7.

5.9 Calculating Quantities from an MD Simulation

There would be no point performing MD simulations unless they allowed us to obtain useful information about the system being studied. In this section, we discuss some of the common quantities that can be extracted from MD, and how to obtain them.

5.9.1 Energies

The potential energy V is calculated simply from the sum of all interactions in the system. It is common to also separately record the individual components of V, for example the bond and angular terms, for further analysis. The kinetic energy K is calculated from the velocities of each atom, using

$$K = \frac{1}{2} \sum_{i=1}^{n} m_i \mathbf{v}_i \cdot \mathbf{v}_i. \tag{5.62}$$

The total energy T is simply the sum of potential and kinetic energy,

$$T = V + K. \tag{5.63}$$

5.9.2 Diffusion Constant

The diffusion constant of a molecule can be estimated from the mean squared displacement (MSD) of its centre of mass. The mean squared displacement is given as a function of time as

$$\mathrm{MSD(t)} = \langle |\mathbf{r}(\mathrm{t}) - \mathbf{r}(0)|^2 \rangle. \tag{5.64}$$

A generic example of a MSD curve for an ideal gas is shown in Fig. 5.5. Here, the first part of the curve is nonlinear. This is because for short times, the molecules are relatively unobstructed and will move in an approximately straight line at constant velocity. Their displacement is therefore proportional to time, and the MSD is proportional to time squared, giving a parabolic shape. The particles are said to be undergoing *ballistic motion* in this part of the curve. The second part of the curve is linear. From his work on Brownian motion, Einstein showed that the diffusion constant is related to the MSD by

$$\mathrm{MSD(t)} = 6\mathrm{Dt} + \mathrm{C}. \tag{5.65}$$

Therefore, the diffusion constant can be estimated as one sixth the gradient of the MSD curve in this linear region.

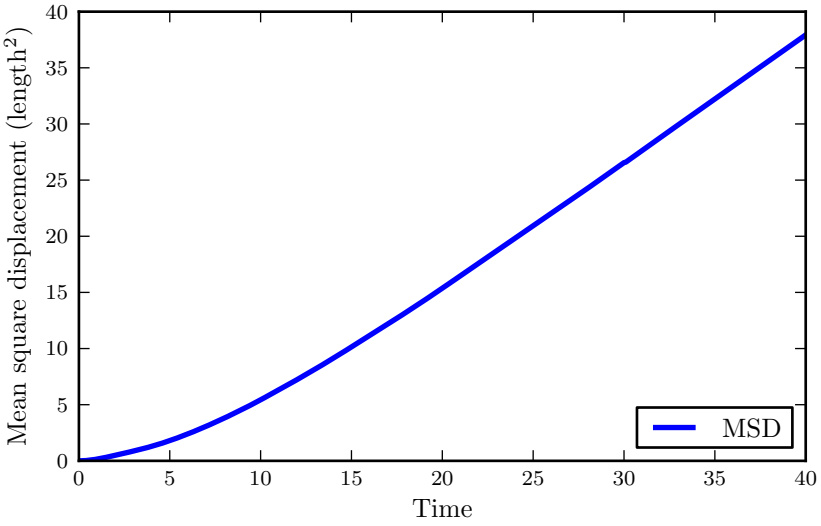


Figure 5.5: An example MSD curve.

5.9.3 Pair Correlation Function

The pair correlation function defined in Eqn. (3.51) can be expressed in a more easy to calculate form as

$$g(r) = \frac{V}{4\pi r^2 N^2} \left\langle \sum_i \sum_{j \neq i} \delta(r - r_{ij}) \right\rangle, \tag{5.66}$$

where V is the volume, N is the number of particles, δ is the Dirac delta function and r_{ij} is the scalar separation of particles i and j. The average is taken over all trajectory frames in a simulation.

The algorithm for calculating the pair correlation function numerically is as follows. First, pick a small value of dr, the shell thickness. Loop over each value of r we want to calculate $g(r)$ at, and

1. Consider each particle one at a time. Count the number of particles which are between r and $r + dr$ away, a spherical shell of thickness dr.
2. Normalise the total count by dividing by the number of particles N.
3. Normalise the volume by dividing by $4\pi r^2 dr$, the shell volume.
4. Finally, divide by the number density ρ.

Care must be taken to account for periodic boundary conditions when calculating inter–particle distances.

5.9.4 Pressure

Pressure is not trivial to calculate in an MD simulation. One method [59] is based on the *Clausius virial function*,

$$W(\mathbf{r}_1, \ldots, \mathbf{r}_N) = \sum_{i=1}^{N} \mathbf{r}_i \cdot \mathbf{F}_i. \tag{5.67}$$

We must make use of the virial theorem, which relates the average kinetic energy to the virial function:

$$2\langle K \rangle = -\langle W \rangle. \tag{5.68}$$

To prove this, note that the average of W can be obtained from a MD trajectory as

$$\langle W \rangle = \lim_{t\to\infty} \frac{1}{t} \int_0^t d\tau \sum_{i=1}^{N} \mathbf{r}_i(\tau) \cdot m_i \ddot{\mathbf{r}}_i(\tau), \tag{5.69}$$

where we have inserted Newton's second law. Integrating by parts, this becomes

$$\begin{aligned} \langle W \rangle &= \lim_{t\to\infty} \frac{1}{t} \sum_{i=1}^{N} m_i \int_0^t \mathbf{r}_i(\tau) \cdot \mathbf{a}_i(\tau) d\tau \\ &= \lim_{t\to\infty} \frac{1}{t} \sum_{i=1}^{N} m_i \left([\mathbf{r}_i \cdot \mathbf{v}_i]_0^t - \int_0^t |\mathbf{v}_i|^2 d\tau \right). \end{aligned} \tag{5.70}$$

The first term here is bounded, if we assume the position vectors are bounded by the periodic box, and the velocities are also bounded. This must be the case if we are to avoid infinite kinetic energies. Then the limit as $t \to \infty$ causes the first term to vanish. We are therefore left with

$$\begin{aligned} \langle W \rangle &= -\lim_{t\to\infty} \frac{1}{t} \int_0^t \sum_{i=1}^{N} m_i |\mathbf{v}_i|^2 d\tau \\ &= -\lim_{t\to\infty} \frac{1}{t} \int_0^t 2K(\tau) \\ &= -2\langle K \rangle. \end{aligned} \tag{5.71}$$

This proves the virial theorem. From the equipartition formula Eqn. (5.18), we then find

$$\langle W \rangle = -N_{\text{deg}} k_B T. \tag{5.72}$$

To continue, let us split the force on each particle into two components,

$$\mathbf{F}_i = \mathbf{F}_i^{\text{int}} + \mathbf{F}_i^{\text{ext}}, \tag{5.73}$$

where $\mathbf{F}_i^{\text{int}}$ is the internal force arising from interatomic interactions, and $\mathbf{F}_i^{\text{ext}}$ is the external force exerted by the (possibly fictional) container walls. The virial can then be written as

$$\begin{aligned} W &= \sum_{i=1}^{N} \mathbf{r}_i \cdot (\mathbf{F}_i^{\text{int}} + \mathbf{F}_i^{\text{ext}}) \\ &= \sum_{i=1}^{N} \mathbf{r}_i \cdot \mathbf{F}_i^{\text{int}} + \mathbf{r}_i \cdot \mathbf{F}_i^{\text{ext}} \\ &= W^{\text{int}} + W^{\text{ext}}. \end{aligned} \tag{5.74}$$

Let us consider W^{ext} in isolation. Splitting this into Cartesian components,

$$W^{\text{ext}} = \sum_{\alpha=1}^{3} \sum_{i=1}^{N} r_i^\alpha F_i^\alpha \tag{5.75}$$

Taking the average,

$$\langle W^{\text{ext}} \rangle = \sum_{\alpha=1}^{3} \left\langle \sum_{i=1}^{N} r_i^\alpha F_i^\alpha \right\rangle. \tag{5.76}$$

Let us imagine that the container is an orthorhombic box with side lengths L_x, L_y and L_z respectively, with its corner located at the origin. The mean force on each side of the box is the pressure P times the area, so that Eqn. (5.76) becomes

$$\begin{aligned} \langle W^{\text{ext}} \rangle &= 0 \times (PL_xL_y) + 0 \times (PL_yL_z) + 0 \times (PL_xL_z) \\ &\quad + L_z(-PL_xL_y) + L_x(-PL_yL_z) + L_y(-PL_xL_z) \\ &= -3PL_xL_yL_z \\ &= -3PV. \end{aligned} \tag{5.77}$$

Combining Eqns. (5.72) and (5.74), we then find

$$\langle W \rangle = \left\langle \sum_{i=1}^{N} \mathbf{r}_i \cdot \mathbf{F}_i^{\text{int}} \right\rangle - 3PV = -N_{\text{deg}} k_B T$$
$$\implies PV = \frac{1}{3} N_{\text{deg}} k_B T + \frac{1}{3} \left\langle \sum_{i=1}^{N} \mathbf{r}_i \cdot \mathbf{F}_i^{\text{int}} \right\rangle. \tag{5.78}$$

This formula allows us to estimate the pressure from the volume, temperature and interatomic forces, all of which we know from the simulation. Also note that in the case of an ideal gas, the particles are noninteracting, so $\mathbf{F}_i^{\text{int}} = 0$ for all i and this formula reduces to

$$PV = N k_B T, \tag{5.79}$$

the familiar equation of state for an ideal gas.

It is sometimes desirable to control the pressure in a simulation, for example to replicate experimental conditions in the NPT ensemble. Many barostat algorithms exist for this purpose [2], but we have not used them in this work.

5.10 Normal Mode Analysis

It is often useful to understand the collective motion of atoms in a molecule. A technique called normal mode analysis allows us to decompose a complex molecular configuration into a weighted sum of these collective motions.

In a system of N atoms, let us define the displacement, force and acceleration of atom i as $\mathbf{x}_i$, $\mathbf{f}_i$ and $\mathbf{a}_i$ respectively. The mass of atom i is given by m_i. The potential energy of the system is a function of the particle positions only, and is denoted $V(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N)$. Let us define the following vectors and matrices:

$$\mathbf{F} = \begin{pmatrix} \mathbf{f}_1 \\ \vdots \\ \mathbf{f}_N \end{pmatrix}, \quad \mathbf{X} = \begin{pmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_N \end{pmatrix}, \quad \mathbf{A} = \begin{pmatrix} \mathbf{a}_1 \\ \vdots \\ \mathbf{a}_N \end{pmatrix}.$$
$$\mathbf{M} = \begin{pmatrix} m_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & m_N \end{pmatrix}. \tag{5.80}$$

We can then write Newton's second law as

$$\mathbf{F} = \mathbf{MA}. \tag{5.81}$$

If the displacements of the atoms from equilibrium are small enough, the restoring forces are given by Hooke's law,

$$\mathbf{F} = -\frac{\partial V}{\partial \mathbf{X}} = -\mathbf{KX}, \tag{5.82}$$

where $\mathbf{K}$ is the force constant matrix, also know as the Hessian. Differentiating this with respect to $\mathbf{X}$, we obtain

$$\mathbf{K} = \frac{\partial^2 V}{\partial \mathbf{X}^2} = \begin{pmatrix} \frac{\partial^2 V}{\partial \mathbf{X}_1^2} & \frac{\partial^2 V}{\partial \mathbf{X}_1 \mathbf{X}_2} & \cdots & \frac{\partial^2 V}{\partial \mathbf{X}_1 \mathbf{X}_N} \\ \frac{\partial^2 V}{\partial \mathbf{X}_2 \mathbf{X}_1} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ \frac{\partial^2 V}{\partial \mathbf{X}_N \mathbf{X}_1} & \cdots & \cdots & \frac{\partial^2 V}{\partial \mathbf{X}_N^2} \end{pmatrix}, \tag{5.83}$$

where each element in $\mathbf{K}$ is given by the 3×3 matrix

$$K_{ij} = \begin{pmatrix} \frac{\partial^2 V}{\partial x_i \partial x_j} & \frac{\partial^2 V}{\partial x_i \partial y_j} & \frac{\partial^2 V}{\partial x_i \partial z_j} \\ \frac{\partial^2 V}{\partial y_i \partial x_j} & \frac{\partial^2 V}{\partial y_i \partial y_j} & \frac{\partial^2 V}{\partial y_i \partial z_j} \\ \frac{\partial^2 V}{\partial z_i \partial x_j} & \frac{\partial^2 V}{\partial z_i \partial y_j} & \frac{\partial^2 V}{\partial z_i \partial z_j} \end{pmatrix}. \tag{5.84}$$

Equating Newton's second law with Hooke's law, we find

$$\mathbf{M}\ddot{\mathbf{X}} = -\mathbf{KX}. \tag{5.85}$$

Let us make the substitution $\mathbf{Y} = \sqrt{\mathbf{M}}\mathbf{X}$. Substituting into Eqn. 5.85, we find

$$\begin{aligned} &\sqrt{\mathbf{M}}\ddot{\mathbf{Y}} = -\mathbf{K}\mathbf{M}^{-1/2}\mathbf{Y} \\ &\implies \ddot{\mathbf{Y}} = -\mathbf{M}^{-1/2}\mathbf{K}\mathbf{M}^{-1/2}\mathbf{Y} = -\tilde{\mathbf{K}}\mathbf{Y}. \end{aligned} \tag{5.86}$$

Let us assume a solution for $\mathbf{Y}$ in the form

$$\mathbf{Y}(t) = \mathbf{A}\cos(\omega t + \phi), \tag{5.87}$$

where $\mathbf{A}$ is a constant vector, and ω and ϕ are an unknown frequency and phase shift respectively. Substituting into Eqn. (5.86), we find

$$\tilde{\mathbf{K}}\mathbf{A} = \omega^2\mathbf{A}$$
$$\implies (\tilde{\mathbf{K}} - \omega^2\mathbf{I})\mathbf{A} = 0. \tag{5.88}$$

So our assumed solution is correct as long as $\det(\tilde{\mathbf{K}} - \omega^2\mathbf{I}) = 0$. Eqn. (5.88) is an eigenvector equation, which can be solved by diagonalisation or other methods. Solving it will give us $3N$ different normal mode frequencies ω_α, with $3N$ corresponding eigenvectors $\mathbf{e}_\alpha$.

Now we can write the general solution of Eqn. (5.86) as

$$\mathbf{Y}(t) = \sum_{\alpha=1}^{3N} c_\alpha \mathbf{e}_\alpha \cos(\omega_\alpha t + \phi_\alpha), \tag{5.89}$$

so the $3N$-dimensional vector $\mathbf{Y}$, which describes the displacements of all the N atoms as a function of time, can be expressed as an expansion in terms of the normal mode amplitudes c_α.

Many MD codes contain built–in functionality for calculating the normal modes of a molecular system. This is done by computing the Hessian matrix numerically, using the potential energy of the chosen force field. The Hessian is then diagonalised to find the eigenvalues and eigenvectors. The Hessian matrix is symmetric, so various tricks can be used to perform fast diagonalization, even for very large molecules.

Chapter 6

A System of Two Dimers

In this chapter, we will examine a very simple system, namely a pair of interacting dimers in one dimension. We will then coarse–grain the dimers into a pair of CG particles using a variety of different CG methods. The system is simple enough that the coarse–graining can be performed analytically. This simple model will then provide insight when we perform coarse–graining on a more complex system in Chapters 8 and 9.

6.1 The Microscopic System

The microscopic system we are going to coarse grain is shown in Figure 6.1. Two dimers, each of natural length a_0, are constrained to point along the x axis. The centre of mass of the left-hand dimer is at the origin and the centre of mass of the right-hand dimer at $x = d$. Because the dimers interact, their lengths a (which must be the same) need not equal a_0. The atoms in the left-hand dimer are at $x_-^L = -a/2$ and $x_+^L = a/2$; the atoms in the right-hand dimer are at $x_-^R = d - a/2$ and $x_+^R = d + a/2$.

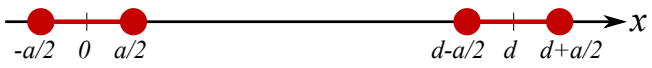


Figure 6.1: Two dimers, each of length a. The centre of mass of the left-hand dimer is at the origin and that of the right-hand dimer at $x = d$. Both dimers are constrained to point along the x axis.

The intra-dimer bonding energy of two atoms at positions x_1 and x_2 is given

by

$$V_{\text{dimer}}(x_1, x_2) = u_{\text{dimer}}(x_2 - x_1). \tag{6.1}$$

Atoms in different dimers, at positions x_1 and x_2, interact via a long-ranged potential

$$V_{\text{lr}}(x_1, x_2) = u_{\text{lr}}(x_2 - x_1), \tag{6.2}$$

the minima of which occur at $x_2 - x_1 = \pm b$.

For simplicity, we assume that the intra-dimer chemical bonding energy is a harmonic function of the dimer length,

$$u_{\text{dimer}}(a) = \frac{1}{2}k(a - a_0)^2, \tag{6.3}$$

where k is the spring constant and a_0 the equilibrium bond length. This assumption should be accurate as long as the inter-dimer forces are small compared to the force required to stretch a single dimer significantly. A typical choice for $u_{\text{lr}}(x)$ is the Lennard-Jones potential,

$$u_{\text{LJ}}(x) = \epsilon \left[\left(\frac{b}{x} \right)^{12} - 2 \left(\frac{b}{x} \right)^{6} \right], \tag{6.4}$$

where ϵ is the well depth. This is illustrated in Figure 6.2, along with its first and second derivatives. The Lennard-Jones force exerted on an atom at x_2 by another atom at x_1 is

$$f_{\text{LJ}}(x_2 - x_1) = -u'_{\text{LJ}}(x_2 - x_1) = 12\epsilon \left(\frac{b^{12}}{(x_2 - x_1)^{13}} - \frac{b^6}{(x_2 - x_1)^7} \right). \tag{6.5}$$

Note that this equation yields the correct sign for the force on atom 2 regardless of whether x_2 is less than or greater than x_1. The force on atom 1, given by $f_{\text{LJ}}(x_1 - x_2)$, is equal and opposite to the force on atom 2.

The total potential energy of the system of two dimers when the centre-of-mass spacing is fixed equal to d is

$$V_{\text{total}}(d, a) = 2u_{\text{dimer}}(a) + 2u_{\text{lr}}(d) + u_{\text{lr}}(d + a) + u_{\text{lr}}(d - a). \tag{6.6}$$

If, for simplicity, we assume that a is small relative to the length scale on which u_{lr} changes significantly, we can expand $u_{\text{lr}}(d \pm a)$ as a Taylor series about $a = 0$ to obtain

$$V_{\text{total}}(d, a) \approx 2u_{\text{dimer}}(a) + 4u_{\text{lr}}(d) + a^2 u''_{\text{lr}}(d). \tag{6.7}$$

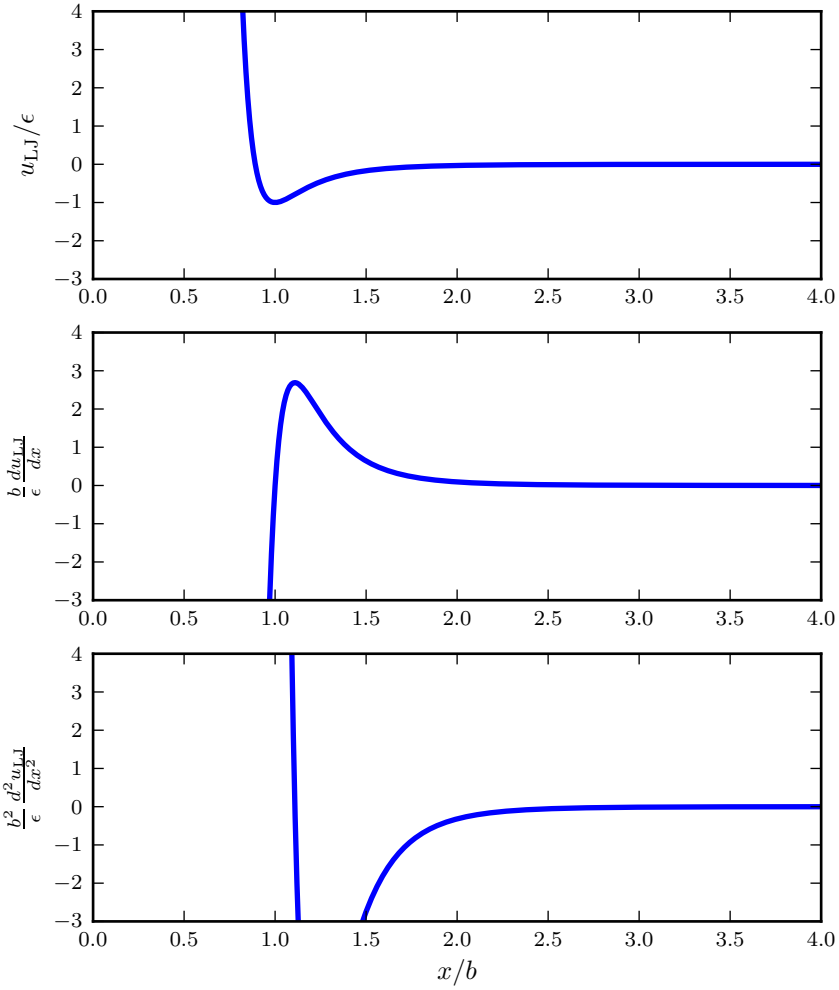


Figure 6.2: The Lennard–Jones potential and its first two derivatives. The cut–off in the bottom panel is due to the choice of scale, and not a physical discontinuity.

This approximation is so convenient that we shall use it frequently from now on.

The forces on the four atoms are

$$f_-^L(d,a) = \quad k(a-a_0) - u'_{\rm lr}(-d) - u'_{\rm lr}(-(d+a)), \tag{6.8}$$

$$f_+^L(d,a) = -k(a-a_0) - u'_{\rm lr}(-d) - u'_{\rm lr}(-(d-a)), \tag{6.9}$$

$$f_-^R(d,a) = \quad k(a-a_0) - u'_{\rm lr}(d) - u'_{\rm lr}(d-a), \tag{6.10}$$

$$f_+^R(d,a) = -k(a-a_0) - u'_{\rm lr}(d) - u'_{\rm lr}(d+a), \tag{6.11}$$

or, if we assume that a is small enough to allow us to Taylor expand,

$$f_-^L(d,a) \approx \quad k(a-a_0) - 2u'_{\rm lr}(-d) + au''_{\rm lr}(-d), \tag{6.12}$$

$$f_+^L(d,a) \approx -k(a-a_0) - 2u'_{\rm lr}(-d) - au''_{\rm lr}(-d), \tag{6.13}$$

$$f_-^R(d,a) \approx \quad k(a-a_0) - 2u'_{\rm lr}(d) - au''_{\rm lr}(d), \tag{6.14}$$

$$f_+^R(d,a) \approx -k(a-a_0) - 2u'_{\rm lr}(d) + au''_{\rm lr}(d). \tag{6.15}$$

6.2 Coarse Graining at Zero Temperature

6.2.1 The Rigid Approach

In the rigid approach, we freeze the dimer bond length a at its equilibrium value a_0 for an isolated dimer. The total potential energy then depends on the dimer spacing d only and is given by

$$\begin{aligned} V_{\rm total}^{\rm rigid}(d) &= 2u_{\rm dimer}(a_0) + 2u_{\rm lr}(d) + u_{\rm lr}(d+a_0) + u_{\rm lr}(d-a_0) \\ &= 2u_{\rm lr}(d) + u_{\rm lr}(d+a_0) + u_{\rm lr}(d-a_0) \\ &\approx 4u_{\rm lr}(d) + a_0^2 u''_{\rm lr}(d). \end{aligned} \tag{6.16}$$

This result can also be derived using the principle of virtual work. Two frozen dimers at centre-of-mass separation d attract or repel each other depending on the value of d. To hold them in position, it is necessary to apply external forces to cancel the long-ranged forces of interaction between the dimers. These

external forces are given by

$$\begin{aligned} f^L_{\rm ext}(d) &= -f^L_-(d,a_0) - f^L_+(d,a_0) \\ &= 2u'_{\rm lr}(-d) + u'_{\rm lr}(-(d+a_0)) + u'_{\rm lr}(-(d-a_0)), \qquad (6.17) \\ f^R_{\rm ext}(d) &= -f^R_-(d,a_0) - f^R_+(d,a_0) \\ &= 2u'_{\rm lr}(d) + u'_{\rm lr}(d-a_0) + u'_{\rm lr}(d+a_0). \qquad (6.18) \end{aligned}$$

Imagine holding the left-hand dimer fixed at the origin and gradually moving the centre of mass of the right-hand dimer in from $x = \infty$ to $x = d$. The externally applied force is adjusted as the right-hand dimer moves to ensure that it exactly cancels the internal force on the centre-of-mass at all times. The work W done by the external force during this process is

$$\begin{aligned} W &= \int_\infty^d f^R_{\rm ext}(x)\,dx \\ &= \int_\infty^d 2u'_{\rm lr}(x) + u'_{\rm lr}(x-a_0) + u'_{\rm lr}(x+a_0)\,dx \\ &= \left[2u_{\rm lr}(x) + u_{\rm lr}(x-a_0) + u_{\rm lr}(x+a_0)\right]_\infty^d \\ &= 2u_{\rm lr}(d) + u_{\rm lr}(d-a_0) + u_{\rm lr}(d+a_0). \qquad (6.19) \end{aligned}$$

The work done must equal the change in potential energy of the system. In this case, since the integration starts at $x = \infty$, where the potential energy is defined to be zero, the change in potential energy is equal to the total potential energy. Hence

$$\begin{aligned} V^{\rm rigid}_{\rm total}(d) &= 2u_{\rm lr}(d) + u_{\rm lr}(d-a_0) + u_{\rm lr}(d+a_0) \\ &\approx 4u_{\rm lr}(d) + a_0^2 u''_{\rm lr}(d), \qquad (6.20) \end{aligned}$$

as above.

6.2.2 The Relaxed Dimer Approach

In the relaxed dimer approach, the bond length a is allowed to adjust to the value $a_{\rm eq}(d)$ that minimises the total potential energy $V_{\rm total}(d,a)$ at each centre-of-mass spacing d. This is expected to be a good approximation if the time scale on which dimer collisions takes place is much longer than the vibrational time period of a single dimer. The approach and scattering of the two dimers then happens slowly enough to allow the bond length to adjust almost adiabatically

and remain very close to its equilibrium value at all times. If the vibrational time scale is comparable to or slower than the collision time, both dimers will be left in vibrationally excited states after the collision and the relaxed dimer approach will be inaccurate.

Setting the differential of Eq. (6.6) with respect to a at constant d equal to zero yields an equation for the relaxed bond length $a_{\rm eq}(d)$ at centre-of-mass separation d:

$$2k(a_{\rm eq}(d) - a_0) + u'_{\rm lr}(d + a_{\rm eq}(d)) - u'_{\rm lr}(d - a_{\rm eq}(d)) = 0. \tag{6.21}$$

This equation is easier to solve if we make the assumption that $a_{\rm eq}(d)$ is small, so that $u'_{\rm lr}(d \pm a_{\rm eq}(d)) \approx u'_{\rm lr}(d) \pm a_{\rm eq}(d)u''_{\rm lr}(d)$. Substituting this approximation back into Eq. (6.21) yields

$$\Delta a_{\rm eq}(d) = a_{\rm eq}(d) - a_0 \approx \frac{-a_0 u''_{\rm lr}(d)}{k + u''_{\rm lr}(d)}. \tag{6.22}$$

The total potential energy of the relaxed dimer with centre-of-mass spacing d is

$$V_{\rm total}^{\rm relaxed}(d) = 2u_{\rm dimer}(a_{\rm eq}(d)) + 2u_{\rm lr}(d) + u_{\rm lr}(d + a_{\rm eq}(d)) + u_{\rm lr}(d - a_{\rm eq}(d)). \tag{6.23}$$

Remembering that $u_{\rm dimer}(a) = \frac{1}{2}k(a - a_0)^2$ and Taylor expanding $u_{\rm lr}(d \pm a_{\rm eq}(d))$ yields

$$\begin{aligned}
V_{\rm total}^{\rm relaxed}(d) &\approx k(\Delta a_{\rm eq}(d))^2 + 4u_{\rm lr}(d) + (a_{\rm eq}(d))^2 u''_{\rm lr}(d) \\
&= k(\Delta a_{\rm eq}(d))^2 + 4u_{\rm lr}(d) + (a_0 + \Delta a_{\rm eq}(d))^2 u''_{\rm lr}(d) \\
&= 4u_{\rm lr}(d) + a_0^2 u''_{\rm lr}(d) + 2a_0 \Delta a_{\rm eq}(d) u''_{\rm lr}(d) + (k + u''_{\rm lr}(d))(\Delta a_{\rm eq}(d))^2 \\
&= V_{\rm total}^{\rm rigid}(d) + 2a_0 \left(\frac{-a_0 u''_{\rm lr}(d)}{k + u''_{\rm lr}(d)} \right) u''_{\rm lr}(d) + (k + u''_{\rm lr}(d)) \left(\frac{-a_0 u''_{\rm lr}(d)}{k + u''_{\rm lr}(d)} \right)^2 \\
&= V_{\rm total}^{\rm rigid}(d) - \frac{(a_0 u''_{\rm lr}(d))^2}{k + u''_{\rm lr}(d)}.
\end{aligned} \tag{6.24}$$

Assuming that the harmonic springs between the two atoms in a dimer are strong compared with the long-ranged interactions between dimers ($k \gg |u''_{\rm lr}(d)|$), we see that $V_{\rm total}^{\rm relaxed}(d) < V_{\rm total}^{\rm rigid}(d)$. This is as expected, of course. The relaxed potential energy tends to the rigid potential energy in the limit as the spring constant $k \to \infty$, again as expected.

Figure 6.3 shows a comparison between the rigid potential and the relaxed potential (as calculated in the small $a_{\rm eq}(d)$ approximation) for two dimers with $\frac{a_0}{b} = 0.1$ and $\frac{kb^2}{2\epsilon} = 10$.

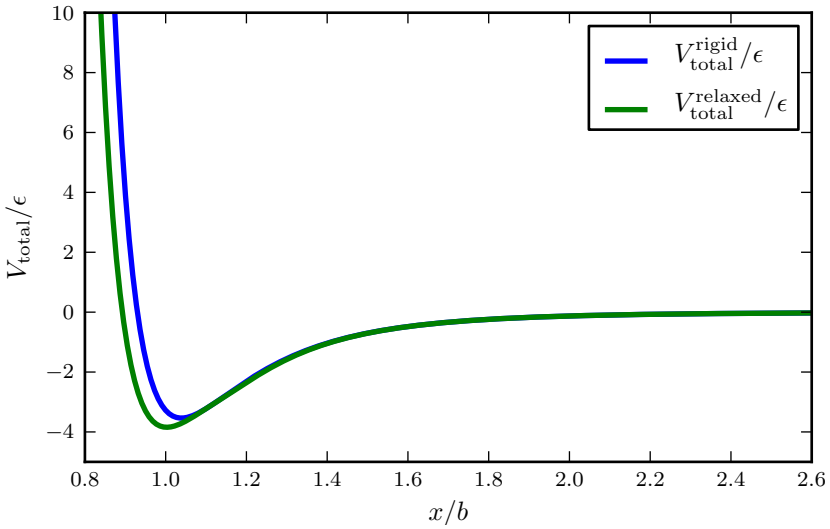


Figure 6.3: The rigid and relaxed potentials for the case when $a_0/b = 0.1$ (the equilibrium dimer bond length is a tenth the equilibrium spacing of two unbonded atoms) and $\frac{kb^2}{2\epsilon} = 10$ (stretching a bond to the equilibrium spacing of two unbonded atoms requires energy 10ϵ).

We can also derive this result using forces. Let us first consider the general case of a system with potential energy $V_{\text{total}}(\boldsymbol{r}) = V_{\text{total}}(q, \xi)$, where $\boldsymbol{r} = (q, \xi)$ is a multi-dimensional position coordinate consisting of "microscopic" coordinates ξ, which we intend to coarse-grain away, and "macroscopic" coordinates q, which we intend to keep. The corresponding relaxed potential is

$$V_{\text{total}}^{\text{relaxed}}(q) = V_{\text{total}}(q, \xi_{\text{eq}}(q)), \tag{6.25}$$

where $\xi_{\text{eq}}(q)$ is determined by solving the equation

$$\frac{\partial V_{\text{total}}(q, \xi)}{\partial \xi} = 0. \tag{6.26}$$

to obtain the relaxed microscopic coordinates for any choice of macroscopic coordinates q. Let us choose a reference point $\boldsymbol{r}_0 = (q_0, \xi_0)$ at which V_{total} is defined to be zero. (In our two-dimer example, $\boldsymbol{r}_0$ is the point $(d = \infty, a = a_0)$.) The potential energy at any other point $\boldsymbol{r} = (q, \xi)$ may be obtained by integration along a path from this reference point:

$$V_{\text{total}}(\boldsymbol{r}) = \int_{\boldsymbol{r}_0}^{\boldsymbol{r}} \boldsymbol{\nabla} V_{\text{total}}(\boldsymbol{r}') \cdot d\boldsymbol{r}'. \tag{6.27}$$

The value of this integral is independent of the path chosen, as long as the start and end points are fixed. Writing the path integral in terms of q and ξ gives

$$V_{\text{total}}(q, \xi) = \int_{(q_0, \xi_0)}^{(q, \xi)} \frac{\partial V_{\text{total}}(q', \xi')}{\partial q'} dq' + \frac{\partial V_{\text{total}}(q', \xi')}{\partial \xi'} d\xi'. \tag{6.28}$$

Suppose now that the microscopic coordinates are fully relaxed at the reference point, $\xi_0 = \xi_{\text{eq}}(q_0)$, and that we choose to evaluate the integral along a path that only passes through other relaxed points $(q', \xi_{\text{eq}}(q'))$, ending at $(q, \xi_{\text{eq}}(q))$. Equation (6.26) tells us that $\partial V_{\text{total}}/\partial \xi$ is zero everywhere on this path, so Eq. (6.28) simplifies to

$$V_{\text{total}}(q, \xi_{\text{eq}}(q)) = \int_{q_0}^{q} \left. \frac{\partial V_{\text{total}}(q', \xi')}{\partial q'} \right|_{\xi' = \xi_{\text{eq}}(q')} dq'. \tag{6.29}$$

This tells us that we can obtain $V_{\text{total}}^{\text{relaxed}}(q) = V_{\text{total}}(q, \xi_{\text{eq}}(q))$ by integrating the forces on the macroscopic coordinates q only, as long as the microscopic coordinates ξ are fully relaxed at every point on the path of integration. No external forces are then required to hold the microscopic coordinates in place, and all of the virtual work is done on the macroscopic coordinates.

We now return to the special case of the two dimers. As in Sec. (6.2.1), imagine applying external forces to counteract the inter-dimer force and make the distance between the centres of mass of the two dimers equal to d. This time, however, as the dimer is not rigid, we have to apply the external forces to the individual atoms, not to the dimer as a whole. To stop the externally applied forces affecting the dimer bond length, we apply half of the total external force on a dimer to each atom in that dimer. Eqs. (6.8) to (6.11) then become

$$f_-^L(d,a) = \frac{1}{2}f_{\rm ext}^L + k(a-a_0) - u'_{\rm lr}(-d) - u'_{\rm lr}(-(d+a)), \tag{6.30}$$

$$f_+^L(d,a) = \frac{1}{2}f_{\rm ext}^L - k(a-a_0) - u'_{\rm lr}(-d) - u'_{\rm lr}(-(d-a)), \tag{6.31}$$

$$f_-^R(d,a) = \frac{1}{2}f_{\rm ext}^R + k(a-a_0) - u'_{\rm lr}(d) - u'_{\rm lr}(d-a), \tag{6.32}$$

$$f_+^R(d,a) = \frac{1}{2}f_{\rm ext}^R - k(a-a_0) - u'_{\rm lr}(d) - u'_{\rm lr}(d+a). \tag{6.33}$$

In equilibrium, all of these forces must be zero. Adding and subtracting Eqs. (6.30) and (6.31) gives

$$f_{\rm ext}^L = 2u'_{\rm lr}(-d) + u'_{\rm lr}(-(d+a)) + u'_{\rm lr}(-(d-a)), \tag{6.34}$$

$$2k(a-a_0) = u'_{\rm lr}(-(d+a)) - u'_{\rm lr}(-(d-a)). \tag{6.35}$$

Since $u_{\rm lr}(x)$ is an even function of x, $u'_{\rm lr}(x)$ must be odd. These equations can therefore be rewritten in the form

$$f_{\rm ext}^L = -2u'_{\rm lr}(d) - u'_{\rm lr}(d+a) - u'_{\rm lr}(d-a), \tag{6.36}$$

$$2k(a-a_0) = -u'_{\rm lr}(d+a) + u'_{\rm lr}(d-a). \tag{6.37}$$

Similarly, adding and subtracting Eqs. (6.32) and (6.33) gives

$$f_{\rm ext}^R = 2u'_{\rm lr}(d) + u'_{\rm lr}(d-a) + u'_{\rm lr}(d+a), \tag{6.38}$$

$$2k(a-a_0) = u'_{\rm lr}(d-a) - u'_{\rm lr}(d+a). \tag{6.39}$$

Eqs. (6.37) and (6.39) are equivalent, implying that both dimers have the same value of a for any given d. They are also the same as Eq. (6.21), which was derived via the energy minimisation approach. The external forces $f_{\rm ext}^L$ and $f_{\rm ext}^R$, as given by Eqs. (6.36) and (6.38), are equal and opposite. This is all as expected.

The work done by the external forces as the centre of mass of the right-hand dimer is moved in from $x = \infty$ to $x = d$ is

$$\begin{aligned} W &= \int_{\infty}^{d} f_{\text{ext}}^{R}(x)\, dx \\ &= \int_{\infty}^{d} 2u'_{\text{lr}}(x) + u'_{\text{lr}}(x - a_{\text{eq}}(x)) + u'_{\text{lr}}(x + a_{\text{eq}}(x))\, dx. \end{aligned} \tag{6.40}$$

Since the external work performed on the system must equal the change in potential energy, this yields

$$V_{\text{total}}^{\text{relaxed}}(d) = \int_{\infty}^{d} 2u'_{\text{lr}}(x) + u'_{\text{lr}}(x - a_{\text{eq}}(x)) + u'_{\text{lr}}(x + a_{\text{eq}}(x))\, dx. \tag{6.41}$$

Remembering that

$$V_{\text{total}}(d, a) = 2u_{\text{dimer}}(a) + 2u_{\text{lr}}(d) + u_{\text{lr}}(d + a) + u_{\text{lr}}(d - a), \tag{6.42}$$

we see that we can also write $V_{\text{total}}^{\text{relaxed}}(d)$ in the form

$$V_{\text{total}}^{\text{relaxed}}(d) = \int_{\infty}^{d} \left.\frac{\partial V_{\text{total}}(x, a)}{\partial x}\right|_{a=a_{\text{eq}}(x)} dx. \tag{6.43}$$

This is exactly analogous to Eq. (6.29), so

$$V_{\text{total}}^{\text{relaxed}}(d) = V_{\text{total}}(d, a_{\text{eq}}(d)), \tag{6.44}$$

as expected. Expressing this result in terms of u_{dimer} and u_{lr} gives

$$V_{\text{total}}^{\text{relaxed}} = 2u_{\text{dimer}}(a_{\text{eq}}(d)) + 2u_{\text{lr}}(d) + u_{\text{lr}}(d + a_{\text{eq}}(d)) + u_{\text{lr}}(d - a_{\text{eq}}(d)), \tag{6.45}$$

in agreement with Eq. (6.23), which was obtained directly from the potential energy.

6.3 Coarse–Graining at Finite Temperature

We now consider the finite temperature case. Firstly, we can no longer assume that both dimers are the same length. Let the left and right dimers have bond lengths a_L and a_R respectively. Eqn. (6.6) for the total potential energy of the system of two dimers becomes

$$\begin{aligned} V_{\text{total}}(d, a_L, a_R) = {} & u_{\text{dimer}}(a_L) + u_{\text{dimer}}(a_R) \\ & + u_{\text{LR}}\left(d + \frac{a_R}{2} + \frac{a_L}{2}\right) + u_{\text{LR}}\left(d - \frac{a_R}{2} + \frac{a_L}{2}\right) \\ & + u_{\text{LR}}\left(d - \frac{a_R}{2} - \frac{a_L}{2}\right) + u_{\text{LR}}\left(d + \frac{a_R}{2} - \frac{a_L}{2}\right). \end{aligned} \tag{6.46}$$

For a fixed centre–of–mass spacing d, both dimers have the same equilibrium length $a_{\rm eq}(d)$. We can therefore write their actual lengths as a displacement from this equilibrium value,

$$\begin{aligned} a_L &= a_{\rm eq}(d) + \Delta a_L, \\ a_R &= a_{\rm eq}(d) + \Delta a_R. \end{aligned} \tag{6.47}$$

Following the approach in Eqn. (6.21), $V_{\rm total}(d, a_L, a_R)$ is at a minimum for fixed d when $a_L = a_R = a_{\rm eq}(d)$. In other words,

$$\left.\frac{\partial V}{\partial a_L}\right|_{a_{\rm eq}(d)} = \left.\frac{\partial V}{\partial a_R}\right|_{a_{\rm eq}(d)} = 0. \tag{6.48}$$

This implies that a two–dimensional Taylor expansion about the minimum in terms of Δa_L and Δa_R yields

$$\begin{aligned} V(d, a_L, a_R) \approx\ & V(d, a_{\rm eq}(d), a_{\rm eq}(d)) \\ & + \frac{1}{2}\left.\frac{\partial^2 V}{\partial {a_L}^2}\right|_{a_{\rm eq}(d)} (\Delta a_L)^2 \\ & + \left.\frac{\partial^2 V}{\partial a_L \partial a_R}\right|_{a_{\rm eq}(d)} \Delta a_L \Delta a_R \\ & + \frac{1}{2}\left.\frac{\partial^2 V}{\partial {a_R}^2}\right|_{a_{\rm eq}(d)} (\Delta a_R)^2. \end{aligned} \tag{6.49}$$

Now

$$\begin{aligned} &\left.\frac{\partial^2 V}{\partial {a_L}^2}\right|_{a_{\rm eq}(d)} = \left.\frac{\partial^2 V}{\partial {a_R}^2}\right|_{a_{\rm eq}(d)} \\ &= k + \frac{1}{4}\left[u''_{\rm LR}(d - a_{\rm eq}(d)) + 2u''_{\rm LR}(d) + u''_{\rm LR}(d + a_{\rm eq}(d))\right], \end{aligned} \tag{6.50}$$

and

$$\left.\frac{\partial^2 V}{\partial a_L \partial a_R}\right|_{a_{\rm eq}(d)} = \frac{1}{4}\left[u''_{\rm LR}(d - a_{\rm eq}(d)) - 2u''_{\rm LR}(d) + u''_{\rm LR}(d + a_{\rm eq}(d))\right]. \tag{6.51}$$

Let us define the spring constant matrix

$$\mathbf{K} = \begin{pmatrix} \left.\frac{\partial^2 V}{\partial {a_L}^2}\right|_{a_{\rm eq}(d)} & \left.\frac{\partial^2 V}{\partial a_L \partial a_R}\right|_{a_{\rm eq}(d)} \\ \left.\frac{\partial^2 V}{\partial a_R \partial a_L}\right|_{a_{\rm eq}(d)} & \left.\frac{\partial^2 V}{\partial {a_R}^2}\right|_{a_{\rm eq}(d)} \end{pmatrix}. \tag{6.52}$$

Then we can write Eqn. (6.46) as

$$V(d, a_L, a_R) \approx V(d, a_{\rm eq}(d), a_{\rm eq}(d)) + \frac{1}{2}(\Delta a_L, \Delta a_R) \begin{pmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{pmatrix} \begin{pmatrix} \Delta a_L \\ \Delta a_R \end{pmatrix}. \tag{6.53}$$

$\mathbf{K}$ is a matrix of the form $\begin{pmatrix} a & b \\ b & a \end{pmatrix}$. This has eigenvalues $\lambda_\pm = a \pm b$, and eigenvectors $e_+ = \frac{1}{\sqrt{2}}(1,1)$ and $e_- = \frac{1}{\sqrt{2}}(1,-1)$. Substituting in the correct values of a and b, we find that

$$\begin{aligned} \lambda_+ &= k + \frac{1}{2}\left[u''_{\rm LR}(d - a_{\rm eq}(d)) + u''_{\rm LR}(d + a_{\rm eq}(d))\right], \\ \lambda_- &= k + u''_{\rm LR}(d). \end{aligned} \tag{6.54}$$

We are now in a position to evaluate the configurational partition function. For a fixed d, it is given by

$$Z(d) = \int \exp\left[-\beta\left(V(d, a_{\rm eq}(d), a_{\rm eq}(d)) + \frac{1}{2}\Delta\mathbf{a}^\intercal\mathbf{K}\Delta\mathbf{a}\right)\right] d^2(\Delta\mathbf{a}), \tag{6.55}$$

where $\Delta\mathbf{a} = (\Delta a_L, \Delta a_R)$ and $d^2(\Delta\mathbf{a}) = d(\Delta a_L)d(\Delta a_R)$. We can factor out the first term from the integral to arrive at

$$Z(d) = e^{-\beta V(d, a_{\rm eq}(d))} \int \exp\left[-\frac{\beta}{2}\left(\Delta\mathbf{a}^\intercal\mathbf{K}\Delta\mathbf{a}\right)\right] d^2(\Delta\mathbf{a}). \tag{6.56}$$

Let us transform to the basis of eigenvectors. This is a unitary transformation with unit Jacobian, so

$$\begin{aligned} Z(d) &= e^{-\beta V(d, a_{\rm eq}(d))} \int \exp\left[-\frac{\beta}{2}\lambda_+ u_+^2 - \frac{\beta}{2}\lambda_- u_-^2\right] du_+ du_- \\ &= e^{-\beta V(d, a_{\rm eq}(d))} \sqrt{\frac{2\pi}{\lambda_+\beta}}\sqrt{\frac{2\pi}{\lambda_-\beta}} \\ &= e^{-\beta V(d, a_{\rm eq}(d))} \frac{2\pi}{\beta}\left[k + \frac{1}{2}u''_{\rm LR}(d - a_{\rm eq}(d)) + \frac{1}{2}u''_{\rm LR}(d + a_{\rm eq}(d))\right]^{-1/2} \\ &\times \left[k + u''_{\rm LR}(d)\right]^{-1/2}. \end{aligned} \tag{6.57}$$

If we assume $a_{\rm eq}(d) << d$, this can be simplified to

$$Z(d) \approx \frac{2\pi}{\beta} e^{-\beta V(d, a_{\rm eq}(d))} \frac{1}{k + u''_{\rm LR}(d)}. \tag{6.58}$$

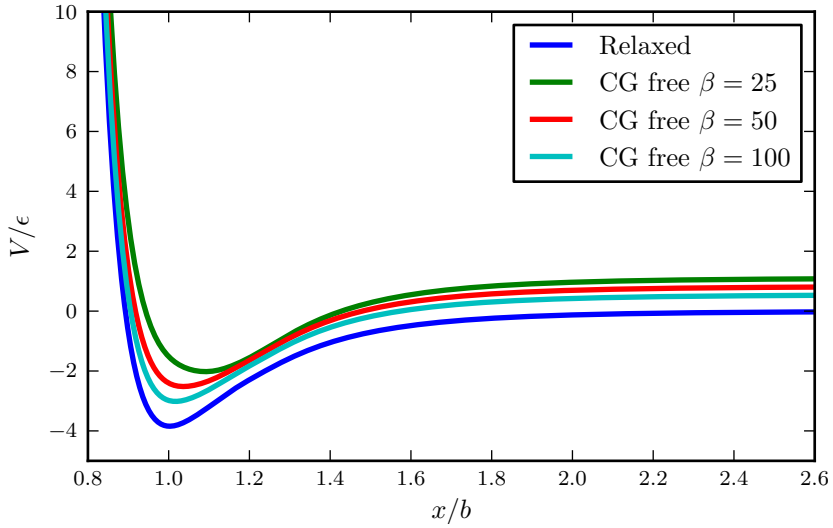


Figure 6.4: The CG free energy from Eqn. (6.59), plotted for a range of values of β. In this plot, $a_0/b = 0.1$ and $\frac{kb^2}{2\epsilon} = 10$. The relaxed potential is plotted for comparison, and represents the zero temperature limit of F_{CG}. F_{CG} itself exhibits a strong temperature dependence which affects the shape of the potential well. There is also a shift in the asymptotic value of the potential at large separations. This is due to the increase in mean energy of an isolated dimer as the temperature increases. It has no effect on the dynamics since the curvature of the potential is not affected.

The CG free energy is then given by

$$\begin{aligned} F_{\mathrm{CG}}(\beta, d) &= -\frac{1}{\beta} \ln Z(d) \\ &= -\frac{1}{\beta} \left[\ln 2\pi - \ln \beta - \beta V(d, a_{\mathrm{eq}}(d)) - \ln(k + u''_{\mathrm{LR}}(d)) \right] \\ &= V(d, a_{\mathrm{eq}}(d)) - \frac{1}{\beta} \ln \left[\frac{2\pi}{\beta(k + u''_{\mathrm{LR}}(d))} \right]. \end{aligned} \tag{6.59}$$

Clearly, this formula reduces to the relaxed potential given in Eqn. (6.23) as the temperature tends to zero (or β tends to infinity). We have plotted F_{CG} for various values of β in Fig. 6.4.

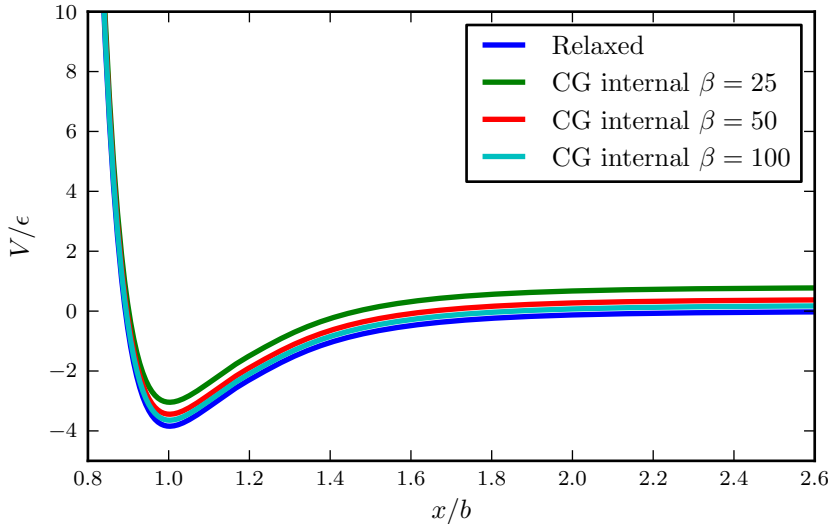


Figure 6.5: The CG internal energy from Eqn. (6.59), plotted for a range of values of β. In this plot, $a_0/b = 0.1$ and $\frac{kb^2}{2\epsilon} = 10$. The relaxed potential is plotted for comparison, and represents the zero temperature limit of U_{CG}. U_{CG} exhibits no temperature dependence, except for a shift which has no effect on the dynamics since the curvature of the potential is not affected.

The CG internal energy is given by

$$\begin{aligned} U_{\mathrm{CG}}(\beta, d) &= -\frac{\partial \ln Z(d)}{\partial \beta} \\ &= -\frac{\partial}{\partial \beta}\left[\ln 2\pi - \ln \beta - \beta V(d, a_{\mathrm{eq}}(d)) - \ln(k + u''_{\mathrm{LR}}(d))\right] \\ &= k_{\mathrm{B}}T + V(d, a_{\mathrm{eq}}(d)). \end{aligned} \tag{6.60}$$

This is equal to the relaxed potential energy, plus a temperature-dependent shift of $k_{\mathrm{B}}T/2$ per degree of freedom that has been integrated out of the system. This shift can have no effect on the dynamics, since it does not affect the curvature of the potential. The internal energy is shifted by this factor of $k_{\mathrm{B}}T/2$ only because we are integrating out harmonic degrees of freedom. In the event that the intramolecular interactions are anharmonic, U_{CG} will in general have a more complicated dependence on temperature. U_{CG} is plotted in Fig. 6.5.

Recalling the expression $F = U - TS$ from thermodynamics, we may calcu-

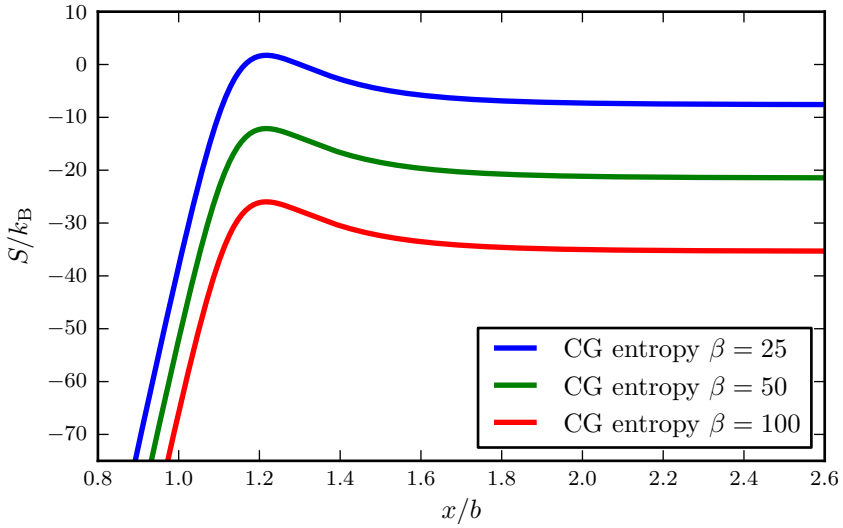


Figure 6.6: The CG entropy from Eqn. (6.61), plotted for a range of values of β. In this plot, $a_0/b = 0.1$ and $\frac{kb^2}{2\epsilon} = 10$. The temperature dependence of F_{CG} is due to the change in shape of the CG entropy as a function of β.

late the CG entropy as

$$\begin{aligned} S_{\mathrm{CG}}(\beta, d)/k_{\mathrm{B}} &= \beta \left[U_{\mathrm{CG}}(\beta, d) - F_{\mathrm{CG}}(\beta, d) \right] \\ &= \ln \left[\frac{2\pi}{\beta(k + u''_{\mathrm{LR}}(d))} \right] + 1. \end{aligned} \tag{6.61}$$

The temperature-dependence of F_{CG} is then due to this entropic term. S_{CG} is plotted for a range of temperatures in Fig. 6.6.

6.4 Kinetic Contribution to Partition Function

We have so far considered only the contribution to the partition function due to the potential energy. We must also consider the kinetic energy if we want to know the full CG Hamiltonian. The atomistic Hamiltonian of the two dimers may be written as

$$H = \frac{1}{2m} \left[(p_L^-)^2 + (p_L^+)^2 + (p_R^-)^2 + (p_R^+)^2 \right] + V(x_L^-, x_L^+, x_R^-, x_R^+), \tag{6.62}$$

where each atom has mass m and the p_i are the atomic momenta. Hamilton's equations imply that

$$\dot{x_i} = \frac{\partial H}{\partial p_i} = \frac{p_i}{m}, \tag{6.63}$$

or that $p_i = m\dot{x_i}$. Therefore, the Lagrangian becomes

$$\begin{aligned} L &= \sum_i p_i \dot{x_i} - H \\ &= \frac{m}{2}\left[(\dot{x_L^-})^2 + (\dot{x_L^+})^2 + (\dot{x_R^-})^2 + (\dot{x_R^+})^2\right] - V(x_L^-, x_L^+, x_R^-, x_R^+). \end{aligned} \tag{6.64}$$

We can define four new variables for the system,

$$\begin{aligned} x_{\text{com}} &= \frac{1}{4}(x_L^- + x_L^+ + x_R^- + x_R^+), \\ d &= \frac{1}{2}(x_R^+ + x_R^-) - \frac{1}{2}(x_L^+ + x_L^-), \\ a_L &= x_L^+ - x_L^-, \\ a_R &= x_R^+ - x_R^-, \end{aligned} \tag{6.65}$$

which represented the centre of mass, dimer separation and left and right dimer lengths respectively. Let us write L in terms of these new variables. We have

$$\begin{aligned} x_R^- + x_R^+ &= 2x_{\text{com}} + d, \\ x_L^- + x_L^+ &= 2x_{\text{com}} - d \end{aligned} \tag{6.66}$$

which implies that

$$\begin{aligned} x_R^+ &= \frac{1}{2}(2x_{\text{com}} + d + a_R), \\ x_R^- &= \frac{1}{2}(2x_{\text{com}} + d - a_R), \\ x_L^+ &= \frac{1}{2}(2x_{\text{com}} - d + a_L), \\ x_L^- &= \frac{1}{2}(2x_{\text{com}} - d - a_L). \end{aligned} \tag{6.67}$$

Then we can write

$$\begin{aligned} &(\dot{x_L^-})^2 + (\dot{x_L^+})^2 + (\dot{x_R^-})^2 + (\dot{x_R^+})^2 \\ &= \frac{1}{2}\left[(\dot{x_L^-} + \dot{x_L^+})^2 + (\dot{x_L^-} - \dot{x_L^+})^2\right] + \frac{1}{2}\left[(\dot{x_R^-} + \dot{x_R^+})^2 + (\dot{x_R^-} - \dot{x_R^+})^2\right] \\ &= \frac{1}{2}\left[(2\dot{x}_{\text{com}} - \dot{d})^2 + \dot{a}_L^2\right] + \frac{1}{2}\left[(2\dot{x}_{\text{com}} + \dot{d})^2 + \dot{a}_R^2\right] \\ &= \frac{1}{2}\left[4\dot{x}_{\text{com}}^2 + 2\dot{d}^2 + \dot{a}_L^2 + \dot{a}_R^2\right]. \end{aligned} \tag{6.68}$$

The Lagrangian in terms of the new variables is then

$$L = \frac{m}{4}\left[4\dot{x}_{\rm com}^2 + 2\dot{d}^2 + \dot{a}_L^2 + \dot{a}_R^2\right] - V(x_{\rm com}, d, a_L, a_R). \tag{6.69}$$

The corresponding momenta $\Pi_{\rm com}, \Pi_d, \Pi_L, \Pi_R$ are then given by

$$\Pi_i = \frac{\partial L}{\partial \dot{q}_I}, \tag{6.70}$$

so that

$$\begin{aligned}
\Pi_{\rm com} &= 2m\dot{x}_{\rm com}, \\
\Pi_d &= m\dot{d}, \\
\Pi_L &= \frac{m}{2}\dot{a}_L, \\
\Pi_R &= \frac{m}{2}\dot{a}_R.
\end{aligned} \tag{6.71}$$

We can write the Lagrangian in terms of these conjugate momenta as

$$L = \frac{1}{m}\left(\frac{\Pi_{\rm com}^2}{4} + \frac{\Pi_d^2}{2} + \Pi_L^2 + \Pi_R^2\right) - V(x_{\rm com}, d, a_L, a_R). \tag{6.72}$$

The Hamiltonian in terms of these conjugate momenta is then

$$H = \frac{1}{m}\left(\frac{\Pi_{\rm com}^2}{4} + \frac{\Pi_d^2}{2} + \Pi_L^2 + \Pi_R^2\right) + V(x_{\rm com}, d, a_L, a_R). \tag{6.73}$$

The kinetic contribution to the partition function is then given by

$$Z_K = \int \exp\left[-\frac{\beta}{m}\left(\frac{\Pi_{\rm com}^2}{4} + \frac{\Pi_d^2}{2} + \Pi_L^2 + \Pi_R^2\right)\right] d\Pi_{\rm com} d\Pi_d d\Pi_L d\Pi_R. \tag{6.74}$$

We are now free to decide which degrees of freedom we would like to remove from the system. Let us remove Π_L and Π_R, to end up with a partition function that is a function of $\Pi_{\rm com}$ and Π_d only,

$$\begin{aligned}
Z_K(\Pi_{\rm com}, \Pi_d) &= \exp\left[-\frac{\beta}{m}\left(\frac{\Pi_{\rm com}^2}{4} + \frac{\Pi_d^2}{2}\right)\right] \int_{-\infty}^{\infty} \exp\left[-\frac{\beta}{m}\left(\Pi_L^2 + \Pi_R^2\right)\right] d\Pi_L d\Pi_R \\
&= \frac{m\pi}{\beta} \exp\left[-\frac{\beta}{m}\left(\frac{\Pi_{\rm com}^2}{4} + \frac{\Pi_d^2}{2}\right)\right].
\end{aligned} \tag{6.75}$$

We find that the kinetic contribution to the internal energy is given by

$$\begin{aligned}
U_K(\Pi_{\rm com}, \Pi_d) &= -\frac{\partial \ln Z_k}{\partial \beta} \\
&= \frac{1}{\beta} + \frac{\Pi_{\rm com}^2}{4m} + \frac{\Pi_d^2}{2m}.
\end{aligned} \tag{6.76}$$

This again includes a contribution of $k_{\mathrm{B}}T/2$ for each degree of freedom which has been integrated out. The kinetic contribution to the free energy is given by

$$\begin{aligned} F_K(\Pi_{\mathrm{com}}, \Pi_d) &= -\frac{1}{\beta} \ln Z_K \\ &= -\frac{1}{\beta} \ln \left(\frac{m\pi}{\beta} \right) + \frac{\Pi_{\mathrm{com}}^2}{4m} + \frac{\Pi_d^2}{2m}. \end{aligned} \tag{6.77}$$

Chapter 7

Atomistic Simulation Results

In order to judge the success or failure of a particular CG model of a system, it is necessary to have some form of benchmark to compare the model against. In our case, this benchmark will be a fully atomistic MD simulation of a system of C_{60} molecules. In this chapter, we discuss these simulations and the various properties we calculate to compare with the CG simulations.

We chose the molecule buckminsterfullerene, or C_{60}, for further study, since its spherical symmetry makes the first question trivial. The correct choice for the spatial unit is clearly the centre of mass (see Fig. 7.1). C_{60} has the added bonus of being relatively small, containing only 60 atoms. This allows us to perform fully atomistic simulations in a relatively small amount of time, providing us with ample simulation data to benchmark our CG potentials against. Let us now discuss some particulars of C_{60}.

7.1 Buckyballs

The buckminsterfullerene molecule, more commonly known as a buckyball, was first discovered by Harry Kroto et al. in 1985 [60]. They were awarded the Nobel Prize in Chemistry in 1996 for this discovery. Buckyballs consist of 60 carbon atoms arranged at the vertices of a truncated icosahedron, and have a diameter of approximately 0.7 nm. The molecule consists of 32 faces, 20 hexagons and

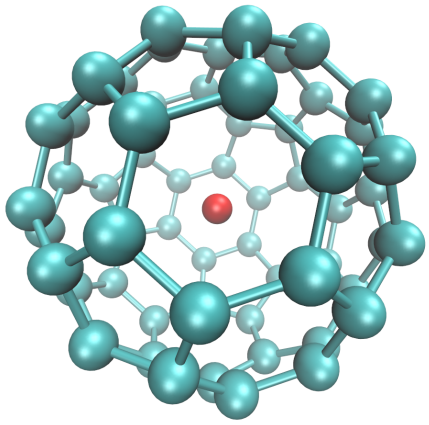


Figure 7.1: A single C_{60} molecule containing 60 carbon atoms with CG representation shown. The red dot at the centre of mass is the position of the CG bead. It has no physical extent but is drawn as a sphere for clarity.

12 pentagons. The average bond length is 0.14 nm. The molecule forms a face–centred cubic crystal structure with a lattice constant of 1.415 nm, and sublimes to the gas phase at approximately 900K [61].

We will take buckyballs as our prototypical test case for coarse-graining, for the reasons mentioned earlier. Our aim is to start with the simplest CG model one could think of, and successively build up layers of complexity by incorporating more physics into the model. At each stage, we hope to quantify what effect these changes have on quantities measured from the CG simulation. We will use reference fully atomistic MD simulations to benchmark these quantities against.

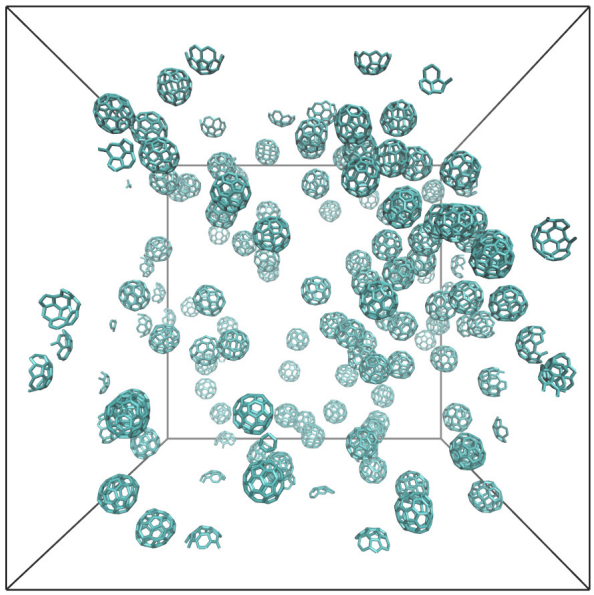


Figure 7.2: Snapshot of an MD simulation of 125 buckyballs, at a temperature of 1800K and density of 100 amu/nm^3.

7.2 Atomistic Simulation Results

All simulations were carried out using the GROMACS 4.6.2 software package [62–66] unless otherwise stated.

Our first benchmark simulation is of 1000 buckyballs at a temperature of 2500K and density of 100 amu/nm^3 for 100 ps (see Fig. 7.2). This is hot enough to see diffusive motion, and dense enough that plenty of collisions between buckyballs occur. The density was controlled by using periodic boundary conditions with a cubic simulation cell.

When choosing the time step to use in an MD simulation, there are two important measurements which must be made when testing conservation of total energy: the magnitude of the energy fluctuations and the rate of energy drift over time. Both of these errors are caused to an extent by truncation of the

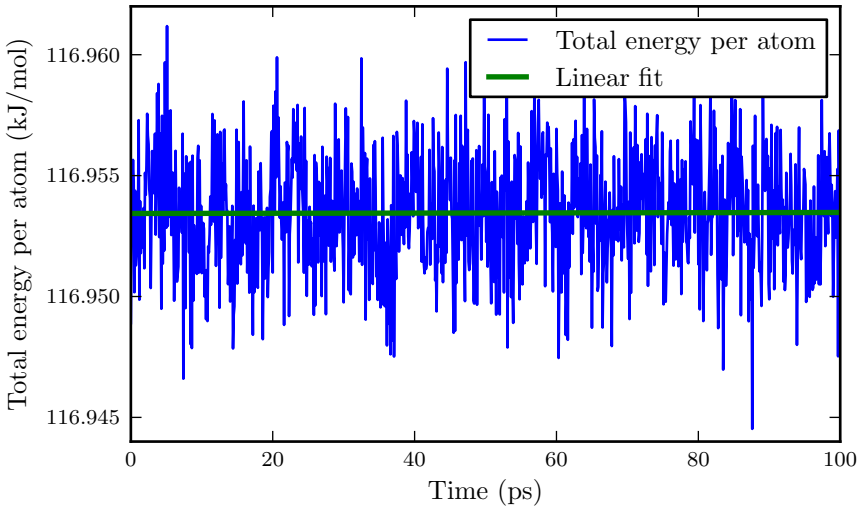


Figure 7.3: Total energy per atom as a function of time during a 100 ps simulation. A linear fit to this data is also shown. Instantaneous energy fluctuations about the mean are approximately 7.6×10^{-5} kJ/mol. Total energy drift per atom is 4.0×10^{-7} kJ/mol per ps, calculated from the gradient of the linear fit.

Taylor series expansion used in the integration algorithm, and can be controlled by reducing the size of the time step. It is desirable to keep both errors as small as possible, whilst still using a large enough time step to be able to perform a useful simulation with available computational resources. A more detailed discussion of energy conservation in MD simulations can be found in [67]. We chose a time step of 1 fs in our simulations. This results in a standard error in the total energy per atom of 7.6×10^{-5} kJ/mol, and an energy drift per atom of 4.0×10^{-7} kJ/mol per ps (see Fig. 7.3). This energy drift per atom is equivalent to a temperature gain per atom of 5.3×10^{-10} K per ps, a negligible amount considering the system temperature is 2500K.

A cutoff radius of 1.1 nm was used for the nonbonded interactions in our force field, with a neighbour list cutoff length of 1.2 nm. A Langevin thermostat with a time constant of 2 ps was used during an initial 100 ps NVT simulation to bring the system to the target temperature of 2500K. It was then switched off, and all statistics were gathered from an NVE simulation with no thermostat

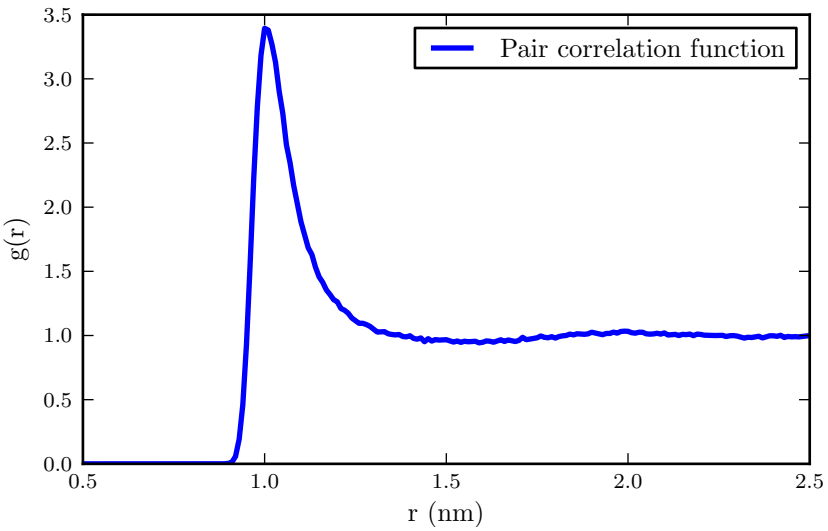


Figure 7.4: Pair correlation function from system of 1000 buckyballs.

applied.

The pair correlation function for the buckyball centre of masses is shown in Fig. 7.4. The position of the first peak here is 1.0 nm, corresponding to the minimum energy separation of two buckyballs. The first minimum and second peak are visible at approximately 1.5 nm and 2.0 nm respectively. This shape is characteristic of a diffusive gas. The mean squared displacement of the buckyball centre of masses is shown in Fig. 7.5. The diffusion constant for a C_{60} molecule can be estimated from the gradient of this plot using the Einstein relation, and is $(193.5 \pm 3.3) \times 10^{-5}$ cm^2/s.

Also of interest is how the pressure behaves for our system. In Fig. 7.6, we see that pressure fluctuates wildly from timestep to timestep in an MD simulation. However, there is still a well–defined mean pressure of 44.9 ± 1.7 bar.

The normal mode frequencies for a single minimised C_{60} molecule are shown in Fig. 7.7. There are several degenerate modes, including the three rotational and three translational modes which all have eigenvalue zero.

For the purposes of coarse-graining, it would be useful to be able to rank the importance of different normal modes in a molecule. Modes with a large amplitude, or modes with large fluctuations in amplitude, could be considered good candidates for being kept in a CG model. Modes whose amplitudes re-

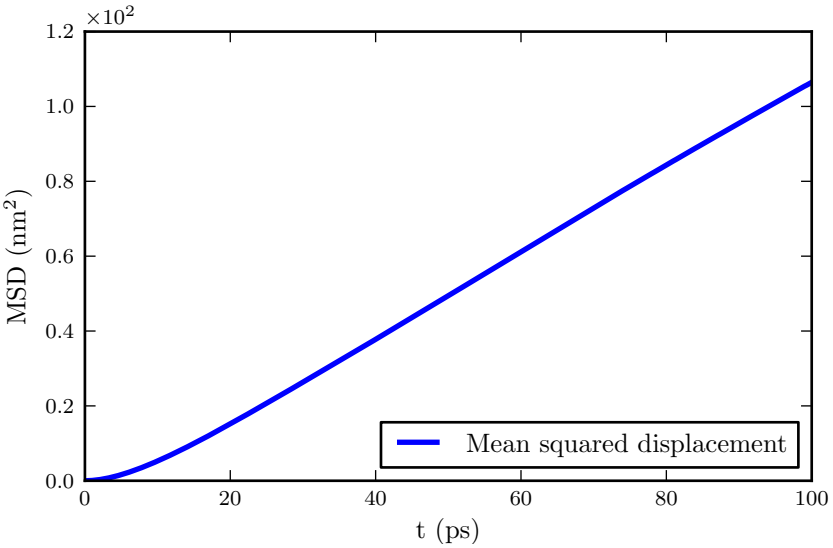


Figure 7.5: Mean squared displacement of buckyball centres of mass.

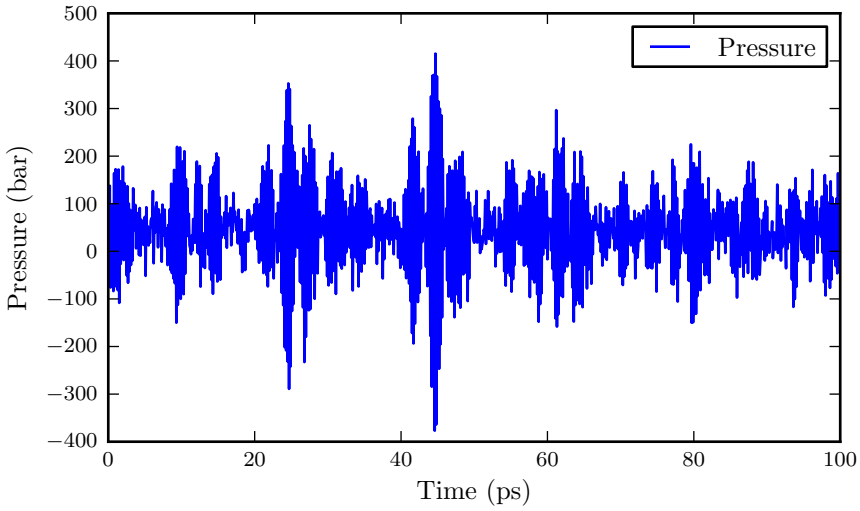


Figure 7.6: Instantaneous pressure as a function of time from system of 1000 buckyballs at 2500K.

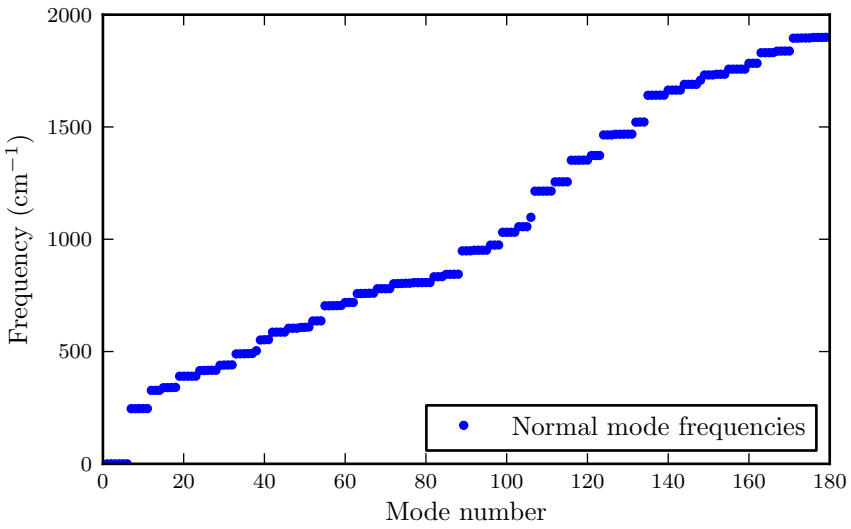


Figure 7.7: The 180 normal mode frequencies of a single C_{60} molecule. Points are joined as a guide to the eye.

mained fairly constant could be ignored and left out entirely as explicit degrees of freedom.

We took the test case of two buckyballs colliding in a head-on collision (see Fig. 7.8), and devised a procedure for analysing the excitations of the normal modes at each stage of the collision, hoping that this would provide some insight into which modes are essential for a CG simulation of buckyballs. The procedure works as follows.

- Analyse the positions and velocities of every atom in one of the buckyballs at each time step in the simulation, before, during and after a collision occurs.
- Place a reference structure, the energy-minimised static buckyball, at the origin. The centre of mass of the simulated buckyball is shifted back to the origin at every time step, so the atomic displacements can be compared with the reference structure.
- Take account of the possibility that the buckyball has rotated, by finding the best fit rotation which brings the simulated buckyball back to the same

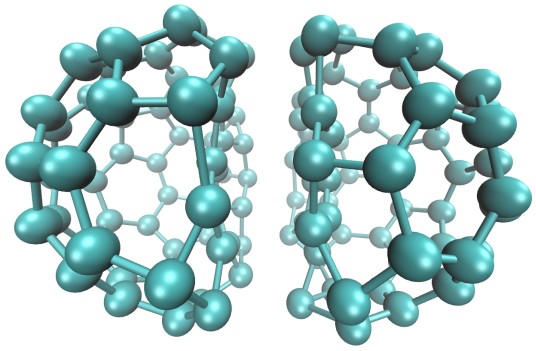


Figure 7.8: Two buckyballs at the beginning of a 100 Å /ps head-on collision along the x axis. This is an unrealistically energetic collision, corresponding to a temperature of approximately 3 million K, intended to exaggerate the deformation of the buckyballs.

orientation as the reference buckyball. The algorithm outlined in [68] was used to find this optimal rotation. It uses the quaternion representation for rotations, and a Newton-Raphson minimisation scheme to find the rotation which minimises the root-mean-square distance (RMSD) between the reference and simulated buckyball.

- Once the simulated buckyball has been translated and rotated back as close to the reference buckyball as possible, compute the displacement of each atom from its reference counterpart.

To compute the normal mode amplitudes c_α at each time step, recall Eqn. (5.89) and take its first derivative:

$$\begin{aligned}\mathbf{Y}(t) &= \sum_{\alpha=1}^{3N} c_\alpha \mathbf{e}_\alpha \cos(\omega_\alpha t + \phi_\alpha), \\ \dot{\mathbf{Y}}(t) &= -\sum_{\alpha=1}^{3N} c_\alpha \mathbf{e}_\alpha \omega_\alpha \sin(\omega_\alpha t + \phi_\alpha).\end{aligned} \tag{7.1}$$

Since the $\mathbf{e}_\alpha$ are orthonormal,

$$\begin{aligned}\mathbf{Y}(t)\cdot\mathbf{e}_\alpha &= c_\alpha\cos(\omega_\alpha t+\phi_\alpha),\\ \dot{\mathbf{Y}}(t)\cdot\mathbf{e}_\alpha &= -c_\alpha\omega_\alpha\sin(\omega_\alpha t+\phi_\alpha).\end{aligned} \tag{7.2}$$

Square both sides and add to arrive at

$$c_\alpha^2 = (\mathbf{Y}\cdot\mathbf{e}_\alpha)^2 + \frac{(\dot{\mathbf{Y}}\cdot\mathbf{e}_\alpha)^2}{\omega_\alpha^2}. \tag{7.3}$$

This formula is not valid for the 6 normal modes with eigenvalues of zero. For these 6 modes, we arbitrarily set c_α to zero. The other 174 values of c_α can be found at each time step of the simulation using Eqn. (7.3). Making a movie of these amplitudes, it is possible to watch the evolution of the normal mode amplitudes as a function of time during the simulation.

Fig. 7.9 shows a comparison between the normal mode amplitudes before and after a 100 Å /ps collision. It is clear that the five-fold degenerate normal modes with a frequency of approximately 46 THz are the most highly excited by the collision. These modes correspond to "squashing" of the buckyball, and we would expect them to become excited due to the large amount of axial deformation visible in Fig. 7.8. This suggests that the "squashing" modes would be a good candidate for inclusion in a CG model of buckyballs. Many of the higher frequency modes are not excited very much during collision, and they may be candidates for exclusion.

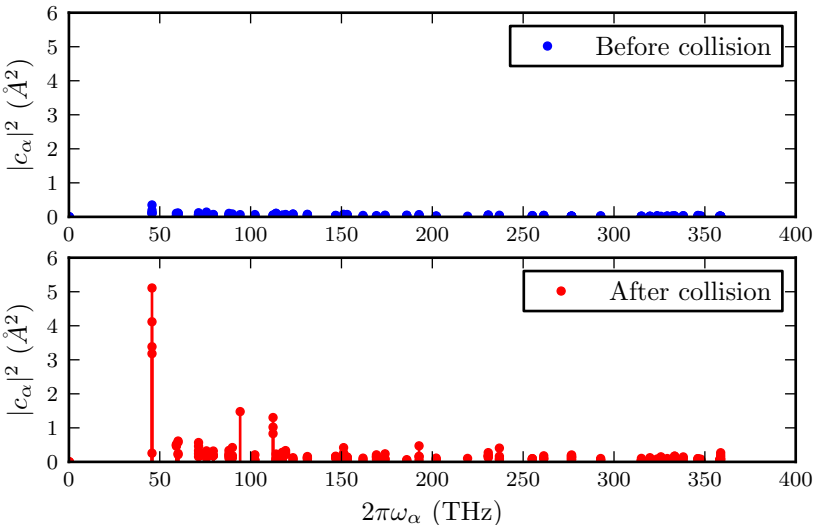


Figure 7.9: Comparison of normal mode amplitudes at two different time steps in a simulation of buckyball collision. The blue curve shows the amplitudes before collision has taken place, where the centre of mass separation of the two buckyballs is 15 Å . The red curve shows the amplitudes 150 fs later, just after collision has occurred.

Chapter 8

Generating Coarse–Grained Potentials

In Chapter 6, we derived CG potentials for a simple system of two dimers. Of course, for any CG method to be useful, we must be able to apply it to a more realistic system. We will now study how CG potentials can be generated for our test system of buckyballs, using the analytical results for two dimers as a reference point.

8.1 The Girifalco Potential

There have been attempts to generate CG potentials for buckyballs in the past. Girifalco derives an approximate analytic CG interaction potential between C_{60} molecules [69]. He first assumes that the carbon atoms interact via a Lennard-Jones potential,

$$V(x) = -\frac{A}{x^6} + \frac{B}{x^{12}}, \tag{8.1}$$

where x is the separation of the atoms, $A = 0.0019266$ kJ/mol $(\text{nm})^6$ and $B = 3.357787 \times 10^{-6}$ kJ/mol $(\text{nm})^{12}$ are constants obtained by matching to experiment (note that A and B are defined the opposite way round to normal). He then makes the approximation that each C_{60} molecule can be represented by a sphere, with the carbon atoms uniformly "smeared out" across its surface. He computes the potential energy of interaction between the two spheres using

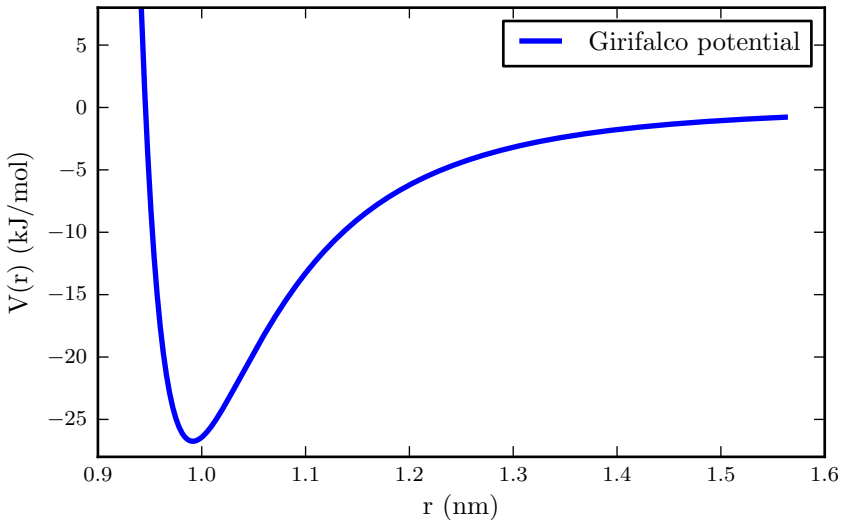


Figure 8.1: Plot of the Girifalco potential Eqn. (8.2), using parameters from [69].

surface integration (see Appendix A for a detailed derivation). The result is the potential

$$V(r) = -\alpha \left[\frac{1}{s(s-1)^3} + \frac{1}{s(s+1)^3} - \frac{2}{s^4} \right] \tag{8.2}$$

$$+\beta \left[\frac{1}{s(s-1)^9} + \frac{1}{s(s+1)^9} - \frac{2}{s^{10}} \right], \tag{8.3}$$

where $s = r/2a$, $2a = 0.71$ nm is the C_{60} diameter, $\alpha = N^2 A/12(2a)^6$ where $N = 60$, and $\beta = N^2 B/90(2a)^{12}$. This potential is plotted in Fig. 8.1.

We can recalculate the Girifalco potential using parameters from our force field. To do this, we perform a conjugate gradients energy minimisation of a single buckyball with our parameters from the OPLS–AA force field, to find its equilibrium structure. By calculating the mean distance of each atom from the buckyball's centre of mass, we obtain the mean radius of the buckyball. For our force field, this corresponds to a buckyball diameter of 0.70 nm, which is 0.01 nm smaller than the value used in Girifalco's paper. The values of A and B we used for the Lennard-Jones force between carbon atoms are $A = 0.0023449$ kJ/mol $(\text{nm})^6$ and $B = 4.693426 \times 10^{-6}$ kJ/mol $(\text{nm})^{12}$ respectively. The potential is plotted in Fig. 8.2, using both Girifalco's parameters and our

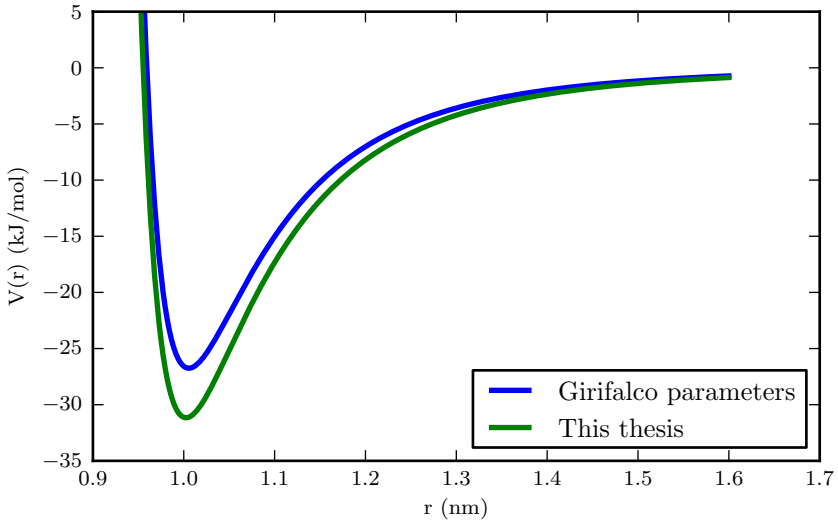


Figure 8.2: Plot of the Girifalco potential using parameters from the paper, and using parameters from our force field.

own. Our parameters give a slightly deeper potential well than Girifalco's, because the Lennard–Jones well depth ϵ used in our force field is slightly larger, but otherwise the two curves are qualitatively similar.

8.2 The Rigid Approximation

The derivation of the Girifalco formula assumes the buckyballs are perfectly spherical. We can test this assumption by treating the buckyballs as rigid bodies with the correct geometry, and evaluating the potential numerically. Two buckyballs are placed along the x–axis with their centre of masses 0.9 nm apart. Each buckyball is rotated independently by a random angle, and the Lennard-Jones interactions between the two molecules are calculated using the values of A and B from OPLS–AA. This is repeated for 100 random rotations, and the mean potential energy for this separation is computed. This is then repeated for increasing centre of mass separations up to 1.6 nm, to build up a pair potential (see Fig. 8.3). We see that the rigid approximation is in agreement to within error with the analytic formula. This is to be expected, since the points of a truncated icosahedron all lie on the surface of a sphere, and we are simply

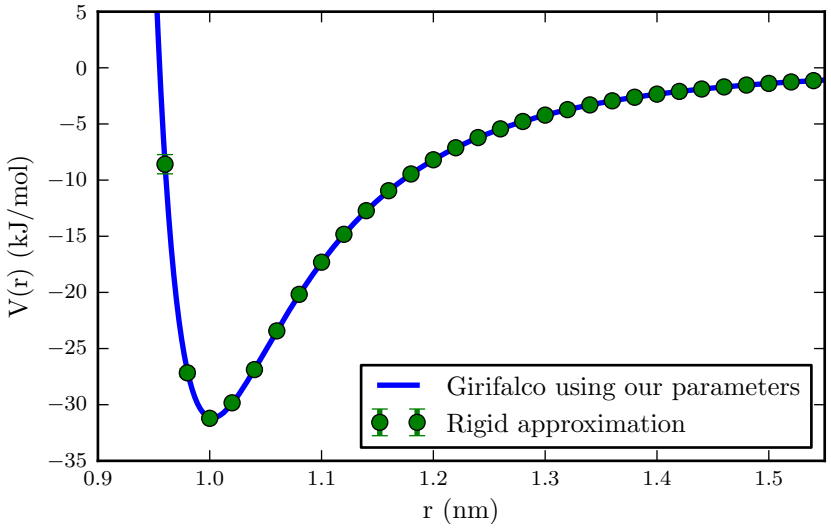


Figure 8.3: Rigid approximation to the Girifalco potential using our parameters, indicated by circles. Corresponding analytic curve plotted with solid lines for comparison.

calculating this spherical average numerically.

This rigid approximation used by Girifalco corresponds to keeping the bond length of each dimer rigidly fixed in the example of two dimers. As one would expect, we will see that this approximation is valid in simulations performed at low temperatures, but becomes increasingly bad as the temperature is increased.

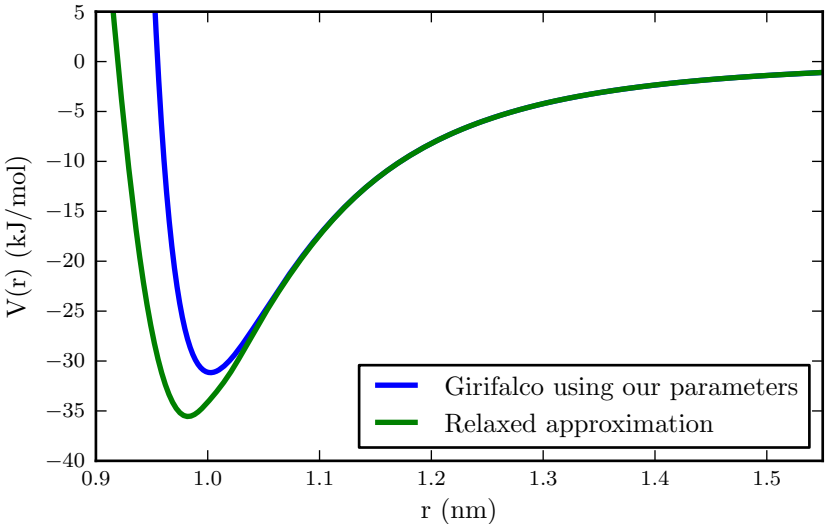


Figure 8.4: Relaxed approximation to the Girifalco potential. A cubic spline has been fitted to the relaxed data for clarity. The relaxed potential here was generated without the use of cut–offs in the Lennard–Jones potential.

8.3 The Relaxed Approximation

We can improve on our assumption that the buckyballs are rigid, inflexible objects by allowing them to relax according to the properties defined in our force field. For each fixed buckyball separation, we run 100,000 steps of conjugate gradient energy minimisation to find the lowest energy configuration for that separation. We repeat this 100 times with a random initial buckyball orientation for each separation. We then calculate the mean force between buckyball centre of masses across these 100 mimisations, fit a cubic spline to the force and integrate it to obtain a potential. The resulting potential curve is shown in Fig. 8.4. This plot was generated without using cutoffs in the Lennard–Jones interaction between carbon atoms, and we see that the relaxed potential minimum is lower than the Girifalco potential. In Fig. 8.5, the relaxed potential was regenerated with the use of cutoffs in the potential. This results in an upwards shift of the relaxed potential. This second version generated with cutoffs is used from now on for consistency with our atomistic force field.

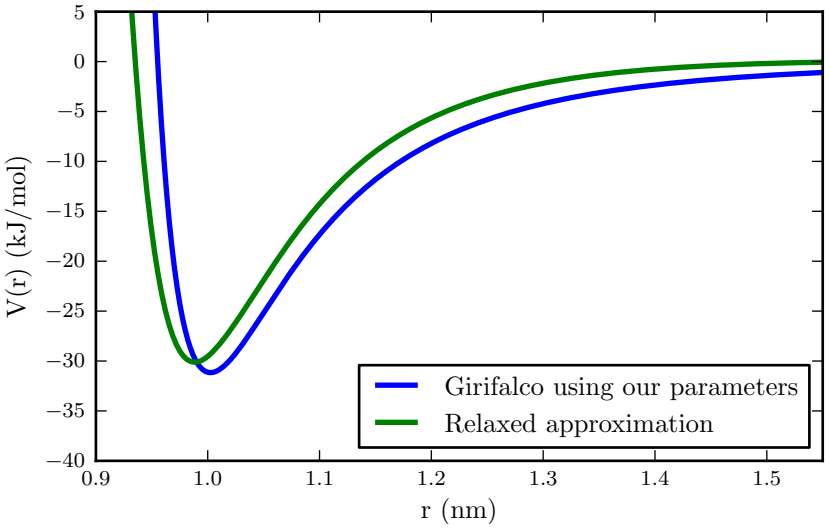


Figure 8.5: Relaxed approximation to the Girifalco potential. A cubic spline has been fitted to the relaxed data for clarity.

The position of the minimum in the relaxed curve has moved to the left by around 0.03 nm. This is because the buckyballs are now free to "squash up" closer to each other, shifting the equilibrium separation of their centres of mass closer.

This approach corresponds to allowing the two dimers to relax to their minimum energy configuration when the separation between their centres of mass is held fixed.

8.4 Constrained Pair Dynamical Simulations

The rigid and relaxed methods described so far essentially amount to zero temperature methods for generating the CG potential. These potentials may not be suitable for use in simulations at higher temperatures. We therefore need to find a numerical approach to generating a CG potential which is analogous to the finite temperature case for two dimers, described in 6.3. In the next sections, we will describe methods for doing this, as well as the technical challenges which must be overcome.

8.4.1 The Brute Force Approach

In this approach, two buckyballs are kept at a fixed separation using the SHAKE constraint algorithm, and thermalised using the Langevin thermostat for 100 ps. The atomic positions and velocities from the last timestep in the thermalised run are used as input for a longer NVE run, during which the average internal energy is calculated. This procedure is repeated at increasing buckyball separations, to build up a curve of internal energy as a function of separation. We will refer to potentials generated via direct measurement of the internal energy as *internal energy potentials* in what follows.

There are two main problems with this method. The first is that the standard deviation on the temperature in an NVT simulation is given by

$$\sigma = T\sqrt{\frac{2}{n_{\mathrm{deg}}}}, \tag{8.4}$$

where n_{deg} is the number of degrees of freedom in the system. In our case of two constrained buckyballs,

$$n_{\mathrm{deg}} = (2 \times 60 \times 3) - 3 - 1 = 356, \tag{8.5}$$

because there are 2 buckyballs, 60 atoms per buckyball, 3 spatial directions, we must subtract 3 because of the initial removal of linear momentum, and we must subtract 1 because of the single constraint between buckyballs. For even a very low temperature of 100K, this results in a standard deviation in the temperature of 7.50K. What this means is that, when we end our NVT run the final configuration of velocities will give rise to a temperature that may not be very close to the target temperature of 100K. When we initialise an NVE run from this configuration, the total energy will therefore be either increased or decreased by a certain amount. If we now repeat the procedure for a buckyball pair with a slightly larger separation, again the temperature will not be exactly 100K, and this will result in a discrepancy between the total energies of the two separations (see Fig. 8.6).

The second problem with this method is that, particularly at close separations, it is possible that the buckyballs could become locked in an energetically favourable configuration. If this happens, we will not be correctly sampling all possible orientations. We can solve both these problems by performing many

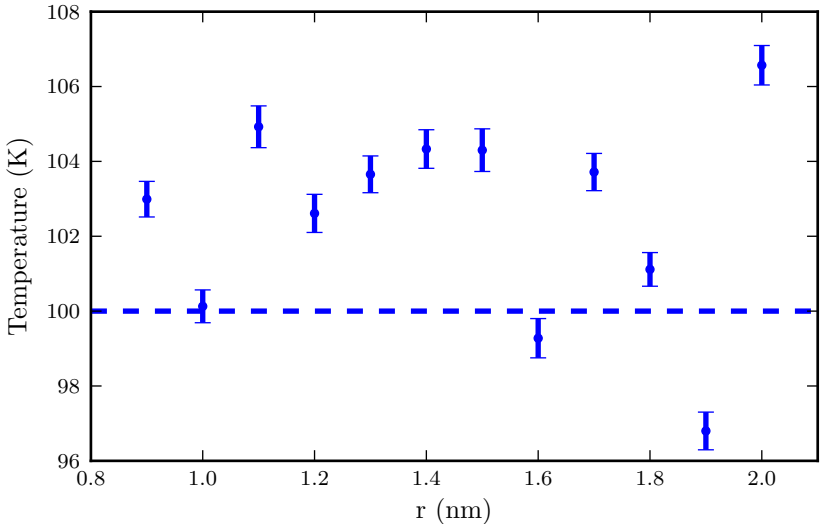


Figure 8.6: Mean temperatures from 12 independent MD simulations of constrained buckyball pairs at different separations, nominally at 100K. Error bars are plus/minus 1 standard error.

simulations for each separation, each starting with a different random orientation of the buckyballs in the pair, and a different total energy sampled from the Boltzmann distribution at temperature T. For 100 simulations, we obtain the potential in Fig. 8.7. This data is not smooth enough to make a good cubic spline fit to. It therefore cannot be used as a potential to perform CG simulations with, since any potential needs to be very smooth in order to avoid energy conservation artefacts. A possible solution to this problem is to perform many more simulations at each separation to reduce the size of the error bars. However, we found that an impractically large number of simulations was needed to achieve a smooth fit. The problem only gets worse if we try to generate the potential at higher temperatures. In the next section, we discuss some alternative solutions.

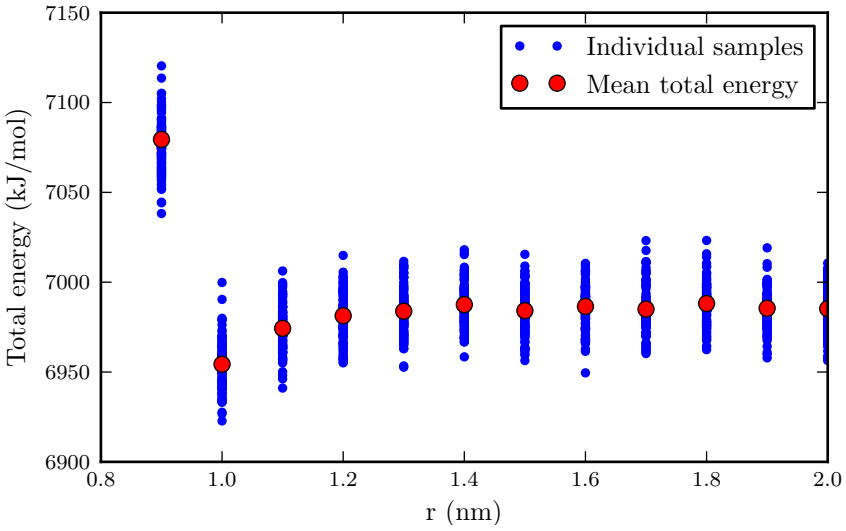


Figure 8.7: Mean total energy from constrained pair simulation. The individual results from the 100 simulations at each separation are plotted in blue for comparison.

8.4.2 Generating Free Energy Potentials from Forces

An alternative method to the brute force approach described above is to examine the mean inter–buckyball force during a simulation, and then integrate this force to obtain a potential. Since this integral yields a two–body approximation to the potential of mean force, it represents the average work required to bring two buckyballs from infinity to a separation of r. We therefore refer to CG potentials generated using this force integration method as *free energy potentials*.

Since the force between buckyballs is a derivative quantity, we find that its statistical error is much smaller than attempting a direct measurement of the internal energy by the brute force approach. This allows us to produce a smooth potential which is suitable for use in a CG simulation within a reasonable number of repeats.

We now discuss how to implement the force integration method in practice. Firstly, we need to calculate the mean inter–buckyball force at each separation. Let us label the pair of buckyballs as buckyball 1 and buckyball 2. We will consider the force on the centre of mass of buckyball 1 at a specific time step t in the simulation. This is given by the sum of the forces on each atom comprising the molecule,

$$\mathbf{f}_{\mathrm{com}}(t) = \sum_{i=1}^{60} \mathbf{f}_i(t). \tag{8.6}$$

Let the centre of mass positions of buckyball 1 and 2 be given by $\mathbf{r}_1$ and $\mathbf{r}_2$ respectively. Then the distance vector between centres of mass is given by $\mathbf{r}_{12} = \mathbf{r}_2 - \mathbf{r}_1$. The magnitude of the COM force along this distance vector is given by

$$f_{\mathrm{com}}(t) = \frac{\mathbf{f}_{\mathrm{com}} \cdot \mathbf{r}_{12}}{|\mathbf{r}_{12}|}. \tag{8.7}$$

It can be seen from this equation that if $\mathbf{f}_{\mathrm{com}} \cdot \mathbf{r}_{12}$ is positive (negative), the COM force between buckyballs is attractive (repulsive). The mean COM force over one simulation run j consisting of n_{steps} time steps is then

$$\langle f_{\mathrm{com}} \rangle_j = \frac{1}{n_{\mathrm{steps}}} \sum_{t=1}^{n_{\mathrm{steps}}} f_{\mathrm{com}}(t). \tag{8.8}$$

The mean COM force over all n_{runs} independent simulation runs is then

$$\langle f_{\text{com}} \rangle = \frac{1}{n_{\text{runs}}} \sum_{j=1}^{n_{\text{runs}}} \langle f_{\text{com}} \rangle_j \,. \tag{8.9}$$

We also calculate the standard error of this quantity. This process is repeated for each COM spacing we wish to evaluate the mean force at. We then perform a weighted univariate cubic spline fit to the data, using the standard errors as weights, with a standard routine from the scientific programming library Scipy [70] in the Python programming language. An example of the result of this procedure for 100 simulation runs at each separation is shown in Fig. 8.8).

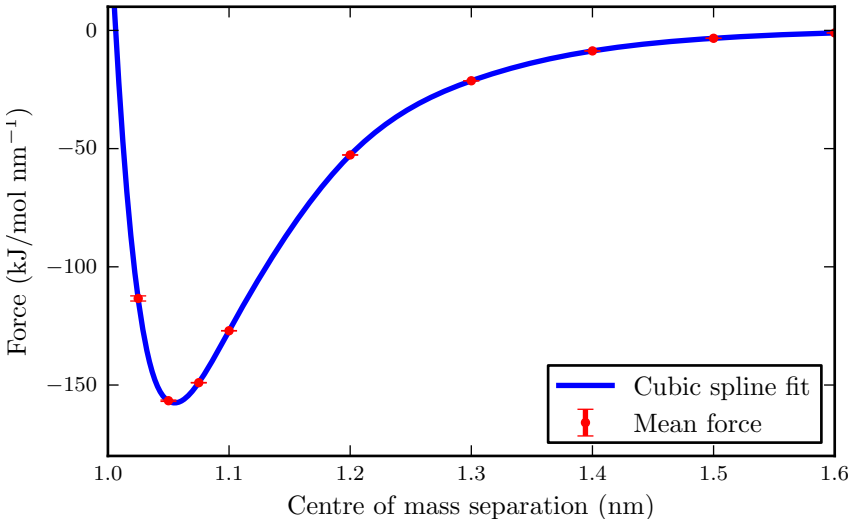


Figure 8.8: Mean force between buckyballs generated from 100 constrained pair simulations at each separation for buckyballs at 2500K. Weighted spline fit to data is shown as solid line.

We now seek a way to generate a CG potential from this force data, which we will refer to as $f(r)$. Let us assume it corresponds to some potential energy function $V(r)$ by the standard formula

$$f(r) = -\frac{dV}{dr}. \tag{8.10}$$

A naive application of the fundamental theorem of calculus tells us that

$$\int_0^r f(r')dr' = -\int_0^r \frac{dV}{dr'}dr' = V(0) - V(r). \tag{8.11}$$

However, since we cannot evaluate the force at $r = 0$ numerically (and it would be infinite), it is impossible to calculate the integral here in order to evaluate the potential at r. A better solution would be to change the limits of integration to integrate inwards from infinity, so that

$$-\int_{\infty}^{r} \frac{dV}{dr'} dr' = V(\infty) - V(r). \tag{8.12}$$

Assuming the potential at infinity is zero, we find that

$$V(r) = -\int_{\infty}^{r} f(r') dr'. \tag{8.13}$$

We may now evaluate the potential by integrating the area under the cubic spline fit, provided out data extends at least to the point where the force becomes zero.

We carried out this force integration procedure to generate free energy potentials at a range of different temperatures (see Fig. 8.9). We also plot the relaxed approximation potential for comparison.

It can be seen that the relaxed approximation appears to be the zero temperature limit of this method. This is to be expected, since running a constrained pair dynamical simulation at a temperature close to 0K will result in negligible thermal deformation of the buckyballs. The structure attained by buckyballs held at a fixed close separation will be the minimum energy configuration, provided sufficient time steps are used to allow relaxation to take place.

It can also be seen that the position of the minimum in the free energy potentials shifts to the right as the temperature is increased. This is because the effective mean radius of the buckyballs increases with temperature, resulting in an equilibrium COM separation with the COMs slightly further apart.

Another interesting feature of this plot is that the well depth of the potentials shifts upwards as the temperature is increased. A qualitative explanation for this is that the higher the temperature is, the easier buckyballs find it to escape becoming bound together. The only way this behaviour can be reflected in a pair potential generated at one specific temperature is for the well to become shallower as a function of increasing temperature.

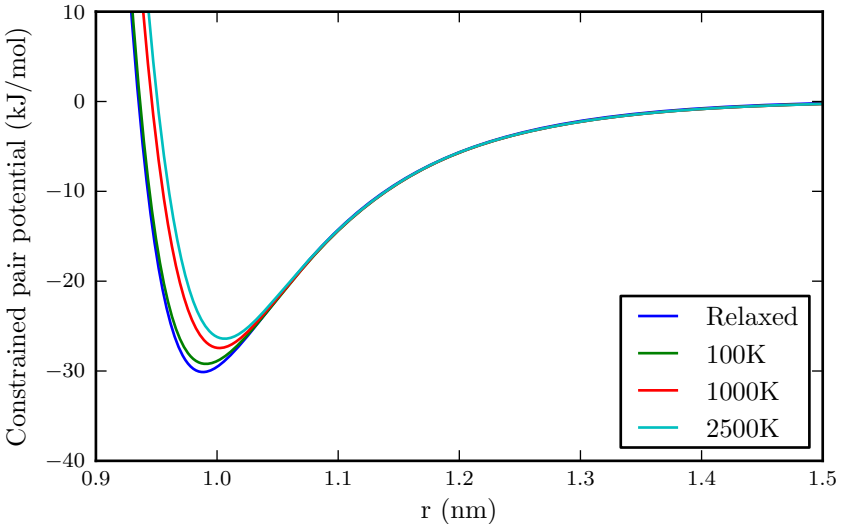


Figure 8.9: Constrained pair free energy potential generated at a range of temperatures. The relaxed approximation potential is plotted for comparison.

8.4.3 Generating Internal Energy Potentials from Free Energy Potentials

Previously, we discussed the difficulties associated with trying to evaluate the internal energy potential directly via the brute force approach. Fortunately, the free energy potentials calculated by integrating the force provide an indirect way to accomplish this. Recall from thermodynamics the formula

$$U(\beta) = F(\beta) + \beta \frac{\partial F}{\partial \beta}, \tag{8.14}$$

or, in terms of temperature,

$$U(T) = F(T) - T \frac{\partial F}{\partial T}. \tag{8.15}$$

This shows that if we know the free energy at a range of temperatures T, we can calculate the derivative $\partial F/\partial T$, and hence the internal energy potential $U(T)$. To calculate the derivative, it helps to examine the free energy as a function of temperature at a fixed value of buckyball separation. This can be found by

considering the free energies for one point on the x axis taken from Fig. 8.8. Some representative examples are plotted in Fig. 8.10.

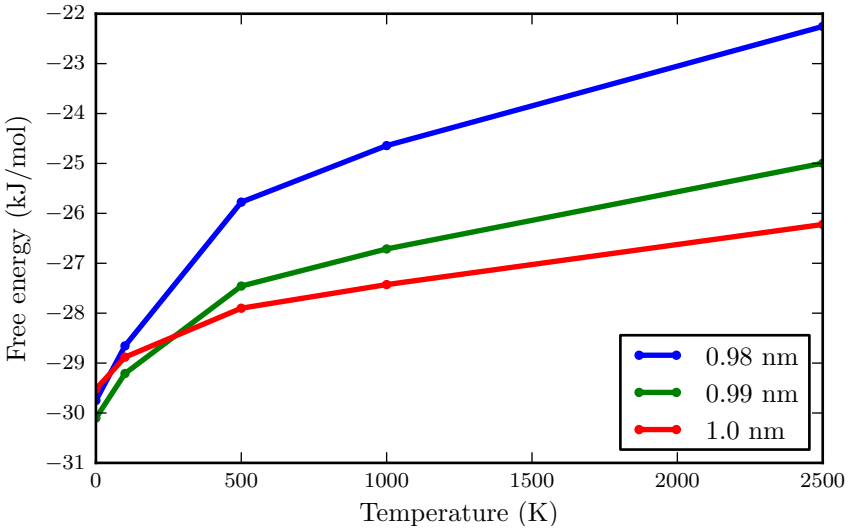


Figure 8.10: Free energy as a function of temperature for three buckyball centre of mass spacings chosen near the potential minimum.

It is clear that F is an approximately linear function of β for temperatures between 500K and 2500K. This allows us to calculate $U(T)$ for any temperature between 500K and 2500K, by applying Eqn. (8.15) individually for each buckyball separation. The resulting internal energy potentials are plotted in Fig. 8.11, and are virtually independent of temperature. This must be the case, since by differentiating Eqn. (8.15) with respect to T for a particular separation, we find

$$\frac{dU}{dT} = \frac{dF}{dT} - \frac{dF}{dT} - T\frac{d^2F}{dT^2} = -T\frac{d^2F}{dT^2}, \tag{8.16}$$

and in the region where F is linear with respect to T, $d^2F/dT^2 = 0$, and so

$$\frac{dU}{dT} = 0 \implies U(T) \text{ is constant.} \tag{8.17}$$

Here, it should be stressed that U is not strictly the thermodynamic internal energy of the system, but rather a function $U(r, T)$ of the separation of two buckyballs. Hence, the formula above is not implying that the specific heat capacity of the system is strictly zero.

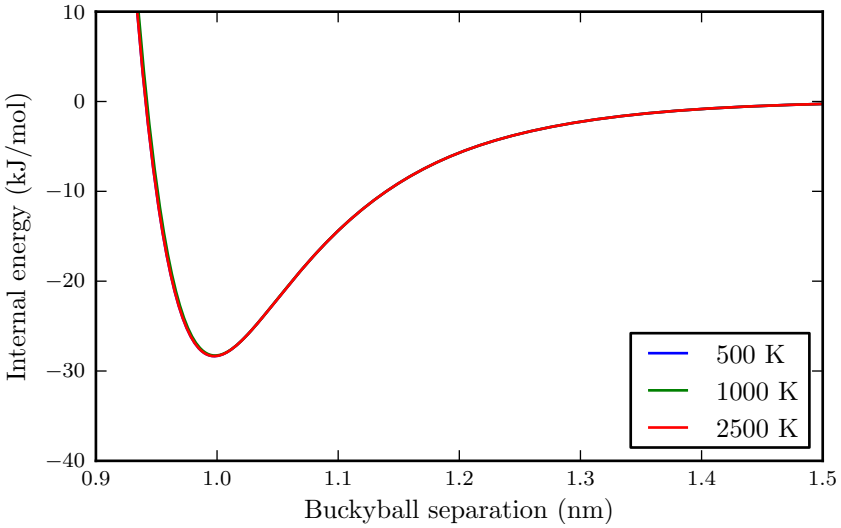


Figure 8.11: Internal energy potentials derived from free energy potentials at a range of temperatures. The three temperatures are virtually indistinguishable.

The free energy and internal energy potentials are plotted together in Fig. 8.12 for two representative temperatures, 500K and 2500K. It can be seen that the difference between the free and internal curves grows as the temperature is increased.

We now compare properties calculated in a fully atomistic MD simulation with CG simulations using the free and internal energy potentials generated at 2500K. The results are summarised in Table 8.1. The internal energy potential underestimates the true pressure by 4 bar. It also yields the largest diffusion constant. For comparison, the pair correlation functions are plotted together in Fig. 8.13. The internal energy potential yields a first maximum in the pair correlation function which is too high, a manifestation of its shallower potential well depth. This can also be seen from the overestimation of the coordination number when using the internal energy potential.

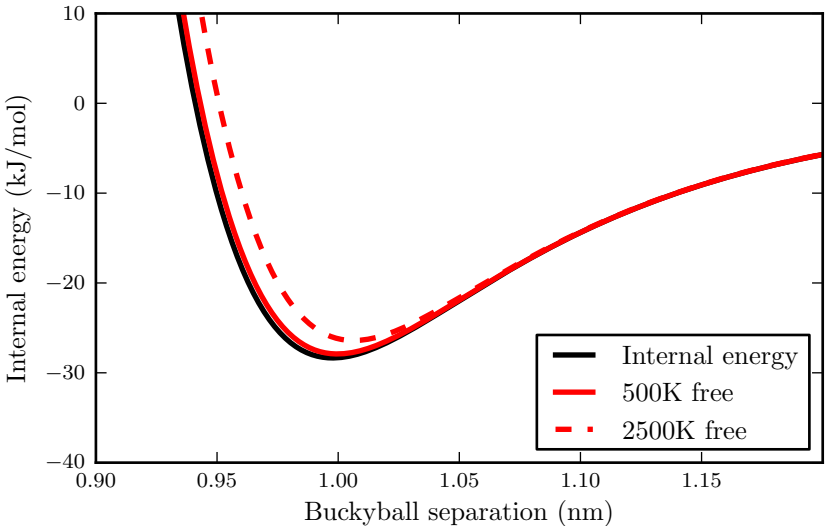


Figure 8.12: The internal energy potential plotted alongside free energy potentials at 500K and 2500K. The difference grows with increasing temperature.

	Pressure (bar)	Coordination number	D (10^{-5} cm^2/s)
Atomistic	44.9 ± 1.7	2.46	193.5
Free	43.0 ± 0.2	2.46	196.1
Internal	40.8 ± 0.2	2.53	202.9

Table 8.1: Comparison of properties calculated from 100 ps simulation of 1000 buckyballs at a temperature of 2500K. Coordination number obtained by integrating $4\pi r^2 \rho g(r)$ between 0.75 and 1.6 nm.

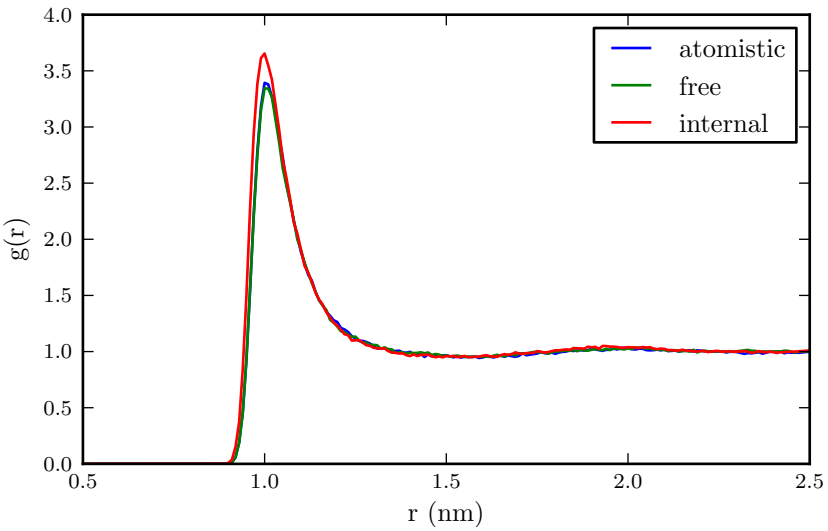


Figure 8.13: Comparison of pair correlation functions calculated from 100 ps simulation of 1000 buckyballs at a temperature of 2500K.

8.5 Unconstrained Dynamical Simulations

In the previous sections, we explored some of the problems with generating a CG potential directly from a pair of constrained buckyballs. An alternative approach to this problem is to generate a potential from a MD simulation of a periodic box full of many molecules, avoiding the use of contraints entirely. By doing so, we will be including the effect of many–body interactions in the pair potential. This approach is very similar to the force matching method applied by Voth et al. in [18].

We have written a program to extract CG pair potentials from a simulation of a box of buckyballs. As input, this program requires a trajectory file from an NVE simulation of the box. This trajectory file must contain the positions, velocties and forces of each atom at regular intervals. Our program also requires a list of the force field and simulation parameters used in the original simulation.

The program begins by extracting the trajectory of a particular buckyball pair from the original trajectory file. For each frame, we use GROMACS to compute the potential energy of the buckyball pair in isolation, as well as the force between buckyballs, resolved along the vector between their centres of mass. It should be stressed that no further MD simulation is performed on the isolated pair; instead, the atomic positions generated from the original ensemble are used.

This process is repeated for every distinct pair of buckyballs in the original simulation. For N buckyballs, and a trajectory file containing n_{frames} coordinate snapshots, the total number of independent force and potential energy measurements we have is then

$$n_{\text{measurements}} = \frac{1}{2}N(N-1)n_{\text{frames}} \tag{8.18}$$

The number of data points therefore has a fast growth rate of $O(N^2)$. It suffices to consider a relatively small number of trajectory frames for large N, provided the simulation snapshots are taken sufficiently far apart to be decorrelated in time.

Next, the program bins the pair separations into histogram bins. The corresponding forces are also binned, and the mean forces in each bin are used to generate a plot of force against separation (see Fig. 8.14). A cubic spline fit to this data is made, and the area under the spline curve is integrated analytically

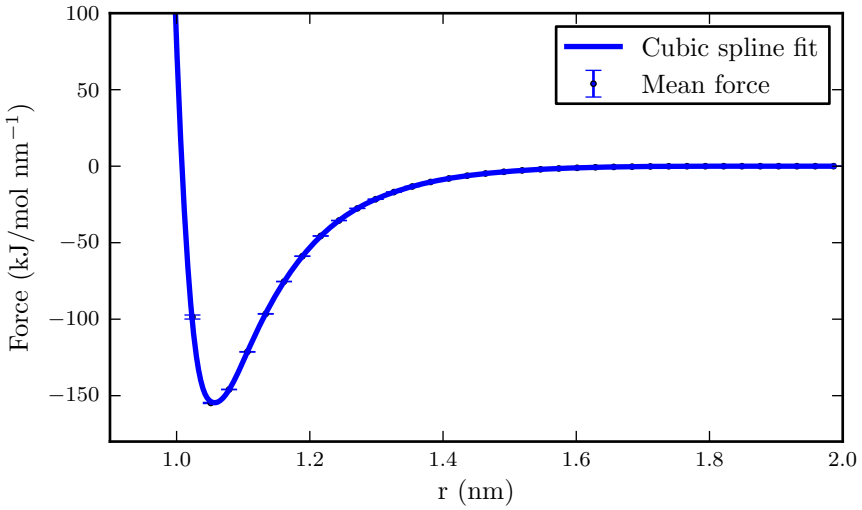


Figure 8.14: Mean force from histogram bins, generated from an MD simulation of a box of 1000 buckyballs at a density of 700 amu/nm^3 and temperature of 2500K. 10 trajectory frames were used, taken at 10 ps intervals. Cubic spline fit to data also shown.

from the end of the data range inwards, producing a free energy potential curve (see Fig. 8.15).

There are some technical challenges involved in this approach. Firstly, when binning the buckyball pair separations, there are naturally less pairs with very close separations. This is because the buckyballs repel each other at short distances, making it less likely to find trajectory snapshots of two buckyballs close together. This makes the statistical error on the leftmost data points in Fig. 8.14 larger than elsewhere. We can overcome this by simply sampling more trajectory snapshots, or considering simulations of more buckyballs. Also, this problem is reduced by increasing the density that the original MD simulation is performed at, which increases the probability of finding two buckyballs close together. Equivalently, running the original MD simulation at a lower temperature will improve the statistics since buckyballs will spend more time on average trapped inside the potential well. The constrained pair approach described in the previous sections is at an advantage here, since we can explicitly constrain buckyballs very close together.

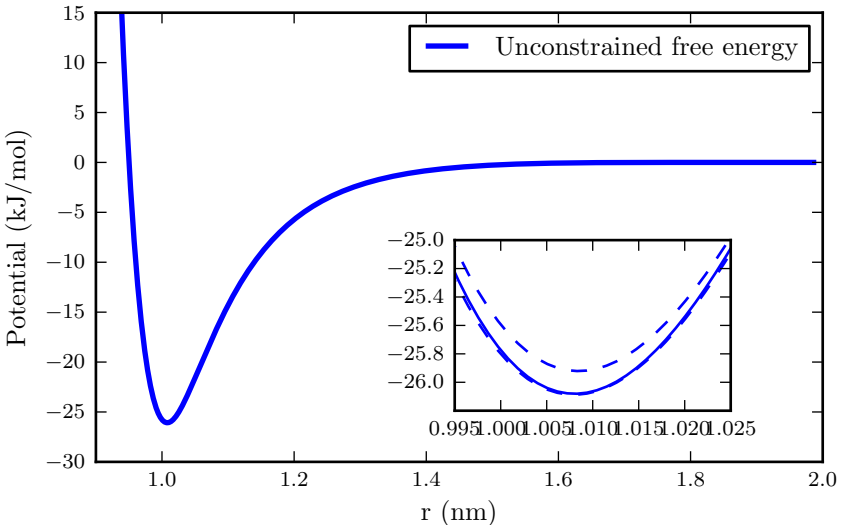


Figure 8.15: Free energy curve obtained from integrating spline fit in Fig. 8.14 from end of data range inwards. Inset is a magnification of the potential minimum. Dotted lines indicate the integral of a spline fitted to the minimum and maximum error bars in Fig. 8.14.

Secondly, the extraction of individual pair trajectories from the original trajectory file is very time consuming. Figs. 8.14 and 8.15 were generated from an NVE MD simulation of 1000 buckyballs over 100 ps, at a temperature of 2500K and density of 700 amu/nm^3. Trajectory snapshots at 10 ps intervals were taken, giving 10 frames of data to analyse. Eqn. (8.18) tells us that there are 499,500 distinct pairs to consider. On an Intel Core i7 processor, this procedure takes around 24 hours. While this may be negligible compared to the time taken to run an equivalent large MD simulation, the potential has to be regenerated at each distinct temperature we wish to perform a CG simulation at. This can add up to a substantial amount of computer time used purely to generate the CG potentials, and for relatively small systems it may actually be more efficient to run the fully atomistic MD simulations.

A separate problem is the issue of measuring the internal energy potential using this separation binning method. From the simulation described above, an attempt to do this is shown in Fig. 8.16. Clearly, this data is unconverged and it is impossible to fit a sensible spline curve. We found that it would take an impractically long time to converge this data to a point where an acceptable spline fit could be made. Fortunately, we can still use the approach outlined in 8.4.3 to generate internal energy potentials indirectly from the free energy curves.

In Fig. 8.17, we plot the potential in Fig. 8.15 generated from this unconstrained technique at a density of 700 amu/nm^3. We also plot an unconstrained potential generated at 100 amu/nm^3, alongside the equivalent constrained pair potential obtained from the method described in 8.4.2. The two approaches produce very similar results which are indistinguishable to within line width. This suggests that many–body effects do not have a major effect on buckyball interactions.

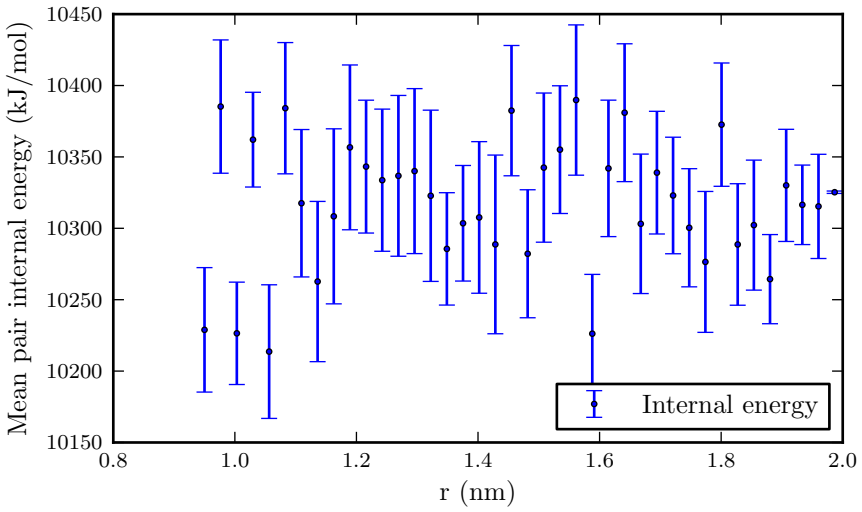


Figure 8.16: Mean internal energy calculated from pair binning method.

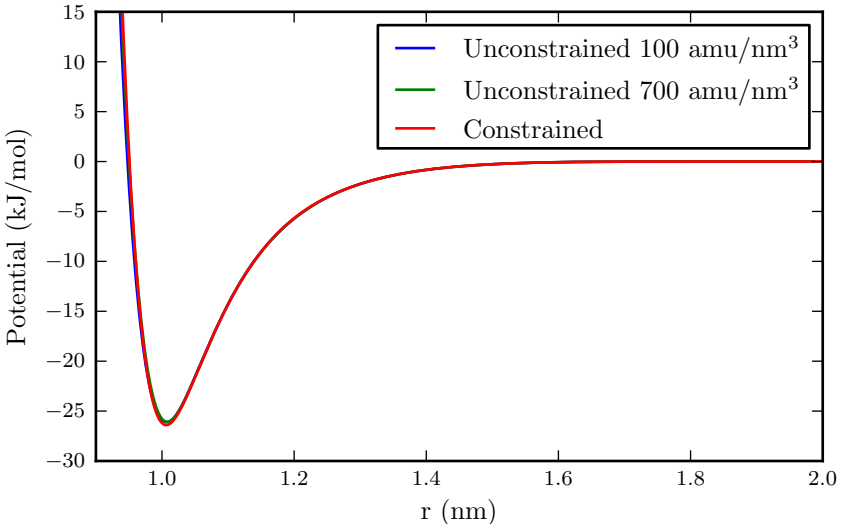


Figure 8.17: Free energy curve obtained from integrating spline fit in Fig. 8.14 from end of data range inwards, obtained at 2500K at two different densities. Constrained pair potential generated at 2500K from previous section plotted for comparison.

8.6 The Entropic Approach

In Chapter 6, we proved for a simple system of two dimers that the CG free energy of the dimer pair is equal to the relaxed energy at a fixed separation d, plus an entropic contribution due to the vibrational degrees of freedom in the dimers (see Eqn. (6.59)). This suggests a potentially very useful method of generating a CG free energy potential at finite temperature for a more complex system.

Firstly, we perform a constrained energy minimisation of a pair of molecules at fixed separation d, and repeat for many values of d to generate a relaxed potential $V_{\text{relaxed}}(d)$ as outlined in Section 6.2.2. We then calculate the normal mode frequencies ν_i of the constrained pair of molecules using the method described in Section 5.10. In general, these frequencies will change as a function of separation between the molecules, so the normal mode calculation must be repeated at each separation we are interested in. If we assume the normal modes are harmonic, which is always the case at low temperatures, then the entropy of each normal mode is simply the entropy of a classical harmonic oscillator S_{sho}, given in Eqn. (3.39). The total vibrational entropy of the molecular pair at separation d is then given by

$$S_{\text{tot}}(d) = \sum_{i}^{n_{\text{modes}}} S_{\text{sho}}(\nu_i(d)). \tag{8.19}$$

We may then calculate an entropic potential by analogy with the formula $F = U - TS$ as

$$V_{\text{entropic}}(d) = V_{\text{relaxed}}(d) - TS_{\text{tot}}(d). \tag{8.20}$$

We expect this potential to coincide with one generated via the constrained pair approach, to the extent that the harmonic approximation is a good one.

This entropic method has a major advantage over the constrained pair approach. Instead of performing numerous dynamical simulations of a constrained molecular pair and averaging to obtain a potential, in the entropic method it is sufficient to perform one energy minimisation and one normal mode calculation at each separation. In general, this makes the entropic potential much faster to generate than the constrained pair potential. Additionally, the constrained pair method must be repeated every time we wish to generate a potential at a new temperature. This is not necessary in the entropic method, since we have

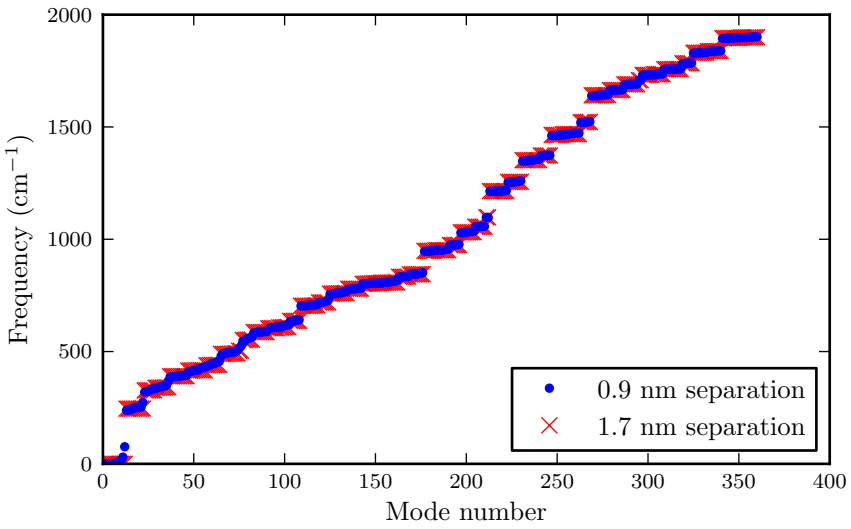


Figure 8.18: Normal mode frequencies of a constrained pair of buckyballs, at two different centre of mass separations. There are 360 modes in the system, the first six of which are translational and rotational modes with eigenvalues of zero.

analytic forms for all the temperature–dependent terms. This makes it trivial to calculate an entropic potential suitable for use at any temperature we wish.

We now demonstrate the application of the entropic approach to our system of buckyballs. The normal mode frequencies of the constrained buckyball pair are plotted for two different buckyball separations in Fig. 8.18. The difference between frequencies at different separations is more clearly shown in Fig. 8.19 The entropy calculated at various buckyball separations is plotted in Fig. 8.20 for three different temperatures. This entropic term is then subtracted from the relaxed potential shown in Fig. 8.5 to produce the entropic potential shown in Fig. 8.21 The entropic potential is not in complete agreement with the constrained pair potential generated at 10,000 K. We used the normal mode analysis program which comes with GROMACS to generate the normal mode frequencies. This program does not support intermolecular constraints during normal mode analysis, which may explain the discrepancy between the two potentials.

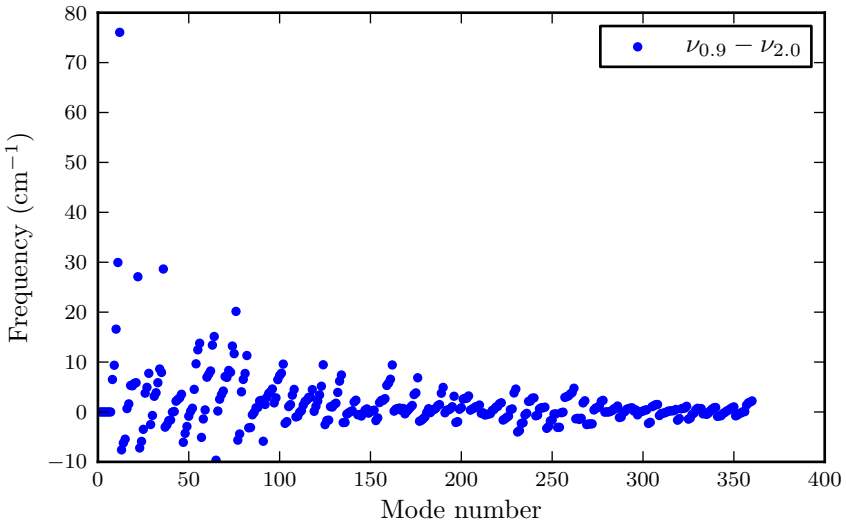


Figure 8.19: Difference between normal mode frequencies calculated at separations of 0.9 and 2.0 nm. The most significant differences are present in the low frequency modes.

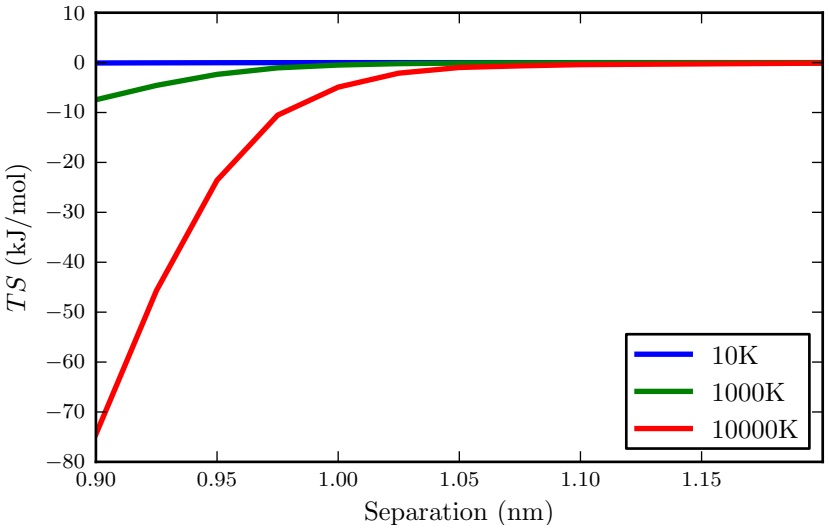


Figure 8.20: Entropy as a function of buckyball separation for three different temperatures. Each curve has been shifted to 0 by subtracting the entropy at infinite separation as a function of T. This shift has no effect on the dynamics.

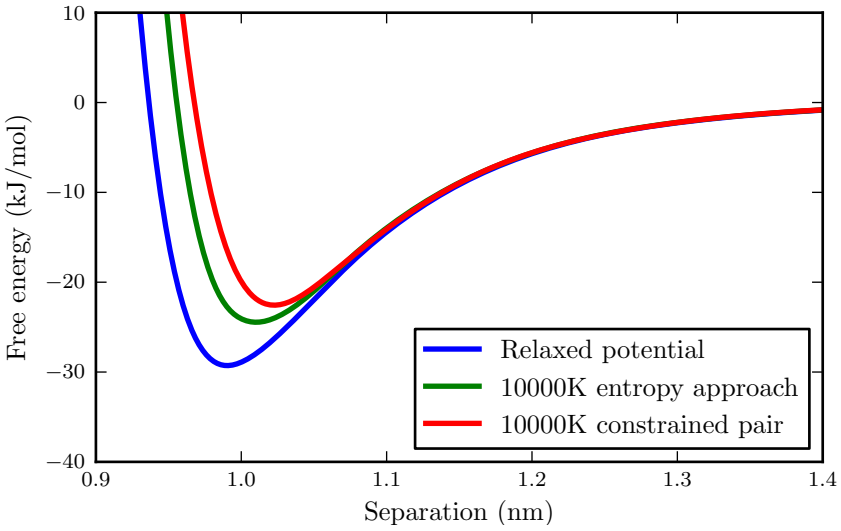


Figure 8.21: Entropic potential generated at three different temperatures. The constrained pair potential generated at 10,000 K is plotted for comparison.

8.7 Using Coarse–Grained Potentials in Simulations

In previous sections, we have generated coarse–grained potential curves using several different techniques. In order to use one of these potentials to perform a simulation, we must convert it into a format usable by GROMACS. The software requires that the potential and force be specified on a uniform grid of separations, with a grid spacing of 0.0005 nm for double precision calculations.

The first requirement is that the potential $V(r)$ must tend to 0 as r tends to infinity. This is not the case for some of our potentials, where the internal energy of the buckyballs themselves is included and therefore the potential is not zero at infinity. To rectify this, we can calculate the mean value of the potential at large separations, and subtract this from the whole curve.

The second step is to ensure the potential is smooth. We generally only know the value of the potential at a discrete set of points, so it is necessary to fit a smooth function to the data. We used a weighted cubic spline fit, where the inverse standard error of each data point was used as a weight. This ensures that points with large standard errors are weakly weighted in the fit. An alternative would be to use an interpolated spline fit, which passes through each data point, but this can lead to sudden sharp changes in the potential. This is undesirable as it will cause energy conservation problems in the final simulation.

The next problem is that GROMACS requires the potential to be defined for all possible values of r, from $r_{\min} = 0$ all the way to the maximum possible separation $r_{\max}$. Generally we cannot calculate values of the potential all the way down to $r_{\min} = 0$, as the simulation becomes numerically unstable when the buckyballs are forced very close together. Instead, we must compute the CG potential at the closest separation possible, and then fit a repulsive wall function at this point. We used a linear function with the same gradient as the potential at this point. The precise functional form is not important, as two CG particles will never actually get this close to each other during the course of a normal simulation because of the strong repulsive forces between them.

Under periodic boundary conditions in a cubic box of side length a, the maximum possible separation of two particles is half the length of the body diagonal,

$$r_{\max} = \frac{\sqrt{3}}{2}a. \tag{8.21}$$

It would be computationally wasteful to calculate the potential at such large separations, since we know the interaction strength is effectively zero. Instead, it suffices to compute the potential up to the cutoff radius r_{cut}, and then pad the rest of the data with zeros.

There is an additional problem with r_{cut} that needs to be considered. In the atomistic simulations, we used a cutoff radius of 1.1 nm for Lennard–Jones interactions between carbon atoms. In the CG simulations, we are instead considering interactions between *buckyballs*. If the centres of mass of two buckyballs are 1.1 nm away, there will still be carbon atoms on each buckyball within interaction range of each other. The attraction will in fact only go to zero when the centre of mass separation $r = r_{\text{cut}} + 2a$, where a is the buckyball radius. Because of this, we use an extended cutoff radius of $1.1 + 0.7 = 1.8$ nm in the CG simulations.

GROMACS performs no postprocessing of the tabulated potential file it is given. This causes a problem at the cutoff radius of 1.8 nm, because if the potential is not exactly zero at r_{cut}, there will be a discrete jump in the potential. This causes an infinity in the force $F(x)$, which in turn causes energy conservation problems. The solution is to add a switch function to the force, which causes the force to taper smoothly to zero at r_{cut}. Let x_b and x_e be the positions to begin and end switching the force, respectively. We define a switching function $s(x)$ as

$$s(x) = a(x - x_b)^2 + b(x - x_b)^3, \tag{8.22}$$

where

$$\begin{aligned} a &= -\frac{3}{\gamma^2}F(x_e) + \frac{1}{\gamma}F'(x_e), \\ b &= \frac{2}{\gamma^3}F(x_e) - \frac{1}{\gamma^2}F'(x_e), \\ \gamma &= x_e - x_b. \end{aligned} \tag{8.23}$$

We choose a cubic function for $s(x)$ since this allows us to make the first and second derivative of the force continuous at the boundaries. Then the force is modified as follows:

$$F(x) = \begin{cases} F(x) & \text{if } x < x_b, \\ F(x) + s(x) & \text{if } x_b \leq x \leq x_e, \\ 0 & \text{if } x > x_e. \end{cases} \tag{8.24}$$

This in turn causes a modification to the potential V(x) over the whole range. Recall

$$F(x) = -\frac{dV}{dx}. \tag{8.25}$$

By the fundamental theorem of calculus,

$$V(b) - V(a) = -\int_a^b F(x')dx'. \tag{8.26}$$

Since we are stipulating that $V(\infty) = 0$, this becomes

$$V(x) - V(\infty) = V(x) = -\int_\infty^x F(x')dx'. \tag{8.27}$$

From Eqn. (8.24), we can find the potential by integrating from infinity inwards towards zero. Clearly,

$$V(x) = -\int_\infty^x 0 \; dx' = 0 \text{ if } x > x_e. \tag{8.28}$$

If $x_b \leq x \leq x_e$,

$$\begin{aligned} V(x) &= 0 - \int_{x_e}^x F(x') + s(x')dx' \\ &= V(x) - V(x_e) - S(x) + S(x_e), \end{aligned} \tag{8.29}$$

where $S(x)$ is the definite integral of $s(x)$, again by the fundamental theorem of calculus. Finally, if $x < x_b$,

$$\begin{aligned} V(x) &= 0 - \int_{x_e}^{x_b} F(x')dx' - \int_{x_b}^x F(x')dx' \\ &= V(x) - V(x_e) + S(x_e) - S(x_b). \end{aligned} \tag{8.30}$$

Clearly, modifying the force with a switch function causes the potential to be shifted across its entire range up to r_{cut}. Generally, this shift is small, and is a

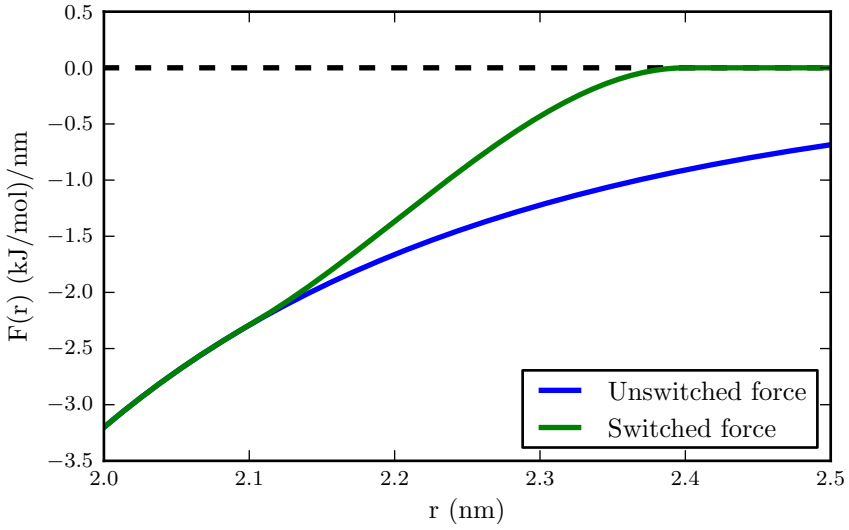


Figure 8.22: Switched and unswitched versions of the Lennard–Jones force for $x_b = 2.1$ and $x_e = 2.4$ nm. The switched version tends smoothly to zero at the cutoff.

small price to pay to avoid energy conservation problems due to the finite cutoff radius. In Figs. 8.22, 8.23 and 8.24, we show examples of the switch function being applied to the Lennard–Jones potential.

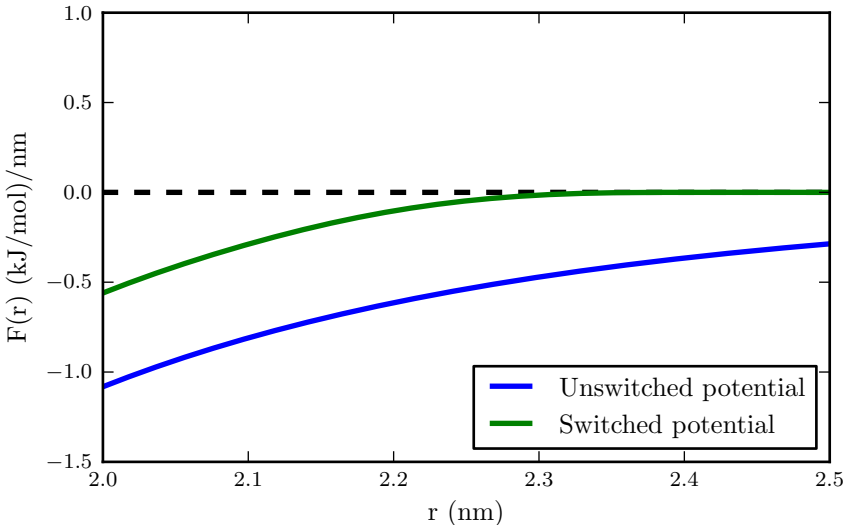


Figure 8.23: Switched and unswitched versions of the Lennard–Jones potential for $x_b = 2.1$ and $x_e = 2.4$ nm. The switched version tends smoothly to zero at the cutoff.

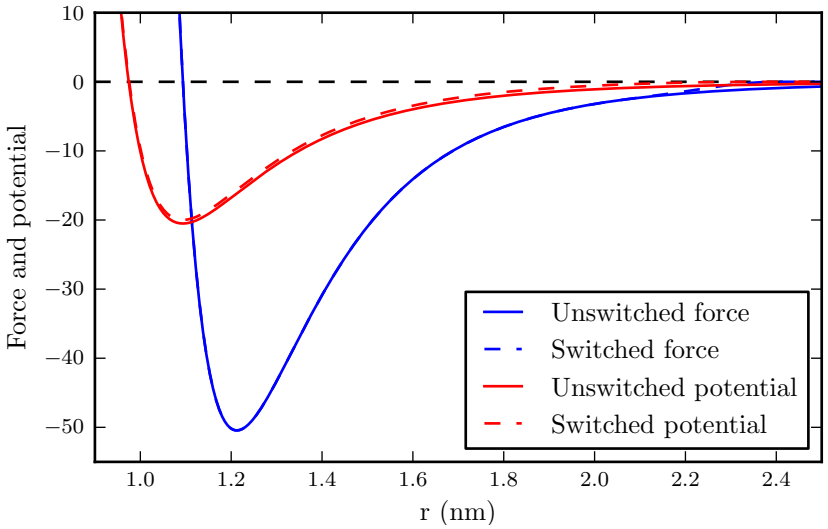


Figure 8.24: Switch force and potential on the same plot. Applying the switch affects the potential across the entire switching range, but the effect is small compared to energy conservation problems which occur with the unswitched potential.

Chapter 9

Coarse–Grained Simulation Results

In the previous chapter, we discussed several different methods for generating CG potentials for the C_{60} molecule. We now turn to the question of which methods may be most appropriate for use under a given set of simulation conditions, and whether any general recommendations about which potential to use can be made. In all cases this is done by comparing results from MD simulations using the CG potentials with reference fully atomistic MD simulations.

9.1 Inadequacy of Rigid and Relaxed Potentials at High T

Ideally, we would like to test the rigid and relaxed CG potentials we have generated against fully atomistic MD simulations of C_{60} performed at low temperatures, since these CG potentials are generated at zero temperature and we therefore expect them to perform best in this domain. Unfortunately, with our MD forcefield for buckyballs (OPLS–AA), at temperatures below around 2500K the C_{60} molecules tend to aggregate into a cluster. This is effectively a nanodroplet surrounded by vapour. This causes problems when calculating the quantities we are using as benchmarks. For example, we calculate pressure as a function of density, but we obtain negative pressures for the cluster of bucky-

balls. Modifying the density also has virtually no effect on the cluster until very high densities are reached, since the cluster only takes up a very small fraction of space in the simulation cell until the cell size is reduced sufficiently.

We also measure diffusion constants, but these are effectively zero inside the cluster for any density we choose. There are also equilibration problems with running the fully atomistic MD simulations long enough to generate a cluster of C_{60} at equilibrium. The CG simulations can of course be run for much longer to obtain an equilibrium cluster, but then we are not making a fair comparison with the reference system.

In short, it is not practical to compare our CG models with fully atomistic simulations at low temperatures. Instead, we are restricted to temperatures of approximately 2500K and above with our force field, when the buckyballs are in the diffusive regime for a given density. In Figs. 9.1, 9.2 and 9.3 we have plotted the pressure, coordination number and diffusion constants of C_{60} at a temperature of 2500K as a function of density.

In Fig. 9.1, it is clear that the rigid and relaxed potentials are poor at reproducing the correct pressure of the atomistic simulations, except at very low densities where the buckyballs essentially act as free particles. This is because the potential well in the CG potentials is too deep, which causes the buckyballs to spend more time on average trapped inside the well. This in turn causes the virial term in Eqn. (5.78) to decrease, lowering the pressure.

In Fig. 9.2, it is clear that both the rigid and relaxed CG potentials significantly overestimate the coordination number. This is again because the well depth in the pair potential is too deep. This causes a pronounced peak in the pair correlation function at the position of the potential minimum, which is too high (see Figs. 9.4 and 9.5).

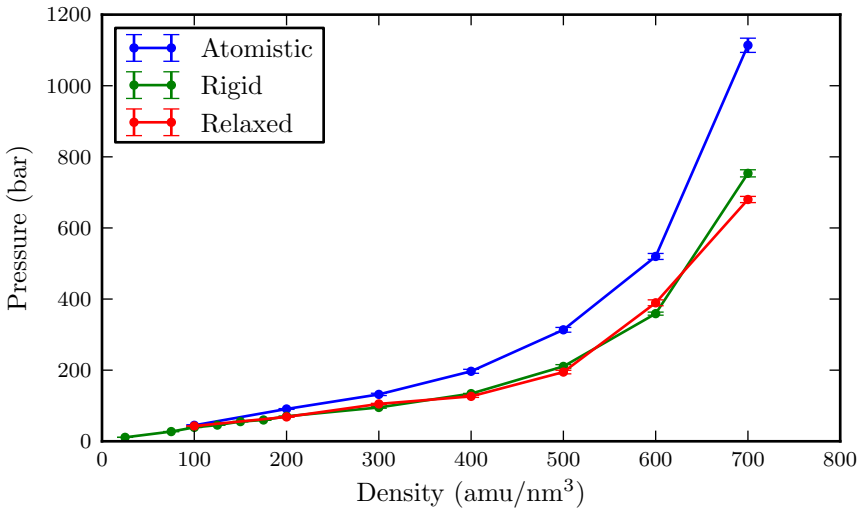


Figure 9.1: Pressure of C_{60} as a function of density at 2500K.

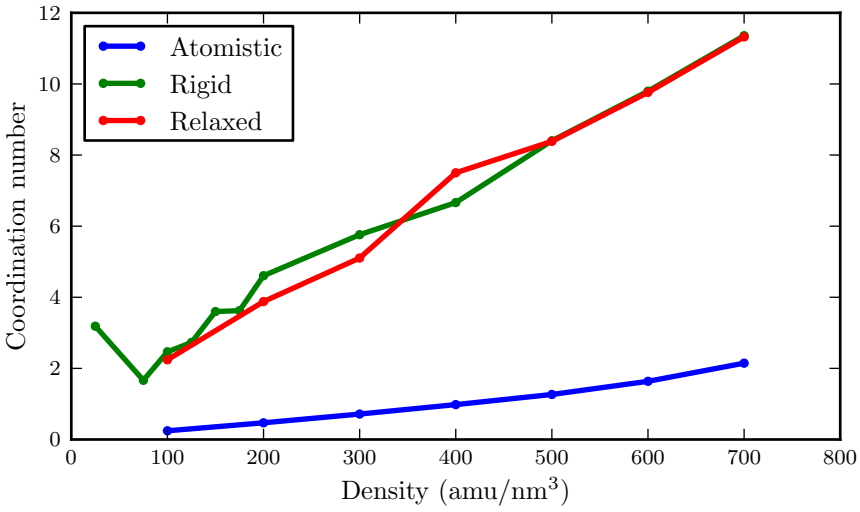


Figure 9.2: Coordination number of C_{60} as a function of density at 2500K.

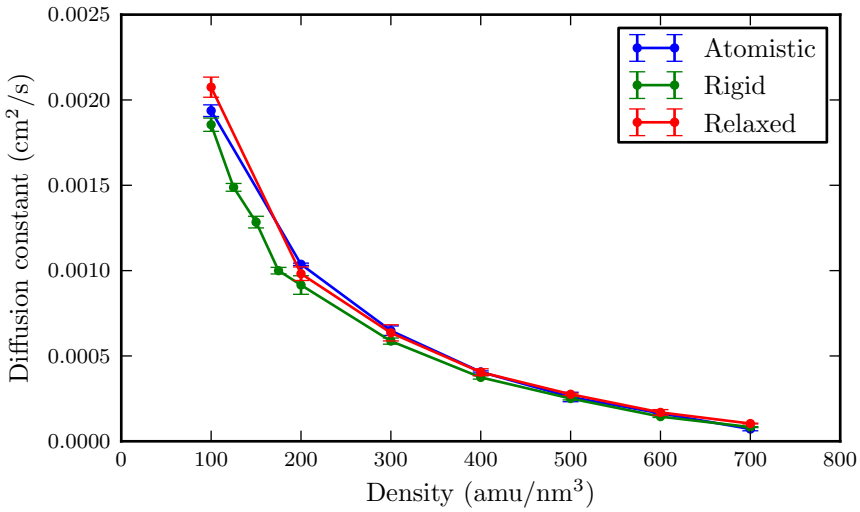


Figure 9.3: Diffusion constant of C_{60} as a function of density at 2500K.

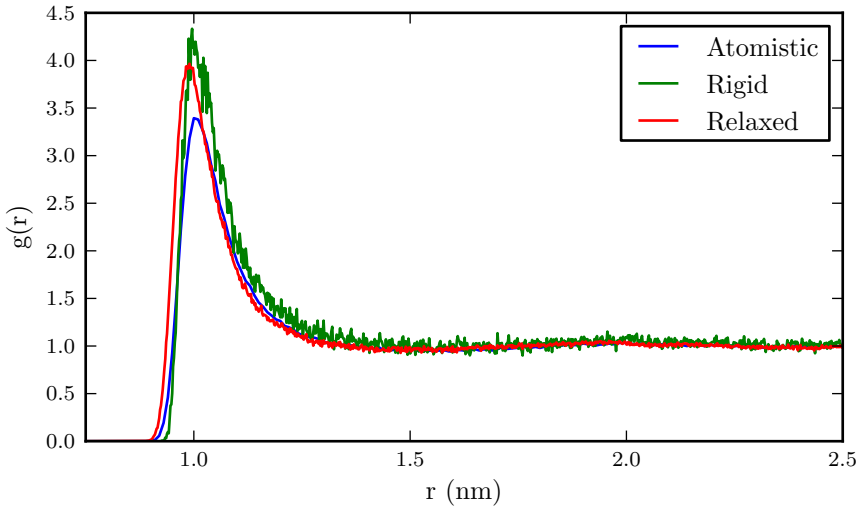


Figure 9.4: Pair correlation function of C_{60} at a density of 100 amu/nm^3 and temperature of 2500K. The first peak height is too high for both CG potentials, causing an incorrect coordination number.

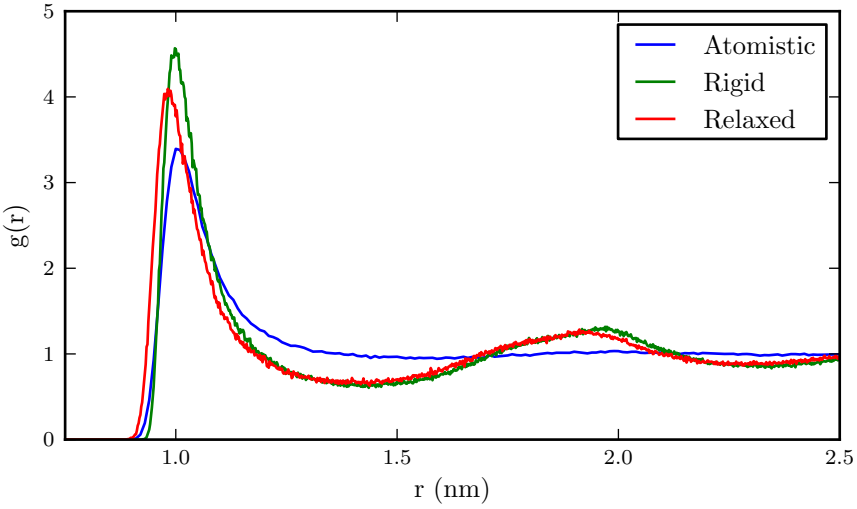


Figure 9.5: Pair correlation function of C_{60} at a density of 700 amu/nm^3 and temperature of 2500K. At this high density, both CG potentials exhibit completely incorrect structure compared to the reference MD simulation.

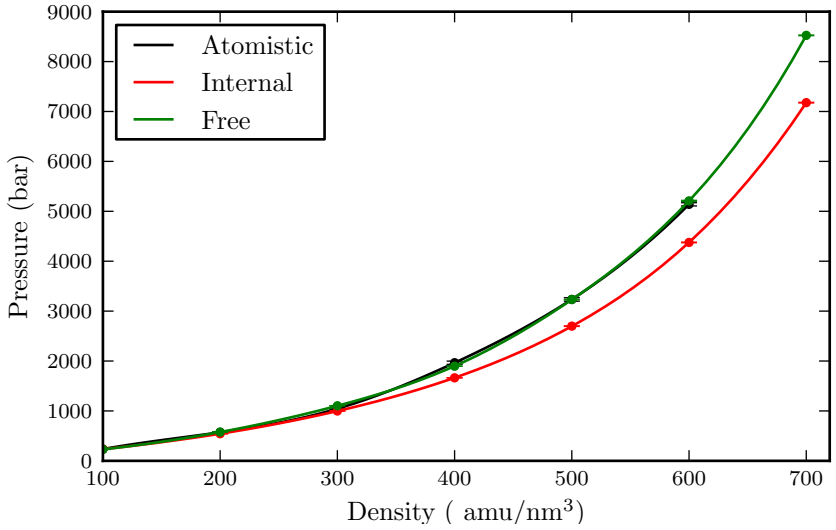


Figure 9.6: Pressure as a function of density at a temperature of 10,000K. The internal energy CG potential fails to reproduce correct pressures at high densities. Solid lines are spline fit to data as a guide to the eye.

9.2 Inconsistency Between Free and Internal Energy Potentials

In Fig. 8.11, we demonstrated that the internal energy CG potential is effectively independent of temperature. However, from Fig. 8.9 it is clear that the free energy CG potential depends strongly on temperature, and only the zero temperature limit of the free energy potential coincides with the internal energy potential. This means that we would expect the internal energy potential to provide an increasingly worse approximation to the full atomistic potential as the temperature is increased. To show that this is the case, we performed a simulation of 1000 buckyballs for 100 ps at a temperature of 10,000 K, both fully atomistically and with free and internal energy CG potentials. This very high temperature was used to exaggerate the differences between the free and internal energy potentials.

In Fig. 9.6, it is clear that the free energy potential correctly reproduces the pressure measured in the fully atomistic MD simulation across the whole range of densities studied. However, the internal energy potential substantially

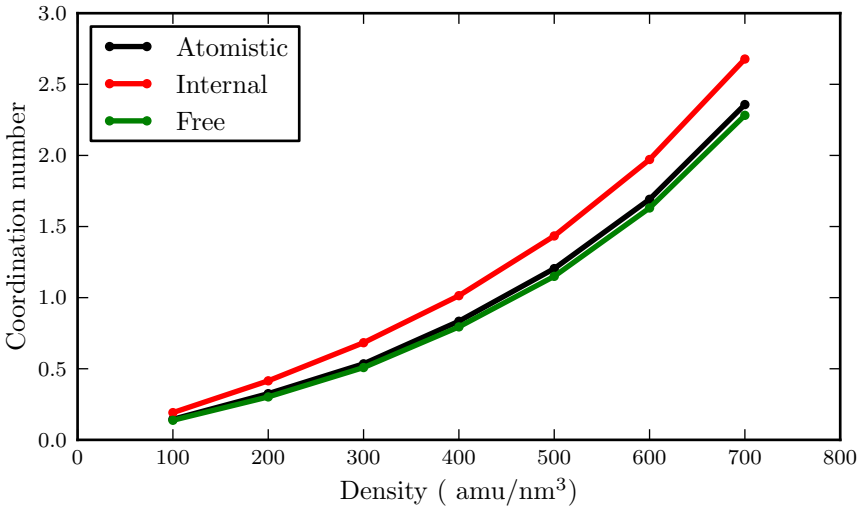


Figure 9.7: Coordination number as a function of density at a temperature of 10,000K.

underestimates the correct pressure at high densities. This is because the effect of the shallow well depth in the internal energy potential is most acute at high densities, when the buckyballs are very close together.

In Fig. 9.7, the free energy potential is again in close agreement with the fully atomistic MD simulation. The internal energy potential substantially overestimates the coordination number over the whole density range. This is because the first peak heights in the pair correlation function are too high when using the internal energy potential, due to the deep potential well.

In Fig. 9.8, it appears that both the free and internal energy CG potentials are equally good at predicting the correct diffusion constant.

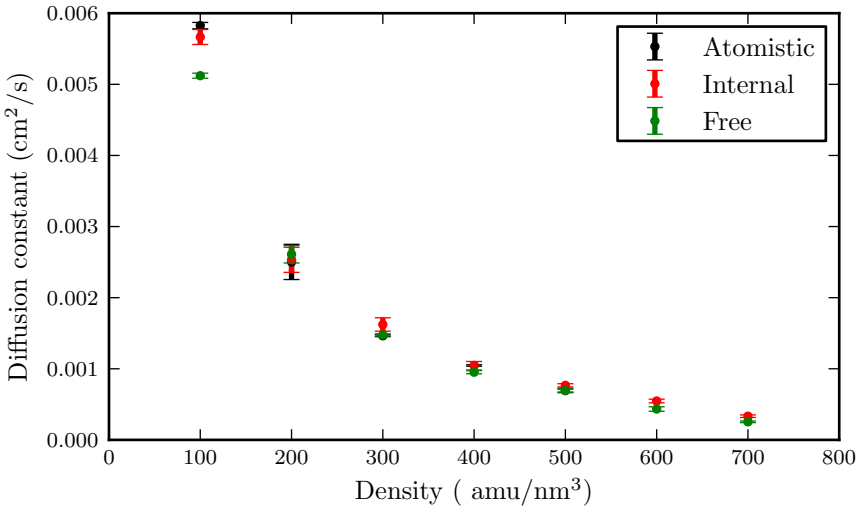


Figure 9.8: Diffusion constant as a function of density at a temperature of 10,000K.

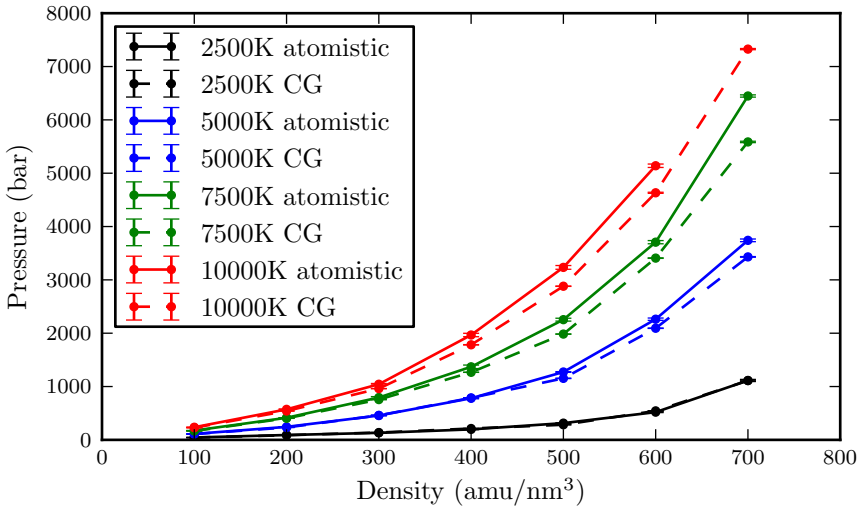


Figure 9.9: Pressure against density for a range of temperatures, using a constrained pair CG potential generated at 2500K in each case.

9.3 Transferability of CG Potentials

The CG potentials we have used so far have each been generated under a specific set of experimental conditions; for example, the constrained pair CG potentials were each generated at a specific temperature. It is important to ask how relevant these potentials are when they are used to run simulations under a different set of experimental conditions. This property is commonly referred to as the *transferability* of the CG potential [71].

In the following figures, we chose a constrained pair CG potential generated at 2500K, and used it to run simulations at 5000K, 7500K and 1000K. The results are compared against fully atomistic simulations at the corresponding temperatures. In Fig. 9.9, it can be seen that the CG simulation pressure is in almost perfect agreement with the fully atomistic simulation at 2500K, the temperature the CG potential was generated at. As the temperature is increased, the CG potential yields a progressively worse approximation to the correct atomistic pressure. In Fig. 9.10, again it can be seen that the CG potential performs well at 2500K, but progressively less well at higher temperatures. In Fig. 9.11, the CG potential generated at 2500K performs relatively well across the whole

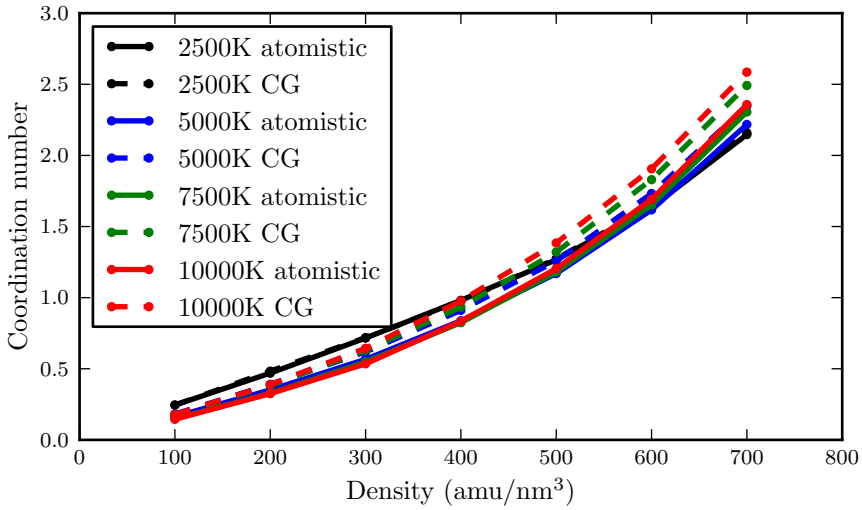


Figure 9.10: Coordination number against density for a range of temperatures, using a constrained pair CG potential generated at 2500K in each case.

range of temperatures we tested. This agrees with our earlier observation that the diffusion constant is relatively insensitive to the CG potential used.

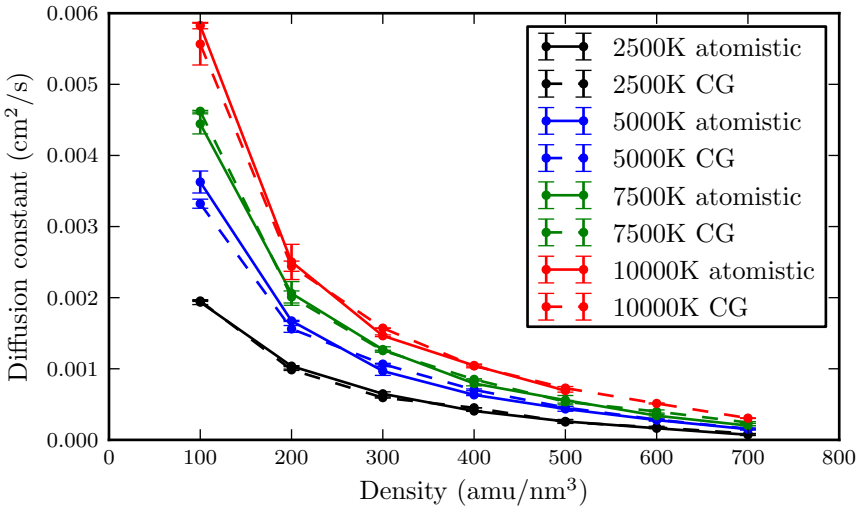


Figure 9.11: Diffusion constant against density for a range of temperatures, using a constrained pair CG potential generated at 2500K in each case.

9.4 Importance of Many–Body Effects

In Figs. 9.12, 9.13 and 9.14 we have compared our fully atomistic MD simulations with unconstrained pair CG potential and constrained pair CG potential results. All potentials were generated at 2500K, and the unconstrained CG potential was generated at a density of 700 amu/nm^3.

In the plot of diffusion constants, both CG potentials perform equally well with respect to the atomistic simulations. This is consistent with our previous results that diffusion rates are relatively insensitive to the potential used. In the pressure and coordination plots, we see that both CG methods produce very similar results to the atomistic MD simulations. This is consistent with our findings that, at least in the case of buckyballs, the CG potentials generated via the constrained pair and unconstrained box methods are very similar. This suggests that many body effects are not strong for this system.

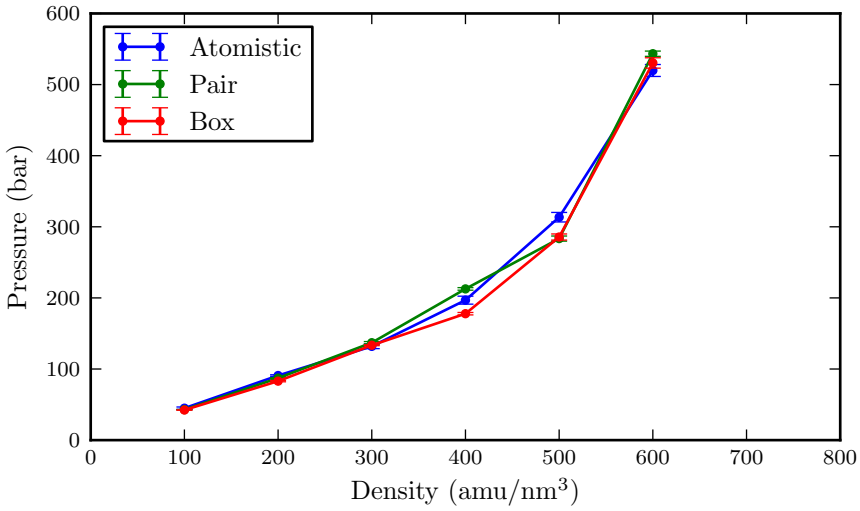


Figure 9.12: Comparison of pressure from atomistic results, constrained pair CG potential and unconstrained CG potential. All potentials were generated at 2500K, and the unconstrained CG potential was generated at a density of 700 amu/nm^3. All simulations were performed at 2500K.

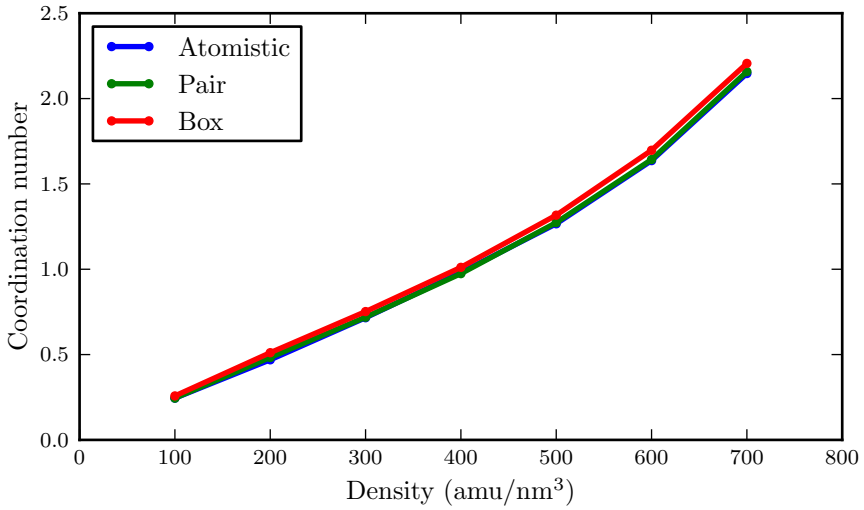


Figure 9.13: Comparison of coordination number from atomistic results, constrained pair CG potential and unconstrained CG potential. All potentials were generated at 2500K, and the unconstrained CG potential was generated at a density of 700 amu/nm^3. All simulations were performed at 2500K.

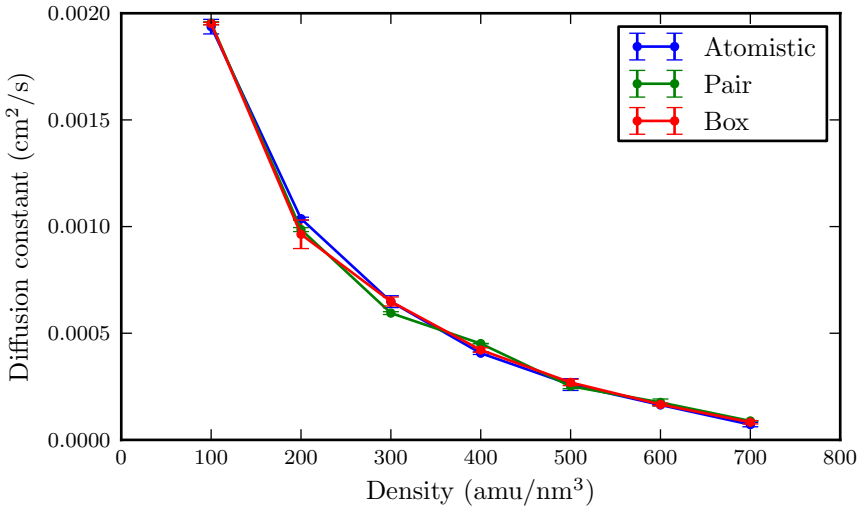


Figure 9.14: Comparison of diffusion constant from atomistic results, constrained pair CG potential and unconstrained CG potential. All potentials were generated at 2500K, and the unconstrained CG potential was generated at a density of 700 amu/nm^3. All simulations were performed at 2500K.

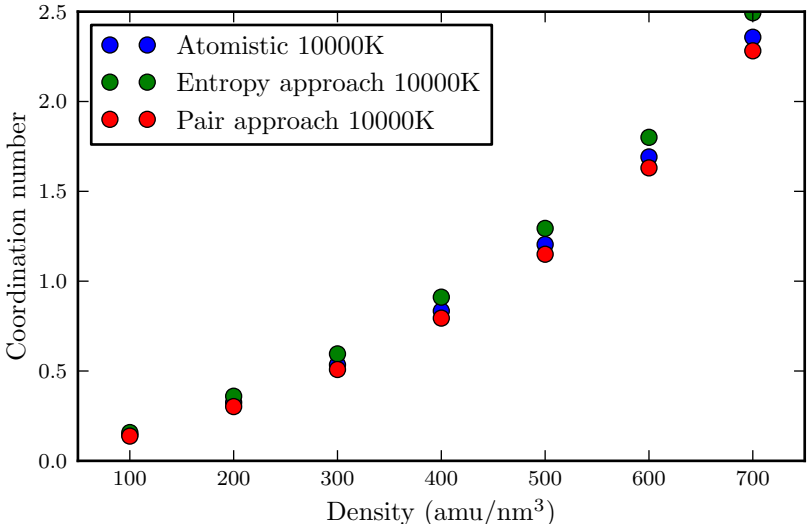


Figure 9.15: Coordination number at 10,000K from fully atomistic simulations, constrained pair CG potential and entropic CG potential.

9.5 Entropy of Buckyball Pairs

In this section, we compare the entropic approach CG potential generated at 10,000K with the constrained pair CG potential generated at the same temperature. From Fig. 8.21, it is clear that the entropic CG potential has a deeper well than the constrained pair CG potential. This manifests itself in Fig. 9.15 as a coordination number which is too high compared to the reference atomistic simulation, and a pressure which is too low at high densities as seen in Fig. 9.16.

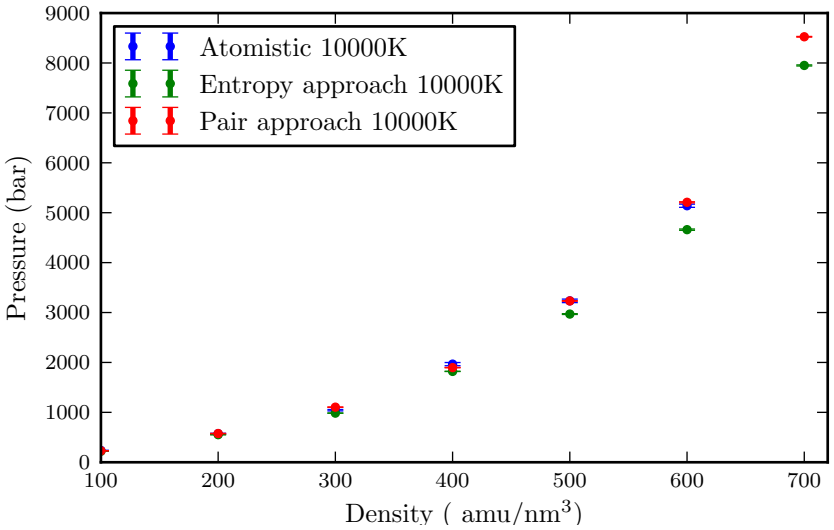


Figure 9.16: Pressure at 10,000K from fully atomistic simulations, constrained pair CG potential and entropic CG potential.

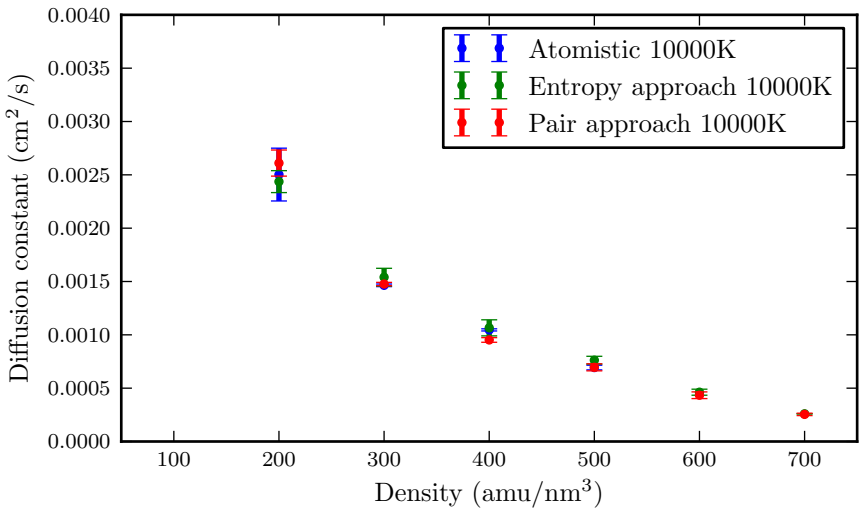


Figure 9.17: Diffusion constant at 10,000K from fully atomistic simulations, constrained pair CG potential and entropic CG potential.

9.6 Computational Efficiency

There would be no benefit to coarse–graining if it did not lead to one of our stated goals, which is to dramatically reduce the amount of computer time required to simulate a given system. According to our GROMACS simulations, on average 91% of the total simulation time is spent evaluating interatomic forces, so this is the most relevant part of the simulation to examine. To get an idea of what degree of speedup to expect, it is instructive to consider the amount of calculations needed to evaluate the interatomic forces in a fully atomistic MD simulation.

In the case of the OPLS–AA force field we have used for our simulations of buckyballs, within each molecule there are 90 harmonic bond terms, 180 angular terms and 360 dihedral terms, for a total of 630 intermolecular force calculations per buckyball. In addition, if we ignore cutoffs and assume each carbon atom interacts with every other carbon atom via a Lennard–Jones force, there are $\frac{1}{2}N(N-1)$ distinct nonbonded pair interactions to calculate. For our simulations of 1000 buckyballs, there are then a total of approximately 1.8×10^9 force calculations per time step. In contrast, in the CG simulation there are only approximately 5×10^5 distinct nonbonded pair interactions to calculate, and no bonded interactions.

To benchmark our simulations, we ran both a fully atomistic and CG NVE simulation of 1000 buckyballs for 100 ps at 10,000K on a PC workstation with a 3.4 GHz Intel Core i7 processor. All simulations were run using only one CPU core, and each simulation was repeated 5 times to obtain an average run time.

The fully atomistic simulations took 348 ± 1 minutes to complete on average, whereas the CG simulations took an average of 18 ± 1 seconds. This makes the CG simulations approximately 1,160 times faster than the equivalent fully atomistic simulation, a very substantial speedup.

Chapter 10

Discussion and Conclusion

10.1 Summary

In Chapter 3, we showed that it is possible to derive the CG partition function analytically for a system where the interactions between particles can be described by a harmonic potential. Once the CG partition function is known, all information about the statistical mechanical state of the system can be easily derived. We did this in two ways: via the standard delta function approach found in the literature [45], and via a new method of selective integration.

This analytical solution for the CG partition function is valuable as an aid to understanding, but unfortunately it is not applicable to more realistic systems. This is because in a system of interacting molecules, the *intermolecular* forces (often described by the Lennard-Jones potential, for example) are strongly anharmonic.

Fortunately, in many commonly used MD force fields, the *intramolecular* forces can be described entirely by harmonic terms, such as the bond stretch, angle stretch and dihedral terms described in Section 5.5.2. This suggests a practical method for coarse–graining: integrate out the harmonic intramolecular terms in the partition function, which will result in a modified intermolecular force in the CG model. This is exactly what we did in Chapter 6 for the simple example of two interacting dimers.

For more complicated molecules, such as buckyballs, it is necessary to resort to numerical techniques to obtain an approximation to the full CG partition

function. To make this problem tractable, and also to produce a potential which is efficiently usable in practice, we must restrict ourselves to considering pair potentials only. In Chapter 8, we described several methods of doing this.

The simplest approach is the rigid approximation, where the internal degrees of freedom of each molecule are held completely rigid. The potential is then generated by performing a spherical average of the interaction energy of the two rigid molecules constrained at a fixed separation. This is exactly the approach used by Girifalco in [69]. Because of the spherical symmetry of the C_{60} molecule, he was able to perform the spherical averaging procedure analytically to derive a formula for the potential. We arrived at the same result by performing the spherical average numerically. Of course, our numerical approach could be applied to any non–spherical molecule. We would expect this to work well for relatively small, "stiff" molecules. It is unlikely to work well for large, soft biomolecules such as polymers and proteins, where conformational changes of the molecules as they approach each other are very important in determining the strength of interaction.

The logical next step is to allow the molecules to find their minimum energy configuration, for a given fixed centre of mass spacing. This is the relaxed approximation. While this is perhaps a more physical approach than insisting the molecules remain rigid, we found that the relaxed potential still gives inadequate agreement with fully atomistic MD simulations performed at high temperatures. A different approach is therefore needed.

To capture the temperature dependence of the CG potential, we next proposed the constrained pair approach, which uses dynamical simulations of molecules constrained at a fixed separation to calculate the mean interaction energy between molecules. We found that a brute force approach to obtaining this mean energy was prohibitively expensive. Fortunately, it is instead possible to sample the mean force acting between the centres of mass of the molecules. This force can then be integrated to obtain the free energy CG potential, which exhibits a strong dependence on the temperature of the underlying constrained MD simulation. From this, we proposed a way of estimating the internal energy potential. The internal energy is nearly independent of temperature, apart from a constant shift factor which has no effect on the dynamics. This is due to the harmonic nature of the interactions between the microscopic degrees of freedom. The internal energy potential therefore suffers from the same disadvantages as

the rigid and relaxed methods when applied to high temperature simulations.

The constrained pair approach is simple to implement, and it can be used to generate a potential in a matter of hours on a standard desktop PC. We found that free energy potentials generated in this manner produced very close agreement to the benchmark fully atomistic simulations when calculating diffusion constants, pressures and coordination numbers (or equivalently radial distribution functions), provided the potential was regenerated at each specific temperature we wished to study. The CG simulations themselves are computationally cheap to perform, which is why the time taken to generate each potential becomes an important consideration.

Next, we discussed an alternative method, based on an unconstrained dynamical simulation of C_{60} molecules in a periodic simulation box. This method is essentially the same as the most common application of the force matching technique [18]. We were hopeful that this would avoid the sampling problems associated with a brute force calculation of the internal energy potential, and also provide a more computationally efficient method to generate a potential for each temperature. Unfortunately, we found that neither of these were the case. The number of trajectory snapshots required to obtain a small enough statistical error to fit smooth cubic splines to is very large. An additional problem is that the number of configurations of molecules which are close together is very low due to the repulsive nature of the Lennard–Jones force at short distances. This makes an accurate measurement of the potential at short distances quite difficult. These problems can be mitigated to some extent by presupposing a functional form for the CG potential (for example a Lennard–Jones like interaction), and using the binned force data to optimize parameters in this fit. We found the shape of our inter–buckyball potentials to be poorly fitted by commonly used nonbonded potentials such as the Lennard–Jones, Buckingham and Mie potentials, so this was not a promising solution for us. It is also significantly more difficult to implement the unconstrained box approach in computer code than the other methods we have discussed. This is because care has to be taken over the force binning due to periodic boundary conditions, and the process of recomputing the energies between distinct buckyball pairs requires careful scripting.

We also found that the importance of many–body effects is negligible, at least for the case of C_{60}. It is therefore easy to recommend the constrained pair

approach over the unconstrained box approach, since it provides quantitative agreement with reference fully atomistic simulations, is easier to implement and takes less computer time to generate the CG potentials.

The final method we discussed was the entropy approach. This idea was inspired by the finite temperature case of the two dimers in Section 6.3. Assuming the internal energy potential is independent of temperature (which is the case in our work), we showed that the free energy CG potential is simply the relaxed potential, plus a temperature–dependent shift which is due to the vibrational entropy of each dimer.

This suggests an approach that can be applied to a more realistic molecule. We can always calculate the normal modes of a molecule. Treating each mode as a simple harmonic oscillator (a valid assumption at low temperatures), we can then estimate the vibrational entropy of the molecule. This is then added to the relaxed CG potential to obtain the CG free energy at any temperature we wish.

The entropy approach is appealing for two reasons. Firstly, numerically all that is required is one energy minimisation and one normal mode calculation for each centre of mass separation. This makes the method fast to implement in practice. Secondly, since we have analytic formulae for the entropy as a function of temperature, these calculations have to be performed only once, regardless of how many temperatures we would like to study. This makes the amount of time spent generating the potentials negligible when compared to either the constrained or unconstrained approach.

When we applied the entropy method to C_{60}, we generated potentials which are not in exact agreement with the equivalent constrained pair potentials. We suspect this is due to the difficulty of calculating the normal modes of a pair of molecules coupled through a constraint. This is the subject of further investigation.

It is also important to emphasise that regardless of the method used to generate the CG potential, all the CG simulations were at least three orders of magnitude faster than their fully atomistic MD counterparts. This speed increase comes from several different areas. Firstly, the number of degrees of freedom in the CG simulations is reduced by a factor of 60 from the atomistic simulations in the case of buckyballs. This makes computation of neighbour lists faster, and drastically reduces the number of pair interactions that need

to be calculated. Secondly, the CG potentials are supplied in tabulated format on a fine mesh. GROMACS then uses spline interpolation to calculate the CG potential for any intermediate particle separation. This effectively allows the CG potential to be implemented as a lookup table in memory, which is much faster than an analytic calculation of a force term. Thirdly, since the fastest degrees of freedom (the carbon–carbon bond stretch) have been integrated out of the CG system, a larger discrete time step can be used for integrating the equations of motion, without affecting the energy conservation of the simulations. We found that a time step three times larger (3 fs) could be used in the CG simulations whilst still maintaining acceptable energy conservation, although in the results reported a time step of 1 fs was still used to maintain consistency with the fully atomistic simulation. In CG simulations of other biomolecular systems, time steps of 10 fs and higher have been employed [72].

10.2 Future Work

In this work, we have introduced several different techniques for generating CG pair potentials between molecules. Since we were focused on developing and testing the methodology of these techniques, we were only able to apply them to a simple test system of buckyballs. We have not even scratched the surface of possible applications, but now that the methodology is in place, it should be relatively simple to extend it to more complicated molecules. Molecules such as water and hydrocarbons which exhibit directionality would be interesting systems to investigate, to see if the constrained pair approach is capable of capturing the effect of orientation in the CG potential.

In biological simulation, it is very rare to consider molecules in vacuum as we have done here. Instead, almost all simulations are carried out in the presence of (implicit or explicit) water molecules. The constrained pair approach suggests a way of including the solvent environment in the CG potential: we simply sample the centre of mass force between two constrained molecules with the solvent molecules explicitly present in the simulations. This should lead to a modification in the vacuum CG potential due to excluded volume effects, charge screening and so on.

Any of the approaches we have discussed (except the unconstrained box method) could easily be applied to a pair consisting of two different molecular

species. Another important issue which has not been discussed in this work is the use of multiple CG sites per molecule. We would expect the dynamics of a long polymer chain, for example, to be poorly described by a CG model consisting of only one CG site. A simple extension to the methods we have described would allow us to handle these multi-site models. Firstly, we decide which atoms in each molecule correspond to a given CG site. The position of the CG site for a given group of atoms is most easily chosen to be the centre of mass of that group, since the force on that CG site is then simply the sum of the forces on the atoms in the group (the centre of geometry of the group is an alternative choice). We then perform our constrained pair approach, individually sampling the forces between CG sites on opposite molecules as the simulation progresses. We then need to calculate the mean intramolecular force between each pair of sites on a single molecule. This can either be calculated during the course of the constrained pair simulation, or on a single unconstrained molecule in vacuum. To mitigate sampling issues, these intramolecular forces may have to be used to fit parameters in a standard functional form for a bonded interaction, such as a harmonic spring term between neighbouring sites. This allows us to build up an entire CG force field for a set of complex molecules by generating sequences of pair potentials between CG sites.

10.3 Conclusion

In this work, we have presented a systematic investigation of different coarse–graining methods and how to apply them to MD simulations. This requires an understanding of the underlying theory of coarse–graining, especially the evaluation of the CG partition function. We have done this analytically for a harmonic system, and in the case of a system of two dimers. However, in general, we are only able to obtain approximations to the full CG partition function in more realistic systems. Nevertheless, we are able to obtain quantitative agreement with reference fully atomistic MD simulations, using simple pair potential approximations to the true many–body CG potential. We have developed several new strategies for generating these pair potentials. We found that the constrained pair approach provides an optimal balance between ease of generation and agreement with reference fully atomistic simulations, provided careful consideration of temperature transferability is undertaken. An impor-

tant point to make is that no CG methodology can be blindly applied to a new system. Instead, each CG methodology must be evaluated on its own merits, and we must be careful to preserve enough information about the system to calculate the properties we are interested in. No CG methodology is capable of preserving every detail of the underlying microscopic model.

Bibliography

[1] M. P. Allen and D. J. Tildesley, *Computer Simulation of Liquids*. Oxford Science Publications, 1987.

[2] D. Frenkel and B. Smit, *Understanding Molecular Simulation*. Academic Press, Inc., 2nd ed., 2001.

[3] W. Jorgensen, "Development and testing of the OPLS all-atom force field on conformational energetics and properties of organic liquids," *J. Am. Chem. Soc.*, vol. 7863, no. 15, pp. 11225–11236, 1996.

[4] J. Diemand, R. Angélil, K. K. Tanaka, and H. Tanaka, "Large scale molecular dynamics simulations of homogeneous nucleation," *J. Chem. Phys.*, vol. 139, p. 074309, Aug. 2013.

[5] D. E. Shaw, K. J. Bowers, E. Chow, M. P. Eastwood, D. J. Ierardi, J. L. Klepeis, J. S. Kuskin, R. H. Larson, K. Lindorff-Larsen, P. Maragakis, M. A. Moraes, R. O. Dror, S. Piana, Y. Shan, B. Towles, J. K. Salmon, J. P. Grossman, K. M. Mackenzie, J. A. Bank, C. Young, M. M. Deneroff, and B. Batson, "Millisecond-scale molecular dynamics simulations on Anton," in *Proc. Conf. High Perform. Comput. Networking, Storage Anal. - SC '09*, (New York, New York, USA), p. 1, ACM Press, 2009.

[6] G. E. Moore, "Cramming more components onto integrated circuits, Reprinted from Electronics, volume 38, number 8, April 19, 1965, p. 114," *IEEE Solid-State Circuits Newsl.*, vol. 20, pp. 33–35, Sept. 2006.

[7] S. Kohlhoff, "Crack propagation in bcc crystals studied with a combined finite element and atomistic model," *Phil. Mag. A*, vol. 64, pp. 851–878, 1991.

[8] B. Kasemo, "Biological surface science," *Surf. Sci.*, vol. 500, pp. 656–677, 2002.

[9] E. Tadmor, M. Ortiz, and R. Phillips, "Quasicontinuum analysis of defects in solids," *Philos. Mag. A*, vol. 73, pp. 1529–1563, 1996.

[10] V. Krishna, G. S. Ayton, and G. A. Voth, "Role of protein interactions in defining HIV-1 viral capsid shape and stability: a coarse-grained analysis," *Biophys. J.*, vol. 98, pp. 18–26, Jan. 2010.

[11] M. E. Tuckerman, B. J. Berne, and G. J. Martyna, "Molecular-Dynamics algorithm for multiple time scales - systems with long-range forces," *J. Chem. Phys.*, vol. 94, pp. 6811–6815, May 1991.

[12] A. Laio and M. Parrinello, "Escaping free-energy minima.," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 99, pp. 12562–6, Oct. 2002.

[13] W. G. Noid, "Perspective: Coarse-grained models for biomolecular systems.," *J. Chem. Phys.*, vol. 139, p. 090901, Sept. 2013.

[14] S. Riniker, J. R. Allison, and W. F. van Gunsteren, "On developing coarse-grained models for biomolecular simulation: a review.," *Phys. Chem. Chem. Phys.*, vol. 14, pp. 12423–30, Sept. 2012.

[15] E. Brini, E. A. Algaer, P. Ganguly, C. Li, F. Rodríguez-Ropero, and N. F. A. van der Vegt, "Systematic coarse-graining methods for soft matter simulations a review," *Soft Matter*, vol. 9, p. 2108, July 2013.

[16] H. I. Ingólfsson, C. A. Lopez, J. J. Uusitalo, D. H. de Jong, S. M. Gopal, X. Periole, and S. J. Marrink, "The power of coarse graining in biomolecular simulations," *Wiley Interdiscip. Rev. Comput. Mol. Sci.*, vol. 4, pp. 225–248, May 2014.

[17] S. Izvekov and G. A. Voth, "A Multiscale Coarse-Graining Method for Biomolecular Systems," *J. Phys. Chem. B*, vol. 109, pp. 2469–73, Feb. 2005.

[18] S. Izvekov and G. A. Voth, "Multiscale coarse graining of liquid-state systems," *J. Chem. Phys.*, vol. 123, p. 134105, Oct. 2005.

[19] W. G. Noid, J.-W. Chu, G. S. Ayton, V. Krishna, S. Izvekov, G. A. Voth, A. Das, and H. C. Andersen, "The multiscale coarse-graining method. I. A rigorous bridge between atomistic and coarse-grained models.," *J. Chem. Phys.*, vol. 128, p. 244114, June 2008.

[20] W. G. Noid, P. Liu, Y. Wang, J.-W. Chu, G. S. Ayton, S. Izvekov, H. C. Andersen, G. A. Voth, V. Krishna, and A. Das, "The multiscale coarse-graining method. II. Numerical implementation for coarse-grained molecular models," *J. Chem. Phys.*, vol. 128, p. 244115, June 2008.

[21] A. Das and H. C. Andersen, "The multiscale coarse-graining method. III. A test of pairwise additivity of the coarse-grained potential and of new basis functions for the variational calculation.," *J. Chem. Phys.*, vol. 131, p. 034102, July 2009.

[22] V. Krishna, W. G. Noid, and G. A. Voth, "The multiscale coarse-graining method. IV. Transferring coarse-grained potentials between temperatures," *J. Chem. Phys.*, vol. 131, no. 2, p. 24103, 2009.

[23] Q. Shi, S. Izvekov, and G. A. Voth, "Mixed atomistic and coarse-grained molecular dynamics: simulation of a membrane-bound ion channel.," *J. Phys. Chem. B*, vol. 110, pp. 15045–8, Aug. 2006.

[24] J. Zhou, I. F. Thorpe, S. Izvekov, and G. A. Voth, "Coarse-grained peptide modeling using a systematic multiscale approach.," *Biophys. J.*, vol. 92, pp. 4289–303, June 2007.

[25] S. Izvekov, A. Violi, and G. A. Voth, "Systematic coarse-graining of nanoparticle interactions in molecular dynamics simulation.," *J. Phys. Chem. B*, vol. 109, pp. 17019–24, Sept. 2005.

[26] D. Reith, M. Pütz, and F. Müller-Plathe, "Deriving effective mesoscale potentials from atomistic simulations," *J. Comput. Chem.*, 2003.

[27] F. Müller-Plathe, "Coarse-graining in polymer simulation: From the atomistic to the mesoscopic scale and back," *ChemPhysChem*, vol. 3, pp. 754–769, 2002.

[28] P. Májek and R. Elber, "A coarse-grained potential for fold recognition and molecular dynamics simulations of proteins.," *Proteins*, vol. 76, pp. 822–36, Oct. 2009.

[29] G. Srinivas, X. Cheng, and J. C. Smith, "A Solvent-Free Coarse Grain Model for Crystalline and Amorphous Cellulose Fibrils," *J. Chem. Theory Comput.*, vol. 7, pp. 2539–2548, Aug. 2011.

[30] A. P. Lyubartsev and A. Laaksonen, "Calculation of effective interaction potentials from radial distribution functions: A reverse Monte Carlo approach," *Phys. Rev. E*, vol. 52, pp. 3730–3737, Oct. 1995.

[31] A. P. Lyubartsev and A. Laaksonen, "Osmotic and activity coefficients from effective potentials for hydrated ions," *Phys. Rev. E*, vol. 55, pp. 5689–5696, May 1997.

[32] A. P. Lyubartsev and A. Laaksonen, "Reconstruction of pair interaction potentials from radial distribution functions," *Comput. Phys. Commun.*, vol. 121-122, pp. 57–59, Sept. 1999.

[33] A. P. Lyubartsev and A. Laaksonen, "Effective potentials for ion-DNA interactions," *J. Chem. Phys.*, vol. 111, p. 11207, Dec. 1999.

[34] A. Lyubartsev and S. Marčelja, "Evaluation of effective ion-ion potentials in aqueous electrolytes," *Phys. Rev. E*, vol. 65, p. 041202, Mar. 2002.

[35] A. Lyubartsev, A. Mirzoev, L. Chen, and A. Laaksonen, "Systematic coarse-graining of molecular models by the Newton inversion method," *Faraday Discuss.*, vol. 144, p. 43, Oct. 2010.

[36] M. S. Shell, "The relative entropy is fundamental to multiscale and inverse thermodynamic problems," *J. Chem. Phys.*, vol. 129, no. 14, p. 144108, 2008.

[37] A. Chaimovich and M. S. Shell, "Coarse-graining errors and numerical optimization using a relative entropy framework," *J. Chem. Phys.*, vol. 134, no. 9, p. 94112, 2011.

[38] M. S. Shell, "Systematic coarse-graining of potential energy landscapes and dynamics in liquids.," *J. Chem. Phys.*, vol. 137, p. 084503, Aug. 2012.

[39] A. Pritchard-Bell and M. S. Shell, "Smoothing protein energy landscapes by integrating folding models with structure prediction.," *Biophys. J.*, vol. 101, pp. 2251–9, Nov. 2011.

[40] S. P. Carmichael and M. S. Shell, "A new multiscale algorithm and its application to coarse-grained peptide models for self-assembly.," *J. Phys. Chem. B*, vol. 116, pp. 8383–93, July 2012.

[41] M. Praprotnik, L. Delle Site, and K. Kremer, "Adaptive resolution molecular-dynamics simulation: changing the degrees of freedom on the fly.," *J. Chem. Phys.*, vol. 123, p. 224106, Dec. 2005.

[42] V. Ruhle, C. Junghans, A. Lukyanov, K. Kremer, and D. Andrienko, "Versatile object-oriented toolkit for coarse-graining applications," *J. Chem. Theory Comput.*, pp. 3211–3223, 2009.

[43] F. Reif, *Fundamentals of Statistical and Thermal Physics*. McGraw-Hill, 1985.

[44] H. B. Callen, *Thermodynamics and an Introduction to Thermostatistics*. Wiley, 2nd ed., 1985.

[45] R. E. Rudd and J. Q. Broughton, "Coarse-grained molecular dynamics: Nonlinear finite elements and finite temperature," *Phys. Rev. B*, vol. 72, p. 144104, Oct. 2005.

[46] R. E. Rudd and J. Q. Broughton, "Coarse-grained molecular dynamics and the atomic limit of finite elements," *Phys. Rev. B*, vol. 58, Sept. 1998.

[47] W. C. Swope, "A computer simulation method for the calculation of equilibrium constants for the formation of physical clusters of molecules: Application to small water clusters," *J. Chem. Phys.*, vol. 76, no. 1, p. 637, 1982.

[48] P. Hünenberger, "Thermostat algorithms for molecular dynamics simulations," *Adv. Comput. Simul.*, pp. 105–149, 2005.

[49] 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.*, vol. 81, no. 8, p. 3684, 1984.

[50] S. Nose, "A unified formulation of the constant temperature molecular dynamics methods," *J. Chem. Phys.*, vol. 81, p. 511, July 1984.

[51] W. Hoover, "Canonical dynamics: Equilibrium phase-space distributions," *Phys. Rev. A*, vol. 31, pp. 1695–1697, Mar. 1985.

[52] J. Jones, "On the determination of molecular fields. II. From the equation of state of a gas," *Proc. R. Soc. London Ser. A*, vol. 106, no. 738, pp. 463–477, 1924.

[53] T. Darden, L. Perera, L. Li, and L. Pedersen, "New tricks for modelers from the crystallography toolkit: the particle mesh Ewald algorithm and its use in nucleic acid simulations," *Structure*, vol. 7, pp. R55–R60, Mar. 1999.

[54] M. Daw and M. Baskes, "Embedded-atom method: Derivation and application to impurities, surfaces, and other defects in metals," *Phys. Rev. B*, vol. 29, pp. 6443–6453, June 1984.

[55] W. Jorgensen and J. Tirado-Rives, "The OPLS potential functions for proteins, energy minimizations for crystals of cyclic peptides and crambin," *J. Am. Chem. Soc.*, vol. 110, no. 6, 1988.

[56] J. Ryckaert and G. Ciccotti, "Numerical integration of the cartesian equations of motion of a system with constraints: molecular dynamics of n-alkanes," *J. Comput. Phys.*, vol. 341, 1977.

[57] A. Bailey, C. Lowe, and A. Sutton, "Efficient constraint dynamics using MILC SHAKE," *J. Comput. Phys.*, vol. 227, pp. 8949–8959, Oct. 2008.

[58] H. C. Andersen, "Rattle : a velocity version of the SHAKE algorithm for molecular dynamics calculations," *J. Comput. Phys.*, vol. 52, pp. 24–34, 1983.

[59] F. Ercolessi, "A molecular dynamics primer." `http://www.fisica.uniud.it/~ercolessi/md/md/md.html`, 8th February 2014.

[60] H. Kroto, J. Heath, S. O'Brien, R. Curl, and R. Smalley, "C60: Buckminsterfullerene," *Nature*, vol. 318, no. 14, pp. 162–163, 1985.

[61] Chemspider, "Buckminsterfullerene." `http://www.chemspider.com/Chemical-Structure.110185.html`, Last accessed: 1st March 2014.

[62] H. Berendsen, D. van der Spoel, and R. van Drunen, "GROMACS: A message-passing parallel molecular dynamics implementation," *Comput. Phys. Commun.*, vol. 91, pp. 43–56, Sept. 1995.

[63] E. Lindahl, B. Hess, and D. V. D. Spoel, "GROMACS 3.0: a package for molecular simulation and trajectory analysis," *J. Mol. Model.*, pp. 306–317, 2001.

[64] D. Van Der Spoel, E. Lindahl, B. Hess, G. Groenhof, A. E. Mark, and H. J. C. Berendsen, "GROMACS: fast, flexible, and free.," *J. Comput. Chem.*, vol. 26, pp. 1701–18, Dec. 2005.

[65] B. Hess, C. Kutzner, D. van der Spoel, and E. Lindahl, "GROMACS 4: Algorithms for Highly Efficient, Load-Balanced, and Scalable Molecular Simulation," *J. Chem. Theory Comput.*, vol. 4, pp. 435–447, Mar. 2008.

[66] S. Pronk, S. Páll, R. Schulz, P. Larsson, P. Bjelkmar, R. Apostolov, M. R. Shirts, J. C. Smith, P. M. Kasson, D. van der Spoel, B. Hess, and E. Lindahl, "GROMACS 4.5: a high-throughput and highly parallel open source molecular simulation toolkit.," *Bioinformatics*, vol. 29, pp. 845–54, Apr. 2013.

[67] G. Ciccotti and W. G. Hoover, *Molecular dynamics simulation of statistical-mechanical systems.* North-Holland, 1986.

[68] P. Liu and D. K. Agrafiotis, "Fast determination of the optimal rotational matrix for macromolecular superpositions," *J. Comput. Phys.*, pp. 2–4, 2010.

[69] L. A. Girifalco, "Molecular properties of C60 in the gas and solid phases," *J. Phys. Chem.*, vol. 96, no. 2, pp. 858–861, 1991.

[70] E. Jones, T. Oliphant, P. Peterson, *et al.*, "SciPy: Open source scientific tools for Python." `http://www.scipy.org`.

[71] A. Villa, C. Peter, and N. van der Vegt, "Transferability of nonbonded interaction potentials for coarse-grained simulations: Benzene in water," *J. Chem. Theory Comput.*, pp. 2434–2444, 2010.

[72] M. Winger, D. Trzesniak, R. Baron, and W. F. van Gunsteren, "On using a too large integration time step in molecular dynamics simulations of coarse-grained molecular models.," *Phys. Chem. Chem. Phys.*, vol. 11, pp. 1934–41, Mar. 2009.

Appendix A

Derivation of the Girifalco Potential

Let us assume each sphere has a radius a and mass m spread evenly over its surface. The mass per unit area, σ, is $m/(4\pi a^2)$. The energy of interaction between two point masses m_1 and m_2 is $m_1 m_2/r^n$, where $n \geq 1$. We start by placing one sphere, which we call "sphere 1", at the origin as shown in Fig. A.1. Our initial aim is to calculate the potential at the point P a distance r ($> a$) from the centre of the sphere. For convenience, we choose to place P on the z-axis. All points on the sphere in the ring between polar angles θ and $\theta + d\theta$ are a distance

$$\begin{aligned} d(\theta) &= \sqrt{(r - a\cos\theta)^2 + (a\sin\theta)^2} \\ &= \sqrt{r^2 + a^2 - 2ar\cos\theta} \end{aligned}$$

from the point P. The area of the ring is

$$\text{circumference} \times \text{width} = 2\pi a \sin\theta \times a d\theta.$$

Hence, the ring contributes

$$dV = \frac{2\pi\sigma a^2 \sin\theta d\theta}{[d(\theta)]^n}$$

to the potential at P. The total potential at P, a distance r from the centre of

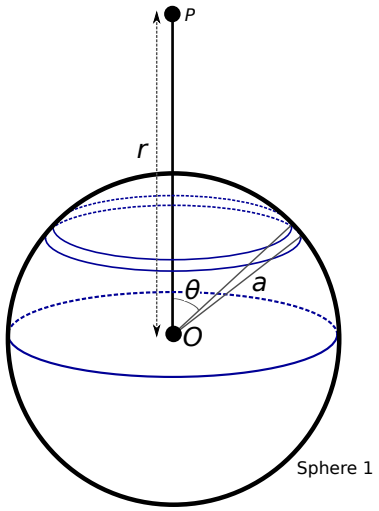


Figure A.1: Sphere 1 at the origin. We will calculate the potential at the point P on the z-axis a distance r ($> a$) from the centre of the sphere.

sphere 1, is

$$\begin{aligned} V(r) &= \int_0^\pi \frac{2\pi a^2 \sigma \sin\theta d\theta}{(r^2+a^2-2ar\cos\theta)^{n/2}} \\ &= \int_{-1}^1 \frac{2\pi a^2 \sigma du}{(r^2+a^2+2aru)^{n/2}} \qquad \text{(where } u = -\cos\theta\text{)} \\ &= \left[\frac{2\pi a^2\sigma}{(1-n/2)2ar(r^2+a^2+2aru)^{(n/2-1)}}\right]_{-1}^1 \\ &= \frac{2\pi a\sigma}{(n-2)r}\left\{\frac{1}{(r-a)^{n-2}} - \frac{1}{(r+a)^{n-2}}\right\}. \end{aligned}$$

Now that we have the potential due to one sphere, we can work out its interaction with the other sphere, which we call "sphere 2". As shown in Fig. A.2, we place the centre of sphere 2 at the origin and the centre of sphere 1 a distance R ($> 2a$) along the z axis. All points in the ring of area $2\pi a^2 \sin\theta d\theta$ on sphere 2 are the same distance,

$$\begin{aligned} r(\theta) &= \sqrt{(R-a\cos\theta)^2 + (a\sin\theta)^2} \\ &= \sqrt{R^2+a^2-2aR\cos\theta}, \end{aligned}$$

from the centre of sphere 1 and therefore feel the same potential. The potential

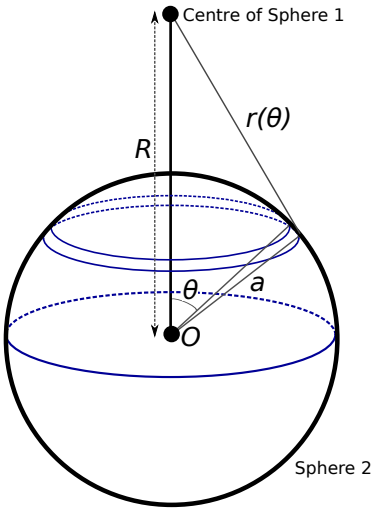


Figure A.2: Sphere 2 is now at the origin, where it feels the potential of sphere 1. The centre of sphere 1 is a distance R ($> 2a$) along the $+z$ axis.

energy of interaction of the ring with sphere 1 is thus

$$\begin{aligned} dU &= V(r(\theta)) \times 2\pi a^2 \sigma \sin\theta d\theta \\ &= \frac{4\pi^2 a^3 \sigma^2}{(n-2)r(\theta)} \left\{ \frac{1}{(r(\theta)-a)^{n-2}} - \frac{1}{(r(\theta)+a)^{n-2}} \right\} \sin\theta d\theta. \end{aligned}$$

Noting that $r^2 = R^2 + a^2 - 2aR\cos\theta$ and hence that

$$2rdr = 2aR\sin\theta d\theta,$$

we can rewrite this as

$$dU = \frac{4\pi^2 a^2 \sigma^2}{(n-2)R} \left\{ \frac{1}{(r-a)^{n-2}} - \frac{1}{(r+a)^{n-2}} \right\}$$

and hence obtain

$$\begin{aligned} U(R) &= \text{potential energy of interaction at spacing } R \\ &= \int_{R-a}^{R+a} \frac{4\pi^2 a^2 \sigma^2}{(n-2)R} \left\{ \frac{1}{(r-a)^{n-2}} - \frac{1}{(r+a)^{n-2}} \right\} dr. \end{aligned}$$

This gives

$$U(R) = \frac{4\pi^2 a^2 \sigma^2}{(n-2)(n-3)R} \left\{ \frac{1}{(R+2a)^{n-3}} + \frac{1}{(R-2a)^{n-3}} - \frac{2}{R^{n-3}} \right\}.$$

Taking $n = 6$ for the first term in the Lennard–Jones potential, this becomes

$$\begin{aligned} U(R) &= \frac{(2\pi a\sigma)^2}{12R}\left[\frac{1}{(R+2a)^3} + \frac{1}{(R-2a)^3} - \frac{2}{R^3}\right] \\ &= \frac{(\pi\sigma)^2}{12(2a)^2}\left[\frac{1}{s(s+1)^3} + \frac{1}{s(s-1)^3} - \frac{2}{s^4}\right], \end{aligned} \tag{A.1}$$

where $s = R/2a$. In this case, $\sigma = N/4\pi a^2$ where $N = 60$ is the number of atoms in the buckyball, so we find the prefactor

$$\alpha = \frac{(\pi\sigma)^2 A}{12(2a)^2} = \frac{N^2 A}{12(2a)^6}. \tag{A.2}$$

Applying a similar procedure for the $n = 12$ term, we find the total potential

$$V(r) = -\alpha\left[\frac{1}{s(s-1)^3} + \frac{1}{s(s+1)^3} - \frac{2}{s^4}\right] \tag{A.3}$$

$$+\beta\left[\frac{1}{s(s-1)^9} + \frac{1}{s(s+1)^9} - \frac{2}{s^{10}}\right], \tag{A.4}$$

where $s = r/2a$, $2a = 0.71$ nm is the C_{60} diameter, $\alpha = N^2A/12(2a)^6$ where $N = 60$, and $\beta = N^2B/90(2a)^{12}$.

Appendix B

The Conjugate Gradients Algorithm

In order to explain this method, it will be useful to switch to more formal notation. Let us consider a vector $\mathbf{x} \in \mathbb{R}^{3n}$. Let $f : \mathbb{R}^{3n} \to \mathbb{R}$ be a scalar function of the form

$$f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\intercal \mathbf{A}\mathbf{x} - \mathbf{x}^\intercal \mathbf{b}, \tag{B.1}$$

where $\mathbf{A} \in \mathbb{R}^{3n \times 3n}$ is a real matrix and $\mathbf{b} \in \mathbb{R}^{3n}$. We want to find a minimum $\mathbf{x} = \mathbf{x}^*$ of this function, that is where $\mathbf{g}(\mathbf{x}^*) = \nabla f(\mathbf{x}^*) = 0$. This equation is a quadratic form, and the results below strictly hold only for such forms. However, in the vicinity of a local minimum, we may expand many functions using a multi–dimensional Taylor series into a quadratic form, so the method can be useful for more general functions. The gradient of Eqn. (B.1) is given by

$$\mathbf{g} = \mathbf{A}\mathbf{x} - \mathbf{b}. \tag{B.2}$$

In this notation, the steepest descent algorithm above can be expressed as follows. The gradient $\mathbf{g}$ always points in the direction of steepest increase of f, so we choose our search direction to be $-\mathbf{g}$, the direction of steepest decrease. We move along this vector until a minimum of f is found in this direction. We then recalculate the gradient, and repeat until a minimum is found. Let $\mathbf{x}_{k-1}$ be the position at the start of iteration k. The gradient for this iteration is then

$\mathbf{g}_k = \mathbf{A}\mathbf{x}_{k-1} - \mathbf{b}$. The search direction is $\mathbf{p}_k = -\mathbf{g}_k$ and we want to find a scalar α_k such that

$$\mathbf{x}_k = \mathbf{x}_{k-1} + \alpha_k \mathbf{p}_k \tag{B.3}$$

is a minimum of $f(\mathbf{x})$ along the search direction. The new gradient $\mathbf{g}_{k+1}$ must be orthogonal to the previous search direction $\mathbf{p}_k$, so that

$$\begin{aligned} 0 = \mathbf{p}_k^\intercal \mathbf{g}_{k+1} &= \mathbf{p}_k^\intercal (\mathbf{A}\mathbf{x}_k - \mathbf{b}) \\ &= \mathbf{p}_k^\intercal [\mathbf{A}(\mathbf{x}_{k-1} + \alpha_k \mathbf{p}_k) - \mathbf{b}] \\ &= \mathbf{p}_k^\intercal (\mathbf{A}\mathbf{x}_{k-1} - \mathbf{b}) + \alpha_k \mathbf{p}_k^\intercal \mathbf{A}\mathbf{p}_k. \end{aligned} \tag{B.4}$$

This implies that

$$\alpha_k = -\frac{\mathbf{p}_k^\intercal (\mathbf{A}\mathbf{x}_{k-1} - \mathbf{b})}{\mathbf{p}_k^\intercal \mathbf{A}\mathbf{p}_k} = -\frac{\mathbf{p}_k^\intercal \mathbf{g}_k}{\mathbf{p}_k^\intercal \mathbf{A}\mathbf{p}_k}. \tag{B.5}$$

For the steepest descent choice of search direction, this becomes

$$\alpha_k = \frac{\mathbf{g}_k^\intercal \mathbf{g}_k}{\mathbf{g}_k^\intercal \mathbf{A}\mathbf{g}_k}. \tag{B.6}$$

Now we may consider the more complicated case of the conjugate gradients method. The difference between gradients at two different iterations is given by

$$\mathbf{g}_k - \mathbf{g}_j = (\mathbf{A}\mathbf{x}_{k-1} - b) - (\mathbf{A}\mathbf{x}_{j-1} - b) = \mathbf{A}(\mathbf{x}_{k-1} - x_{j-1}). \tag{B.7}$$

For consecutive iterations, $j = k - 1$ and

$$\mathbf{g}_k = \mathbf{g}_{k-1} + \mathbf{A}(\mathbf{x}_{k-1} - \mathbf{x}_{k-2}) = \mathbf{g}_{k-1} + \alpha_{k-1}\mathbf{A}\mathbf{p}_{k-1}, \tag{B.8}$$

where we have used Eqn. (B.3). The set of search directions $\{\mathbf{p}_k\}$ are said to be conjugate directions with respect to $\mathbf{A}$ if

$$\mathbf{p}_k^\intercal \mathbf{A}\mathbf{p}_j = 0 \;\forall\; j \neq k. \tag{B.9}$$

We can prove these conjugate directions are linearly independent by contradiction. First, assume there is a set of scalars $\{\mu_i\}$ which are not all zero, for which

$$\sum_{i=1}^{k} \mu_i \mathbf{p}_i = 0. \tag{B.10}$$

Multiply both sides by $\mathbf{p}_j^\intercal \mathbf{A}$, and the conjugacy condition implies that

$$\mu_j \mathbf{p}_j^\intercal \mathbf{A} \mathbf{p}_j = 0 \; \forall \; j. \tag{B.11}$$

$\mathbf{A}$ is positive definite, so the only solution is that μ_j is zero for all j. This contradicts the original assumption and so the conjugate directions are linearly independent.

We now need to construct a set of conjugate directions from a general set of directions $\{\mathbf{u}_k\}$. This can be done using the formulae

$$\begin{aligned} \mathbf{p}_1 &= \mathbf{u_1}, \\ \mathbf{p}_k &= \mathbf{u}_k + \sum_{i=1}^{k-1} \beta_i^{(k)} \mathbf{p}_i \text{ for } k > 1, \end{aligned} \tag{B.12}$$

where

$$\beta_i^{(k)} = -\frac{\mathbf{u}_k^\intercal \mathbf{A} \mathbf{p}_i}{\mathbf{p}_i^\intercal \mathbf{A} \mathbf{p}_i} \text{ for } i < k. \tag{B.13}$$

We can prove this by induction. Assume that k conjugate directions have already been found. We must show that the next direction $\mathbf{p}_{k+1}$ constructed according to the above formula is conjugate to all k previous directions.

$$\begin{aligned} \mathbf{p}_j^\intercal \mathbf{A} \mathbf{p}_{k+1} &= \mathbf{p}_j^\intercal \mathbf{A} \mathbf{u}_{k+1} + \sum_{i=1}^{k} \beta_i^{(k+1)} \mathbf{p}_j^\intercal \mathbf{A} \mathbf{p}_i \\ &= \mathbf{p}_j^\intercal \mathbf{A} \mathbf{u}_{k+1} - \sum_{i=1}^{k} \frac{\mathbf{u}_{k+1}^\intercal \mathbf{A} \mathbf{p}_i}{\mathbf{p}_i^\intercal \mathbf{A} \mathbf{p}_i} \mathbf{p}_j^\intercal \mathbf{A} \mathbf{p}_i \\ &= \mathbf{p}_j^\intercal \mathbf{A} \mathbf{u}_{k+1} - \frac{\mathbf{u}_{k+1}^\intercal \mathbf{A} \mathbf{p}_j}{\mathbf{p}_j^\intercal \mathbf{A} \mathbf{p}_j} \mathbf{p}_j^\intercal \mathbf{A} \mathbf{p}_j, \\ &= \mathbf{p}_j^\intercal \mathbf{A} \mathbf{u}_{k+1} - \mathbf{u}_{k+1}^\intercal \mathbf{A} \mathbf{p}_j \\ &= 0, \end{aligned} \tag{B.14}$$

since all the other terms in the sum vanish due to the definition of conjugacy, and $\mathbf{A}$ is symmetric. This implies $\mathbf{p}_{k+1}$ is indeed conjugate to the previous k directions. $\mathbf{p}_1$ and $\mathbf{p}_2$ are trivially verified to be conjugate directions, which completes the proof by induction.

Since the conjugate directions are linearly independent, any other vector may be written as a linear combination of them. In particular, the vector from the initial point $\mathbf{x}_0$ to the minimum $\mathbf{x}^*$ can be written as

$$\mathbf{x}^* - \mathbf{x}_0 = \sum_{i=1}^{3n} \tilde{\alpha}_i \mathbf{p}_i. \tag{B.15}$$

We can use the conjugacy condition to determine the set of coefficients $\{\tilde{\alpha}_k\}$,

$$\begin{aligned}\tilde{\alpha}_k &= \frac{\mathbf{p}_k^\intercal \mathbf{A}(\mathbf{x}^* - \mathbf{x}_0)}{\mathbf{p}_k^\intercal \mathbf{A}\mathbf{p}_k} \\ &= -\frac{\mathbf{p}_k^\intercal \mathbf{g}_1}{\mathbf{p}_k^\intercal \mathbf{A}\mathbf{p}_k},\end{aligned} \tag{B.16}$$

since $\mathbf{A}\mathbf{x}_0 = \mathbf{g}_1$ and $\mathbf{A}\mathbf{x}^* = 0$. It is therefore possible to reach $\mathbf{x}^*$ from $\mathbf{x}_0$ in $m \leq 3n$ steps, where the kth step is given by

$$\mathbf{x}_k = \mathbf{x}_{k-1} + \tilde{\alpha}_k \mathbf{p}_k. \tag{B.17}$$

Going one step back and multiplying both sides by $\mathbf{A}$, we find

$$\mathbf{A}\mathbf{x}_{k-1} = \mathbf{A}\mathbf{x}_0 + \sum_{i=1}^{k-1} \tilde{\alpha}_i \mathbf{A}\mathbf{p}_i, \tag{B.18}$$

or

$$\mathbf{g}_k = \mathbf{g}_1 + \sum_{i=1}^{k-1} \tilde{\alpha}_i \mathbf{A}\mathbf{p}_i. \tag{B.19}$$

Taking the scalar product with $\mathbf{p}_k$ and using the conjugacy condition implies that $\mathbf{p}_k^\intercal \mathbf{g}_1 = \mathbf{p}_k^\intercal \mathbf{g}_k$. Inserting this into Eqn. (B.16) and comparing with Eqn. (B.6) shows that $\tilde{\alpha}_k = \alpha_k$. This implies that the steps from $\mathbf{x}_0$ to $\mathbf{x}^*$ along the conjugate directions proceed via points which are minima along each search direction.

Taking the scalar product with $\mathbf{p}_j$ instead gives

$$\mathbf{p}_j^\intercal \mathbf{g}_k = \mathbf{p}_j^\intercal \mathbf{g}_1 + \alpha_j \mathbf{p}_j^\intercal \mathbf{A}\mathbf{p}_j = 0 \tag{B.20}$$

by inserting Eqn. (B.16). This shows that $\mathbf{g}_k$ is orthogonal to all previous search directions.

In the method of conjugate gradients, we choose our general directions $\mathbf{u}_k$ to be equal to the steepest descent directions $\mathbf{g}_k$. Note that these are not exactly the same directions as in the steepest descent algorithm, as a different set of points will be visited. In this case, Eqn. (B.13) becomes

$$\beta_i^{(k)} = \frac{\mathbf{g}_k^\intercal \mathbf{A}\mathbf{p}_i}{\mathbf{p}_i^\intercal \mathbf{A}\mathbf{p}_i} \text{ for } i < k. \tag{B.21}$$

This can be rewritten using Eqn. (B.8) as

$$\mathbf{p}_i^\intercal \mathbf{A}\mathbf{p}_i \beta_i^{(k)} = \frac{1}{\alpha_i} \mathbf{g}_k^\intercal (\mathbf{g}_{i+1} - \mathbf{g}_i). \tag{B.22}$$

By orthogonality of the gradients, $\beta_i^{(k)} = 0$ for $i < k-1$ and the only non-vanishing coefficient is $\beta_{k-1}^{(k)} = \beta_k$,

$$\beta_k = -\frac{\mathbf{g}_k^\intercal \mathbf{g}_k}{\mathbf{p}_{k-1}^\intercal \mathbf{g}_{k-1}} = \frac{\mathbf{g}_k^\intercal \mathbf{g}_k}{\mathbf{g}_{k-1}^\intercal \mathbf{g}_{k-1}} = \frac{|\mathbf{g}_k|^2}{|\mathbf{g}_{k-1}|^2}. \tag{B.23}$$

This is know as the Fletcher–Reeves formula. The method starts with a steepest descent step, and the next search direction is a linear combination of the current gradient and the previous search direction,

$$\mathbf{p}_k = -\mathbf{g}_k + \beta_k \mathbf{p}_{k-1}. \tag{B.24}$$

Appendix C

Copyright Permissions

AIP PUBLISHING LLC LICENSE
TERMS AND CONDITIONS

Jun 19, 2014

All payments must be made in full to CCC. For payment instructions, please see information listed at the bottom of this form.

License Number	3384270822599
Order Date	May 08, 2014
Publisher	AIP Publishing LLC
Publication	Journal of Chemical Physics
Article Title	Adaptive resolution molecular-dynamics simulation: Changing the degrees of freedom on the fly
Author	Matej Praprotnik,Luigi Delle Site,Kurt Kremer
Online Publication Date	Dec 14, 2005
Volume number	123
Issue number	22
Type of Use	Thesis/Dissertation
Requestor type	Student
Format	Print and electronic
Portion	Photograph/Image
Title of your thesis / dissertation	Coarse-Grained Molecular Dynamics
Expected completion date	Jun 2014
Estimated size (number of pages)	180
Total	0.00 GBP