Elements of Molecular Dynamics

W. Smith

Elements of Molecular Dynamics

W. Smith

Formerly
Computational Scientist at the Computational
Science and Engineering Department,
Daresbury Laboratory, Warrington, U.K.
and
Visiting Professor at the Department of Materials
Science and Engineering, University of
Sheffield, U.K.
and
Visiting Professor at The Institute of
Pharmaceutical Innovation,
University of Bradford, U.K.

First Edition January 2014

Version 1.1

To
K. Singer and E.T. Lloyd
- A small return on time
invested!

Acknowledgements

I have learned much from CCP5 colleagues down the years and have passed a lot of it on here. It is almost impossible after thirty years or so to recall everyone who gave me advice, so I would first like to thank the CCP5 community in its entirety for its openness and generosity. That said, there are a few people whose instruction I do recall and I am grateful to the following who unknowingly have contributed to this book!

Tim Forester, Ilian Todorov, Maurice Leslie, Ian Bush, Laurence Ellison, Florian Mueller-Plathe, Konrad Singer, David Heyes, David Fincham, Les Woodcock, David Adams, David Brown, Mike Gillan, Mike Allen, Dominic Tildesley, Mark Pinches, Dennis Rapaport, Andrew Raine, Maurice Dixon, Mark Rodger, David Quigley, Mark Tuckerman, Ross Brown, Pierre-Andre Cazade, Patrice Bordat, John Harding, Duncan Harris, Paul Smith, Henry Boateng, Martyn Guest, Robert Harrison.

This document was prepared using OpenOffice Writer and Zotero Bibliography software.

Version History

Version 1.0: Released January 2014.

Version 1.1: Released June 2015. Modifications made to Chapter 10, sections 10.3.1, 10.5.3 and 10.5.4, with the aim of improving and clarifying the text.

Table of Contents

Chapter 1

Molecular Dynamics Basics

1.1 Introduction

At heart molecular dynamics is an attempt to solve the classical equations of motion for a system composed of atoms and molecules with the aim of obtaining the *time evolution* of the system. The method is most often applied to *condensed phase* systems i.e. systems in the solid or liquid state, where the objective is to learn something about how the bulk properties of the system arise from the molecular basis. The time evolution aspect of molecular dynamics distinguishes it from the Monte Carlo method, in which the molecular system evolves through a stochastic or random-walk process rather than a true dynamical process. This gives molecular dynamics a handle on *time dependent* properties, which means it is particularly useful for exploring *transport properties*, such as diffusion, thermal conductivity and viscosity, or aspects of spectroscopy, such as solvation induced spectral shifts, or the kinetics of chemical processes – all areas in which the dependence on time is the key factor.

That molecular dynamics describes the time evolution of molecular systems using classical mechanics is perhaps unexpected. After all, the atomic scale is where quantum mechanical effects are supposedly important. In most applications however, this is not an issue. The matter is determined by the thermal de Broglie wavelength of the constituent atoms at the temperature of interest:

$$\Lambda = \sqrt{\frac{\beta h^2}{2\pi m}} \tag{1.1}$$

where $\hbar$ is *Planck's constant*, m is the atomic mass and $\beta = 1/k_B T$, where T is temperature and k_B is *Boltzmann's constant.* The wavelength is very often small enough for the atoms to be regarded as essentially classical [1] – meaning that no significant diffraction effects or quantum tunnelling is expected.

Nevertheless, it is prudent, when dealing with the lighter elements – in particular hydrogen at low absolute temperatures - to check that quantum effects can be safely neglected. There are time dependent methods for dealing with systems that manifest quantum behaviour, but they are often difficult to apply and hard to interpret. This explains why conventional, classical molecular dynamics is much preferred. But if time dependence is unimportant and the focus is on thermodynamical and structural properties, the Path Integral Monte Carlo (PIMC) method [2] is arguably the best many-particle quantum method available for condensed phase systems.

Molecular dynamics is, above all else, a *computational* method and its application represents a *simulation* of the system of interest. It is necessarily computational

because an analytical solution for the dynamics of many interacting particles is mathematically impossible. Thus the solution offered by molecular dynamics is inevitably *numerical* in nature. The equations of motion are cast into a *discrete* form, such that a knowledge of the atomic positions, velocities and forces at a time, t, allows the determination (or at least the estimation) of the positions and velocities at a short time, $t+\Delta t$, later. From these new positions, new atomic forces are calculated and then a further step in time may be made. This stepping process is generally referred to as a *time step* and a typical simulation is between several thousand and several million time steps in duration. Given also that the number of atoms in a simulation can range from a few hundred to a few million, the huge computational demand of this procedure is all too clear. The time step is normally of the size $1 \sim 10$ *femtoseconds* (fs), where one femtosecond is 10^{-15} seconds, an exceedingly small number that reflects the time scale on which atomistic dynamics takes place.

But what is the point of such a simulation? It is apparent that most simulations will generate a bewildering amount of data – as a minimum the positions in 3D space of all the atoms in the system at many time steps in the simulation, and often the atomic velocities and forces and associated thermodynamic data (e.g. energy, pressure *etc.*) as well. Such data are unlikely to be directly useful, except perhaps where they are streamed to rendering devices to make movies of the molecular motion – a procedure that does not itself guarantee the identification of anything scientifically significant. Rather, the best practical option for dealing with such a vast volume of data is to process it *statistically* so that averaged properties of the system can be obtained. Thus the averaged system energy is equivalent to the internal energy U defined in thermodynamics and the averaged kinetic energy is equivalent to the temperature T in Kelvin, and so on. The fluctuations of these properties, as they instantaneously depart from their average values during the simulation, can also yield significant system properties. Similarly, important time dependent properties may be obtained through the calculation of appropriate *time correlation functions*. This all belongs to the subject of *statistical mechanics*, and it is the methodology of this subject that provides the tools for calculating most of what molecular dynamics (and indeed Monte Carlo) simulations can usefully produce. By this approach a vast number of meaningful physical properties can be determined.

Technically speaking, the averaging procedures used in molecular dynamics are examples of *Boltzmann* averaging, which means the results are obtained by time averaging the data from a single simulation over a long period of time. This is distinct from *Gibbs* averaging in which the results are an average over many distinct but equivalent replicas of the same system at some instant of time. In principle this shouldn't make much difference, but in practice it means that we need to be sure that the time dependence inherent in molecular dynamics is not having an obscure effect on our averages. At the simplest level, it is not meaningful to calculate an average of a property that is changing systematically with time, but it *is* meaningful if the property merely shows fluctuations in value without a systematic drift. Another key issue here is *correlation*, which implies that sequential values of some property depend in some way on the preceding values. These correlations are not necessarily harmful and are indeed expected in real systems, rather it is that their presence in the data can invalidate the basis of the statistical mechanical treatment. The removal, or at least detection, of correlation in the data is sometimes crucial to validating the

results of a simulation.

The potential of molecular dynamics as a useful scientific probe is readily apparent and therein lies a reason to learn about the subject. However there are some misconceptions that need to be addressed at the outset. It should be said that the method is not without problems or limitations. It is essential to become familiar with these lest its exploitation become a colossal waste of time and resources. It does not stand apart from mainstream science as a purely theoretical tool obviating the need for experimental study or verification. It is often reliant on empirical data, most particularly in the description of inter-atomic forces. It is sometimes reliant on assumptions that are dangerous to accept without further proof, such as the occurrence of equilibrium and energetic equipartition, or even the validity of the numerically approximate trajectories it generates. On the other hand there is a great deal of experience abroad to confirm molecular dynamics as a useful and productive tool in the right hands and the theoretical understanding underpinning its implicit assumptions is advancing all the time. It can be argued that the basis of molecular dynamics is as valid as that of any laboratory experiment. For these reasons it is much better to think of molecular dynamics as a potent *experimental* tool, often providing insight and data that cannot be obtained in any other way.

This book is primarily about the methodology of molecular dynamics and how it is applied, with discussion of the problems and pitfalls. It is not about any particular area of application, but it is intended to be widely useful. The book is not intended to win the reader over to the "molecular dynamics camp" in preference to any other approach, but to show how it can be used appropriately and to best effect.

1.2 A Molecular Dynamics Simulation

The essence of what molecular dynamics is like in application is best illustrated by an example. We consider one of the simplest cases and also one of its earliest practical applications – that of a noble gas in the liquid state [3], [4]. The essential features of a simulation will be introduced and will provide the basis for a more technical description later in the book.

1.2.1 The Model System

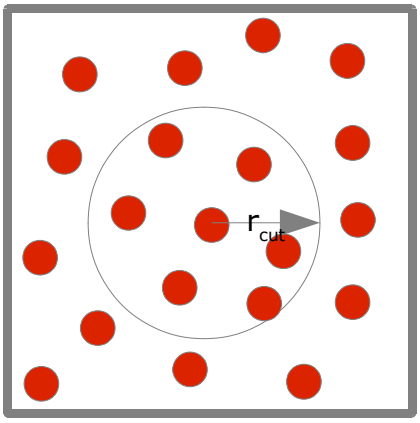

Figure 1.1: A simple molecular dynamics simulation.

Figure 1.1 gives some idea of what the system looks like. We have a number of noble gas atoms, represented by red spheres (100~1000 atoms is sufficient for most systems of this kind), inside a `box' of some sort, which is called the *simulation cell*. The simulation cell is almost always three dimensional, but showing it in two dimensions makes it easier to see what's going on. The walls of the cell are not solid; they are virtual walls, without substance, which define a unit of repetition in three dimensional space called a *periodic cell,* which is the basis of a *periodic boundary condition* or PBC [1].

1.2.2 The Periodic Boundary Condition

The idea behind a PBC is that the periodic cell is assumed to be replicated in the same way as a unit cell in crystallography, with the effect that the whole of space is filled. This is shown in figure 1.2, where the atomic velocities are indicated by the arrows attached to each atom. This is a two dimensional analogue of the three dimensional case. The system is in fact infinite in extent, so this is just a subsection of the whole. Each cell shown is an exact replica of all the others, so the contents and dynamics of all cells are the same. We therefore need only pay attention to one of these cells, such as the shaded central cell, and we need only describe the dynamics of the atoms in this cell to describe the motion of *all* the atoms in the infinite system. This follows naturally from the fact that all images of a given particle have the same arrangement of surrounding atoms, which gives rise to the same forces, and the same dynamics. Should an atom leave the cell due to the dynamics, an image of the atom, with identical dynamical properties, enters the cell on the opposite side and the contents of the cell remain unchanged. Thus, by exploiting the periodicity, all the information required to propagate the dynamics of the cell contents can be calculated from the current atomic configuration of the cell.

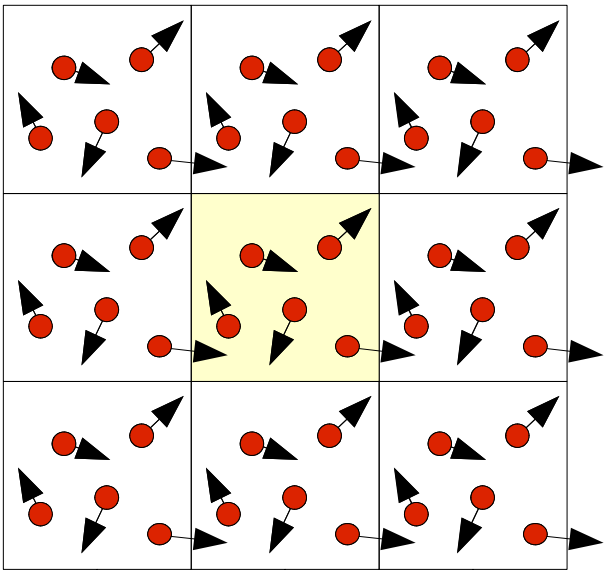

Figure 1.2: The Periodic Boundary Condition

The purpose of the periodic boundary is to create a system without surfaces. Since the

objective is to study the properties of the bulk system, not one unduly affected by surface forces, which are known to be strong. Of course, in a real system, this precise periodicity is absent (even in a periodic crystal the instantaneous positions and velocities of the periodic image atoms are not the same) and this sets some spatial limitations on the properties that can be calculated. It is worth noting that there are many possible forms of periodic boundary, besides the simple cubic PBC used in this example. Simulations without periodic boundaries are also sometimes used.

1.2.3 The Inter-atomic Potential

The atoms in our system are of course interacting with each other and this is expressed in terms of an *empirical pair potential*, a typical form for which is the Lennard-Jones potential [1]:

$$\Phi(r_{ij}) = 4\varepsilon\left(\left(\frac{\sigma}{r_{ij}}\right)^{12} - \left(\frac{\sigma}{r_{ij}}\right)^{6}\right) \tag{1.2}$$

This potential is a function of the distance r_{ij} between two atoms, which is the modulus (i.e. length) of the vector $\vec{r}_{ij}$ defined by the relation:

$$r_{ij} = |\vec{r}_{ij}| = |\vec{r}_j - \vec{r}_i|. \tag{1.3}$$

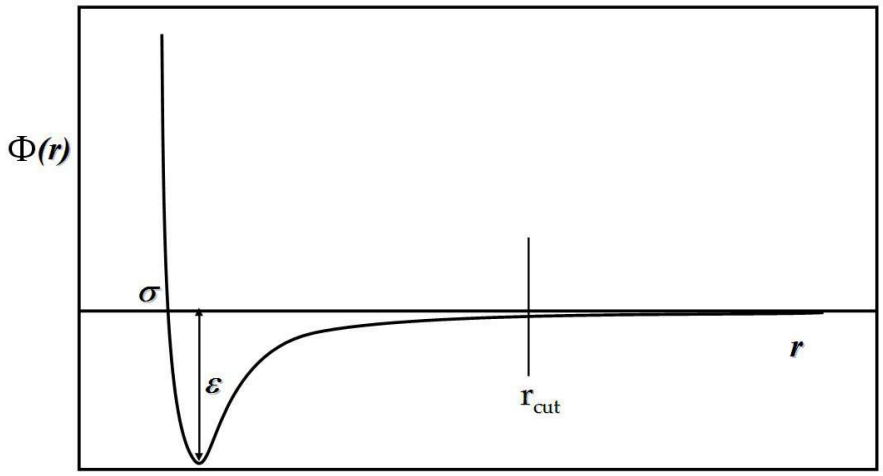

Figure 1.3: The Lennard-Jones Pair Potential

The parameter σ defines the *length scale* of the potential and corresponds to the effective diameter of the atoms. The parameter ε defines the *energy scale* which corresponds to the interaction strength of the potential. The potential is empirical because it has an assumed form and the parameters σ and ε are fitted to experimental data. Typically the potential, which is plotted in figure 1.3, has both repulsive and attractive parts. The r^{-12} term defines the short ranged repulsive component and the r^{-6} term defines the longer ranged attractive component. The first represents the repulsion experienced by the atoms when their electronic shells overlap and the second represents the so called *London dispersion interaction*. Both

are quantum mechanical in origin and are approximated by the given mathematical forms. Between these two extremes lies a point of equilibrium – the minimum shown in figure 1.3. Most empirical pair potentials have a behaviour similar to this, though the mathematical form may be different.

A key point to stress here is that the interaction is a *pair potential*. Every atom is deemed able to interact with every other through such pairwise expressions. The possibility of a collective or many-body interaction, is set aside as an unnecessary complication. This assumption proves to be a valid one for this system, but for some systems, for example metals, this is not a safe assumption. The adoption of pair force potentials means that the maximum number of interactions we must evaluate in a system composed of N atoms is $N(N-1)/2.$ Therefore the *order of complexity* of the simulation is said to be $N^2,$ which is the highest polynomial dependence of the simulation on the number of atoms and is a good indicator of what the computational cost of the simulation will be.

1.2.4 The Potential Cut-off

In principle pair potentials have infinite range – no matter how large the separation between the atom pair the potential has a non-zero value. However, it is found that for potentials like the Lennard-Jones potential, the value becomes negligible once the separation gets beyond a few multiples of the distance $\sigma,$ so it is usual to apply a cut-off condition: for any pair of atoms separated by a specified distance r_{cut} the interaction is assumed to be zero and nothing is calculated. In figure 1.1 the cut-off for an atom is shown by the circle (representing a sphere in three dimensions) and is marked by a vertical line in figure 1.3.

Applying this cut-off has, hopefully, small effects on the subsequent simulation, but there is clearly a judgement to be made about where it can be safely drawn. It will be appreciated from figure 1.3 that using too short a cut-off will create a discontinuity in the energy calculation, which means that an artificial energy fluctuation will arise in the simulation. More subtly, there is also a discontinuity in the calculated pair force, which can give rise to artefacts in the calculated system properties. So ideally, the cut-off needs to be large enough to mitigate these effects. Using a cut-off also means that the calculated system energy is missing contributions from the longer ranged interactions, as is the calculated system pressure. However corrections are easy to obtain based on the overall system density and the size of the cut-off. Provided the density of the system is constant, the corrections are also constants, and can be added in later.

1.2.5 The Minimum Image Convention

When using a cut-off in conjunction with a periodic boundary condition, it is usual to set the cut-off so that the net interaction experienced by each atom includes contributions from only one image of the other atoms in the system. This is known as the *minimum image convention*: only the closest images of any two atoms contribute to the calculated energy (and force). This is primarily an operational choice, since using more than one image makes handling the periodic boundary condition more complicated and therefore more expensive computationally. The minimum image convention is shown in figure 1.4, where it is image atom j' rather than j that

interacts with atom i.

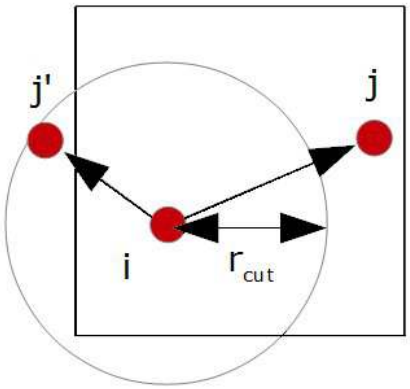

Figure 1.4: The Minimum Image Convention

1.2.6 The Inter-atomic Force

The pair potential gives rise to a force acting between the atoms i and j. The force on the atom j due to atom i is calculated as a derivative:

$$\vec{f}_{ij}(r_{ij}) = -\left(\frac{1}{r_{ij}}\right)\left(\frac{d}{dr_{ij}}\Phi(r_{ij})\right)\vec{r}_{ij} \tag{1.4}$$

Note that the force $\vec{f}_{ij}(r_{ij})$ is a vector quantity with three components in the x, y and z directions, while the original potential $\Phi(r_{ij})$ is a scalar. The force atom j exerts on atom i, $\left(\vec{f}_{ji}(r_{ij})\right)$ has the same magnitude as this, but acts in the opposite direction - as demanded by Newton's third law of motion. Note also that the force acts along the direction of the vector $\vec{r}_{ij}$, which is a property of spherical atoms. For non-spherical entities this is not necessarily the case, but these are rarely used.

The pair force obtained from the Lennard-Jones potential (1.2) is

$$\vec{f}_{ij} = \left(\frac{24\varepsilon}{r_{ij}^2}\right)\left(2\left(\frac{\sigma}{r_{ij}}\right)^{12} - \left(\frac{\sigma}{r_{ij}}\right)^{6}\right)\vec{r}_{ij} \tag{1.5}$$

The pair force given in equation (1.4) is not the full force acting on atom j. All the other atoms within the cut-off range r_{cut} of atom j will also exert a force. Thus the net force acting of atoms j is the vector sum of these:

$$\vec{f}_j = \sum_{(i \neq j)}^{(r_{ij} < r_{cut})} \vec{f}_{ij}(r_{ij}). \tag{1.6}$$

1.2.7 The Initial Conditions

The positions and velocities of the atoms at the start of the simulation are initial conditions that we must specify. The atom positions, for example, may be taken from a crystal structure or from a previous, related simulation, perhaps one performed at a different temperature. The options are many and varied and in many cases are quite simple to implement. Liquid or amorphous structures can be started from crystals which melt when a high temperature is specified. In this case care must be taken to ensure that melting has actually occurred, perhaps by monitoring an order parameter appropriate for the crystal structure and seeing if this diminishes with time. Alternatively a linear mean squared displacement can be a useful indicator of diffusion taking place in a liquid state. But no test is absolutely fool proof and a battery of tests is advisable. On *no account* should a structure be set up by simply allocating atoms randomly within the simulation cell, unless mayhem is preferred to science.

The initial velocities can be generated from a sequence of random numbers and, after correcting for bulk translational motion (so that the system as a whole does not move in space), then scaled so that system kinetic energy is appropriate for the required system temperature. More sophisticated starting procedures use random numbers selected from a Gaussian distribution to generate starting velocities. By this means, one hopes, the thermodynamic equilibrium can be attained more quickly. It is sometimes necessary to remove bulk rotation from the system, particularly if there is no periodic boundary to inhibit this.

There are many things to consider when initiating a new simulation and it is not possible to cover everything here. But it is most important to think carefully about the issue beforehand. The vast majority of simulations that go wrong do so because they have been set up badly. As a rule, simulations should be started gently and nudged cautiously to where they are required to go. No attempt should be made to "force" the simulation violently in the desired direction.

1.2.8 The Equations of Motion

With this information we can apply Newton's equations of motion [5] to determine the subsequent motion of the atoms:

$$\begin{aligned} \frac{d\vec{v}_j}{dt} &= \frac{1}{m_j}\vec{f}_j, \qquad \text{(a)} \\ \frac{d\vec{r}_j}{dt} &= \vec{v}_j \qquad\quad \text{(b)} \end{aligned} \tag{1.7}$$

The quantity m_j is the mass of the atom j. The equations (1.7) are of course the analytical expressions for Newton's equations and, as mentioned previously, need to be written in a discrete form for molecular dynamics work. There are many different discrete forms that can be used, each with its own characteristics, such as stability and accuracy, but here we take a particularly simple form known as the leapfrog algorithm:

$$\vec{v}_j(n+1/2)=\vec{v}_j(n-1/2)+\frac{\Delta t}{m_j}\vec{f}_j(n), \quad \text{(a)}$$
$$\vec{r}_j(n+1)=\vec{r}_j(n)+\Delta t\,\vec{v}_j(n+1/2). \quad \text{(b)} \qquad (1.8)$$

The equations (1.8)(a) and (1.8)(b) respectively advance or update the atomic velocity and position over a discrete time step Δt . In (1.8)(a) the force at the $n'th$ time step is used to update the velocity at the $(n-1/2)'th$ time step to the $(n+1/2)'th$ time step, which is at a time interval Δt later. (The use of half time steps for the velocity is a particular feature of this algorithm, as will be explained in chapter 4, but introduces no real difficulty here.) In (1.8)(b) the position at the $n'th$ time step is updated to the $(n+1)'th$ time step using the velocity from the $(n+1/2)'th$ time step. The calculation of the new atomic positions then allows a fresh calculation of the atomic forces, from which a further step in time can be taken. Clearly the repetition of this sequence, for each time step, allows the simulation to proceed for any required period of time and usually this is many thousands of time steps.

1.2.9 The Establishment of Equilibrium

When a simulation starts from a fresh configuration of atoms, it is extremely unlikely that such an artificially generated system is correct for the temperature, density or pressure required for the system. We say that it is not representative of the *thermodynamic state* of the system, or equivalently, not in *thermodynamic equilibrium*. For this reason it is invariably necessary to run the simulation for a number of times steps, ranging from a few to many thousand depending on the system, so that the system has time to relax into an equilibrium state. That the system will eventually do so is a consequence of thermodynamics – all systems evolve to a state of maximum entropy, and once there, are in thermodynamic equilibrium. However there are no guarantees that this will happen quickly and it is important to ensure that equilibrium has been attained before extracting scientific data from the simulation. In this Lennard-Jones system, this *equilibration time* is usually of the order of a thousand time steps or so, but in some problematic cases it can take very much longer. The equilibration period is an important aspect of all molecular dynamics simulations and must be given due attention if meaningful results are to be obtained. The absence of drift in the values of system properties is a key indication of equilibration. As a minimum, the system kinetic energy (or temperature) and configuration energy should be seen to fluctuate around fixed average values and the fluctuations in the two should be of similar magnitude.

Inevitably, when a simulation starts, the system temperature immediately deviates from that required. Almost always (unless something peculiar is going on), the temperature rises as the system adopts a lower configuration energy. Whatever occurs, it can be offset by scaling the atomic velocities at regular intervals until no obvious drift in temperature occurs. After which the scaling is stopped to allow the system dynamics to proceed normally. Such *ad hoc* scaling should only be applied during the equilibration period.

1.2.10 The Calculation of System Properties

After equilibration, as the simulation proceeds, we may accumulate data to describe

the properties of the system. As mentioned above these are almost always averages of some kind. Basic properties include the system temperature, configuration energy and virial (which is needed to calculate the pressure), all of which can easily be calculated `on the fly' as the simulation proceeds. Simple formulas exist for these properties:

1. Temperature: (T)

$$T = \frac{2}{3k_B}\langle K \rangle, \tag{1.9}$$

where $\langle K \rangle$ is the average system *kinetic energy* and k_B is *Boltzmann's constant* (1.38 x 10^{-23} J K^{-1}). The angular brackets $\langle \ldots \rangle$ represent an average taken over the full equilibrated simulation.

2. Kinetic energy: (K)

$$K = \frac{1}{2}\sum_{i=1}^{N} m_i \dot{r}_i^2, \tag{1.10}$$

where N is the number of atoms in the simulation.

3. Configuration energy: (Φ_{conf})

$$\Phi_{conf} = \sum_{i=2}^{N} \sum_{j<i} \phi_{ij}(r_{ij}), \tag{1.11}$$

which is the form appropriate for pair potentials.

4. Virial: (Ψ)

$$\Psi = -\sum_{i=1}^{i<N} \sum_{j>i} \vec{r}_{ij} \cdot \vec{f}_{ij}(r_{ij}), \tag{1.12}$$

5. Pressure: (P)

$$P = \frac{1}{3V}\left(2\langle K \rangle - \langle \Psi \rangle\right) \tag{1.13}$$

where V is the system volume.

Some properties, particularly time dependent ones, take the form of *correlation functions* and are more complicated in nature. These are probably best calculated from data that is written out and stored on computer disk at prescribed time intervals. These are discussed later in chapter 3.

However, there are two particularly important properties that need an early mention: the *radial distribution* function (or RDF) and the *mean-squared displacement* (or

MSD). The RDF is an example of a (static) *structural* property that describes the way atoms are arranged in the system. The mean-squared displacement is an example of a *time-dependent* property and describes the atomic diffusion that takes place in the system. These two important functions are described in detail in chapter 3.

One very important property that is not immediately accessible to molecular dynamics simulation, as it has been described so far, is the *thermodynamic free energy*. By the same token *entropy* is not immediately accessible either. This is because these are not properties obtainable through sampling individual configurations of the system, but instead are a collective property of *all* the possible configurations the system can access. Fortunately, this does not mean that molecular dynamics cannot be used to determine the free energy *difference* between thermodynamic states, but generally speaking a single simulation of system is not sufficient to provide this. Rather, special techniques are required for this, which we shall encounter later in chapter 10.

This concludes the outline of the simulation of the Lennard-Jones system. In the rest of the book we shall look into the subjects raised in much greater detail.

1.3 Some Recommended Textbooks

In no particular order we present some textbooks that are extremely useful to anyone working in the molecular simulation field. They have been chosen for their thoroughness, clarity and the insight they provide.

1. *Computer Simulation of Liquids*, M.P. Allen and D.J. Tildesley, Oxford Clarendon Press (1987). Now a classic text, this provides one of the best introductions to the theory and practice of molecular simulation.

2. *Understanding Molecular Simulation*, D. Frenkel and B. Smit, Academic Press (2002). Another outstanding textbook, which offers valuable insight into some of the more advanced aspects of molecular simulation.

3. *The Art of Molecular Dynamics Simulation*, D.C. Rapaport, Cambridge University Press (2004). A detailed description of molecular simulation as a practical art, written by an expert programmer. Particularly useful to C programmers.

4. *Theory of Simple Liquids*, J.-P. Hansen and I.R. McDonald, Academic Press (1986). Another classic text presenting the theoretical basis for molecular theory. The insight provided is not restricted to liquids. Not an easy read, but very rewarding to persistent readers.

5. *Classical Mechanics*, H. Goldstein, Addison Wesley (1980). One of the best accounts of modern classical mechanics, rigorous and readable.

Chapter 2

The Mechanics of Molecular Dynamics

2.1 Introduction

In this chapter we present the elements of classical mechanics and the equations of motion on which molecular dynamics is founded. The treatment is more descriptive than rigorous and full derivations will not be given, unless they are either brief or illuminating. The justification for this approach is that a deep mathematical understanding is not as essential to molecular dynamics practice as is an insight into what it all actually *means*. This is not intended to discourage the reader from seeking out the derivations in the many excellent text books available on the subject, as there is much that can be learned from doing so. We wish merely to provide a less intimidating introduction and leave the worthy mathematics to authors better equipped to provide it.

2.2 Classical Mechanics

In molecular dynamics the physical systems we are concerned with are composed of atoms and molecules. So, without loss of generality, we confine our discussion of classical mechanics to such entities. The "particles" often referred to in dynamical theory are here simply atoms, which are considered to be spheres interacting via forces based on empirical potential energy functions, as was seen in the example presented in section 1.2.3. Sometimes these atoms are gathered into molecules, which means they are connected by spring-like "bonds" with flexible "bond angles", which are also described by empirical potentials. At other times the molecules can be partly or wholly rigid, but are almost always composed of spherical atoms held in some sort of framework. Rigidity in this case can be defined by the requirement that every pair of atoms in the structure maintains a constant distance of separation. At other times we may define molecules that posses rigid bonds, but variable bond angles, so that they retain *conformational flexibility*, which is often more important to molecular behaviour than bond vibration. In the most complicated cases all of these structural *motifs* can be present in the same system. In this chapter we outline the mathematics required to model the dynamics of such structures.

2.3 Newtonian Mechanics

Newton's approach to classical mechanics [5] (pp.1-11) is fully adequate to compute the dynamics of a system composed of simple atoms interacting through a centro-symmetric potential like the Lennard-Jones potential we saw in section 1.2.3. In principle the Newtonian prescription can provide a complete account of the dynamics of *any* system. However, it most readily lends itself to a description based on *Cartesian coordinates,* which are used to define the atomic positions, velocities and forces. For computational purposes this is convenient for the construction of vectors and arrays, which are highly efficient to work with. However, the more complicated molecular structures are not best described by Cartesian coordinates. In these cases

there are better coordinate systems that can be employed, which may include vibration or molecular rotation. Nevertheless, the conceptual directness of Newton's approach is appealing and most of the molecular dynamics programs that have been written to date are unashamedly Newtonian in concept and Cartesian in design and cope with the more complicated systems by modification and extension.

We shall describe Newton's equations of motion assuming the system consists of N atoms. Molecules are permissible, provided they are free of rigid components. In which case Newton's equations of motion take their simplest form with a minor change of notation. In the N atom system the $i'th$ atom is located in space by three components $(x_i, y_i, z_i),$ of a position vector $(\vec{r}_i),$ making a total of $3N$ coordinates which may be represented as a super-vector $(\vec{R}_{3N})$ with $3N$ components $(x_1, x_2, \ldots, x_{3N}),$ such that the components of $\vec{r}_i$ are now identified as the components $(x_{3i}, x_{(3i+1)}, x_{(3i+2)}).$ Newton's equations of motion can then be written as

$$\frac{dv_i}{dt} = \frac{f_i}{m_i}, \qquad \text{(a)}$$
$$\frac{d\,x_i}{dt} = v_i. \qquad \text{(b)} \qquad (2.1)$$

in which m_i represents the atomic mass and f_i the $i'th$ component of force. As was shown in section 1.2.6, the force arises from the interaction potential, as a derivative with respect to the coordinate. Equation (2.1)(a) shows how the velocity $\vec{v}_i$ is changed by the force and equation (2.1)(b) shows how the position x_i is changed by the velocity. The dynamics of the system is formally obtained by integration of $3N$ sets of these differential equations.

As an example of applying Newton's approach consider the motion of a projectile under the influence of gravity, as shown in figure 2.1

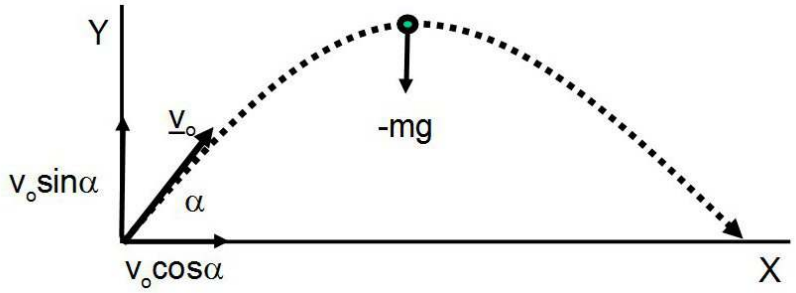

Figure 2.1: The Motion of a Projectile

The geometry of this system in confined to the two-dimensional XY plane, as shown. Directions X and Y represent the horizontal and vertical directions respectively. The

position of the projectile is given by the vector $\vec{r}=(x,y)$ and the force acting on it is the gravitational force $\vec{f}=(0,-mg),$ with m as the projectile mass and $-g$ as the acceleration due to gravity. The projectile is initially set at the position $\vec{r}_0=(x_0,0)$ and is given an initial velocity $\vec{v}_0=(v_o\cos(\alpha),v_o\sin(\alpha)).$ In this example the x and y components of the vectors are separable, so we may write (and integrate) Newton's equations as follows:

$$\begin{aligned} m\ddot{x}=0 \;\rightarrow\; \dot{x}=v_o\cos(\alpha) \;\rightarrow\; x=x_0+t\,v_0\cos(\alpha), & \quad \text{(a)} \\ m\ddot{y}=-mg \;\rightarrow\; \dot{y}=v_0\sin(\alpha)-g\,t \;\rightarrow\; y=t\,v_0\sin(\alpha)-\frac{g\,t^2}{2}, & \quad \text{(b)} \end{aligned} \tag{2.2}$$

where the constants of integration at each stage are supplied by the initial conditions. From these results it is easy to show that the trajectory of the particle is a parabola, as Galileo first proved. This simple example, much beloved of secondary school mathematicians, is the archetypal Newtonian dynamical system and the forerunner of the advanced dynamics of planetary systems and indeed the molecular dynamics simulation of atoms.

2.4 Langrangian Mechanics

Newton's approach works fine when the systems are relatively simple, but when physical *constraints* are present, such as rigid bonds or rigid molecules, then some way must be found to incorporate these into the scheme. Without constraints the $3N$ coordinates of the atomic positions are able to vary independently and the system is said to have $3N$ *degrees of freedom*. But with each constraint added (i.e. each rigid bond or fixed atomic pair separation in a rigid molecule) the number of degrees of freedom is reduced by one[1] and Newtons' equations of motion (2.1) must allow for this. With ingenuity system specific solutions are often possible, but a more general approach is desirable. To do this we need to go beyond a purely Cartesian description to one in which any practical set of coordinates may be used. This is the basis of Lagrange's approach [5] (pp.16-21) which introduces the idea of *generalised coordinates*.

Generalised positions $\{q_j\}$ represent any convenient coordinates defining the arrangement of the atoms or molecules in space. Such coordinates may, of course, be Cartesian, but we may also, for example, use instead angles that specify molecular orientations or bond angles, and so on. The number of these coordinates must be just sufficient to describe the arrangement completely and must be completely independent, so each can vary without affecting the others. For example, to define the arrangement of a diatomic molecule in space, we may use the three Cartesian coordinates of the molecule's centre of mass, the two Euler angles that define its orientation in space and the length of the bond as appropriate generalised coordinates. The derivative of the generalised positions with respect to time defines their velocities $\{\dot{q}_j\},$ where the `dot' notation signifies a single derivative with the respect to time.

Unlike Newtonian mechanics, in which the fundamental concept is force, Lagrangian

1 This assumes there are no redundant constraints i.e. more constraints than is strictly necessary to fix the rigid molecular geometry.

mechanics is based on energy. There are two energy components that we need to consider: the kinetic energy and the potential (or configuration) energy. We encountered both of these in section 1.2.10, where they appeared in a Cartesian description. Now we need to describe them in terms of generalised coordinates.

We start with Cartesian coordinates. It is apparent that, even in a system with constraints any Cartesian coordinate may be written as a function of the generalised coordinates:

$$x_i \equiv x_i\left(q_1, q_2, \ldots, q_{3\mathrm{N}}\right). \tag{2.3}$$

Note that there are up to $3\mathrm{N}$ generalised coordinates. But if there are constraints acting, some of these will be constants and the corresponding equation of motion may be ignored.

From equation (2.3) we obtain the corresponding velocity by the normal rules of differentiation:

$$\dot{x}_i \equiv \frac{dx_i}{dt} = \sum_{j=1}^{3\mathrm{N}} \left(\frac{\partial x_i}{\partial q_j} \right) \dot{q}_j . \tag{2.4}$$

Equations (2.3) and (2.4) are *transformations* from generalised coordinates to their Cartesian counterparts and their usefulness stems from the fact that it is generally easier to first describe a physical system in Cartesian coordinates and then convert to the generalised coordinates to handle the dynamics. In particular, the kinetic energy has a much simpler Cartesian form.

The kinetic energy (K) of a system of N atoms is given by

$$K(\dot{x}_i) = \frac{1}{2} \sum_{i=1}^{3\mathrm{N}} m_i \dot{x}_i{}^2 . \tag{2.5}$$

Conversion of this to the corresponding generalised form requires substituting (2.4) into (2.5) which gives

$$K\left(\{q_j\}, \{\dot{q}_j\}\right) = \frac{1}{2} \sum_{i=1}^{3\mathrm{N}} m_i \left(\sum_{j=1}^{3\mathrm{N}} \left\{ \frac{\partial x_i}{\partial q_j} \right\} \dot{q}_j \right)^2 . \tag{2.6}$$

Note that, when generalised coordinates are used, the kinetic energy becomes a function of both velocity $\dot{q}_j$ *and* position q_j. This arises because of the partial derivatives in (2.4). The equation appears to imply a tedious amount of analytical work to get the final expression, but in practice it is unusual for a coordinate x_i to depend on more than a few of the generalised coordinates q_j. In which case the equation becomes much simpler to handle. The transformation (2.4) is somewhat

over-pessimistic in this respect. Equation (2.6) should be regarded as the most general form.

The potential energy is written in the Cartesian form as $\Phi\left(\{x_i\}\right)$. Note that this has no dependence on the velocities $\{\dot{q}_j\}$, which means we are assuming no *dissipative* (i.e. frictional) forces operate in the system. Such systems are said to be *conservative*. To convert this to the generalised form $\Phi\left(\{q_j\}\right)$ is straightforward. All that is required is the substitution of every coordinate x_i appearing in the formula for the potential (whatever that may be) by the transform (2.3). Formally we may write this as

$$\Phi\left(\{x_i\}\right)\Rightarrow\Phi\left(\{q_i\}\right) \tag{2.7}$$

This may, or may not, be a tedious thing to do. In molecular dynamics this is rarely difficult because it is common practice to define the potential function already in terms of the generalised coordinates, since they offer a more natural description than the Cartesian. For example pair potentials are described in terms of the inter-atomic distance rather than atomic positions. Bond angle potentials are similarly given as functions of angles, so a transformation of the form of the potential is not required. However, bearing in mind the previous comment that molecular dynamics programs are most often cast in a Cartesian form, a knowledge of the reverse transformation of (2.7) is a practical necessity.

The mechanics of Lagrange starts with a function: the *Lagrangian*, which is defined as the instantaneous difference between the kinetic energy of a dynamical system and its potential energy. This is written as

$$L\left(\{q_j\},\{\dot{q}_j\}\right)=K\left(\{q_j\},\{\dot{q}_j\}\right)-\Phi\left(\{q_j\}\right). \tag{2.8}$$

Lagrange's equations of motion can be derived from the Lagrangian by a *variational principle*. It can be shown that when dynamical systems evolve in time, the path taken is such that the time integral of the Lagrangian function is *extremal*, meaning it is either a maximum or a minimum, which is equivalent to the first derivative of the Lagrangian along the path being zero. Formally, for a system evolving from time t_1 to the time t_2, this is written as

$$\delta\int_{t_1}^{t_2} L\left(\{q_j\},\{\dot{q}_j\}\right)dt=0. \tag{2.9}$$

This condition is sufficient to allow the derivation of the equations of motion for a system defined by generalised coordinates. We quote the result below, which represents Lagrange's equations of motion:

$$\frac{d}{dt}\left(\frac{\partial L\left(\{q_j\},\{\dot{q}_j\}\right)}{\partial \dot{q}_j}\right)=\frac{\partial L\left(\{q_j\},\{\dot{q}_j\}\right)}{\partial q_j}, \quad \text{for all} \quad j. \tag{2.10}$$

Lagrange's equations of motion may be derived directly from Newtonian mechanics, but the derivation from the *variational principle* is both elegant and powerful. Incidentally, it is a simple exercise to show that (2.10) holds for Cartesian coordinates. Using $\{x_i\}$ and $\{\dot{x}_i\}$ in place of $\{q_j\}$ and $\{\dot{q}_j\}$ (which is straight-forward for a system without constraints,) quickly reduces the equations to the Newtonian form (2.1).

As an example of Lagrange's approach we consider a model homo-nuclear diatomic molecule in a two dimensional space, as shown in figure 2.2.

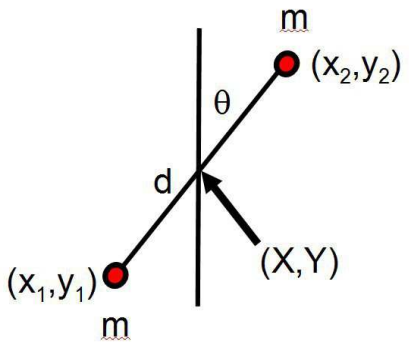

Figure 2.2: A Model Homo-nuclear Diatomic Molecule

The molecule is composed of two atoms of mass m at positions $\vec{r}_1=(x_1,y_1)$ and $\vec{r}_2=(x_2,y_2).$ The centre of mass of the molecule is thus at $\vec{R}=(X,Y)$ where $X=(x_1+x_2)/2$ and $Y=(y_1+y_2).$ The atoms are separated by a distance $d=\left((x_1-x_2)^2+(y_1-y_2)^2\right)^{(1/2)}$ where d is the bond-length and the bond makes an angle θ with the *y*-direction, so that $\tan(\theta)=(x_1-x_2)/(y_1-y_2).$ The four quantities X,Y,d and θ represent a convenient set of coordinates with which to define the configuration of the system, since the pair (X,Y) defines the position of the molecule, d the bond-length and θ the orientation of the molecule. These become the generalised coordinates $\{q_i : i=1,\ldots,4\}$ replacing the original four: x_1, y_1, x_2 and $y_2.$ Following Lagrange's prescription we must express the original coordinates in terms of the generalised ones, thus:

$$\begin{aligned} x_1&=-\frac{d}{2}\sin(\theta)+X, \quad & x_2&=\frac{d}{2}\sin(\theta)+X, \quad && \text{(a)} \\ y_1&=-\frac{d}{2}\cos(\theta)+Y, \quad & y_2&=\frac{d}{2}\cos(\theta)+Y. && \text{(b)} \end{aligned} \tag{2.11}$$

The time derivatives of these are

$$\dot{x}_1 = -\frac{\dot{d}}{2}\sin(\theta) - \frac{d}{2}\cos(\theta)\dot{\theta} + \dot{X}, \quad \dot{x}_2 = \frac{\dot{d}}{2}\sin(\theta) + \frac{d}{2}\cos(\theta)\dot{\theta} + \dot{X}, \quad \text{(a)}$$
$$\dot{y}_1 = -\frac{\dot{d}}{2}\cos(\theta) + \frac{d}{2}\sin(\theta)\dot{\theta} + \dot{Y}, \quad \dot{y}_2 = \frac{\dot{d}}{2}\cos(\theta) - \frac{d}{2}\sin(\theta)\dot{\theta} + \dot{Y}. \quad \text{(b)} \qquad (2.12)$$

The kinetic energy, K, written in terms of these velocities, is easily shown to be

$$K = \frac{m}{2}\left(\dot{x}_1^{\,2} + \dot{x}_2^{\,2} + \dot{y}_1^{\,2} + \dot{y}_2^{\,2}\right) = m\dot{X}^{\,2} + m\dot{Y}^{\,2} + \frac{1}{4}m\dot{d}^{\,2} + \frac{1}{4}md^2\dot{\theta}^{\,2}. \qquad (2.13)$$

The only source of potential energy in this system is the bond between the atoms, which we can write as $\Phi = \Phi(d)$, and therefore the Lagrangian function $L = K - \Phi$ is fully defined. From Lagrange's equation (2.10) we have the following equations of motion:

$$\frac{1}{2}m\ddot{d} = -\acute{\Phi}(d) + \frac{1}{2}md\,\dot{\theta}^{\,2}, \quad \text{(a)}$$
$$d\,\ddot{\theta} + 2\dot{d}\,\dot{\theta} = 0, \quad \text{(b)} \qquad (2.14)$$
$$\ddot{X} = 0, \quad \ddot{Y} = 0, \quad \text{(c)}$$

where $\acute{\Phi}(d)$ in (2.14)(a) is the first derivative of $\Phi(d)$ with respect to d. The first of these equations is evidently a modified vibrational equation of motion – augmented by a centrifugal term (second on the right) which incorporates the effects of spinning the molecule. The second equation describes the rotational motion of the molecule, augmented by the effects of bond vibration (second term on the left). Note there are no forces operating in this equation. The last two equations show there is no acceleration of the molecule as a whole, since no external forces are operating. The exact solution of these equations depends upon the form of $\Phi(d)$. It should also be noted that if the bond length d is fixed, the equations simplify to describe the free rotation of a rigid diatomic molecule.

These equations introduce an important concept: the separation of the modes of motion of a molecule into translational, rotational and vibrational components. This separation is invariably used to describe the motions of molecules in general.

2.5 Hamiltonian Mechanics

Hamilton's mechanics also makes use of generalised coordinates. The main distinction from Lagrange's mechanics is the use of the generalised *momentum* p_i as a key dynamical variable. The generalised momentum is defined as a derivative of the Lagrangian function:

$$p_j = \frac{\partial L\left(\{q_k\},\{\dot{q}_k\}\right)}{\partial \dot{q}_j}. \qquad (2.15)$$

This is described as the momentum *conjugate* to the generalised position. The momentum may be used to define a new function, the *Hamiltonian:*

$$H\left(\{q_j\},\{p_j\}\right)=\sum_{i=1}^{3N} p_i\dot{q}_i - L\left(\{q_j\},\{\dot{q}_j\}\right), \tag{2.16}$$

in which $H\left(\{q_j\},\{p_j\}\right)$ is not explicitly dependant on the velocities $\{\dot{q}_j\}$ since these are nominally functions of the coordinates $\{q_j\}$ and momenta $\{p_i\}$. The Hamiltonian function is the basis for a new set of equations to describe the dynamical motion of the system:

$$\begin{aligned} \frac{dq_j}{dt} &= \frac{\partial H\left(\{q_k\},\{p_k\}\right)}{\partial p_j} \quad \text{(a)} \\ \frac{dp_j}{dt} &= -\frac{\partial H\left(\{q_k\},\{p_k\}\right)}{\partial q_j} \quad \text{(b)} \end{aligned} \tag{2.17}$$

The equations (2.17) are *Hamilton's equations of motion*. Once again, by replacing the generalised coordinates q_j and $\dot{q}_j$ in equations (2.15) to (2.17) with their Cartesian counterparts, Hamilton's equations can easily be shown to be identical to Newton's.

An important property of the Hamiltonian stems from the fact that for a conservative system, the potential function $\Phi\left(\{q_j\}\right)$ is not dependent on the velocities $\{\dot{q}_j\}$ so equation (2.15) becomes

$$p_j=\frac{\partial K\left(\{q_k\},\{\dot{q}_k\}\right)}{\partial \dot{q}_j}, \tag{2.18}$$

and since it is evident from (2.6) that the kinetic energy K is a quadratic function of velocities $\{\dot{q}_j\}$, it is easy to show that the first term right of equation (2.16) is in fact $2K$. So it follows from the definition of the Lagrangian in (2.8) that

$$H\left(\{q_j\},\{p_j\}\right)=K\left(\{q_j\},\{\dot{p}_j\}\right)+\Phi\left((q_j)\right), \tag{2.19}$$

which means that the Hamiltonian is identical to the *total energy* of a conservative system.

Lagrange's and Hamilton's equations are equivalent ways of describing the dynamics of a system, since both are couched in generalised coordinates. However, the identification of the Hamiltonian with the total energy of the system means it makes frequent appearance in statistical mechanics, and from that association, the Hamiltonian acquires a much wider significance and power.

To demonstrate Hamilton's approach we consider again the homo-nuclear diatomic molecule in figure 2.2, but with an additional potential: we shall assume that the atoms of the molecule are confined in space by a quadratic potential of the form

$$\Phi(x, y)=\frac{k}{2}\left(x^2+y^2\right), \tag{2.20}$$

which operates in addition to the bond potential $\Phi(d)$ we encountered earlier. We may use the same generalised coordinates as before, which means we can obtain the momenta conjugate to these by using the operation (2.18) with the kinetic energy definition (2.13) to give

$$p_X=2m\dot{X}, \quad p_Y=2m\dot{Y}, \quad p_\theta=\frac{1}{2}md^2\dot{\theta}, \quad p_d=\frac{1}{2}m\dot{d} \tag{2.21}$$

in terms of which the kinetic energy becomes

$$K=\frac{p_X^2}{4\mathrm{m}}+\frac{p_Y^2}{4\mathrm{m}}+\frac{p_\theta^2}{md^2}+\frac{p_d^2}{m}. \tag{2.22}$$

As mentioned already, the potential energy of the system is now

$$\Phi=\Phi(d)+\Phi(x_1, y_1)+\Phi(x_2, y_2) \tag{2.23}$$

where we retain, for convenience, the Cartesian description of the second and third terms on the right. The Hamiltonian H is the sum of $K+\Phi$ and we may use Hamilton's equations (2.17) together with the relations (2.21) to derive the equations of motion (2.24).

$$\begin{aligned}
&\dot{X}=\frac{p_X}{2\mathrm{m}}, \quad \dot{Y}=\frac{p_Y}{2\mathrm{m}}, \quad \dot{\theta}=\frac{2p_\theta}{md^2}, \quad \dot{d}=\frac{2p_d}{m}, && \text{(a)}\\
&\dot{p}_X=-kx_1-kx_2, \quad \dot{p}_Y=-ky_1-ky_2, && \text{(b)}\\
&\dot{p}_\theta=-kx_1(y_1-Y)+ky_1(x_1-X)-kx_2(y_2-Y)+ky_2(x_2-X), && \text{(c)}\\
&\dot{p}_d=-\dot{\Phi}(d)+\frac{2\mathrm{p}_\theta^2}{md^3}-\frac{1}{d}\left(kx_1(x_1-X)+ky_1(y_1-Y)+kx_2(x_2-X)+ky_2(y_2-Y)\right). && \text{(d)}
\end{aligned} \tag{2.24}$$

What do these equations mean? Equations (2.24)(a) are merely a rearrangement of (2.21). The remaining equations can be more easily interpreted if it is recognised that terms like $-kx_i$ represent the x-component of the force on atom i due to the confining potential (2.20) with a corresponding interpretation applying to all similar

terms. We shall represent these external[2] atomic forces by the vectors $\vec{f}_i$. With this identification we may represent both equations (2.24)(b) vectorially as

$$\dot{\vec{p}}_M = \vec{f}_1 + \vec{f}_2 \tag{2.25}$$

where $\vec{p}_M$ is the momentum of the molecule's centre of mass. This equation shows that the motion of the molecule's centre of mass derives from a vector sum of all the external forces acting on the molecule – an important result.

Equation (2.24)(c) may be written as

$$\dot{p}_\theta = \vec{f}_1 \times (\vec{r}_1 - \vec{R}) + \vec{f}_2 \times (\vec{r}_2 - \vec{R}) \tag{2.26}$$

where $\vec{R}$ is the vector (X, Y) locating the molecule's centre of mass and $\vec{r}_i$ is the vector (x_i, y_i) locating the $i'th$ atom. The operation $\vec{a} \times \vec{b}$ represents a *vector product*. The right side of equation (2.26) can be seen to be the sum of vectors which are the products of atomic forces and the corresponding displacements of the atoms from the molecular centre. Such products are known as *torques* and are the rotational equivalent of a force. The calculation of torques is a major requirement for general purpose molecular dynamics programs.

Equation (2.24)(d) can be written as

$$\dot{p}_d = -\dot{\Phi}(d) + \frac{2\,p_\theta^2}{m d^3} + \frac{1}{d}\left(\vec{f}_1 \cdot (\vec{r}_1 - \vec{R}) + \vec{f}_2 \cdot (\vec{r}_2 - \vec{R})\right). \tag{2.27}$$

Comparing (2.27) with equation (2.14)(a) we see essentially the same modified vibrational equation of motion for the bond, with the first two terms on the right providing the same dynamical contributions as before. The third term provides the forces due to the external confinement potential - as components directed along the bond, as can be seen from the scalar product of the atomic forces with the displacement of the atoms from the centre of mass.

The equations (2.25) to (2.27) thus provide insight into how the atomic forces are combined with the molecular structure description to obtain the dynamics of the molecule in terms of translational, rotational and vibrational components. For reasons to do with practical molecular dynamics however, it has to be said that if the atoms of a molecule are linked through flexible bonds, it is not necessary to switch to a molecular representation of the dynamics. It is perfectly acceptable to handle the dynamics in terms of the constituent atoms and treat the different sources of force as separate contributions to the dynamics of the atoms. In computational terms this is both convenient and simple to implement. Nevertheless, there are models of molecules which are not totally flexible and then it is more sensible to consider the dynamics of whole molecules, in which the rotational and translational motion are

2 That is, forces originating external to the molecule.

handled in a manner similar to that shown here.

2.6 Rigid Body Dynamics

While it is clear from the preceding sections that the Hamiltonian and Lagrangian approaches to classical dynamics are extremely general, the Newtonian approach has an appealing directness. So for the relatively simple molecular models that are often dealt with in molecular dynamics a Newtonian approach is most commonly adopted. This does not diminish the value of the Lagrange and Hamilton approaches in the context of molecular dynamics, as they have uses beyond dynamics alone, as will become evident in later chapters. But meanwhile we look at Newtonian dynamics as our method of choice for the dynamics of molecules.

We have mentioned in section 2.2 the kinds of molecular models we are mostly concerned with. Simple non-bonded atoms are directly amenable to a Newtonian treatment, as we have seen in the example of the noble gas in section 1.2 . In principle the same methods apply to molecules that consist of atoms held together by flexible bonds. Molecules that have rigid bonds, but otherwise flexible bond angles, can be handled by a modification of this basic approach, known as the SHAKE algorithm[6], which we describe in chapter 4, section 4.5. The remaining molecular model to consider is one in which every atom maintains a fixed distance from every other, though the molecule is free to move in space and to rotate. Dynamically this is known as a rigid body and the dynamical treatment of such an object is the concern of this section.

We first define the total mass $M,$ of a molecule composed of N_a atoms as

$$M = \sum_{i=1}^{N_a} m_i . \tag{2.28}$$

We also define the centre of mass $\vec{R}$ of the molecule as

$$\vec{R} = \frac{1}{M} \sum_{i=1}^{N_a} m_i \vec{r}_i . \tag{2.29}$$

It follows from (2.29) that

$$\ddot{\vec{R}} = \frac{1}{M} \sum_{i=1}^{N_a} m_i \ddot{\vec{r}}_i . \tag{2.30}$$

Newton's equation of motion for the $i'th$ atom in the molecule is

$$m_i \ddot{\vec{r}}_i = \vec{f}_i + \sum_{j \neq i}^{N_a} \vec{f}_{ij} , \tag{2.31}$$

where $\vec{f}_i$ is the force acting on the atom from *outside* the molecule i.e. from other molecules in the system, and $\vec{f}_{ij}$ is the force on atom i from other atoms in the molecule. Note we are assuming here that these forces are pair forces, which is not in general true, but for rigid bodies it leads to a result that generally is true. Summing (2.31) for all atoms in the molecule gives

$$\sum_{i=1}^{N_a} m_i \ddot{\vec{r}}_i = \sum_{i=1}^{N_a} \vec{f}_i + \sum_{i=1}^{N_a} \sum_{j \neq i}^{N_a} \vec{f}_{ij}. \tag{2.32}$$

From Newton's third law we have

$$\vec{f}_{ij} = -\vec{f}_{ji}, \tag{2.33}$$

so the pair forces cancel and (2.32) becomes

$$\sum_{i=1}^{N_a} m_i \ddot{\vec{r}}_i = \sum_{i=1}^{N_a} \vec{f}_i, \tag{2.34}$$

which, using (2.30), can be written as

$$M \ddot{\vec{R}} = \vec{F}, \tag{2.35}$$

where

$$\vec{F} = \sum_{i=1}^{N_a} \vec{f}_i. \tag{2.36}$$

Equation (2.36) shows that the centre of mass of a molecule moves *translationally* as a single particle of mass M under the influence of the total force $\vec{F}$. This is what we saw in the examples in the sections 2.4 and 2.5 preceding.

With regard to rotational motion, we begin by defining the *angular momentum* (which is also called the *moment of momentum*) $\vec{J}$, of the molecule around its centre of mass as

$$\vec{J} = \sum_{i=1}^{N_a} \vec{d}_i \times m_i \dot{\vec{d}}_i, \tag{2.37}$$

where $\vec{d}_i$ is the vector defining the displacement of the atom, i, from the centre of mass, $\vec{R}$, i.e.

$$\vec{d}_i = \vec{r}_i - \vec{R}, \tag{2.38}$$

where $\vec{r}_i$ is the atom's global location. The vector $\dot{\vec{d}}_i$ is the time derivative of $\vec{d}_i$.

The derivative of (2.37) with respect to time is

$$\dot{\vec{J}} = \frac{d}{dt}\sum_{i=1}^{N_a}(\vec{d}_i \times m_i \dot{\vec{d}}_i) = \sum_{i=1}^{N_a}\dot{\vec{d}}_i \times m_i \dot{\vec{d}}_i + \sum_{i=1}^{N_a}\vec{d}_i \times m_i \ddot{\vec{d}}_i. \tag{2.39}$$

The first term on the right of (2.39) involves a vector product of two identical vectors and is therefore zero. The second term on the right can be expanded using Newton's second law

$$\dot{\vec{J}} = \sum_{i=1}^{N_a}\vec{d}_i \times m_i \ddot{\vec{d}}_i = \sum_{i=1}^{N_a}\vec{d}_i \times \left(\vec{f}_i + \sum_{j\neq i}^{N_a}\vec{f}_{ij}\right). \tag{2.40}$$

Now, it is easily shown that

$$\vec{d}_i \times \vec{f}_{ij} = -\vec{d}_j \times \vec{f}_{ji}, \tag{2.41}$$

so that the terms involving the pair forces cancel and (2.40) becomes

$$\dot{\vec{J}} = \sum_{i=1}^{N_a}\vec{d}_i \times \vec{f}_i = \vec{\tau}, \tag{2.42}$$

where $\vec{\tau}$ is the *molecular torque* or *turning force* about the molecule's centre of mass.

Perhaps surprisingly, nothing we have derived so far assumes the molecule is actually rigid! We now make that assumption by declaring an obvious relationship between the velocity of the atoms with respect to the centre of mass, $\dot{\vec{d}}_i$, and the angular velocity $\vec{\omega}$, of the molecule as a whole:

$$\dot{\vec{d}}_i = \vec{\omega} \times \vec{d}_i. \tag{2.43}$$

The angular momentum $\vec{J}$ given in (2.37) may therefore be written as

$$\vec{J} = \sum_{i=1}^{N_a}\vec{d}_i \times m_i \dot{\vec{d}}_i = \sum_{i=1}^{N_a}\vec{d}_i \times m_i(\vec{\omega} \times \vec{d}_i) = \sum_{i=1}^{N_a} m_i(\vec{\omega}(\vec{d}_i \cdot \vec{d}_i) - \vec{d}_i(\vec{d}_i \cdot \vec{\omega})). \tag{2.44}$$

In the last expression on the right we have made use of the standard decomposition of a vector triple product:

$$\vec{a}\times(\vec{b}\times\vec{c})=(\vec{a}\cdot\vec{c})\vec{b}-(\vec{a}\cdot\vec{b})\vec{c}\,. \tag{2.45}$$

It is informative now to split the vector equation (2.44) into its components as follows

$$\begin{aligned}
J^x&=\sum_{i=1}^{N_a} m_i\Big(\omega^x\left\{(d_i^x)^2+(d_i^y)^2+(d_i^z)^2\right\}-d_i^x\left\{\omega^x d_i^x+\omega^y d_i^y+\omega^z d_i^z\right\}\Big),\\
J^y&=\sum_{i=1}^{N_a} m_i\Big(\omega^y\left\{(d_i^x)^2+(d_i^y)^2+(d_i^z)^2\right\}-d_i^y\left\{\omega^x d_i^x+\omega^y d_i^y+\omega^z d_i^z\right\}\Big),\\
J^z&=\sum_{i=1}^{N_a} m_i\Big(\omega^z\left\{(d_i^x)^2+(d_i^y)^2+(d_i^z)^2\right\}-d_i^z\left\{\omega^x d_i^x+\omega^y d_i^y+\omega^z d_i^z\right\}\Big).
\end{aligned} \tag{2.46}$$

Then it is apparent that (2.46) can be written in matrix form as follows

$$\begin{bmatrix}J^x\\J^y\\J^z\end{bmatrix}=\begin{bmatrix}I^{xx}\,I^{xy}\,I^{xz}\\I^{yx}\,I^{yy}\,I^{yz}\\I^{zx}\,I^{zy}\,I^{zz}\end{bmatrix}\begin{bmatrix}\omega^x\\\omega^y\\\omega^z\end{bmatrix}\quad\text{or}\quad\vec{J}=\boldsymbol{I}\,\vec{\omega} \tag{2.47}$$

where we have defined a matrix, $\boldsymbol{I}$, which is known as the *moment of inertia tensor*, with the elements

$$\begin{aligned}
I^{xx}&=\sum_{i=1}^{N_a} m_i\left(\{d_i^y\}^2+\{d_i^z\}^2\right), &\quad I^{xy}&=I^{yx}=-\sum_{i=1}^{N_a} m_i d_i^x d_i^y,\\
I^{yy}&=\sum_{i=1}^{N_a} m_i\left(\{d_i^z\}^2+\{d_i^x\}^2\right), &\quad I^{yz}&=I^{zy}=-\sum_{i=1}^{N_a} m_i d_i^y d_i^z,\\
I^{zz}&=\sum_{i=1}^{N_a} m_i\left(\{d_i^x\}^2+\{d_i^y\}^2\right), &\quad I^{xz}&=I^{zx}=-\sum_{i=1}^{N_a} m_i d_i^x d_i^z.
\end{aligned} \tag{2.48}$$

In condensed form these elements may be written as

$$I^{\alpha\beta}=\sum_{i=1}^{N_a} m_i\left(\delta_{\alpha\beta}d_i^2-d_i^\alpha d_i^\beta\right), \tag{2.49}$$

where $d_i^2=\vec{d}_i\cdot\vec{d}_i$ and α,β represent Cartesian components x,y,z *etc.* It is important to note that $\boldsymbol{I}$ is a *symmetric matrix*, meaning that $I^{\alpha\beta}=I^{\beta\alpha}$.

Using the equation (2.47) we may re-write the rotational equation of motion (2.42) as

$$\frac{d}{dt}(\boldsymbol{I}\,\vec{\omega}) = \vec{\tau}. \tag{2.50}$$

This is the equation we need to solve to obtain the rotational dynamics of a rigid molecule.

A loose description of equation (2.50) is that $\vec{\tau}$ represents the turning force on the molecule and that the product $\boldsymbol{I}\,\vec{\omega}$ represents the molecular "rotational momentum", where $\vec{\omega}$ is the equivalent of velocity and $\boldsymbol{I}$ the equivalent of mass. This makes (2.50) seem to be a variant of Newton's second law of motion. However we cannot ignore the fact that $\boldsymbol{I}$ is a tensor, not a scalar, and this means the equation does not behave in the same way as Newton's. The main problem is that, unlike mass, $\boldsymbol{I}$ is not a constant since the elements of the matrix change with the orientation of the molecule in space. Nevertheless in what follows we exploit the analogy to tackle the problem head-on and show that a practical solution of (2.50) is actually possible. Later however we shall return to the more established approaches in which most of the literature is couched.

Performing the differentiation in (2.50) gives the following form

$$\vec{\tau} = \boldsymbol{I}\,\dot{\vec{\omega}} + \dot{\boldsymbol{I}}\,\vec{\omega}, \tag{2.51}$$

where we recognize that $\boldsymbol{I}$ changes with time. Its derivative, $\dot{\boldsymbol{I}}$, is obtained by differentiating (2.49) which leads to

$$\dot{I}^{\alpha\beta} = -\sum_{i=1}^{N_a} m_i\left(d_i^{\alpha}\dot{d}_i^{\beta} + d_i^{\beta}\dot{d}_i^{\alpha}\right), \tag{2.52}$$

in which we have recognised that d_i^2 is a constant with zero time derivative. The velocity $\dot{\vec{d}}_i$ is given by equation (2.43) which may be expanded into its components and inserted into (2.52) to give

$$\dot{I}^{xy}=\dot{I}^{yx}=-\sum_{i=1}^{N_a} m_i\left(d_i^x\left(\omega^z d_i^x-\omega^x d_i^z\right)+d_i^y\left(\omega^y d_i^z-\omega^z d_i^y\right)\right)=\omega^z\left(I^{xx}-I^{yy}\right)-\omega^x I^{xz}+\omega^y I^{yz},$$
$$\dot{I}^{yz}=\dot{I}^{zy}=-\sum_{i=1}^{N_a} m_i\left(d_i^y\left(\omega^x d_i^y-\omega^y d_i^x\right)+d_i^z\left(\omega^z d_i^x-\omega^x d_i^z\right)\right)=\omega^x\left(I^{yy}-I^{zz}\right)-\omega^y I^{yx}+\omega^z I^{zx},$$
$$\dot{I}^{zx}=\dot{I}^{xz}=-\sum_{i=1}^{N_a} m_i\left(d_i^z\left(\omega^y d_i^z-\omega^z d_i^y\right)+d_i^x\left(\omega^x d_i^y-\omega^y d_i^x\right)\right)=\omega^y\left(I^{zz}-I^{xx}\right)-\omega^z I^{zy}+\omega^x I^{xy},$$
$$\dot{I}^{xx}=-\sum_{i=1}^{N_a} 2\,m_i d_i^x\left(\omega^y d_i^z-\omega^z d_i^y\right)=2\left(\omega^y I^{xz}-\omega^z I^{xy}\right),$$
$$\dot{I}^{yy}=-\sum_{i=1}^{N_a} 2\,m_i d_i^y\left(\omega^z d_i^x-\omega^x d_i^z\right)=2\left(\omega^z I^{yx}-\omega^x I^{yz}\right),$$
$$\dot{I}^{zz}=-\sum_{i=1}^{N_a} 2\,m_i d_i^z\left(\omega^x d_i^y-\omega^y d_i^x\right)=2\left(\omega^x I^{zy}-\omega^y I^{zx}\right),$$
(2.53)

Thus at any given instant of time, every quantity in equation (2.51) is known apart from $\vec{\omega}$. The equation may be rearranged into

$$\dot{\vec{\omega}}=\boldsymbol{I}^{-1}\left(\vec{\tau}-\dot{\boldsymbol{I}}\,\vec{\omega}\right),\tag{2.54}$$

which can be integrated numerically over a time step to update the angular velocity $\vec{\omega}$. Then it only remains to update the vectors $\vec{d}_i$ using equation (2.43) to complete the integration of the molecular rotation over the time step. This almost seems like a practical scheme, except that equation (2.43) must be integrated *exactly* if the magnitude of $\vec{d}_i$ is to be preserved, which is essential to retain the integrity of the molecular structure. A naïve numerical integration, accurate to some finite order, simply won't do. However a practical numerical scheme for rotational motion employing equations (2.54) and (2.43) is possible, and is presented in chapter 4, section 4.6.1.2 .

While a direct approach undoubtedly works as a numerical scheme, it appears to have been overlooked by the majority of program writers, who almost universally employ methods based on the analytical approach of Euler. This requires the use of two distinct frames of reference: a laboratory frame, in which the whole simulated system is located, and a local frame that is exclusive to the molecule in question. The purpose of this is to cast the dynamical problem into a frame where the moment of inertia tensor is always diagonal, and therefore constant. It also provides a convenient way to define the orientation of a molecule in the laboratory frame.

2.7 The Principal Moments of Inertia

The difficulty in handling the matrix $\boldsymbol{I}$ is alleviated by the fact that for all rigid molecules there exists a particular orientation for which the matrix has a *diagonal* form in which $I^{\alpha\beta}=0$ when $\alpha\neq\beta$. The remaining diagonal elements of $\boldsymbol{I}$ namely, $I^{xx}, I^{yy}, I^{zz},$ are then called the *principal moments of inertia*. The orientation in which this occurs provides a convenient *reference orientation*, relative to which all

other orientations of the molecule may be defined. We shall refer to this as the *principal orientation*. This property of rigid molecules is the key to solving the equation of motion (2.50). Finding this orientation and the principal moments of inertia is therefore our first consideration when defining a new rigid molecule. The procedure for doing this is as follows.

- Firstly the molecule is described in an arbitrary orientation. The positions of all the atoms are described by displacement vectors $\vec{d}_i$ defined with respect to the molecular centre of mass, $\vec{R}$. From these vectors and the masses m_i of each of the atoms, the moment of inertia tensor $\boldsymbol{I}$ may be constructed from equations (2.48).

- We now perform a mathematical operation on $\boldsymbol{I}$ known as a *diagonalization*. There are many standard procedures for this, but for a real, symmetric matrix like $\boldsymbol{I}$ the *Jacobi* procedure is particularly convenient. It is a standard utility available in coded form in every library of linear algebra routines. We need not go into details, but what this procedure amounts to is the iterative construction of a matrix $\boldsymbol{R}$ and its transpose $\tilde{\boldsymbol{R}}$ (for which $\tilde{R}^{\alpha\beta}=R^{\beta\alpha}$) such that $\boldsymbol{I}^p$, the diagonalized form of $\boldsymbol{I}$, is given by the matrix product

$$\boldsymbol{I}^p=\tilde{\boldsymbol{R}}\,\boldsymbol{I}\,\boldsymbol{R}. \tag{2.55}$$

  The outcome of the Jacobi procedure is thus the matrices $\boldsymbol{I}^p$ and $\boldsymbol{R}$.

- The diagonal elements of $\boldsymbol{I}^p$ constitute the required principal moments of inertia. The matrix $\boldsymbol{R}$ on the other hand is a *rotation matrix*, which rotates the molecule from the principal orientation into the arbitrary orientation in which the tensor $\boldsymbol{I}$ was originally defined. If vector $\vec{d}_i^{\,p}$ is the atomic displacement with respect to the molecular centre of mass in the principal frame and $\vec{d}_i$ is the same vector described in the laboratory frame, then

$$\vec{d}_i=\boldsymbol{R}\,\vec{d}_i^{\,p}, \tag{2.56}$$

  Note that equation (2.56) may be inverted in the following manner

$$\vec{d}_i^{\,p}=\boldsymbol{R}^{-1}\vec{d}_i=\tilde{\boldsymbol{R}}\,\vec{d}_i \tag{2.57}$$

  where we have used the property of all rotation matrices that $\tilde{\boldsymbol{R}}=\boldsymbol{R}^{-1}$ (so that $\boldsymbol{R}\,\tilde{\boldsymbol{R}}=\tilde{\boldsymbol{R}}\,\boldsymbol{R}=\mathbf{1}$, where $\mathbf{1}$ is the unit matrix).

It should now be evident that if we start with the molecule in the principal orientation, where the atomic positions are given by the vectors $\vec{d}_i^{\,p}$, a rotation matrix can be constructed to rotate the molecule into any desired orientation, as defined by the vectors $\vec{d}_i$. In the process, the moment of inertia tensor changes from diagonal to non-diagonal. This means that the rotation matrix $\boldsymbol{R}$ effectively *defines* the

orientation of the molecule and, as we have seen, we can always calculate the rotation matrix for an arbitrary orientation by diagonalising the moment of inertia tensor. However, the full rotation matrix requires nine real numbers to specify fully, which is inconvenient and inefficient. A more economical way to specify the orientation was derived by Euler, who showed that any orientation can be defined by (at most) three parameters - the so called Euler angles, from which the rotation matrix can be subsequently constructed.

2.8 Molecular Orientation: The Euler Angles

Euler's angles are obtained by considering what happens to local coordinate axes when they are rotated with respect to the laboratory reference frame, the local axes being represented by unit vectors $\vec{i}, \vec{j}$ and $\vec{k}$ directed along the positive x, y and z directions respectively. We start with the molecule in the principal orientation, so the local axes are parallel with the laboratory axes. Then, in sequence, we apply three rotations about selected *local* axes, without using the same axis twice consecutively. The angles through which the turns are made constitute the Euler angles. There are twelve possible sequences for rotating three axes. The important point is that, once a particular sequence is chosen it must be adhered to as a convention and obeyed in all subsequent applications, otherwise serious inconsistencies arise. Throughout this book we use the so-called *x-convention,* which we now describe.

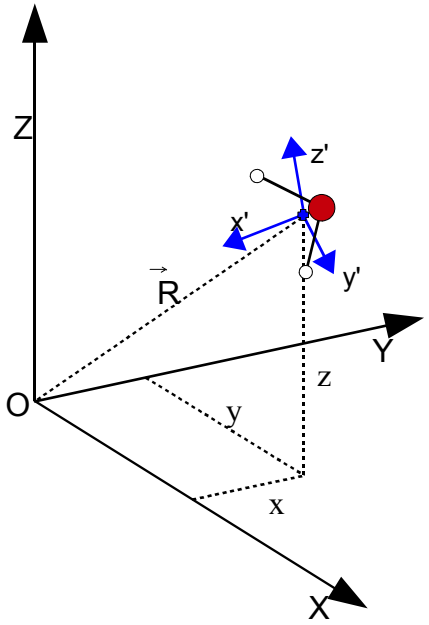

Figure 2.3: Molecular Orientation – Local and Laboratory Frames

We start with the molecule (which can be anywhere in the laboratory reference frame) in the principal orientation . Then a set of orthogonal, Cartesian coordinate axes are embedded in the molecule with the origin of all axes at the molecule's centre of mass. This is the so-called *local coordinate system*, or *local frame of reference*, which can be used to define the positions of the molecule's atoms with respect to the centre of

mass. If the molecule is rotated about its centre of mass, the local reference frame rotates with it, so the local atomic coordinates do not change and in consequence the matrix I remains diagonal in this frame, but the atomic coordinates with respect to the laboratory reference frame certainly do change, and the matrix I becomes non-diagonal. The circumstances are shown in figure 2.3, where the local axes of an example water molecule are rotated in the laboratory frame.

The x-convention sequence of rotations is shown in figure 2.4. In part (a) a rotation is made about the z axis vector $\vec{k}$ through an angle ϕ. The angle is positive if it represents a clockwise rotation when viewed in the direction of $\vec{k}$ along the axis of rotation. This rotation moves the x axis vector $\vec{i}$ to $\vec{i}'$ and the y axis vector $\vec{j}$ to $\vec{j}'$, but the z axis vector $\vec{k}$ is unchanged, so $\vec{k}'=\vec{k}$. Angle ϕ is thus the first Euler angle. Before the rotation the axes of the local and laboratory frames are the same, since the molecule is in the principal orientation. After the rotation they are no longer the same; the local axes have new directions given by $\vec{i}', \vec{j}', \vec{k}'$ in the laboratory frame. We can express the new vectors in terms of $\vec{i}, \vec{j}, \vec{k}$ as follows

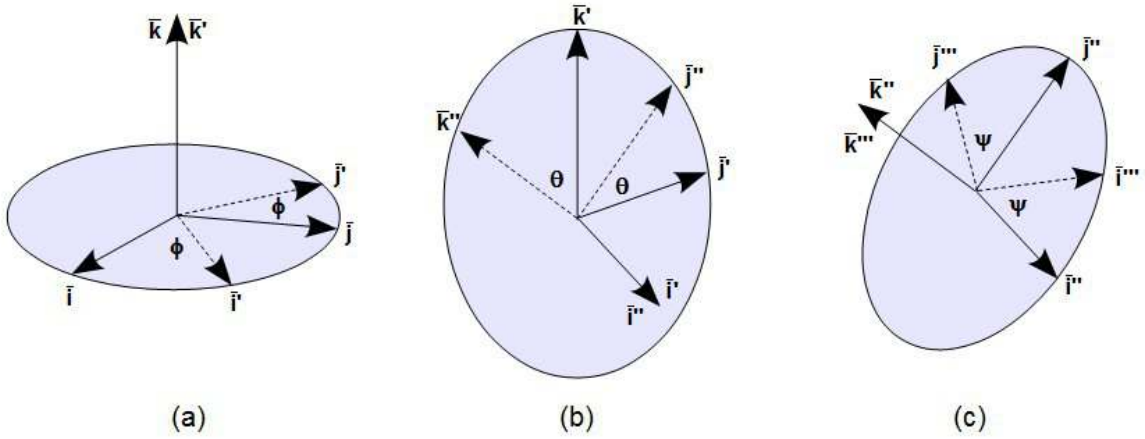

Figure 2.4: The Euler Angles in the x-Convention

$$\begin{aligned}\vec{i}' &= \vec{i}\cos\phi + \vec{j}\sin\phi,\\ \vec{j}' &= -\vec{i}\sin\phi + \vec{j}\cos\phi,\\ \vec{k}' &= \vec{k}.\end{aligned} \tag{2.58}$$

In part (b) a rotation through angle θ is made about the $\vec{i}'$ vector. Again, θ is positive if the angle of rotation is clockwise when viewed in the direction of vector $\vec{i}'$. Vector $\vec{j}'$ becomes vector $\vec{j}''$ and $\vec{k}'$ becomes $\vec{k}''$. Vector $\vec{i}'$ is unchanged by the rotation. Angle θ is the second Euler angle. This gives us a new set of

relations:

$$\begin{aligned}\vec{i}'' &= \vec{i}',\\ \vec{j}'' &= \vec{j}'\cos\theta + \vec{k}'\sin\theta,\\ \vec{k}'' &= -\vec{j}'\sin\theta + \vec{k}'\cos\theta.\end{aligned} \tag{2.59}$$

Lastly, in part (c) the rotation is through the angle ψ about the vector $\vec{k}''$ and is positive if clockwise when viewed along the vector direrction. Now $\vec{i}''$ becomes $\vec{i}'''$ and $\vec{j}''$ becomes $\vec{j}'''$, with $\vec{k}''$ unchanged. ψ is the final Euler angle and our final set of relations is

$$\begin{aligned}\vec{i}''' &= \vec{i}''\cos\psi + \vec{j}''\sin\psi,\\ \vec{j}''' &= -\vec{i}''\sin\psi + \vec{j}''\cos\psi,\\ \vec{k}''' &= \vec{k}''.\end{aligned} \tag{2.60}$$

The combined effect of these successive rotations is obtained by substituting equations (2.58) into equations (2.59) and then substituting the results into equations (2.60), which leads to

$$\begin{aligned}\vec{i}''' &= \vec{i}\,(\cos\psi\cos\phi - \sin\psi\cos\theta\sin\phi) + \vec{j}\,(\cos\psi\sin\phi + \sin\psi\cos\theta\cos\phi) + \vec{k}\sin\psi\sin\theta,\\ \vec{j}''' &= -\vec{i}\,(\sin\psi\cos\phi + \cos\psi\cos\theta\sin\phi) - \vec{j}\,(\sin\psi\sin\phi - \cos\psi\cos\theta\cos\phi) + \vec{k}\cos\psi\sin\theta\\ \vec{k}''' &= \vec{i}\sin\theta\sin\phi - \vec{j}\sin\theta\cos\phi + \vec{k}\cos\theta.\end{aligned} \tag{2.61}$$

This formula can be used to define *any* orientation of the rigid molecule, starting from the molecule in the principal frame and applying the appropriate Euler rotations.

To be clear what equation (2.61) means: it describes the *rotated local axes* of the molecule, represented by vectors $\vec{i}''', \vec{j}''', \vec{k}'''$, in terms of the *fixed laboratory axes* represented by the vectors $\vec{i}, \vec{j}, \vec{k}$ which, prior to the rotation, coincided with the principal axes of the molecule.

After the rotation (2.61) a vector $\vec{d}_i^{\,p}$ in the local frame of the molecule, with components $\left(d_i^{p\,x}, d_i^{p\,y}, d_i^{p\,z}\right)$ is represented in the laboratory frame as the vector $\vec{d}_i$ with components $\left(d_i^x, d_i^y, d_i^z\right)$. The relationship between the two is given by

$$d_i^x\vec{i} + d_i^y\vec{j} + d_i^z\vec{k} = d_i^{p\,x}\vec{i}''' + d_i^{p\,y}\vec{j}''' + d_i^{p\,z}\vec{k}'''. \tag{2.62}$$

Substituting into the right of this relation the equations (2.61) and isolating components of $\vec{i}, \vec{j}, \vec{k}$ gives

$$
\begin{aligned}
d_i^x &= (\cos\psi\cos\phi - \sin\psi\cos\theta\sin\phi)\, d_i^{p\,x} - (\sin\psi\cos\phi + \cos\psi\cos\theta\sin\phi)\, d_i^{p\,y} + \sin\theta\sin\phi\, d_i^{p\,z}, \\
d_i^y &= (\cos\psi\sin\phi + \sin\psi\cos\theta\cos\phi)\, d_i^{p\,x} - (\sin\psi\sin\phi - \cos\psi\cos\theta\cos\phi)\, d_i^{p\,y} - \sin\theta\cos\phi\, d_i^{p\,z}, \\
d_i^z &= \sin\theta\sin\psi\, d_i^{p\,x} + \sin\theta\cos\psi\, d_i^{p\,y} + \cos\theta\, d_i^{p\,z},
\end{aligned}
\tag{2.63}
$$

which in matrix form is

$$
\vec{d}_i = \boldsymbol{R}\, \vec{d}_i^{\,p}, \tag{2.64}
$$

where the components $R^{\alpha\beta}$ are to be determined from their correspondence with equation (2.63):

$$
\boldsymbol{R} = \begin{bmatrix}
(\cos\psi\cos\phi - \sin\psi\cos\theta\sin\phi), & -(\sin\psi\cos\phi + \cos\psi\cos\theta\sin\phi), & \sin\theta\sin\phi \\
(\cos\psi\sin\phi + \sin\psi\cos\theta\cos\phi), & -(\sin\psi\sin\phi - \cos\psi\cos\theta\cos\phi), & -\sin\theta\cos\phi \\
\sin\theta\sin\psi, & \sin\theta\cos\psi, & \cos\theta
\end{bmatrix}
\tag{2.65}
$$

Incidentally, the coefficients in (2.61) can also be written as

$$
\begin{aligned}
\vec{i}^{\,\prime\prime\prime} &= \vec{i}\, R^{xx} + \vec{j}\, R^{yx} + \vec{k}\, R^{zx}, \\
\vec{j}^{\,\prime\prime\prime} &= \vec{i}\, R^{xy} + \vec{j}\, R^{yy} + \vec{k}\, R^{zy}, \\
\vec{k}^{\,\prime\prime\prime} &= \vec{i}\, R^{xz} + \vec{j}\, R^{yz} + \vec{k}\, R^{zz}.
\end{aligned}
\tag{2.66}
$$

These results show that the rotation matrix we have derived using Euler's angles is the same as that defined by diagonalising the moment of inertia tensor. Furthermore, since we now know what the elements of the matrices $\boldsymbol{R}$ and $\tilde{\boldsymbol{R}}$ are, it is a straightforward matter to show that $\tilde{\boldsymbol{R}} = \boldsymbol{R}^{-1}$.

To summarise, this section has established the relationships between Euler's angles, the rotation matrix $\boldsymbol{R}$ and the local and laboratory reference frames. These relations provide a framework for calculating the dynamics of a rotating rigid molecule.

2.9 Euler's Rotational Equations of Motion for a Rigid Molecule

To obtain Euler's equations of motion we start by multiplying both sides of equation (2.51) by the rotation matrix $\tilde{\boldsymbol{R}}$ to give

$$
\tilde{\boldsymbol{R}}\, \vec{\tau} = \tilde{\boldsymbol{R}}\, \boldsymbol{I}\, \boldsymbol{R}\, \tilde{\boldsymbol{R}}\, \dot{\vec{\omega}} + \tilde{\boldsymbol{R}}\, \dot{\boldsymbol{I}}\, \boldsymbol{R}\, \tilde{\boldsymbol{R}}\, \vec{\omega}, \tag{2.67}
$$

where we have also inserted the harmless product $\boldsymbol{R}\,\tilde{\boldsymbol{R}} = \boldsymbol{1}$. The purpose of this operation is to cast all vectors and tensors into the principle frame, and so obtain

$$
\vec{\tau}^{\,p} = \boldsymbol{I}^{\,p}\, \dot{\vec{\omega}}^{\,p} + \dot{\boldsymbol{I}}^{\,p}\, \vec{\omega}^{\,p}, \tag{2.68}
$$

where the matrix $\boldsymbol{I}^p$ is of course diagonal. However it is important to note that the matrix $\boldsymbol{R}$ can only diagonalise the moment of inertia tensor at *one specific instant of time and* at an instant later, as the body rotates, $\boldsymbol{R}$ is no longer the correct matrix for this task. This means that, as the body moves, we need to keep updating $\boldsymbol{R}$ for the new orientation. We describe how to do this later. For now we simply note that in equation (2.68) the moment of inertia tensor is diagonal for an instant only.

To proceed further we need to process the matrix $\dot{\boldsymbol{I}}^p$. This is just the matrix $\dot{\boldsymbol{I}}$ represented in the principal frame. So taking equation (2.52) as our starting point we can write directly

$$\dot{I}^{p\alpha\beta} = -\sum_{i=1}^{N_a} m_i \left(d_i^{p\alpha} \dot{d}_i^{p\beta} + d_i^{p\beta} \dot{d}_i^{p\alpha} \right), \tag{2.69}$$

where $\dot{\vec{d}}_i^{\,p}$ is given by

$$\dot{\vec{d}}_i^{\,p} = \vec{\omega}^p \times \vec{d}_i^{\,p}. \tag{2.70}$$

Expanding (2.70) and substituting the result into (2.69) leads to a result similar to (2.53) except that the zero off-diagonal elements of $\boldsymbol{I}^p$ simplify the result, which is

$$\begin{aligned} \dot{I}^{pxy} = \dot{I}^{pyx} &= \omega^{pz} \left(I^{pxx} - I^{pyy} \right), \\ \dot{I}^{pyz} = \dot{I}^{pzy} &= \omega^{px} \left(I^{pyy} - I^{pzz} \right), \\ \dot{I}^{pzx} = \dot{I}^{pxz} &= \omega^{py} \left(I^{pzz} - I^{pxx} \right), \\ \dot{I}^{pxx} = \dot{I}^{pyy} &= \dot{I}^{pzz} = 0. \end{aligned} \tag{2.71}$$

Substituting this result into (2.68) and rearranging, finally gives Euler's rotational equations of motion,

$$\begin{aligned} \tau^{px} &= I^{pxx} \dot{\omega}^{px} - \omega^{py} \omega^{pz} \left(I^{pyy} - I^{pzz} \right), \\ \tau^{py} &= I^{pyy} \dot{\omega}^{py} - \omega^{pz} \omega^{px} \left(I^{pzz} - I^{pxx} \right), \\ \tau^{pz} &= I^{pzz} \dot{\omega}^{pz} - \omega^{px} \omega^{py} \left(I^{pxx} - I^{pyy} \right), \end{aligned} \tag{2.72}$$

which may be integrated numerically to update $\vec{\omega}^p$. Note however that this result is expressed in the local frame of reference in which $\boldsymbol{I}$ was instantaneously diagonal. In consequence a new rotation matrix must now be found, so that the new orientation of the local reference frame of the molecule can be defined. This problem is solved in terms of the Euler angles, as follows.

Angle ϕ represents a rotation about the vector $\vec{k}$, so an angular velocity $\vec{\omega}_\phi$ about $\vec{k}$ is given by

$$\vec{\omega}_{\phi} = \dot{\phi}\,\vec{k} \tag{2.73}$$

Likewise θ is a rotation about the vector $\vec{i}'$, so we have

$$\vec{\omega}_{\theta} = \dot{\theta}\,\vec{i}', \tag{2.74}$$

and finally, ψ is a rotation about $\vec{k}''$ and so

$$\vec{\omega}_{\psi} = \dot{\psi}\,\vec{k}''. \tag{2.75}$$

In general the angular velocity $\vec{\omega}$ can be represented as a vector sum of these:

$$\vec{\omega} = \vec{\omega}_{\phi} + \vec{\omega}_{\theta} + \vec{\omega}_{\psi} = \dot{\phi}\,\vec{k} + \dot{\theta}\,\vec{i}' + \dot{\psi}\,\vec{k}''. \tag{2.76}$$

We now express $\vec{k}, \vec{i}'$ and $\vec{k}''$ in (2.76) exclusively in terms of $\vec{i}''', \vec{j}''', \vec{k}'''$, defining the local frame. To do this we require the (inverses of) equations (2.58) to (2.60), which on substitution into (2.76) leads us to

$$\vec{\omega}^{p} = \vec{i}'''(\dot{\phi}\sin\theta\sin\psi + \dot{\theta}\cos\psi) + \vec{j}'''(\dot{\phi}\sin\theta\cos\psi - \dot{\theta}\sin\psi) + \vec{k}'''(\dot{\phi}\cos\theta + \dot{\psi}) \tag{2.77}$$

or more explicitly as the components

$$\begin{aligned} \omega^{p^x} &= \dot{\phi}\sin\theta\sin\psi + \dot{\theta}\cos\psi, \\ \omega^{p^y} &= \dot{\phi}\sin\theta\cos\psi - \dot{\theta}\sin\psi, \\ \omega^{p^z} &= \dot{\phi}\cos\theta + \dot{\psi}. \end{aligned} \tag{2.78}$$

We can now invert these equations to obtain

$$\begin{aligned} \dot{\phi} &= \omega^{p^x}\frac{\sin\psi}{\sin\theta} + \omega^{p^y}\frac{\cos\psi}{\sin\theta}, \\ \dot{\theta} &= \omega^{p^x}\cos\psi - \omega^{p^y}\sin\psi, \\ \dot{\psi} &= -\omega^{p^x}\frac{\sin\psi}{\tan\theta} - \omega^{p^y}\frac{\cos\psi}{\tan\theta} + \omega^{p^z}. \end{aligned} \tag{2.79}$$

These are the equations of motion of the Euler angles. The updated components of vector $\vec{\omega}^{p}$ obtained by integrating (2.72) are used together with the currently known Euler angles ϕ, θ, ψ in (2.79) to update the Euler angles to the new molecular orientation. Once these updated angles are obtained a new rotation matrix $\boldsymbol{R}$ can be constructed. This completes the prescription of Euler equations of motion.

Unfortunately, from the perspective of numerical integration, Euler's approach has a

problem, which can be seen in equations (2.79) where the denominators appearing in the first and last equations take a zero value if $\theta = 0.$ When this happens the equations become indeterminate and the numerical integration breaks down. To cure this problem Evans and Murad [7] introduced quaternions into molecular dynamics methodology. We shall discuss these next.

2.10 Quaternions and the Quaternion Equations of Motion

In the context of molecular dynamics quaternions can be thought of as an ordered set of four parameters $\breve{q} = (q_0, q_1, q_2, q_3)$ which define the orientation of a rigid body. In wider mathematics they are significant for having a unique algebra of their own, but they have been superseded by conventional vectors and are no longer much used. The advantage of using quaternion parameters to specify orientation is that the resulting equations of motion are free from the singularities that arise with Euler angles.

Quaternion parameters are defined in terms of the Euler angles ϕ, θ, ψ as follows

$$\begin{aligned} q_0 &= \cos(\theta/2)\cos((\phi+\psi)/2), \\ q_1 &= \sin(\theta/2)\cos((\phi-\psi)/2), \\ q_2 &= \sin(\theta/2)\sin((\phi-\psi)/2), \\ q_3 &= \cos(\theta/2)\sin((\phi+\psi)/2). \end{aligned} \tag{2.80}$$

Since, at most, only three parameters are needed to define molecular orientation there is a redundancy in using four and this requires that the four parameters have a conserved norm:

$$|\breve{q}|^2 = q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1. \tag{2.81}$$

The rotation matrix can expressed in terms of the quaternion parameters as

$$\boldsymbol{R} = \begin{bmatrix} (q_0^2+q_1^2-q_2^2-q_3^2) & 2(q_1q_2-q_0q_3) & 2(q_1q_3+q_0q_2) \\ 2(q_1q_2+q_0q_3) & (q_0^2-q_1^2+q_2^2-q_3^2) & 2(q_2q_3-q_0q_1) \\ 2(q_1q_3-q_0q_2) & 2(q_2q_3+q_0q_1) & (q_0^2-q_1^2-q_2^2+q_3^2) \end{bmatrix} \tag{2.82}$$

which can be proved by inserting the definitions of the parameters from (2.80) into (2.82) and comparing the result with the elements in matrix (2.65).

We require an equation of motion for the quaternions to replace the problematic equations (2.79). The form this takes is rather simple:

$$\begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & q_0 \end{bmatrix} \begin{bmatrix} 0 \\ \omega^{p}{}^{x} \\ \omega^{p}{}^{y} \\ \omega^{p}{}^{z} \end{bmatrix}, \quad \text{or} \quad \dot{\breve{q}} = \frac{1}{2}\boldsymbol{Q}\breve{\omega}^{p}, \tag{2.83}$$

where we have defined $\breve{\omega}^p$ as a four-component equivalent of $\vec{\omega}^p$. The inverse of this equation is

$$\begin{bmatrix} 0 \\ \omega^{p\,x} \\ \omega^{p\,y} \\ \omega^{p\,z} \end{bmatrix} = 2 \begin{bmatrix} q_0 & q_1 & q_2 & q_3 \\ -q_1 & q_0 & q_3 & -q_2 \\ -q_2 & -q_3 & q_0 & q_1 \\ -q_3 & q_2 & -q_1 & q_0 \end{bmatrix} \begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_2 \end{bmatrix}, \quad \text{or} \quad \breve{\omega}^p = 2\boldsymbol{Q}^{-1}\dot{\breve{q}} \tag{2.84}$$

from which it is apparent that matrix $\boldsymbol{Q}^{-1}$ is the transpose of matrix $\boldsymbol{Q}$ i.e. $\boldsymbol{Q}^{-1} = \tilde{\boldsymbol{Q}}$. Proof of (2.83) is tedious. It involves first expanding (2.80) into trigonometric functions of individual half-angles and taking the time derivatives to obtain relations between the components of $\dot{\breve{q}}$ and the derivatives $\dot{\phi}, \dot{\theta}, \dot{\psi}$. Then it is necessary to substitute in the relations (2.79), having also first expanded these into half-angle trigonometric functions. Then through a process of elimination and contraction, a 4×3 sub-matrix of $\boldsymbol{Q}$ is obtained that relates the four components of $\dot{\breve{q}}$ to the three components of $\vec{\omega}$. To complete the 4×4 matrix, the time derivative of the normalisation condition (2.81):

$$q_0\dot{q}_0 + q_1\dot{q}_1 + q_2\dot{q}_2 + q_3\dot{q}_3 = 0 \tag{2.85}$$

is incorporated to construct the first column of $\boldsymbol{Q}$. Relation (2.83) is simply the inverse of this result.

The equation of motion (2.83) contains no inherent singularities and in consequence enables a stable integration of $\dot{\breve{q}}$. All that is required are the current quaternion parameters $\breve{q}$ and the updated angular velocity $\vec{\omega}^p$. Once $\breve{q}$ has been updated, a new version of $\boldsymbol{R}$ can be calculated. Thus overall the method is an extension of Euler's approach, but avoiding problems with singularities.

2.11 The Kinetic Energy of a Rigid Molecule

The kinetic energy of a rigid molecule differs from that of a single atom in that it has rotational as well as translational components. In this section we provide expressions for these.

We start with the kinetic energy of a general molecule composed of N_a atoms:

$$K = \frac{1}{2}\sum_{i=1}^{N_a} m_i \dot{\vec{r}}_i \cdot \dot{\vec{r}}_i. \tag{2.86}$$

Inserting the time derivative of $\vec{d}_i = \vec{r}_i - \vec{R}$ from (2.38) leads to

$$K=\frac{1}{2}\sum_{i=1}^{N_a} m_i(\dot{\vec{R}}+\dot{\vec{d}}_i)\cdot(\dot{\vec{R}}+\dot{\vec{d}}_i), \tag{2.87}$$

which may be expanded to

$$K=\frac{1}{2}\sum_{i=1}^{N_a} m_i\dot{R}^2+\sum_{i=1}^{N_a} m_i\dot{\vec{R}}\cdot\dot{\vec{d}}_i+\frac{1}{2}\sum_{i=1}^{N_a} m_i\dot{\vec{d}}_i\cdot\dot{\vec{d}}_i. \tag{2.88}$$

In the first term on the right the sum over atomic masses returns the molecular mass M. The middle term sums to zero, since collectively the atoms in a molecule have no net momentum with respect to its centre of mass (i.e. $\sum_{i=1}^{N_a} m_i\dot{\vec{d}}_i=0$). In the third term we use the relation (2.43) which is appropriate for a rigid body to give:

$$K=\frac{1}{2}M\,\dot{R}^2+\frac{1}{2}\sum_{i=1}^{N_a} m_i\dot{\vec{d}}_i\cdot(\vec{\omega}\times\vec{d}_i). \tag{2.89}$$

The scalar triple product in the second term right of (2.89) can be rearranged:

$$K=\frac{1}{2}M\,\dot{R}^2+\frac{1}{2}\sum_{i=1}^{N_a} m_i\vec{\omega}\cdot(\vec{d}_i\times\dot{\vec{d}}_i). \tag{2.90}$$

Rearranging this gives

$$K=\frac{1}{2}M\,\dot{R}^2+\frac{1}{2}\vec{\omega}\cdot\sum_{i=1}^{N_a}(\vec{d}_i\times m_i\dot{\vec{d}}_i), \tag{2.91}$$

which, according to (2.37), is the same as

$$K=\frac{1}{2}M\,\dot{R}^2+\frac{1}{2}\vec{\omega}\cdot\vec{J} \tag{2.92}$$

and from (2.47) this finally becomes

$$K=\frac{1}{2}M\,\dot{R}^2+\frac{1}{2}\vec{\omega}\cdot\boldsymbol{I}\cdot\vec{\omega}. \tag{2.93}$$

The first term on the right of (2.93) we recognise as the *translational kinetic energy* (K_T) of the molecule's centre of mass. The second term is the *rotational kinetic energy* (K_R) of the molecule about the centre of mass. In the rigid molecule's local

reference frame, the rotational kinetic energy can be written as

$$K_R = \frac{1}{2}\left(I^{p\,xx}(\omega^{p\,x})^2 + I^{p\,yy}(\omega^{p\,y})^2 + I^{p\,zz}(\omega^{p\,z})^2 \right). \tag{2.94}$$

2.12 The Rotational Motion of Rigid Linear Molecules

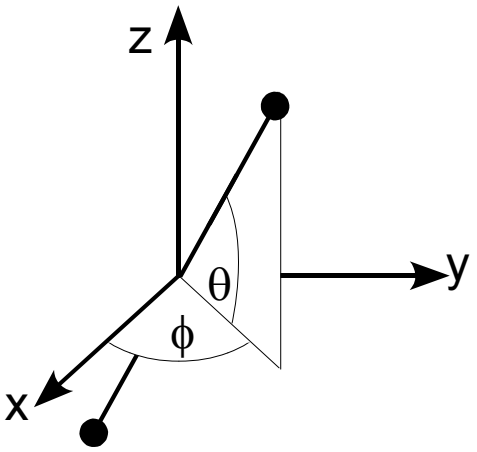

Figure 2.5. The Euler Angles for a Linear (Diatomic) Molecule

Linear molecules do not quite fit into the schemes described above. This is because they possess only two rotational degrees of freedom, a property that derives from the fact that their orientation in space is completely described by just two Euler angles (see figure 2.5). Since the molecules are rigid, they have no internal bond vibration and it is quite easy to derive Lagrange's or Hamilton's equations of motion for a rigid linear diatomic molecule moving in three dimensions, by adapting the two dimensional examples given in sections 2.4 and 2.5 to include the extra Euler angle and to exclude the vibrational dynamics of the bond. We will simply quote the final equations for these cases.

The Lagrangian equations of motion for coordinates $\vec{R}, \phi, \theta$ are

$$\begin{aligned} &\ddot{\vec{R}} = \vec{F}/M, \\ &\ddot{\phi} = 2\tan\theta\,\dot{\theta}\dot{\phi} + \tau^z/(\mu d^2\cos^2\theta), \\ &\ddot{\theta} = -\cos\theta\sin\theta\,\dot{\phi}^2 + (\tau^x\sin\phi - \tau^y\cos\phi)/(\mu d^2), \end{aligned} \tag{2.95}$$

and the Hamiltonian equations for coordinates $\vec{R}, \phi, \theta$ and momenta $\vec{P}, p_\phi, p_\theta$ are

$$\begin{aligned} &\dot{\vec{P}} = \vec{F}, && \dot{\vec{R}} = \vec{P}/M, \\ &\dot{p}_\phi = \tau^z, && \dot{\phi} = p_\phi/(\mu d^2\cos^2\theta), \\ &\dot{p}_\theta = -\mu d^2\sin\theta\cos\theta\,\dot{\phi}^2 + (\tau^x\sin\phi - \tau^y\cos\phi), && \dot{\theta} = p_\theta/(\mu d^2). \end{aligned} \tag{2.96}$$

In these equations M is the total molecular mass $(m_1+m_2),$ with m_1, m_2 defining the atomic masses. The vector $\ddot{\vec{R}}$ is the acceleration of the centre of mass vector $\vec{R}$ which is defined by the relation

$$\vec{R}=M^{-1}(m_1\vec{r}_1+m_2\vec{r}_2), \tag{2.97}$$

where vectors $\vec{r}_1, \vec{r}_2$ define the atomic positions. The force $\vec{F}=\vec{f}_1+\vec{f}_2$ is the vector sum of all the atomic forces $\vec{f}_1, \vec{f}_2$ acting on the molecule. Distance $d=|\vec{d}|=|\vec{r}_2-\vec{r}_1|$ is the length of the molecule. The reduced mass μ is defined by the relation

$$\frac{1}{\mu}=\frac{1}{m_1}+\frac{1}{m_2}. \tag{2.98}$$

The parameters (τ^x, τ^y, τ^z) are the components of the torque vector $\vec{\tau}$ which, like the force $\vec{F}$ is a sum of the individual torque vectors associated with each atom. i.e. $\vec{\tau}=\vec{\tau}_1+\vec{\tau}_2$ where

$$\vec{\tau}_1=\vec{d}_1\times\vec{f}_1, \qquad \vec{\tau}_2=\vec{d}_2\times\vec{f}_2, \tag{2.99}$$

and $\vec{d}_1$ and $\vec{d}_2$ locate the atoms with respect to the molecular centre of mass i.e.

$$\vec{d}_1=-\frac{\mu}{m_1}\vec{d}, \qquad \vec{d}_2=\frac{\mu}{m_2}\vec{d}. \tag{2.100}$$

It is also useful to note the relationship between the atomic positions and these vectors:

$$\vec{r}_i=\vec{R}+\vec{d}_i. \tag{2.101}$$

It is easy to generalize equations (2.95) through to (2.101) for a rigid polyatomic linear molecule composed of (say) N_a atoms. In (2.95) and (2.96) we simply replace the quantity $\mu d^2,$ where it appears, by the *scalar moment of inertia* $I,$ which is defined by

$$I=\sum_{i=1}^{N_a} m_i d_i^2 \tag{2.102}$$

where d_i is the magnitude of the vector $\vec{d}_i$ that locates the $i'th$ atom along the molecular axis (defined by unit vector $\vec{e}$), with respect to the molecular centre of mass:

$$\vec{d}_i = d_i \vec{e}. \tag{2.103}$$

(Note: in this formula d_i may be either positive or negative.) The molecular mass M and centre of mass $\vec{R}$ are as defined in equations (2.28) and (2.29). The force $\vec{F}$ and torque $\vec{\tau}$ are the vector sums of all the atomic forces and torques the molecule experiences

$$\vec{F} = \sum_{i=1}^{N_a} \vec{f}_i, \qquad \vec{\tau} = \sum_{i=1}^{N_a} \vec{\tau}_i. \tag{2.104}$$

The atomic torques being calculated in a manner identical to equations (2.99).

Unfortunately, equations (2.95) and (2.96) are problematic for numerical integration. In both sets the equation of motion for ϕ becomes indeterminate when the $\cos^2\theta$ term in the denominator approaches zero, which can easily arise. For this reason we will take another approach that does not use Euler angles. We will derive equations of motion based on the unit vector $\vec{e}$ that lies along the molecular axis and treat it in a Cartesian fashion. Vector $\vec{e}$ will effectively define the molecular orientation, though it has three components, not two. The redundancy in the parameter set is nullified by the fact that $\vec{e}$ is a vector of unit length – a fact we must keep in mind at all times.

We will use a Lagrangian description. First we specify the kinetic energy K for a linear rigid molecule composed of N_a atoms. In Cartesian coordinates this is

$$K = \frac{1}{2} \sum_{i=1}^{N_a} m_i \dot{r}_i^2. \tag{2.105}$$

The velocity $\dot{\vec{r}}_i$ is obtained from the time derivatives of (2.101) and (2.103), which gives

$$\dot{\vec{r}}_i = \dot{\vec{R}} + d_i \dot{\vec{e}} \tag{2.106}$$

and on inserting this into (2.105) we obtain

$$K = \frac{1}{2} M \dot{R}^2 + \frac{1}{2} I \dot{e}^2 \tag{2.107}$$

where the moment of inertia I is defined in (2.102).

The potential energy Φ we define as

$$\Phi=\Phi(\{\vec{r}_i\}). \tag{2.108}$$

Here $\Phi(\{\vec{r}_i\})$ represents the normal system potential, which is the sum of the potentials of each atom of the molecule at its particular location. In general this is an inter-molecular potential generated by the presence of other molecules, but we need not be specific about the locations of these molecules on this occasion.

The Lagrangian for the system is $L=K-\Phi$ and we may use this in Lagrange's equations (2.10) to obtain the required equations of motion. Given the above definitions of K and Φ these are easily shown to be

$$\begin{aligned} M\ddot{\vec{R}}&=-\frac{\partial\Phi(\{\vec{r}_i\})}{\partial\vec{R}}=-\sum_{i=1}^{N_a}\frac{\partial\Phi(\{\vec{r}_i\})}{\partial\vec{r}_i}=\sum_{i=1}^{N_a}\vec{f}_i=\vec{F}\,,\\ I\ddot{\vec{e}}&=-\frac{\partial\Phi(\{\vec{r}_i\})}{\partial\vec{e}}=-\sum_{i=1}^{N_a}\frac{\partial\Phi(\{\vec{r}_i\})}{\partial\vec{r}_i}d_i=\sum_{i=1}^{N_a}d_i\vec{f}_i. \end{aligned} \tag{2.109}$$

The first of these equations is familiar and is readily integrable. The second effectively describes the rotational motion of the molecule, but we must be careful as it also contains components that tend to change the length of the vector $\vec{e}$, which is not permitted. These disruptive components are legitimate in the dynamical sense – they are a manifestation of the differences in the forces acting on the individual atoms, which act to alter the molecular length and there are also centrifugal forces inherent in the spinning molecule which have a stretching effect. To avoid these effects we must amend the equation to nullify all influences acting along the molecular axis. This is easily done by taking the vector product of $\vec{e}$ with both sides of the equation to give

$$I\,\vec{e}\times\ddot{\vec{e}}=\sum_{i=1}^{N_a}d_i\vec{e}\times\vec{f}_i. \tag{2.110}$$

The term on the right of (2.110) is merely a sum of atomic torques. The effect of performing this operation is to remove any contributions to the acceleration $\ddot{\vec{e}}$ that act along the direction of $\vec{e}$. Thus all changes to $\vec{e}$ are perpendicular to its direction, which is actually an implicit requirement for any constraint. We now note the identity

$$\dot{\vec{\omega}}=\frac{d}{dt}(\vec{e}\times\dot{\vec{e}})=\vec{e}\times\ddot{\vec{e}} \tag{2.111}$$

where $\dot{\vec{\omega}}$ is the derivative of the angular velocity $\vec{\omega}$, which is defined as

$$\vec{\omega}=\vec{e}\times\dot{\vec{e}}. \tag{2.112}$$

Thus (2.110) may be written as

$$I\dot{\vec{\omega}}=\sum_{i=1}^{N_a}\vec{\tau}_i=\vec{\tau}\,. \tag{2.113}$$

Integration of equation (2.113) provides the angular velocity $\vec{\omega}$ and we may then attempt to obtain $\vec{e}$ by integrating

$$\dot{\vec{e}}=\vec{\omega}\times\vec{e}\,, \tag{2.114}$$

but as was mentioned in section 2.6 (when discussing the direct integration of the rotational motion for a general rigid molecule), this type of equation must be solved exactly, not by some numerically approximate method. However a method is provided in section 4.6.1.2 , where the numerical integration of such equations is discussed. The method outlined there is fully viable for handling the rotational dynamics of rigid linear molecules.

Chapter 3

Statistical Mechanics

3.1 Introduction

In laboratory experiments we frequently measure the physical properties of material systems such as temperature (T), pressure (P), internal energy (U) and other *static thermodynamic properties*. Alternatively, we might look at *time dependent properties* such as diffusion, viscosity, thermal or electrical conductivity and determine the corresponding *transport coefficients* that quantify these properties. All these properties are, of course, dependent on the component molecules that form the bulk material, but the way in which this dependence is realised is not immediately apparent. The ultimate purpose of molecular dynamics is to provide an understanding of this dependence.

But as has already been mentioned, calculating the trajectories of many atoms is not, in itself, very useful if all it generates is a vast volume of data. Tools for compressing and analysing the data are essential and statistical mechanics provides them. It is the crucial bridge between the simulated *microscopic* system and the experimental *macroscopic* system. It turns out that almost all the interesting properties of a system are calculated as *averages* of the simulation data and it follows that the *variance* (or *mean-squared fluctuation)* associated with the averaging process is a measure of the statistical error. The fluctuation also turns out to have thermodynamic significance and to be an important factor in defining, and indeed *driving*, time dependent properties.

In practice, the calculation of an average requires that the system is in some sort of stable state i.e. one in which none of its characteristic variables is changing systematically (though it may perhaps be fluctuating around the mean value). This state corresponds to *thermodynamic equilibrium*. It is difficult to define this state in a molecular dynamics simulation, but the maintenance of stable averages for a number of significant properties is taken as a sign of this, if not actually a proof that the equilibrium exists.

Important questions arising at this stage are: what does taking an average actually mean from a physical point of view and how is the average to be obtained? There are two interpretations of what the averaging process means, one is due to Boltzmann and the other to Gibbs. These offer quite different viewpoints on the issue, though they are expected to yield the same results according to a principle called the *Ergodic Hypothesis*. These matters are discussed below. With regard to the calculation of averages, molecular dynamics more or less obliges us to take a Boltzmann view.

3.2 Boltzmann Averaging

In Boltzmann's approach to statistical mechanics, the average of a physical property is regarded as a *time* average. A property $A(t)$ is obtained as a function of time (t) ranging from $t=0$ to $t=T,$ where T is assumed large, and the average value $\bar{A}$ is calculated as an integral

$$\bar{A}=\frac{1}{T\to\infty}\int_0^T A(t)\,dt. \tag{3.1}$$

In a similar way, the variance is calculated as

$$(\delta\bar{A}^2)=\frac{1}{T\to\infty}\int_0^T \left(A(t)-\bar{A}\right)^2 dt. \tag{3.2}$$

The quantity $\left(A(t)-\bar{A}\right)$ in (3.2) is the *deviation* of the instantaneous value $A(t)$ from the average $\bar{A}$. The average of all such deviations is necessarily zero, so taking the square of the deviation guarantees a non-zero and positive result that is a measure of the scatter of the instantaneous values of $A(t)$ around the average.

In molecular dynamics the integral (3.1) is expressed as a numerical approximation:

$$\bar{A}\approx\frac{1}{T_\infty}\sum_{n=1}^{N_S} A_n\Delta t=\frac{1}{N_S}\sum_{n=1}^{N_S} A_n \tag{3.3}$$

in which A_n is the value of the property A taken at the $n'th$ time point in the data, N_S is the number of such points in the data, Δt is the time interval between points and $T_\infty=N_S\Delta t.$ Thus the property $A(t)$ is *sampled* N_S times at intervals of Δt and averaged.

We should define what we mean by *sampling*. Sampling involves taking an atomic configuration from the molecular dynamics trajectory at a time $t=n\Delta t$ and using it to calculate the property of interest (A_n) as a function of the coordinates of the configuration $\left(\{x_i\},\{\dot{x}_i\}\right)$ and adding the value A_n to the sum in (3.3).

The numerical approximation for (3.2) is correspondingly

$$\left(\delta\bar{A}^2\right)\approx\frac{1}{T_\infty}\sum_{n=1}^{N_S}\left(A_n-\bar{A}\right)^2\Delta t=\frac{1}{N_S}\sum_{n=1}^{N_S}\left(A_n-\bar{A}\right)^2. \tag{3.4}$$

Calculating the averages (3.3) and (3.4) is straightforward provided sufficient thought is given to arithmetic precision. Both formulas imply that, computationally, all N_S

values need to be stored for an *a posteriori* calculation of $\bar{A}$ but this is not the case. If the values of $\bar{A}$ and $\left(\delta \bar{A}^2\right)$ up to the $n'th$ data point are known (i.e. $\bar{A}_n$ and $\left(\delta \bar{A}_n^2\right)$) the $(n+1)'th$ values are given by

$$\left(\delta \bar{A}_{n+1}^2\right)=\frac{n}{n+1}\left\{\left(\delta \bar{A}_n^2\right)+\frac{1}{n+1}\left(A_{n+1}-\bar{A}_n\right)^2\right\} \quad \text{(a)}$$

$$\bar{A}_{n+1}=\frac{n}{n+1}\bar{A}_n+\frac{1}{n+1}A_{n+1}. \quad \text{(b)} \qquad (3.5)$$

Starting with $\bar{A}_1=A_1$ and $(\delta \bar{A}_1^2)=0$ these formulas allow all subsequent values to be calculated with high accuracy and without storing all previous values of A_n.

3.3 Gibbs Averaging

Gibbs' approach to statistical mechanics is radically different from Boltzmann's. Instead of sampling the same system at different moments in time, Gibbs' sampling may be regarded as a sampling of many replicas of the same system at the same time. Each system replica has identical chemical contents and is in the same thermodynamic state as the others, but it is assumed that each replica is a different realisation of that state, meaning that its microscopic configuration is different. As a result, despite being formally in the same state, each property of a replica may deviate (i.e. fluctuate) from the average value. The set of equivalent replicas is referred to as an *ensemble*, and the averaging process as an *ensemble average*. The ensemble average of a property A
is given by

$$\langle A\rangle=\frac{1}{N_s}\sum_{n=1}^{N_s}A_n, \qquad (3.6)$$

and the variance by

$$\left\langle \delta A^2\right\rangle=\frac{1}{N_s}\sum_{n=1}^{N_s}\left(A_n-\langle A\rangle\right)^2. \qquad (3.7)$$

It is evident that these formulas closely resemble the Boltzmann equivalents, but the interpretation is is very different. The configurations sampled in this case are not related by time progression. The main advantage of Gibbs' approach is that it provides a basis for statistical mechanics that is free of unsound assumptions, the most important of which is the assumption that the data points used in the averaging are completely independent of each other, which is to say that they are *uncorrelated* by definition. This is a difficult claim to make in Boltzmann's approach, where each configuration is deterministically derived from the one preceding it. It follows that Gibbs' approach provides the more rigorous approach to the averaging process.

3.4 The Ergodic Hypothesis

The Gibbs and Boltzmann approaches to statistical mechanics look very different and we should worry that they may give rise to different quantitative accounts of the same phenomena. However, it is reasonable to expect that they are equivalent provided that certain criteria are met. This assumed equivalence is called the *Ergodic Hypothesis*.

What might the qualifying criteria be? Firstly, we should ensure that enough sampling of the systems is done. Numerically this means that N_S, which appears in the Boltzmann sampling equations (3.3) and (3.4) and the Gibbs sampling equations (3.6) and (3.7), is large (ideally tending to infinity). Secondly we should ensure that all the samples taken are *statistically independent*. This means there is no *correlation* between the data points to bias the results. In the Gibbs approach this is satisfied by definition, as the many replicas of the system are defined as independent. In the Boltzmann approach this is not so easy to establish. The best we can do is ensure that the samples taken are widely separated in time, so that any causal connection between the samples is effectively lost. In practice this approach seems to work and tests can be devised to establish that no significant correlation exists in the data. But the fact remains that successive configurations of the system are generated by dynamical equations and, by definition, are causally related.

Some escape from this dilemma is afforded by the fact that molecular dynamics, which is a practical realisation of Boltzmann's approach, offers only an approximate solution to the equations of motion, so the causality thread running through the trajectories is compromised and correlation is inevitably lost after a while. However, this is a dispiriting argument as it implies that the sampling becomes ideal only at the point where the trajectories are manifestly wrong! A more optimistic assumption is that molecular systems are *dynamically chaotic* and the bulk system loses correlation at a faster rate than the integration algorithms lose contact with the true trajectories of the constituent molecules. There is clearly a lot of hand waving going on here but hopefully we are swimming rather than drowning. As mentioned above, practical methods for removing correlation from the data seem to work and provided we employ them properly, we may be confident that we obtaining meaningful results. There is an important lesson here: since molecular dynamics is a practical realisation of the Boltzmann approach, we must *always* be mindful of possible correlation in the output data and take steps to ensure that it does not prejudice the calculations of the averages and variance, which are the primary results of a simulation.

Throughout this book we will mostly be taking a Gibbs view of the statistical mechanics and assume that the Ergodic Hypothesis is universally valid. But at all times the reader should remember that molecular dynamics has all the pitfalls of Boltzmann averaging.

3.5 Working with Averages

We present here some sample derivations, with useful results, to provide a demonstration of how ensemble averaging works in practice.

3.5.1 Some Simple Properties of Averages

i. The average of the fluctuations is zero. We implied above that this is obvious. Maybe it isn't, so here is the proof.

$$\langle A-\langle A\rangle\rangle=\frac{1}{N_s}\sum_{n=0}^{N_s}\left(A_n-\langle A\rangle\right)=\langle A\rangle-\langle A\rangle=0 \tag{3.8}$$

ii. The variance is the average of the squares minus the square of the average. This is a useful relationship theoretically, though it is not very useful numerically, since it has problems with numerical precision.

$$\left\langle\left(A-\langle A\rangle\right)^2\right\rangle=\frac{1}{N_S}\sum_{n=0}^{N_s}\left(A_n^2-2\mathrm{A}_n\langle A\rangle+\langle A\rangle^2\right)=\langle A^2\rangle-2\langle A\rangle^2+\langle A\rangle^2=\langle A^2\rangle-\langle A\rangle^2 \tag{3.9}$$

3.5.2 The One Dimensional Random Walk

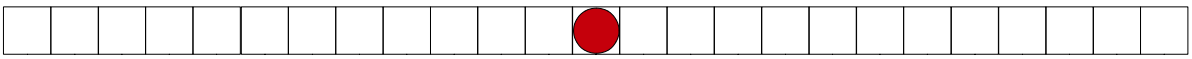

Figure 3.1: The 1D Random Walker

In figure 3.1 we have a linear array of identical cells, the central one of which is occupied by a counter. At regular moments in time the counter is moved at random one cell to the left or right. At some lengthy period later we might expect the counter to be somewhere to the right or left of the centre, though we have no way of knowing precisely where. This is a one dimensional random walker, a common model in statistical mechanics.

Though it is not possible to predict exactly where the counter will be after, say, n moves, it is certainly possible to calculate an average position. To do this we imagine a very large number of replicas of this system (the ensemble), all of which start with the counter in the central cell and each of which follows its own random sequence of n moves (the trajectory). We may use ensemble averaging to determine the outcome.

We define the position (x_n) of a counter after n moves as

$$x_n = \sum_{i=1}^{n} m_i \qquad \text{where} \quad m_i = \pm d. \tag{3.10}$$

Variable d is the cell width (i.e. the "stepping distance"). The ensemble average position $\langle x_n \rangle$ is given by

$$\langle x_n \rangle = \left\langle \sum_{i=1}^{n} m_i \right\rangle = \sum_{i=1}^{n} \langle m_i \rangle = 0. \tag{3.11}$$

Thus the average position, taken over the whole ensemble, is the origin. This is inevitable because the quantity $\langle m_i \rangle$ is effectively a sum of random positive and negative moves $(\pm d)$. Note that this does not mean we expect all (or indeed many) of the replicas to be found at the origin at the end of the experiment. It means that there will be a symmetrical distribution of counters either side of the origin so that 50% of them will be found on the right ("positive" displacement) and 50% on the left ("negative" displacement). The average displacement for the ensemble is thus zero.

Likewise we may calculate the mean-squared position:

$$\begin{aligned} \langle x_n^2 \rangle &= \left\langle \left(\sum_{i=1}^{n} m_i \right)^2 \right\rangle = \left\langle \sum_{i=1}^{n} \sum_{j=1}^{n} m_i m_j \right\rangle = \left\langle \sum_{i=1}^{n} m_i^2 \right\rangle + \left\langle \sum_{i=1}^{n} \sum_{j \neq i}^{n} m_i m_j \right\rangle \\ &= \sum_{i=1}^{n} \langle m_i^2 \rangle + \sum_{i=1}^{n} \sum_{j \neq i}^{n} \langle m_i m_j \rangle = nd^2 + 0. \end{aligned} \tag{3.12}$$

This result follows from the fact that m_i^2 must be the positive quantity $d^2,$ and $\langle m_i m_j \rangle$ is again a sum of randomly positive and negative numbers $(\pm d^2)$ and must be zero in the large n limit. The result shows that the mean-squared displacement of an ensemble of one dimensional random walkers is linear with the number of steps, and if each step were made at regular intervals in time, then it would be linear in time. Linear dependence also arises in two and three dimensional random walkers.

The importance of the random walker in statistical mechanics is that it provides a model for diffusion. In very many systems, the mean-squared displacement (or MSD) is linear with time, although this may not be apparent at short time. An important observation is that a linear MSD implies the diffusion arises from a random walk behaviour and not from deterministic dynamics. We discuss the MSD in a molecular context in section 3.6 .

3.5.3 The Clausius Virial Theorem

The pressure is an important thermodynamic variable in all systems and an ability to calculate it in molecular dynamics is a major requirement. The virial theorem of

Clausius allows us to do this, by introducing an important property, the virial, which is central to the calculation of pressure.

Figure 3.2 represents a (3D) cubic, hard-walled box of side L with N spheres (representing molecules) in continual motion within. The size of the spheres draw is much exaggerated for clarity; in reality they are considered to be negligibly small in relation to the size of the box. Collisions between molecules and with the walls are fully elastic, so that no energy is ever lost from the system.

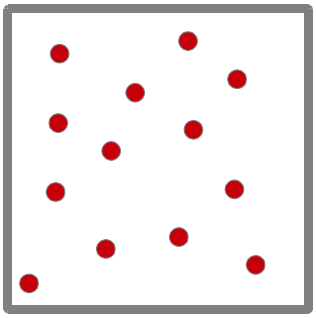

Figure 3.2: Model Hard-walled System

We now imagine an ensemble of such systems and attempt to calculate time derivative of the following ensemble average, which is based on the scalar product of each atoms' position with its momentum.

$$\frac{d}{dt}\left\langle \sum_{i=1}^{N} m_i \vec{r}_i \cdot \dot{\vec{r}}_i \right\rangle = \left\langle \sum_{i=1}^{N} m_i \dot{\vec{r}}_i \cdot \dot{\vec{r}}_i \right\rangle + \left\langle \sum_{i=1}^{N} m_i \vec{r}_i \cdot \ddot{\vec{r}}_i \right\rangle, \tag{3.13}$$

The first term on the right is evidently twice the system kinetic energy, while the second term is related to the particle acceleration $\ddot{\vec{r}}_i$ and therefore the force $\vec{f}_i$. So (3.13) becomes

$$\frac{d}{dt}\left\langle \sum_{i=1}^{N} m_i \vec{r}_i \cdot \dot{\vec{r}}_i \right\rangle = 2K + \left\langle \sum_{i=1}^{N} \vec{r}_i \cdot \vec{f}_i \right\rangle, \tag{3.14}$$

We now split the force into contributions that arise from the inter-atomic forces $\left(\vec{f}_i^{internal}\right)$, and those that arise from collisions with the walls $\left(\vec{f}_i^{external}\right)$. Equation (3.13) is now

$$\frac{d}{dt}\left\langle \sum_{i=1}^{N} m_i \vec{r}_i \cdot \dot{\vec{r}}_i \right\rangle = 2\mathrm{K} + \left\langle \sum_{i=1}^{N} \vec{r}_i \cdot \vec{f}_i^{internal} \right\rangle + \left\langle \sum_{i=1}^{N} \vec{r}_i \cdot \vec{f}_i^{external} \right\rangle, \tag{3.15}$$

and recognising that the external force only acts when the molecules collide with the walls allows us to write this as

$$\frac{d}{dt}\left\langle \sum_{i=1}^{N} m_i \vec{r}_i \cdot \dot{\vec{r}}_i \right\rangle = 2\mathrm{K} - \Psi - 3\mathrm{LPA}, \tag{3.16}$$

where Ψ is a physical property known as the *virial*[3], which is therefore defined as:

$$\Psi = -\left\langle \sum_{i=1}^{N} \vec{r}_i \cdot \vec{f}_i^{internal} \right\rangle. \tag{3.17}$$

In (3.16) we have used the fact that average force between a wall and the colliding molecules is equal to the pressure P multiplied by the area A of each wall acting *inwards*, and that the molecule's position when the collision takes place is $\pm L/2$ (assuming the coordinates $\vec{r}_i$ originate at the centre of the box). The term $3\mathrm{LPA}$ arises from the six faces of the cube. This term may be written as $3PV$ where V is the volume of the box.

Returning to the left side of equation (3.13) we note that there should be no correlation between the position of a molecule and its momentum (except at the wall where its momentum perpendicular to the face is anyway zero) so the scalar product contributing to the ensemble average can be both positive and negative in sign (or zero at the wall). The ensemble average is therefore zero, which means that

$$0 = 2\mathrm{K} - \Psi - 3\mathrm{PV}, \tag{3.18}$$

or, after rearrangement,

$$P = \left(2\mathrm{K} - \Psi\right)/3\mathrm{V}. \tag{3.19}$$

The virial obtained in this case applies to molecules confined within a hard-walled box and is not suitable for calculating the pressure in systems with periodic boundaries. A form which *is* suitable can be obtained by replacing the force $\vec{f}_i^{internal}$ in (3.17) by a sum of inter-atomic forces as shown in equation (1.5) (which applies to pair forces only) and rearranging to give the virial in the form shown in (1.12), where we are able to apply the minimum image convention to the inter-atomic separation $\vec{r}_{ij}$ and thus calculate the virial correctly in a system with a periodic boundary. Fortunately, the formula (3.19) still applies. Similar modifications are possible for systems with higher order forces. The important point is that the virial must be expressed in a form that is independent of the choice of the system centre of coordinates, which is not the case for (3.17). In chapter 7 we shall derive expressions for the virial which are based on thermodynamic methods and avoid this difficulty.

3 Note the virial here and elsewhere in this book is defined as the *negative* of an ensemble average of the scalar product of atomic positions and forces.

3.6 The Mean-squared Displacement

The mean-squared displacement is the quantitative expression of the phenomenon of *diffusion*. It is a process by which atoms located at some position at a given time are found to be in a different position at some time later as the result of random molecular collisions. It is a property of all phases of matter, though in solids it is usually orders of magnitude less than in liquids and gases. To calculate the MSD it is necessary to mark the position of an atom in space at a time zero, $\vec{r}_i(0),$ then at some time t later mark its current position, $\vec{r}_i(t),$ the square of the distance (i.e. displacement) between these two locations is then recorded. The average of the squared displacement, for all atoms in the system at time $t,$ and for all systems in the ensemble, is the mean-squared displacement. Formally this is given by the formula

$$\left\langle r^2(t)\right\rangle = \left\langle \frac{1}{N}\sum_{i=1}^{N}\left|\vec{r}_i(t)-\vec{r}_i(0)\right|^2 \right\rangle, \tag{3.20}$$

where N is the number of atoms in the system. The mean-squared displacement is directly related to experimental diffusion via the Einstein relation:

$$\left\langle r^2(t)\right\rangle = 6\mathrm{Dt} + C. \tag{3.21}$$

This relation, which applies when t is large enough, shows that at long time the MSD is a linear function of time, and the gradient of this function is related the diffusion coefficient $D.$ At short times the MSD is *not* linear in time, and this sometimes shows up in simulation results. For a noble gas system t needs to be typically of order one *picosecond* for linearity to set in. Also, care needs to be taken in how the average (3.20) is calculated. It is formally correct in the Gibbs view, but not necessarily easy to obtain very accurately in molecular dynamics unless the number of atoms N is large and a technique using many time origins is employed.

3.7 Fluctuations

Fluctuations are deviations from the mean values of system properties and are a natural property of all molecular systems. In the Boltzmann approach these appear as momentary deviations from the system average, while in the Gibbs approach they appear as the differences between individual ensemble members. They arise because every Boltzmann time-sampled configuration, or Gibbs sampled ensemble member, has a different atomic arrangement and thus a different partition of the system energy between the component molecules of the system. This would be the case even if the total energy of each configuration was the same. So if individual energy components, such as kinetic or potential energy, were being monitored, these components would inevitably fluctuate.

A simple example of this is presented by the simple harmonic oscillator. During a single oscillation, the kinetic and potential energies vary as shown in figure 3.3, but the sum of the two remains constant. A system composed of many harmonic oscillators vibrating with different frequencies and phases (a model to which most molecular systems approximate), will show a fluctuating total kinetic energy and a fluctuating total potential energy, though overall the system energy will remain constant.

The inevitability of fluctuations does not mean that all properties of a system can fluctuate. Many systems satisfy *conservation laws*, which fix the value of certain properties. For example the laws of thermodynamics demand that energy is conserved and this means that the total energy in an isolated system (i.e. one free of external influences) should stay constant. The total momentum of the system is similarly conserved. The constancy of properties like these provide useful checks on the correctness of molecular dynamics simulations, though it is important to note that numerical approximations inherent in the methodology, together with the limited numerical precision of computers, may give rise to (hopefully negligible) artificial fluctuations.

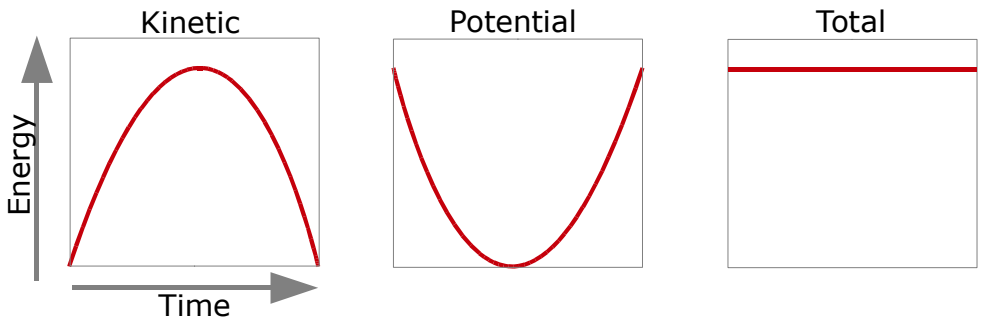

Figure 3.3: Energy Exchange in the Harmonic Oscillator

3.8 Correlation and Correlation Functions

Two sampled variables are said to be correlated if a given value of one of them implies the value of the other i.e. they are somehow *connected*. The variables can be of the same type, or of different types, and sampled at different points in time or space. Random variables (also known as stochastic variables) are by definition uncorrelated. One the other hand, variables which have a causal connection usually exhibit a degree of correlation, since the second may be calculated once the first is known. Correlation implies a causal connection, but this need not be the case. For example, both variables may be causally related to a third variable, but not to each other. In molecular systems correlation tends to decay with time or distance, except perhaps in idealised systems. The decay behaviour is described by a *correlation function* which is characterised by a *correlation time* or *correlation distance*, as appropriate. Molecular systems exhibit both correlated and random behaviours simultaneously, with the correlated behaviour dominating at short times or distances and random behaviour dominating at longer times or larger distances.

Mathematically, a correlation function $C(t)$ of two properties $A(t)$ and $B(t)$, both functions of t (which may be time or distance) defined on an interval $[0, T]$, is

represented by the integral

$$C(t)=\frac{1}{T-t}\int_0^{T-t}\left(A(t+u)-\bar{A}\right)\left(B(u)-\bar{B}\right)du, \qquad (3.22)$$

where $\bar{A}$ and $\bar{B}$ are the average values of $A(t)$ and $B(t)$ over the range $[0,T]$ and $t<T$ It will be noticed that the integrand is a product of the *fluctuations* in properties A and B (rather than their values) taken with arguments differing by the interval $t.$ The integral is an average of this product over the range $[0,T-t]$ with the value of t held constant. Properties $A(t)$ and $B(t)$ may of course be the same, in which case the integral represents an *auto-correlation* function, and when they are different it represents a *cross-correlation* function.

Fluctuations are used in the calculation because the variation in properties A and B as a function of t is entirely described by them. The average values contain no *t*-dependent information and are therefore subtracted. (If this is not done a constant value $\bar{A}\,\bar{B}$ is added to the correlation function.)

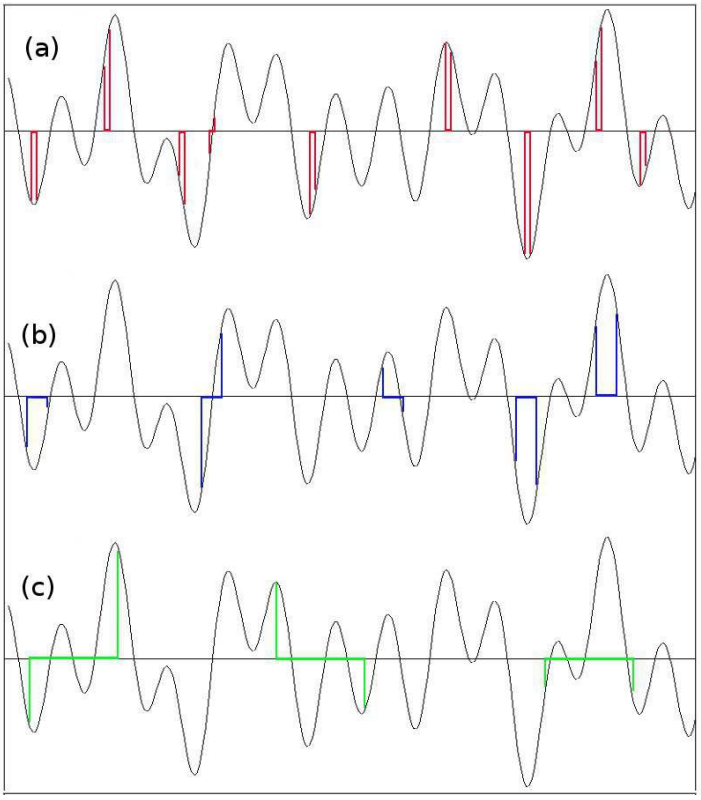

Figure 3.4. Constructing an Auto-correlation Function

The way a correlation integral works is shown in figure 3.4, which represents the construction of an autocorrelation function of a property $A(t)$ in a molecular system in which both correlated and random influences occur. It is also assumed (for

simplicity) that the average value of $A(t)$ is zero. What happens when t is small is shown in panel (a), where pairs of sampled values of the fluctuating function $A(t)$, separated by the interval t, are shown. Because t is small, any pair of values tends to be similar in magnitude and sign, thus the product $A(t+u)A(u)$ tends to be positive over most of the range of $A(t)$, which results in a large positive value for $C(t)$. In panel (b), the sample points have a larger separation and there is now a greater probability that the pairs of values have opposite signs and substantially different values. Thus over the range of $A(t)$ the product $A(t+u)A(u)$ will sometimes be positive and sometimes negative. The integral thus gives a reduced value of $C(t)$ compared to when t is small. Clearly, if the correlation in the original function is persistent, the similarity in values and sign of $A(t+u)$ and $A(u)$ will also persist and the correlation integral will maintain a strong positive value for larger values of t. The function $C(t)$ will therefore decay more slowly when correlation is more persistent. In panel (c) the separation t is now large and the two sampled values of $A(t)$ have essentially random magnitudes and sign. The product $A(t+u)A(u)$ therefore alternates randomly between positive and negative values and the integral returns a value approaching zero. Thus the overall form of the function $C(t)$ shows the degree of correlation between values of variable A separated by the interval t and also indicates the rate at which this correlation decays with increasing t. The same observations apply to cross-correlation functions, which describe the correlation between two different variables.

For molecular dynamics purposes integral (3.22) should be expressed in a discrete form to exploit the data sets $\{A_i\}$ and $\{B_i\}$ sampled at N_S points at intervals of Δt. This is given by

$$C_k = \frac{1}{N_S - k} \sum_{i=0}^{N_s-k-1} \left(A_{(i+k)} - \bar{A}\right)\left(B_i - \bar{B}\right) \qquad with \qquad \{k = 0 \ldots, N_s - 1\}. \tag{3.23}$$

Note that it is essential that the ordering of the data points with respect to the variable t is preserved.

This is a Boltzmann view of correlation, but it may be cast into a Gibbs view by imagining this calculation being performed on many replicas of the system simultaneously to give $\langle C_k \rangle$ rather than C_k, which under the Ergodic Hypothesis can be regarded as the same.

Correlation functions have number of general properties, which at first sight appear somewhat obscure, but turn out to be very useful. We present them here. (Note that to avoid writing the full correlation integral we shall represent (3.22) as: $C(t) = [A(u+t)B(u)]$ in what follows.)

1. The correlation function is independent of the origin. This means that in the integral (3.22) the integration variable u may be replaced by $u+c$ where c is any constant, without affecting the final correlation function i.e.

$$C(t) = [A(u+t)B(u)] = [A(u+t+c)B(u+c)]. \tag{3.24}$$

This is the same as saying that it does not depend on where along the u axis we start to compute the correlation function, the result will always be the same.

2. If $C(t)$ is the correlation function of $A(t)$ and $B(t)$ which have time derivatives $\dot{A}(t)$ and $\dot{B}(t)$ respectively, then

$$C(t)=[\dot{A}(t)B(t)]=-[A(t)\dot{B}(t)] \tag{3.25}$$

and

$$C(t)=[\dot{A}(t)A(t)]=[A(t)\dot{A}(t)]=0. \tag{3.26}$$

These follow from the independence of the origin.

3. The correlation function is symmetric with respect to the variable $t.$ In mathematical terms this means that $C(t)=C(-t).$ This means that any correlation that exists when $t=0$ varies in the same way going forward or backward from that point.

4. If $C(t)$ is the correlation function of $A(t)$ and $B(t)$ and their respective Fourier transforms are $\tilde{C}(f),\ \tilde{A}(f)$ and $\tilde{B}(f)$ e.g.

$$\tilde{C}(f)=\int_{-\infty}^{\infty} C(t)\exp(-2\pi ift)\,dt \tag{3.27}$$

then $\tilde{C}(f)=\tilde{A}^*(f)\tilde{B}(f),$ where $\tilde{A}^*(f)$ is the complex conjugate of $\tilde{A}(f).$ Thus, while integral (3.22) has a complicated construction in the t domain, it is a simple product in the f domain. (Note f is *frequency* when t represents time and *wavenumber* when it represents distance.) This strange relation turns out to be very useful in the theory and practice of molecular dynamics.

3.9 The Velocity Auto-correlation Function

A commonly computed example of a correlation function is the molecular velocity auto-correlation function (VAF), in which the argument of the integral in (3.22) is the scalar product $\vec{v}(t+u)\cdot\vec{v}(u).$ (Note the average molecular velocity is zero, since it may be positive or negative). This correlation function shows how, on average, the velocity vector of a molecule changes with time in response to the presence of other molecules in the system. An example VAF plot of a typical Lennard-Jones liquid is shown in figure 3.5. The function is strongly positive at short time, but diminishes in absolute value as time increases, while showing a distinctly oscillatory behaviour. The decay of the VAF towards zero at longer times is characterised by a *relaxation time* $\tau_c,$
after which correlation is effectively lost.

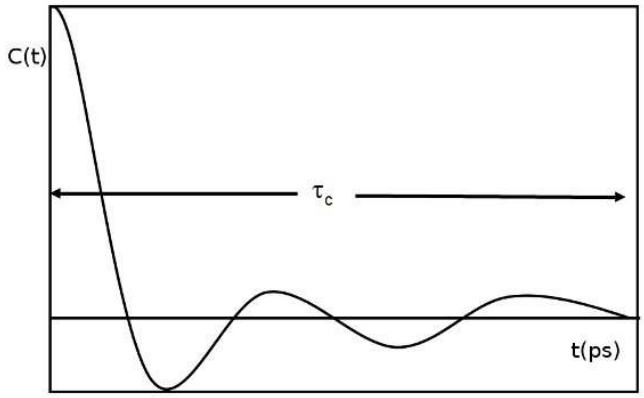

Figure 3.5: A Typical Velocity Auto-correlation Function

The behaviour of this plot can be accounted for in the following way. At short time, a molecule is not much influenced by neighbouring molecules and maintains its initial velocity, giving high correlation with its velocity at zero time. However, gradually the forces from neighbouring molecules take effect and the velocity is perturbed and gradually altered, so the correlation diminishes with longer time. In a liquid, each molecule is surrounded by an effective "shell" of neighbours and at some point a collision with this shell causes it to rebound. This requires the velocity to reverse direction to some extent, so the correlation goes negative. The point at which the VAF first goes to zero is thus measures the average "collision time." Repeated crossings of the time axis suggest that sequential collision events take place, however, since this occurs against a background of rapidly moving molecules, there is an accumulating randomness to the trajectory of the molecule and the overall correlation is bound to diminish to zero. This account is a simplification that obscures other, more subtle, effects that may be taking place, but it is broadly in keeping with our understanding of the underlying molecular behaviour.

3.10 Diffusion and the Green-Kubo Relation

There is a close relationship between the VAF and the MSD that is particularly revealing about molecular processes and worthy of an account here. It starts with the definition of molecular displacement as a function of time and velocity derived from the equation of motion (1.7)(b):

$$\vec{r}(t) = \int_0^t \vec{v}(u)\,du. \tag{3.28}$$

Squaring this gives

$$r^2(t) = \int_0^t \int_0^t \vec{v}(u') \cdot \vec{v}(u)\,du'\,du. \tag{3.29}$$

We may now define a relationship between u and u' as

$$u' = u + s, \tag{3.30}$$

which allows (3.29) to be re-written as

$$r^2(t) = \int_0^t \int_{-u}^{t-u} \vec{v}(u+s) \cdot \vec{v}(u)\, ds\, du. \tag{3.31}$$

Taking an ensemble average of $r^2(t)$ leads to

$$\left\langle r^2(t) \right\rangle = \int_0^t \int_{-u}^{t-u} \left\langle \vec{v}(u+s) \cdot \vec{v}(u) \right\rangle ds\, du. \tag{3.32}$$

Now, the ensemble average $\left\langle \vec{v}(u+s) \cdot \vec{v}(u) \right\rangle,$ which we recognise as the VAF encountered above, should return the same result no matter what the value of $u.$ It is dependent only on the variable s and not on the origin $u.$ So it can be written as $\left\langle \vec{v}(s) \cdot \vec{v}(0) \right\rangle$ without affect. So equation (3.32) becomes

$$\left\langle r^2(t) \right\rangle = \int_0^t \int_{-u}^{t-u} \left\langle \vec{v}(s) \cdot \vec{v}(0) \right\rangle ds\, du. \tag{3.33}$$

To take things further we must change the order of integration, so that the integration over the variable u is done first. This requires a change to the limits of the integration so it covers the same region of the u and s plane:

$$\left\langle r^2(t) \right\rangle = 2 \int_0^t \int_s^t \left\langle \vec{v}(s) \cdot \vec{v}(0) \right\rangle du\, ds. \tag{3.34}$$

Integration over u gives

$$\begin{aligned} \left\langle r^2(t) \right\rangle &= 2 \int_0^t (t-s) \left\langle \vec{v}(s) \cdot \vec{v}(0) \right\rangle ds \\ &= 2t \int_0^t \left\langle \vec{v}(s) \cdot \vec{v}(0) \right\rangle ds - 2 \int_0^t s \left\langle \vec{v}(s) \cdot \vec{v}(0) \right\rangle ds \\ &= 2t I_0 + 2 I_1, \end{aligned} \tag{3.35}$$

where

$$I_0 = \int_0^t \langle \vec{v}(s) \cdot \vec{v}(0) \rangle ds \tag{3.36}$$

and

$$I_1 = \int_0^t s \langle \vec{v}(s) \cdot \vec{v}(0) \rangle ds. \tag{3.37}$$

As we have already noted, the VAF, decays to zero at long time, in which case both integrals I_0 and I_1 settle on constant values as t increases. So in the long-time limit we have

$$\langle r^2(t) \rangle = C_0 t + C_1. \tag{3.38}$$

This result establishes once again that the MSD is linear at long time, but the derivation reveals something more. Firstly it shows that the linearity of the MSD sets in only after the VAF goes to zero, in other words when correlation in the velocities of the molecules has been lost. Thus the linear MSD is a consequence (and signature) of random motion. Secondly it shows that, at short time, the MSD is unlikely to be linear, because the integrals I_0 and I_1 are not constant, indeed these integrals provide a means to determine the nature of the non-linearity of the MSD in this regime. Lastly, we note that the slope of the linear MSD plot determines the so called diffusion coefficient D, through the relation $C_0 = 6\mathrm{D}.$ From this it follows directly that

$$D = \frac{1}{3} \int_0^\infty \langle \vec{v}(s) \cdot \vec{v}(0) \rangle ds. \tag{3.39}$$

This relationship, between the diffusion coefficient D and the integral I_0 is an example of a *Green-Kubo* relation, which is a general relationship between *transport coefficients* and *correlation functions*. It is an extremely important relationship for molecular dynamics work and theory in general.

3.11 The Radial Distribution Function

In condensed phase systems (i.e. solids or liquids) the atoms pack around each other in a characteristic order. In crystalline solids this order is obvious, but in liquids and amorphous solids (such as glasses) this is much harder to see, particularly if the atoms are in motion with respect to each other and, indeed, do not even hold an average mean position. The order in the structure is hidden by the chaos. The RDF represents a means of *projecting out* this hidden order. It is a description of the *average* atomic structure of the system.

The manner in which the RDF is constructed is shown in figure 3.6. At regular intervals in the simulation, each atom is taken in turn and this becomes the centre for

a series of concentric spheres (e.g. as shown in figure 3.6), each an interval Δr larger than the one inside. The volume between consecutive spheres defines a *shell* of atoms displaced around the central atom. The number of atoms in each shell fluctuates as the simulation proceeds due to their motion, but if an average is taken, by using every atom in turn as a centre and by sampling many configurations over the course of the simulation, a constant number is obtained for each shell. This defines the atomic *radial number distribution,* $n(r),$ as a function of the radius $r.$

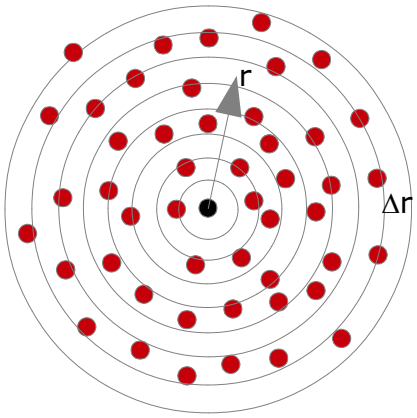

Figure 3.6: Constructing the Radial Distribution Function

Formally this defined as

$$n(r)=\left\langle \frac{1}{N}\sum_{i=1}^{N}\sum_{j\neq i}^{N}\delta\left(|\vec{r}-\vec{r}_{ij}|\right)\right\rangle, \tag{3.40}$$

in which $\delta(x)$ is the *Dirac delta function*[4], which is defined as

$$\delta(x)=\begin{cases}1 & if\ x=0,\\ 0 & if\ x\neq 0.\end{cases} \tag{3.41}$$

In this context $\delta(x)$ is a counting device that counts the occurrences of $r=r_{ij}.$

Dividing $n(r)$ by the volume of the shell and by the average atomic density for the system (ρ) gives the radial distribution function $g(r).$ Mathematically this is approximated by the equation:

$$g(r)=\frac{n(r)}{\rho\, 4\pi r^2 \Delta r}, \tag{3.42}$$

in which the product $4\pi r^2 \Delta r$ is an approximation for the volume of the shell.

4 For more information on the Dirac delta function see Appendix 2.

The RDF describes how, on average, the atoms are arranged around each other in the system. A typical RDF for a liquid system is shown in figure 3.7, in which a number of features are apparent.

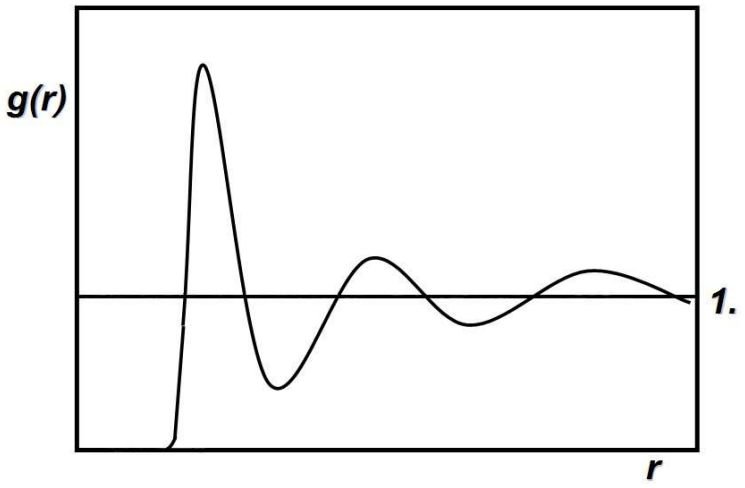

Figure 3.7: A Typical Radial Distribution Function

The peaks in the RDF correspond to the packing of the atoms around an atom in concentric shells. Such packing would not be readily apparent in a `snapshot' of an atomic configuration, as it requires the averaging process to make it clear. The RDF in figure 3.7 is typical for a liquid, in which the peaks are broad and the minima between them are non zero – indicating that atoms may freely interchange between shells, as would be expected for a liquid. In a solid they are sharper and more localised, so interchange of atoms between shells is not very likely. The RDF is zero at small values of r due to the finite size of the atoms. Overlapping of atomic centres is not possible due to the short ranged repulsion evident in the plot of the Lennard-Jones potential in figure 1.3. The location of the first peak compares nicely with the minimum of the potential, though it is not always this clear with other potentials. Another noticeable feature is that at large values of r the RDF tends to the value $1.$ This happens because the the RDF must eventually reach the uniform density of the bulk system (ρ) and as the formula (3.42) shows, the RDF must therefore become unity. The RDF thus reveals the hidden structure within the rapid dynamics of atomic systems and is a very useful property for describing such systems.

The RDF can be compared directly to experiment via the structure factor, $S(k),$ which is obtained from x-ray scattering experiments:

$$S(k)=1+4\pi\rho\int_0^\infty\left(\frac{\sin(kr)}{kr}\right)(g(r)-1)r^2\,dr. \tag{3.43}$$

This integral is little more than a radial Fourier transform of the RDF. The variable k

is the magnitude of the so-called *scattering vector* of the x-rays and ρ is the atomic density.

The RDF also has application in calculating thermodynamic properties. For example in a Lennard-Jones system, which is governed by a potential of the form given in (1.2), the configuration energy (1.11) may be accurately estimated from

$$U_{conf} \approx 2\pi\rho \int_0^\infty V(r)\, g(r)\, r^2\, dr\,, \tag{3.44}$$

in which $V(r)$ represents the Lennard-Jones potential and ρ is once again the average atomic density.

At this point it is useful to recognise that (3.44) provides a means for correcting for the cut-off that is applied to in the potential in a molecular dynamics simulation. If it is assumed that r_{cut} is large enough to ensure that $g(r) \approx 1,$ we may write

$$U_{corr} \approx 2\pi\rho \int_{r_{cut}}^\infty V(r)\, r^2\, dr \tag{3.45}$$

where U_{corr} is the long range correction to the configuration energy.

In a similar way we may obtain the long range correction to the virial as

$$\Psi_{corr} \approx -4\pi\rho \int_{r_{cut}}^\infty \left(\frac{dV(r)}{dr} \right) r^3\, dr. \tag{3.46}$$

3.12 Collective Properties - Correlations in Space and Time

In an atomic system there are correlations that are functions of both space and time. Such correlation functions can be used to describe how the atomic structure of a system collectively responds to the dynamical motions of the constituent atoms. This behaviour is encapsulated by the *van Hove correlation function* [8], which (for a single component system) has the following form:

$$G(\vec{r}\,, t) = \frac{1}{N} \left\langle \sum_{i=1}^{N} \sum_{j=1}^{N} \delta[\vec{r} + \vec{r}_i(0) - \vec{r}_j(t)] \right\rangle \tag{3.47}$$

From the presence of the Dirac delta function it can be seen that this function accumulates non-zero contributions to the ensemble average only when vector $\vec{r}$ equals $\vec{r}_j(t) - \vec{r}_i(0)$ and only if there is a real correlation between the position of atom i at time zero and that of atom j at time t will the average yield a significant result.

It may be useful to have an example of how such a correlation can arise. If we consider a static description of structure of an atomistic system, we can understand this in terms of how atoms pack around an arbitrarily chosen atom. If the chosen atom is removed (perhaps by thermally activated displacement) the original structure will rearrange. However the rearrangement does not happen instantly, but will occur on an appropriate time scale that will vary according to the distances of the atoms surrounding the chosen one. Thus the original event (the displacement of the chosen atom) will be followed by later events at positions displaced from it. These later events are thus correlated in space and time with the original event.

The van Hove correlation function (3.47) can be separated into two contributions. The first is the *self correlation function*:

$$G_s(\vec{r},t)=\frac{1}{N}\left\langle \sum_{i=1}^{N}\delta[\vec{r}+\vec{r}_i(0)-\vec{r}_i(t)]\right\rangle \tag{3.48}$$

and the second is the *distinct correlation function*:

$$G_d(\vec{r},t)=\frac{1}{N}\left\langle \sum_{i=1}^{N}\sum_{j\neq i}^{N}\delta[\vec{r}+\vec{r}_i(0)-\vec{r}_j(t)]\right\rangle . \tag{3.49}$$

The self function (3.48) is concerned with the correlation between the $i'th$ atom at time zero and the same atom at time $t.$ On this basis it may be assumed to be related to the self diffusion of atoms. The distinct function (3.49) correlates the position of the $j'th$ atom at time t with that of the $i'th$ atom at time zero. It is therefore concerned with how the system collectively responds to the motions of individual atoms. This is important in the propagation of effects through the system, such as sound waves. If we set $t=0$ in (3.48), we find that

$$G_s(\vec{r},0)=\delta(\vec{r}). \tag{3.50}$$

Similarly, setting $t=0$ in (3.49) leads to

$$G_d(\vec{r},0)=\rho g(\vec{r}), \tag{3.51}$$

where ρ is the uniform system density and $g(\vec{r})$ is the static radial distribution function. In the long range $(\vec{r}\to\infty)$ we find that $G(\vec{r},t)\to\rho.$

An example of the use of the self correlation function is shown in figure 3.8. Two plots are shown of the *radially averaged* functions $G_s(r,t)$ plotted as functions of r at several different times. The plot on the left shows a series of Gaussian-like functions[5] that spread as time increases. This describes diffusion in a liquid. The plot on the right shows the corresponding behaviour in solid diffusion. The peak at the origin slowly collapses while a second peak grows some distance, $\Delta r,$ away. It shows that diffusion in a solid involves atomic "hopping" - the atom jumps quickly from one position to another, without spending much time at intervening distances. Liquid

5 For more information on Gaussian functions see Appendix 1.

diffusion, on the other hand, is more gradual.

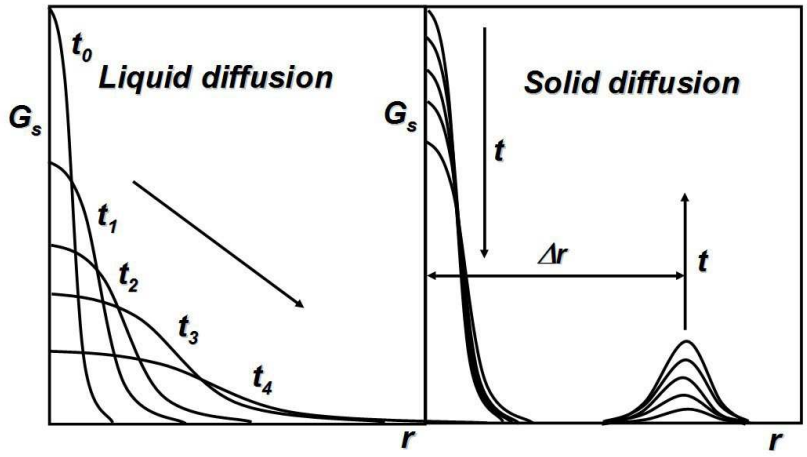

Figure 3.8. The van Hove Self Correlation Function and Diffusion

An example of the distinct correlation function is shown in figure 3.9. This shows a series of plots of the *spherically averaged* functions $G_d(r,t)$ plotted as functions of r at different times. Since they are spherically averaged they appear as a series of radial distribution functions. This again is an example of hopping-diffusion in solids and we see the growth of a peak at the origin as time progresses while the nearest outer peak slowly collapses. This behaviour is consistent with the hopping mechanism – the atom at the origin hops away and atoms from the first shell surrounding the position hop into the vacancy. Thus both the self and distinct correlation functions are useful for discovering the mechanism of diffusion in various system.

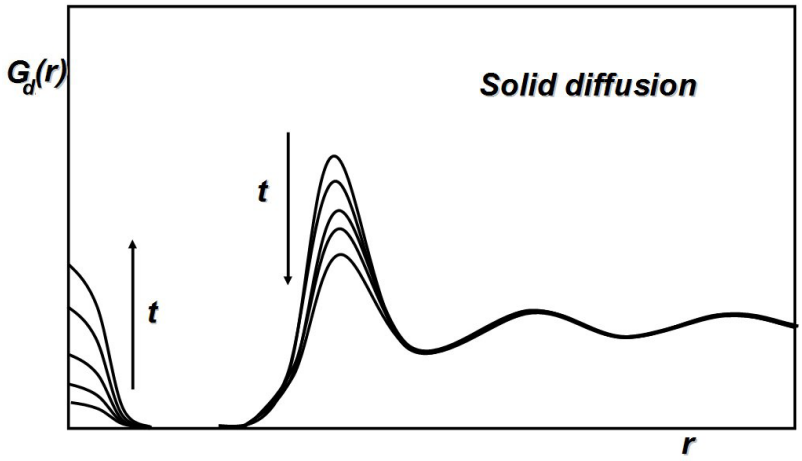

Figure 3.9. The van Hove Distinct Correlation Function and Diffusion

Closely related to the van Hove correlation function is the *dynamic structure factor*, $S(\vec{k},\omega)$, which is a generalisation of the static structure factor given in equation

(3.43). The dynamic structure factor is built from the atomic data in stages in the following manner. Firstly, we define the *atomic density function,* $\rho(\vec{r},t),$ at an instant t as follows:

$$\rho(\vec{r},t)=\sum_{j=1}^{N}\delta\left(\vec{r}-\vec{r}_j(t)\right). \tag{3.52}$$

Next the spatial Fourier transform of the atomic density is obtained:

$$\rho_{\vec{k}}(t)=\int_{-\infty}^{\infty}\rho(\vec{r},t)\exp\left(-i\vec{k}\cdot\vec{r}\right)d\vec{r}, \tag{3.53}$$

in which $\vec{k}$ is a *wave-vector* defined (e.g. for a cubic simulation cell) as

$$\vec{k}=\frac{2\pi}{L}\begin{bmatrix}l\\m\\n\end{bmatrix}, \tag{3.54}$$

where l, m, n are integers and L is the width of the cell. The result of this transform is

$$\rho_{\vec{k}}(t)=\sum_{j=1}^{N}\exp\left(i\vec{k}\cdot\vec{r}_j(t)\right). \tag{3.55}$$

The effect of the Fourier transform is, of course, to project out any spatial periodicities that may exist in the system density, that are commensurate with the wave-vector $\vec{k}$ and which are indicated by the magnitude of the coefficients $\rho_{\vec{k}}(t).$ It is evident from (3.55) that $\rho_{\vec{k}}(t)$ is a *complex* quantity.

We next compute the function:

$$F(\vec{k},t)=\frac{1}{N}\left\langle\rho_{\vec{k}}(t)\rho_{-\vec{k}}(0)\right\rangle. \tag{3.56}$$

The resulting function $F(\vec{k},t)$ is known as the *Intermediate Scattering Function*, which is related to the van Hove correlation function, $G(\vec{r},t),$ via the spatial Fourier transform:

$$F(\vec{k},t)=\int_{-\infty}^{\infty}G(\vec{r},t)\exp(-i\vec{k}\cdot\vec{r})d\vec{r}. \tag{3.57}$$

From this relation we see that $F(\vec{k},t)$ describes how the spatial periodicities in the

van Hove correlation function vary as a function of time. Since $G(\vec{r},t)$ is expected to be an even function of $\vec{r}$, its Fourier transform will be *real.* We may now go one step further and project out the periodicities in time also:

$$S(\vec{k},\omega)=\frac{1}{2\pi}\int_{-\infty}^{\infty}F(\vec{k},t)\exp(i\omega t)dt, \tag{3.58}$$

in which ω is the *angular frequency* $2\pi f$, where f is the frequency. This, finally, is the *dynamic structure factor*. It is evidently the spatial *and* temporal Fourier transform of the van Hove correlation function $G(\vec{r},t)$ and as such represents a complete description of the bulk dynamical properties of the system. It is related to the static structure factor via the equation:

$$S(\vec{k})=\int_{-\infty}^{\infty}S(\vec{k},\omega)d\omega. \tag{3.59}$$

The dynamic structure factor projects out the periodicities in $G(\vec{r},t)$ in both space and time. Such dual periodicities arise in the propagation of sound through the system. An example of this is shown in figure 3.10, which is a plot of $S(\vec{k},\omega)$ as a function of ω for a particular wave-vector $\vec{k}=\{2\pi/L\}(0,0,1)$. The secondary peak at ω_0 is due to a sound wave travelling with a velocity ω_0/k.

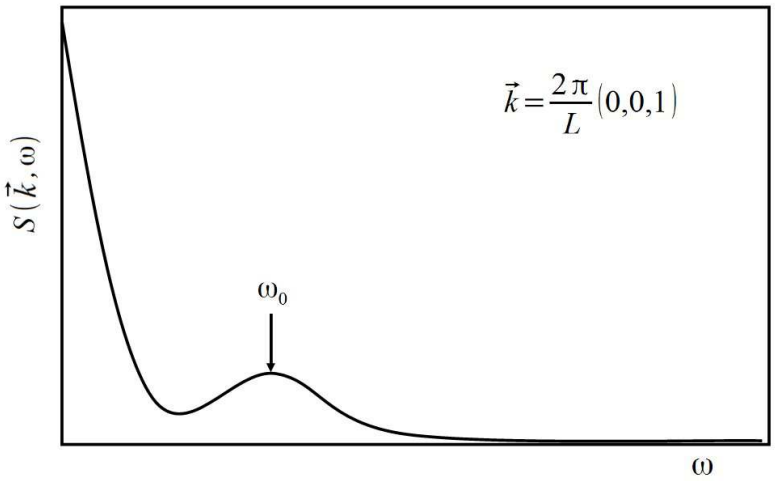

Figure 3.10. Typical plot of $S(\vec{k},\omega)$ Showing the Propagation of a Sound Wave

Sound propagation is intimately linked to the elastic properties of a system, so the dynamic structure factor also affords an insight into these properties. However, the most important aspect of the dynamic structure factor is its central rôle in inelastic neutron scattering. It can be shown that it is directly related to the differential cross section through the formula (see [1] p. 222)

$$\frac{d^2\sigma}{d\Omega\, d\omega} = b^2 N \frac{k_1}{k_0} S(\vec{k}, \omega), \tag{3.60}$$

in which b is the atomic scattering length and $\vec{k}_0$ and $\vec{k}_1$ are the wave-vectors of the incoming and scattered neutrons respectively. Equation (3.60) (and its extension to multi-component systems) thus affords a very deep connection between experiment and molecular dynamics.

3.13 Statistical Error in Simulations

Fluctuations are the main source of statistical error in the properties calculated by molecular dynamics. The standard error (σ) of an average $\langle A \rangle$ is given formally by

$$\sigma^2 = \left(\delta \bar{A}^2\right) / N_s \tag{3.61}$$

where $\left(\delta \bar{A}^2\right)$ is the variance calculated using the formula (3.4) and N_S is the number of sampled data points. This formula should be treated with caution however, as it is essential for the data used in the calculation of the variance to be *uncorrelated* if an accurate determination of the error is to be obtained. Uncritical use of (3.61) can lead to the calculated error being as much as an order of magnitude too small! It is important therefore to be able to ensure that the sampling of the data produces an uncorrelated set fit for calculating the true statistical error.

An obvious method for checking for correlation is to compute the time correlation function for the property of interest. This will show how long the correlation in the data persists through the correlation time τ_{corr}. Once this is known, we may set about re-sampling the data at intervals in excess of this and using the re-sampled set to calculate the error properly. It is not necessary to be too precise in the determination of τ_{corr} provided it is clear that the estimate guarantees the correlation is truly lost on this time scale.

A very practical approach to estimating errors is that described by Flyvbjerg and Petersen [9] and is commonly known as the *blocking method*. This works as follows.

Commencing from a set of data $\left\{A_i : i = 1, \ldots, N_s\right\}$ which has been generated in a time sequence such that: $t = i\Delta t$, we define a blocking operation $T\left(\{A_i\}\right)$ by the operation:

$$T\left(\{A_i\}\right): \ \frac{1}{2}\left(A_{(2i-1)} + A_{2i}\right) \rightarrow A_i'. \tag{3.62}$$

In other words, we take the average of separate, consecutive pairs in the set $\{A_i\}$ and create a new set $\left\{A_i' : i = 1, \ldots, N_s/2\right\}$ (which is half the size of the previous one).

We can also define an operation $T^n\left(\{A_i\}\right)$ to mean the application of operation $T\left(\{A_i\}\right)$ repeatedly n times, reducing the set size by half on each application. (Note: if N_s is an odd number at any stage, the last data point is dropped from the set.) Now, it can be shown that, if the data is *uncorrelated* then after each application of $T\left(\{A_i\}\right)$:

$$\bar{A}' = \bar{A}, \quad \sigma'^2 = \sigma^2 \quad \text{and} \quad \Delta t' = 2\Delta t. \tag{3.63}$$

So, while the average and variance of the data is unchanged, the time interval separating the data points in the new set is doubled. Since our assumption is that the data is correlated at short time, but uncorrelated at long time, with each blocking operation, the growth of Δt guarantees that at some stage there exists a value of n for which $T^n\left(\{A_i\}\right)$ gives an uncorrelated set. When this happens σ^2 will settle on the true value for uncorrelated data and then it remains fixed on that value, having grown from a small number to one significantly larger. (The average is relatively insensitive to increasing *n*.) So the procedure is simple: calculate $\bar{A}$ and σ^2 on the data obtained from the simulation then repeatedly perform the blocking operation $T\left(\{A_i\}\right)$ until σ^2 converges on a fixed value. The converged value is then gives the true statistical error.

Unfortunately, things are never quite this simple. In practice, halving the number of data points with each blocking operation, reduces the accuracy of the estimated error. (The error bar of σ^2 increases as n increases.) This means that smooth convergence is rarely observed. Furthermore it is possible that there is insufficient data for σ^2 to converge within accessible values of n. In general it is advisable to plot σ^2 *versus* n to observe how well the convergence proceeds. If there is no obvious plateau in the plot, the largest obtained value of σ^2 must be taken as the best guess. Despite these difficulties, the blocking method is currently the best known method for dealing with error estimation.

It is not often appreciated, but the blocking method can also be used to provide error bars on correlation functions, mean-squared displacements and other results normally presented in graphical form. The data used to generate each point on the graph plot is inevitably obtained from some sort of averaging process and the key is to apply the blocking method to the estimation of each of these averages. This implies a lot of work for each data point, but the blocking method is highly efficient, and the reward is a statistically reliable means for evaluating the convergence of the plotted data.

3.14 Distribution Functions

Useful though the various system averages and variances are, they are far from presenting a complete understanding of what goes on in a molecular system. A much more powerful idea is that of a *distribution function*. What this represents is a quantitative description of how the energy in an *equilibrated* system is shared out between its components. Once this is known, then not only can the average properties and variances of a system be trivially obtained, but also more significant properties

such as free energy and entropy become amenable to calculation.

The most important distribution function in statistical thermodynamics is the Boltzmann distribution function[6] shown in figure 3.11. It is presented here as a histogram, each vertical bar of which indicates how many molecules possess an energy between the values $E_i - \delta E/2$ and $E_i + \delta E/2,$ where δE is an energy *bin width*, which is small on the scale of full energy range accessible to the system and $\{E_i\}$ is a set of ascending energy levels. This distribution is described by the well-known formula

$$n_i = n_0 \exp(-\beta E_i), \tag{3.64}$$

in which n_0 represents the number or population of molecules on the lowest energy level in the system and n_i the population of the $i'th$ level. The exponent β is given by

$$\beta = 1/k_B T, \tag{3.65}$$

in which T is the absolute temperature and k_B is Boltzmann's constant. The histogram clearly shows that the number of molecules at any given energy diminishes with rising energy. However increasing the temperature T has the effect of increasing the number of high energy molecules. All this is well known.

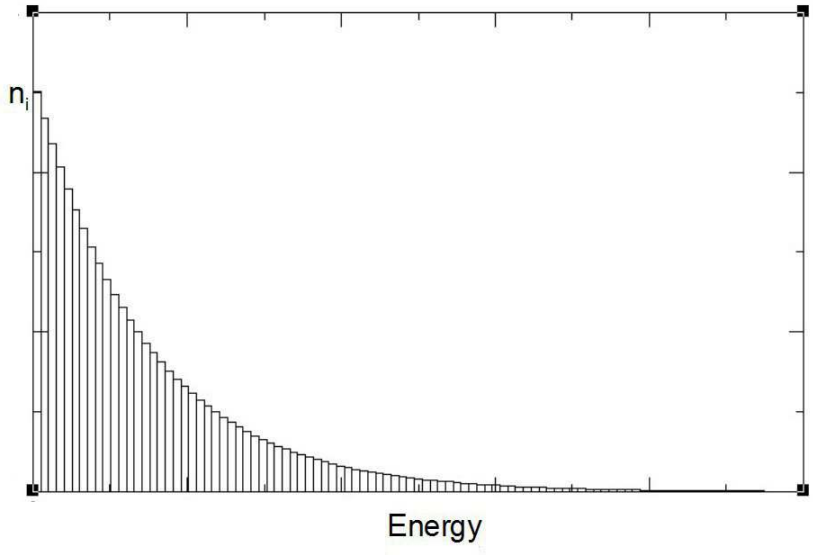

Figure 3.11: The Boltzmann Distribution

There is a wealth of literature that proves the inevitability of Boltzmann's distribution, but a simple computational model shows this emerging quite naturally. The recipe for

6 Another important distribution is the Normal or Gaussian distribution, which also arises in statistical mechanics. See Appendix 1.

a computer program showing this is as follows.

1. The system is composed of a large number of "molecules" (of order 10^5).
2. Each molecule starts with one unit of energy (in arbitrary units).
3. A pair of molecules is chosen at random.
4. The first molecule chosen *transfers* one unit of energy to the second, provided it has any energy units to give.
5. Steps 3 and 4 are repeated *ad infinitum (*i.e. a very large number of times - say 10^9 pair selections).
6. Create a histogram of the number of molecules on each energy E_i level in the system.

This recipe is clearly a very crude model for the interchange of energy between molecules in (say) a gas, but it is not too different in spirit from what is believed to occur in reality. The question is: what is the final distribution of energy units between the molecules after a long time?

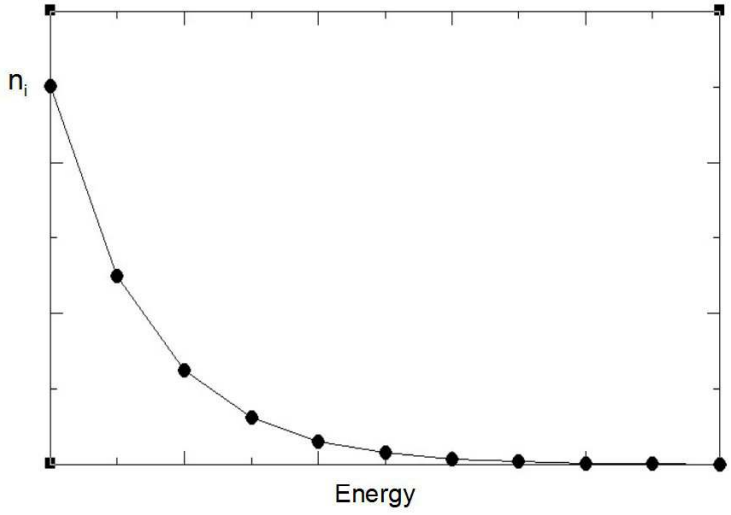

Figure 3.12: The Simulated Boltzmann Distribution

A computer program based on this recipe shows that after a certain time (depending on the number of molecules) the distribution settles on the form shown in figure 3.12. A plot of the logarithm of the number of molecules versus energy shows that the distribution is exponential in form with a negative exponent. All of which strongly suggests that the Boltzmann distribution operates in this model.

How can this be proved? If is assumed that the model does settle on a fixed distribution and that it is a monotonically decaying function with increasing energy (which is what our simulation implies), then it can be established that the distribution must be exponential. The argument runs as follows.

We assume that there are N molecules in the system and that each energy level

E_j is occupied by n_j molecules in the equilibrium state. We examine all the possible energy transfers that can affect the population n_i of a particular level E_i. Since each transfer is restricted to changing a molecule's energy by ± 1 unit, we need to consider four categories of molecule: those on levels E_i, E_{i-1} and E_{i+1} (with populations n_i, n_{i-1} and n_{i+1} respectively), and all the other molecules in the system (which number $M = N - n_i - n_{i-1} - n_{i+1}$ - we'll refer to these as class M molecules).

The two molecules selected for energy transfer are chosen randomly, so the probability of selecting a molecule on any level E_j is simply n_j / N. Thus the probability of selecting a molecule on a level E_j *and* one on level E_k is $(n_j n_k)/N^2$, but if $j = k$ the probability is $(n_j(n_j - 1))/N^2$, (since choosing the first one, reduces the choice of the second). Consider now all the transfers that can occur involving molecules on level E_i :

- Exchanges between molecules on level E_i with those in class M will either increase or decrease the energy of the molecule on level E_i by one unit, so *all* such transfers reduce the population n_i by one molecule. Thus the outcome of such transfers is a change in the population n_i at a rate proportional to the probability of selection i.e. $R_1 = -(n_i M)/N^2$.

- Exchanges between two molecules on level E_i results on *both* molecules moving to new levels (one to level E_{i+1} and one to level E_{i-1}), thus the rate of population change of n_i resulting from this selection is $R_2 = -2(n_i(n_i - 1))/N^2$.

- Exchanges between molecules on levels E_i and E_{i-1} have no effect if the energy is passed from the higher energy molecule to the lower (they simply swap levels), but if the lower gives to the higher, then n_i is reduced by one, as the molecule jumps to level E_{i+1}. These alternatives have equal probability, so the rate of population change of n_i associated with this selection is $R_3 = -(n_i n_{i-1})/(2N^2)$.

- Similarly, transfers between levels E_i and $E_{(i+1)}$ lead to a rate of population change of n_i given by $R_4 = -(n_i n_{i+1})/(2N^2)$.

This is the last of the possible transfers that produces a reduction in the population n_i. Note that no transfer involving molecules on level E_i can possibly result in n_i increasing. The transfers that do produce an increase in population must be those that involve molecules on levels E_{i+1} and E_{i-1} thus:

- Molecules from level E_{i-1} selected with molecules in class M have a 50% chance of being promoted to level E_i. The same is true for molecules on level E_{i+1}. Taking account of the probabilities for this selection to be made, results in the rates of change of population n_i of $R_5 = (n_{i-1} M)/(2N^2)$ and

$R_6 = (n_{i+1} M)/(2N^2).$

- Exchanges between molecule pairs on level E_{i-1} always result on one molecule promoted to level E_i. Similarly for molecule pairs on level E_{i+1}, transfers always result in a demotion to level E_i. The rates associated with these transfers are $R_7 = (n_{i-1}(n_{i-1}-1))/N^2$ and $R_8 = (n_{i+1}(n_{i+1}-1))/N^2$.

- Lastly, molecules on level E_{i-1} selected with molecules on level E_{i+1} have a 50% chance of promoting both molecules to level E_i (or of moving both further away). So the rate of change of population n_i arising from this is $R_9 = (n_{i-1} n_{i+1})/N^2$.

These represent all the transfers that affect the population n_i. Since the distribution is assumed to be stationary, n_i is fixed, which must mean that all these competing transfers must combine with null effect i.e. $\sum_{k=1}^{9} R_k = 0.$

Inserting the above formulas for R_k and rearranging slightly gives:

$$0 = \frac{M}{N^2}\left(-n_i + \frac{n_{i-1}}{2} + \frac{n_{i+1}}{2}\right) + \frac{n_i}{N^2}\left(-2n_i - \frac{n_{i-1}}{2} - \frac{n_{i+1}}{2}\right) + \frac{1}{N^2}\left(n_{i-1}^2 + n_{i+1}^2 + n_{i-1} n_{i+1}\right) + \frac{1}{N^2}\left(2n_i - n_{i-1} - n_{i+1}\right).$$

Now we use the approximation that $n_i = (n_{i-1} + n_{i+1})/2,$ which results in the removal of the first and last terms on the right. We may also drop the factor $1/N^2$ throughout to give $0 = -3n_i^2 + n_{i-1}^2 + n_{i+1}^2 + n_{i-1} n_{i+1}$.

By gathering the squares of n_{i-1} and n_{i+1} we obtain $0 = -3n_i^2 + (n_{i-1} + n_{i+1})^2 - n_{i-1} n_{i+1},$ and using the approximate relation $n_i = (n_{i-1} + n_{i+1})/2,$ once again leads to $0 = n_i^2 - n_{i-1} n_{i+1}.$ This may be rearranged to give the key relationship: $\frac{n_i}{n_{i-1}} = \frac{n_{i+1}}{n_i}.$

Since the choice of level E_i as the focus of this analysis was arbitrary, the same relation must hold for all energy levels. In which case the ratio of the populations of consecutive levels is a constant (say C) and, moving down from level E_i one level at a time, we may write $n_i = C n_{i-1} = C^2 n_{i-2} = C^3 n_{i-3} = \ldots\ldots = C^i n_0,$ where n_0 is the population of the lowest accessible energy level. We note that, for a monotonically decreasing function, the constant C is less than one, in which case we can write $\log_e(C) = -\alpha,$ or $C = \exp(-\alpha)$ where α is some constant. So finally, we have $n_i = n_0 \exp(-\alpha i),$ which defines the *Boltzmann* distribution appropriate to our model. Note that the constant α can be determined from the simulation results – from the slope of a plot of $\log_e(n_i)$ *versus* i. Also the integer i in this equation may be replaced by the expression $E_i/\delta E,$ where δE is the (constant) gap between energy levels, to bring the distribution formula even closer to the Boltzmann form.

This is not, of course, a complete proof. We have not proved that the model *must* converge to a stable distribution or that it *must* be monotonically decreasing with increasing energy; these are merely implied by the simulation result. But we have at least shown that, if these assumptions hold, the Boltzmann distribution is the inevitable result.

3.15 Distribution Functions and System Properties

The idea of a distribution function allows us to think about system properties in a different way. If we consider the Boltzmann distribution shown in figure 3.11, this is a histogram of the population of molecules at each energy level in the system. We may therefore calculate averages of various properties of the system using the idea of a *weighted average*. So for some averaged property $\bar{A}$ we can write

$$\bar{A} = \frac{1}{N} \sum_{i=0}^{i_{max}} n_i A_i , \tag{3.66}$$

where N is the total number of molecules, n_i the population of molecules between the energies $E_i - \delta E/2$ and $E_i + \delta E/2,$ and i_{max} is the highest occupied energy "bin" in the system. The full set of bins in this construction spans the whole of the accessible energy scale. A_i is the value of the property A possessed by all the molecules in bin $i.$ Incidentally we note the obvious fact that

$$N = \sum_{i=0}^{i_{max}} n_i \tag{3.67}$$

which means that (3.66) can be written as

$$\bar{A} = \frac{\sum_{i=0}^{i_{max}} n_i A_i}{\sum_{i=0}^{i_{max}} n_i}. \tag{3.68}$$

Since the populations n_i are given by the Boltzmann distribution (3.64) this can be written as

$$\bar{A} = \frac{\sum_{i=0}^{i_{max}} A_i \exp(-\beta E_i)}{\sum_{i=0}^{i_{max}} \exp(-\beta E_i)}. \tag{3.69}$$

Note the apparent loss of dependence on N in this equation, however it is implicitly retained through the sum limit i_{max}, which ensures all molecules are counted.

Equation (3.66) really only applies when the histogram bins are of finite width δE, but the distribution can also be thought of as a continuous function. This can be done by modifying (3.66) to give

$$\bar{A} = \frac{1}{N} \sum_{i=0}^{i_{max}} \left(\frac{n_i}{\delta E} \right) A_i \delta E \tag{3.70}$$

and (3.67) to give

$$N = \sum_{i=0}^{i_{max}} \left(\frac{n_i}{\delta E} \right) \delta E. \tag{3.71}$$

In both cases the term in brackets represents a *number density* (with respect to the energy) and is normally referred to as the *density of states*. We now assume that as $\delta E \to 0$ then $i_{max} \to \infty$, in which case (3.69) takes the limiting form of a ratio of integrals:

$$\bar{A} = \frac{\int_0^\infty A(E) \exp(-\beta E) dE}{\int_0^\infty \exp(-\beta E) dE} \tag{3.72}$$

in which we have redefined the zero point of energy to be the lowest energy a molecule can possess, hence the definite integrals begin at 0. Note also that the upper limit of the integrals is taken as infinity. This is because the integrals are valid for systems with an infinite number of molecules, in which case molecules may formally approach infinite energy. The form of (3.72) is a very familiar one in statistical mechanics.

From all this we may define a continuous form for the Boltzmann distribution as

$$f(E) dE = \eta \exp(-\beta E) dE \tag{3.73}$$

in which η is a *normalisation constant* defined so that

$$\frac{1}{\eta} = \int_0^\infty \exp(-\beta E) dE, \tag{3.74}$$

which ensures that

$$\int_0^\infty f(E)\,dE = 1. \tag{3.75}$$

In this form $f(E)$ defines a *probability density*, which gives the probability of finding a molecule within the range δE of energy E as $f(E)\delta E.$ Using this representation the average (3.72) becomes

$$\bar{A} = \int_0^\infty A(E)\,f(E)\,dE. \tag{3.76}$$

This form of the average integral is sometimes used in place of (3.72).

It is worth noting that although all the above integrals have energy E as the integrating variable, this is not essential. In defining the number density in equations (3.70) and (3.71) we could equally well have chosen some other variable or variables in the denominator, for example the atomic positions and momenta. This of course has the effect of changing the integrals to multiple integrals over many variables, but it also gives a better handle on the relationship between the atomic configuration and the statistical thermodynamics, as we shall see.

We should also note that as well as using expressions like (3.72) to calculate the average properties of a system, expressions for fluctuations and correlation functions *etc.* can also be derived. We reserve such issues until later.

A comment is probably in order here about the relevance of this discussion (and of the preceding "derivation" of the Boltzmann distribution) to the calculation of system properties in the Boltzmann and Gibbs representations. Our results appear to pertain to a single copy of the system, which is not Gibb's view, nor are they apparently averaged over time, which is Boltzmann's view. It should be clear however, that the distribution arrived at is one that emerges over time and thereafter remains static. Meanwhile, the molecules are assumed to be continually exchanging energy. Thus the distribution may be regarded as the *time averaged* result of all the transfers that take place and averages computed using the distribution are consistent with Boltzmann's approach. Gibb's view is that the distribution emerges from an ensemble of system replicas – the result of looking at the instantaneous energy distribution in each individual replica and averaging the result over all replicas. Clearly we have not done this here, though is is not infeasible to set many simulations of the system running at once and performing such an average. However, since our model system is entirely driven by random energy transfers and there is no correlation between successive events, it is not unreasonable to expect that the results of such an "ensemble" average would be identical to a single system result. If not, one may reasonably ask how an alternative result could ever arise. So with some confidence we may assert that the Boltzmann distribution would result from both ensemble and time averaging.

3.16 Phase Space

Since, in statistical mechanics, we are concerned with relating the properties of a bulk system to the mechanics of the atomic constituents, it is essential that we have a convenient means to describe the instantaneous atomic positions and momenta of the system. This is provided by the concept of *phase space*.

In a space described by Cartesian coordinates, every individual atom, with index i, is located by a position vector $\vec{r}_i=(x_i, y_i, z_i)$ and its momentum is described by a vector $\vec{p}_i=(p^x, p^y, p^z)$. If there are N atoms in the system, there are clearly $6N$ independent components defining the atomic positions and momenta of the entire system. It is therefore useful to propose a $6N$ dimensional space, called *phase space,* in which the whole system is defined by a unique vector $\vec{\Gamma}^N$, the $6N$ components of which are the ordered components of the individual atoms. Thus

$$\vec{\Gamma}^N=(x_1, y_1, z_1, \ldots, x_N, y_N, z_N, p_1^x, p_1^y, p_1^z, \ldots, p_N^x, p_N^y, p_N^z), \tag{3.77}$$

or more conveniently

$$\vec{\Gamma}^N=(x_1, x_2, \ldots, x_{3N}, p_1, p_2, \ldots, p_{3N}), \tag{3.78}$$

in which the individual position and momentum components are distinguished by location in the vector, rather than by symbol. The form

$$\vec{\Gamma}^N=(\vec{r}^N, \vec{p}^N) \tag{3.79}$$

is often used as an abbreviation. The vectors $\vec{r}^N$ and $\vec{p}^N$ represent the positions and momenta respectively of all N atoms in the system.

The vector $\vec{\Gamma}^N$ is of course Cartesian and defines a *single point* in the $6N$ dimensional phase space. As the system evolves in time through dynamical processes, the point traces a path, or trajectory, through phase space. If the path ever crosses itself, then the system is periodic by definition, as the solution of the equations of motion from a given point in phase space is uniquely determined by Newton's laws, so if the system crosses its own path it is destined to repeat the journey prior to this point.

The concept of phase space is a useful one. A dynamical trajectory is a unique path traced through phase space. This offers a useful (and easily visualized) distinction between Boltzmann's and Gibbs' approaches. A Boltzmann averaging process is one which calculates a system property over a single trajectory in phase space – the assumption being that this trajectory explores every region of phase space accessible to the system in the given thermodynamic state and so gives a meaningful average. On the other hand, an ensemble is represented at any given instant by a cluster of

points in phase space, each equivalent to an ensemble "replica" in the sense described previously, and each tracing out its own unique trajectory (see figure 3.13). The Gibbs averaging process is then one which calculates a system property as an average over all the ensemble points at a given instant. The points are assumed to be great in number and representative of the whole of the phase space accessible to the system in the given thermodynamic state. From this viewpoint the density of the points in phase space has meaning – it leads to idea of the *probability density* in phase space, which is discussed below. With regard to time evolution, the dynamics of the cluster of points representing the ensemble is also of interest and offers a route to calculating time dependent properties of the system.

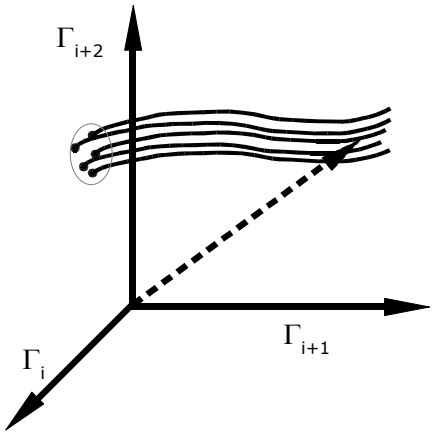

Figure 3.13: The Evolution of an Ensemble in Phase Space

3.17 The Probability Density in Phase Space

In order to generalise the ideas so far presented we now define the *N-particle* probability density in phase space: $f^N(\vec{\Gamma}^N)$. This goes beyond the distribution functions introduced in sections 3.14 and 3.15 , which described the probability of an individual *molecule* possessing a particular energy. Here we are concerned with the probability of finding the system within the volume $d\vec{\Gamma}^N$ around the phase space point $\vec{\Gamma}^N$. We may sample the system either by choosing a particular instant in time along a trajectory, or by choosing a phase space point from the ensemble.

First some notation. The function $f^N(\vec{\Gamma}^N)$ can appear in many forms, depending on how we wish to express the phase space dependence, thus:

$$f^N(\vec{\Gamma}^N) \equiv f^N(\vec{r}^N, \vec{p}^N) \equiv f^N(x_{1,}\ldots, x_{3N}, p_{1,}\ldots, p_{3N}). \tag{3.80}$$

Sometimes the implicit dependence of $\vec{\Gamma}^N$ on time due to its dynamical evolution is expressed, as in

$$f^N(\vec{\Gamma}^N;t)\equiv f^N(\vec{r}^N,\vec{p}^N;t)\equiv f^N(x_{1,}\ldots,x_{3N},p_{1,}\ldots,p_{3N};t), \tag{3.81}$$

which implies that $f^N(\vec{\Gamma}^N)$ is time dependent through $\vec{\Gamma}^N$ rather than explicitly, though in fact for equilibrium systems $f^N(\vec{\Gamma}^N)$ turns out not to be time dependent; it is only $\vec{\Gamma}^N$ that evolves.

There is a normalisation condition[7] on $f^N(\vec{\Gamma}^N)$ which is that

$$\begin{aligned}\int f^N(\vec{\Gamma}^N)\,d\vec{\Gamma}^N &= \int\int f^N(\vec{r}^N,\vec{p}^N)\,d\vec{r}^N\,d\vec{p}^N \\ &= \int\ldots\int f^N(x_1,\ldots,x_N,p_1,\ldots,p_N)\,dx_1\ldots,dx_N,dp_1,\ldots,dp_N = 1.\end{aligned} \tag{3.82}$$

which means that the summed probability of all possible phase space points must be unity. The normalisation integral (3.82) proves to be an important system property in statistical mechanics and it goes by a special name: the *partition function* (sometimes called the *sum over states*). Its importance will become manifest later.

Since molecules of the same species are identical, these formulas contain a great deal of statistically redundant information, so it is useful to collapse them down into lower order functions, such as the *n-particle* form:

$$f^n(\vec{r}^n,\vec{p}^n)=\frac{N!}{(N-n)!}\int\int f^N(\vec{r}^N,\vec{p}^N)\,d\vec{r}^{(N-n)}\,d\vec{p}^{(N-n)}, \tag{3.83}$$

where the ratio of factorials is intended to account for the number of different, but equivalent, ways n atoms can be chosen for the set of N atoms.

This process can be repeated from $n=N$ down to $n=1,$ which gives the *single-particle* probability density:

$$f^1(\vec{r}^1,\vec{p}^1)=N\int\int f^N(\vec{r}^N,\vec{p}^N)\,d\vec{r}^{(N-1)}\,d\vec{p}^{(N-1)}. \tag{3.84}$$

Usually in statistical mechanics the single particle form is the only one that is mathematically tractable, so has much theoretical importance. The two-particle function $f^2(\vec{r}^2,\vec{p}^2),$ is also of great importance since it is related to the radial distribution function described in section 3.11 .

Using the probability density we can formally define an average property of the system in the following way

$$\langle A\rangle=\int A(\vec{\Gamma}^N)\,f(\vec{\Gamma}^N)\,d\vec{\Gamma}^N, \tag{3.85}$$

7 In the sections that follow, it can be assumed that all integrals over phase space variables are intended to range over the whole of phase space.

which is the phase space analogue of equation (3.76). This is of course the Gibbs form for an ensemble average.

The important question however is what mathematical form does $f^N(\vec{\Gamma}^N)$ take? The obvious answer is that it must possess a form based on the Boltzmann distribution. However, the specific form used varies according to the kind of physical system being investigated. In the laboratory experiments are usually performed under prescribed conditions of volume and temperature, or pressure and temperature, or incorporating the exchange of atoms between phases and so on. For each such experiment there is an appropriate ensemble, and for different ensembles a different (though somewhat similar) form of $f^N(\vec{\Gamma}^N)$ is required. We describe a number of such ensembles in the following sections. These are known as the microcanonical, canonical, isothermal-isobaric and the grand canonical ensembles. Note that our use of the ensemble to define the different experimental systems does not mean that a purely Gibbs approach is inevitable. A Boltzmann description of all of these systems can be provided (though in the case of the grand canonical ensemble, not easily!) This is evidenced by the many molecular dynamics techniques in existence to model such systems, all of which use Boltzmann averaging. Nevertheless, to avoid duplication, we shall use Gibbs' nomenclature and formulas to proceed from here.

3.18 The Common Ensembles

3.18.1 The Microcanonical Ensemble (NVE)

The microcanonical ensemble is appropriate for systems with constant energy. In a molecular dynamics context these would be systems with a constant number of atoms, N, in a fixed volume, V, without any kind of thermostat to control the temperature, so the energy of the system, E, is conserved and cannot vary. This is the so-called NVE ensemble. The probability density is given by

$$f^N(\vec{\Gamma}^N)=\frac{h^{-3N}}{N!}\frac{\delta\left(H^N(\vec{\Gamma}^N)-E\right)}{Q^N(V,E)}. \tag{3.86}$$

The Dirac delta function[8], $\delta()$, in this formula ensures that the *N-particle* Hamiltonian function, $H^N(\vec{\Gamma}^N)$, (which equates to the energy of the system) does not deviate from the prescribed energy E. In effect the delta function samples the continuous Hamiltonian function and selects only those phase space vectors $\vec{\Gamma}^N$ for which the Hamiltonian equals energy E. The appearance of Planck's constant, $\hbar$, is unexpected, but is inserted to ensure that the quantum and classical definitions of the probability density are in accord. In the denominator, the factorial $N!$ takes account of the presumed indistinguishability of the atoms and the partition function $Q^N(V,E)$ ensures the normalisation of $f^N(\vec{\Gamma}^N)$. The partition function is thus

8 See Appendix 2 for a description of the Dirac delta function and its properties.

$$Q^N(V,E)=\frac{h^{-3N}}{N!}\int \delta\left(H^N(\vec{\Gamma}^N)-E\right)d\vec{\Gamma}^N. \tag{3.87}$$

Following from equation (3.86) the average value of some property A can now be given as

$$\langle A\rangle=\frac{\int A(\vec{\Gamma}^N)\delta(H^N(\vec{\Gamma}^N)-E)d\vec{\Gamma}^N}{\int \delta(H^N(\vec{\Gamma}^N)-E)d\vec{\Gamma}^N}. \tag{3.88}$$

The practical interpretation of this formula is that, in the numerator, the delta function is selecting points from phase space for which the system energy is E and adds the value of the variable A that corresponds to these points to a sum (the integral). The denominator then provides the number of phase space points sampled to give the final average $\langle A\rangle$.

It has to be admitted that the probability density for the microcanonical ensemble does not much resemble Boltzmann's distribution. This is because the microcanonical ensemble identifies phase space points with the same energy and does not describe a distribution of points with different energies. Nevertheless, an examination of the energy distribution between the molecules of each of these iso-energetic points will show it to be Boltzmann-like in form.

Associated with every physical system is the idea of a *thermodynamic potential.* This is the potential that drives *change* in the system, rather as change in a (damped) mechanical system is driven by its seeking the lowest potential energy. Since the system energy of a microcanonical ensemble is strictly constant, it is apparent that it cannot be this that drives change. It is in fact the entropy (specifically the quantity $-TS$, where T is the temperature and S the entropy) that is the thermodynamic potential.

It is often said that molecular dynamics simulations at constant particle number, volume and energy (NVE) corresponds to the microcanonical ensemble. This is almost true, but in molecular dynamics it is customary to place a constraint on the system to guarantee that the centre of mass of the system is stationary. This removes three degrees of freedom from the system and violates, if in a minor way, the identification with the microcanonical ensemble. Since the same constraint is applied to most ensembles in molecular dynamics, this caveat is practically universal.

3.18.2 The Canonical Ensemble (NVT)

The canonical ensemble applies to systems at constant volume which are thermostatted to maintain a constant *average* temperature, while the sampled temperature fluctuates about the average value. Temperature fluctuation also happens in the microcanonical ensemble, but here the thermostat adds or removes heat

energy from the system to maintain the constant average temperature. Thus in the canonical ensemble the system energy also fluctuates about a mean value. This corresponds to many real experimental systems, where the volume is kept constant in a sealed vessel immersed in a thermostatted heat bath controlling the temperature.

The probability density in this case is

$$f^N(\vec{\Gamma}^N)=\frac{h^{-3N}}{N!}\frac{\exp(-\beta H^N(\vec{\Gamma}^N))}{Q^N(V,T)}, \tag{3.89}$$

where, once again, $H^N(\vec{\Gamma}^N)$ is the *N-particle* Hamiltonian function and now $Q^N(V,T)$ is the canonical partition function, with V and T representing the volume and temperature respectively. The constant β is given by $\beta=1/k_BT$, as usual. This formula shows that the distribution of energy amongst the different phase space points in the canonical ensemble is Boltzmann-like in form.

The partition function for the canonical ensemble is given by

$$Q^N(V,T)=\frac{h^{-3N}}{N!}\int \exp(-\beta H^N(\vec{\Gamma}^N))\,d\vec{\Gamma}^N, \tag{3.90}$$

and the average value of some property A by

$$\langle A\rangle=\frac{\int A(\vec{\Gamma}^N)\exp(-\beta H^N(\vec{\Gamma}^N))\,d\vec{\Gamma}^N}{\int \exp(-\beta H^N(\vec{\Gamma}^N))\,d\vec{\Gamma}^N}. \tag{3.91}$$

If the Hamiltonian function can be separated into independent kinetic and potential energy terms then it is possible to reduce the partition function to an integral purely over atomic positions. First we write

$$Q^N(V,T)=\frac{h^{-3N}}{N!}\int\int \exp(-\frac{\beta}{2}\sum_{i=1}^{N}\frac{p_i^2}{m})\exp(-\beta\Phi(\vec{r}^N))\,d\vec{r}^N\,d\vec{p}^N, \tag{3.92}$$

where $\Phi(\vec{r}^N)$ is the system potential (i.e. configuration) energy, which is a function of the atomic positions only. The term involving the momenta $\{\vec{p}_i\}$ may be integrated analytically[9], using the standard integral

$$\int_{-\infty}^{\infty}\exp(-ax^2)\,dx=\sqrt{\frac{\pi}{a}} \tag{3.93}$$

to give

9 Note that, for simplicity, we have assumed all atoms have the same mass.

$$Q^N(V,T)=\left(\frac{\Lambda^{-3N}}{N!}\right)Z^N(V,T), \tag{3.94}$$

where

$$\Lambda=\sqrt{\frac{\beta h^2}{2\pi m}} \tag{3.95}$$

is the *thermal de Broglie wavelength* and

$$Z^N(V,T)=\int \exp(-\beta\Phi(\vec{r}^N))\,d\vec{r}^N, \tag{3.96}$$

is the so-called *configuration integral*. A similar reduction may be applied to the ensemble average (3.91), if the property A is also purely a function of atomic position. This shows that it is possible to obtain some system properties by averaging over atomic positions only. Most of the useful properties can be handled in this way.

Note that when $\Phi(\vec{r}^N)=0,$ as in an ideal gas, then

$$Z^N(V,T)=V^N \tag{3.97}$$

where V is the system volume.

The thermodynamic potential of the canonical ensemble is the Helmholz free energy. It is derivable from the standard thermodynamic relation

$$F=U-TS=-\frac{1}{\beta}\log\left(Q^N(V,T)\right), \tag{3.98}$$

in which U,T and S are internal energy, temperature and entropy respectively. If the Hamiltonian function can again be separated into independent kinetic and potential energy terms we can rewrite (3.98) as

$$F=-\frac{1}{\beta}\log\left(\frac{\Lambda^{-3N}V^N}{N!}\right)-\frac{1}{\beta}\log\left(\frac{Z^N(V,T)}{V^N}\right) \tag{3.99}$$

or

$$F=F_{id}+F_{ex} \tag{3.100}$$

where

$$F_{id} = -\frac{1}{\beta}\log\left(\frac{\Lambda^{-3N}V^N}{N!}\right) \tag{3.101}$$

is the *ideal gas* contribution to the free energy and

$$F_{ex} = -\frac{1}{\beta}\log\left(\frac{Z^N(V,T)}{V^N}\right) \tag{3.102}$$

is the so-called *excess* free energy contribution, which arises from inter-molecular interactions.

As has been mentioned already, it is not possible to compute free energy directly by a molecular dynamics simulation. This is because the partition function cannot be obtained merely by following a trajectory through phase space – it requires an integral over the whole of phase space. Only average values of properties along a trajectory can be obtained. However, the relations outlined here can be used as the basis for a more sophisticated approach that will provide, after some effort, the difference in free energy between systems, which is sufficient for most purposes. See chapter 10.

3.18.3 The Isothermal-Isobaric Ensemble (NPT)

This ensemble is appropriate for systems that have both a thermostat and a barostat, to control the temperature and pressure respectively. This complies with most chemistry experiments, which are done under constant temperature and pressure conditions. In molecular dynamics this is known as the NPT ensemble, since it conserves the number of atoms (N), the pressure (P), and the temperature (T). The probability density is written as

$$f^N(\vec{\Gamma}^N, V) = \frac{h^{-3N}}{V_0 N!}\frac{\exp(-\beta(H^N(\vec{\Gamma}^N)+PV))}{Q^N(P,T)}. \tag{3.103}$$

The probability density is now a function of the volume V as well as the phase space variables $\vec{\Gamma}^N$, and the Boltzmann term contains the product PV. The mean volume of the system V_0 appears in the denominator as a normalisation constant, since integrals involving $f^N(\vec{\Gamma}^N, V)$ must also be integrated over the volume. This is apparent in the definition of the partition function $Q^N(P,T)$:

$$Q^N(P,T) = \frac{h^{-3N}}{V_0 N!}\int\int \exp(-\beta(H^N(\vec{\Gamma}^N)+PV))\, d\vec{\Gamma}^N\, dV. \tag{3.104}$$

Once again, the average value of some property A is given by the appropriate ensemble average:

$$\langle A\rangle=\frac{\int\int A(\vec{\Gamma}^N)\exp(-\beta(H^N(\vec{\Gamma}^N)+PV))\,d\,\vec{\Gamma}^N dV}{\int\int \exp(-\beta(H^N(\vec{\Gamma}^N)+PV))\,d\,\vec{\Gamma}^N dV}. \tag{3.105}$$

It can be seen from these formulas that the isothermal-isobaric ensemble incorporates the canonical ensemble. This means that the manipulations applied to the canonical ensemble, such as splitting the Hamiltonian into independent kinetic and potential contributions can be applied here also.

The thermodynamic potential of the isothermal-isobaric ensemble is the Gibbs free energy, which is given by

$$G=F+PV=-\frac{1}{\beta}\log(Q^N(P,T)). \tag{3.106}$$

3.18.4 The Grand Canonical Ensemble (μVT)

In the grand canonical ensemble, the number of atoms in the system is allowed to fluctuate, as well as the temperature. The fluctuation is governed by the *chemical potential* μ, which defines the "thermodynamic cost" of inserting an atom into the system. The insertion and removal of species in a chemical system arises naturally in the phenomena of phase equilibria and the grand canonical ensemble provides the statistical mechanical tools for investigation. The probability density for the grand canonical ensemble appropriate for a system consisting of a *single component species* is:

$$f(\vec{\Gamma}^N;N)=\frac{h^{-3N}}{N!}\frac{\exp(\beta N\mu)\exp(-\beta H^N(\vec{\Gamma}^N))}{\Xi(\mu,V,T)}. \tag{3.107}$$

The probability density is now evidently a function the number of particles $N.$ The partition function is

$$\Xi(\mu,V,T)=\sum_{N=0}^{\infty}\frac{h^{-3N}}{N!}\exp(\beta N\mu)\int\exp(-\beta H^N(\vec{\Gamma}^N))\,d\,\vec{\Gamma}^N. \tag{3.108}$$

The summation over N rather than an integral arises because the particles are necessarily discrete.

The calculation of a system property A can be performed using an ensemble average in the same way as the preceding ensembles:

$$\langle A\rangle=\frac{\sum_{N=0}^{\infty}\exp(\beta N\mu)\int A(\vec{\Gamma}^N)\exp(-\beta H^N(\vec{\Gamma}^N))d\vec{\Gamma}^N}{\sum_{N=0}^{\infty}\exp(\beta N\mu)\int\exp(-\beta H^N(\vec{\Gamma}^N))d\vec{\Gamma}^N}, \tag{3.109}$$

where the main thing to note is that the grand canonical ensemble apparently incorporates the canonical ensemble with every conceivable value of the particle number N and all the reductions and manipulations applicable to the canonical expressions are applicable here also.

The Gibbs free energy of a single component system is defined as

$$G=\mu N=F+PV, \tag{3.110}$$

and the thermodynamic potential (the so-called *grand potential*) is

$$\Omega=-PV=-\frac{1}{\beta}\log(\Xi(\mu,V,T)). \tag{3.111}$$

After all this, it is fair to say that the grand canonical ensemble is not commonly simulated by molecular dynamics; it is much more the province of Monte Carlo simulations. This is not to say that suitable algorithms do not exist (see [10]) but it not commonly done. On the other hand the calculation of the chemical potential μ is a frequent requirement and is often done by the methods employed in the Monte Carlo area [11].

3.18.5 Converting Between Ensembles

If we examine Table 3.1 showing the partition functions and thermodynamic potentials, it is evident that they are related.

Table 3.1: The Common Ensembles

Ensemble	Partition Function	Thermodynamic Potential
NVE	$Q^N(V,E)=\frac{h^{-3N}}{N!}\int\delta\left(H^N(\vec{\Gamma}^N)-E\right)d\vec{\Gamma}^N$	$-TS$
NVT	$Q^N(V,T)=\frac{h^{-3N}}{N!}\int\exp(-\beta H^N(\vec{\Gamma}^N))d\vec{\Gamma}^N$	$F=U-TS$
NPT	$Q^N(P,T)=\frac{h^{-3N}}{V_0 N!}\int\int\exp(-\beta(H^N(\vec{\Gamma}^N)+PV))d\vec{\Gamma}^N dV$	$G=F+PV$

μVT	$\Xi(\mu, V, T) = \sum_{N=0}^{\infty} \frac{h^{-3N}}{N!} \exp(\beta N \mu) \int \exp(-\beta H^N(\vec{\Gamma}^N)) d\vec{\Gamma}^N$	$\Omega = -PV = F - \mu N$

By inspection it can be seen that the partition functions Q are related via a hierarchy of Laplace transforms:

$$Q_{(new)} = \int Q_F \exp(-\beta f F) dF \tag{3.112}$$

and the thermodynamic potentials Ψ are related via a hierarchy of additions

$$\Psi_{(new)} = \Psi_F + fF, \tag{3.113}$$

where f represents an *intensive* property of the system (*T, P* or μ) and F is an *extensive* property of the system (*S, V* or *N*).

3.19 Working with the Partition Function

In this section we show, by example, how important the partition function is to statistical mechanics. These examples may be described a "tricks of the trade" since they provide useful and perhaps unexpected tools. We shall focus on the partition function of the canonical ensemble, since it the simplest example that demonstrates all the required outcomes.

3.19.1 Differentiation of $\log(Q^N(V,T))$ by β.

This provides a link between the system energy U and the partition function.

$$\begin{aligned} \frac{\partial \log(Q^N(V,T))}{\partial \beta} &= \frac{1}{Q^N(V,T)} \frac{\partial Q^N(V,T)}{\partial \beta} \\ &= \frac{-1}{Q^N(V,T)} \int \int H^N(\vec{\Gamma}^N) \exp(-\beta H^N(\vec{\Gamma}^N)) d\vec{\Gamma}^N. \end{aligned} \tag{3.114}$$

From which we have

$$U = \langle H^N \rangle = -\frac{\partial \log(Q^N(V,T))}{\partial \beta}. \tag{3.115}$$

Also, the specific heat at constant volume is defined as

$$C_v = \left(\frac{\partial U}{\partial T} \right)_V , \tag{3.116}$$

so it can easily be shown from (3.115) that

$$C_v = -k_B \beta^2 \left(\langle (H^N(\vec{\Gamma}^N))^2 \rangle - \langle H^N(\vec{\Gamma}^N) \rangle^2 \right). \tag{3.117}$$

Equation (3.117) shows that C_V is related to the *fluctuation* in the system energy.

3.19.2 The Mixed Hamiltonian

A mixed Hamiltonian function is one which combines the Hamiltonians from two different systems A and B, thus

$$H_\lambda^{\tilde{N}}(\vec{\Gamma}^{\tilde{N}}) = (1-\lambda) H_A^N(\vec{\Gamma}^N) + \lambda H_B^N(\vec{\Gamma}^N), \tag{3.118}$$

in which $H_A^N(\vec{\Gamma}^N)$ and $H_B^N(\vec{\Gamma}^N)$ are the different Hamiltonians and λ is a "mixing" parameter, which ranges in value from 0 to 1. Note that the mixed Hamiltonian is dependent on the number of atoms $\tilde{N} \leqslant 2N$ which, at most, can be $2N$ atoms but possibly fewer if systems A and B have atoms in common, as is often the case. Inserting the mixed Hamiltonian into the partition function and differentiating with respect to λ gives

$$\begin{aligned} \frac{\partial \log(Q_\lambda^{\tilde{N}}(V,T))}{\partial \lambda} = \\ -\frac{\beta}{Q_\lambda^{\tilde{N}}(V,T)} \int (H_B^N(\vec{\Gamma}^N) - H_A^N(\vec{\Gamma}^N)) \exp(-\beta H_\lambda^{\tilde{N}}(\vec{\Gamma}^{\tilde{N}})) d\vec{\Gamma}^{\tilde{N}} \\ = -\beta \langle \Delta H_{(AB)}^{\tilde{N}}(\vec{\Gamma}^{\tilde{N}}) \rangle_\lambda , \end{aligned} \tag{3.119}$$

In this formula $Q_\lambda^{\tilde{N}}(V,T)$ is the partition function for the mixed Hamiltonian system and the suffix λ on the angular brackets indicates that the ensemble average is calculated using the mixed Hamiltonian with a fixed value of λ.

From the relationships (3.98) and (3.119) we therefore have that

$$\frac{\partial F}{\partial \lambda} = \langle \Delta H_{(AB)}^{\tilde{N}}(\vec{\Gamma}^{\tilde{N}}) \rangle_\lambda \quad \text{or} \quad \Delta F_{(AB)} = \int_{\lambda=0}^{\lambda=1} \langle \Delta H_{(AB)}^{\tilde{N}}(\vec{\Gamma}^{\tilde{N}}) \rangle_\lambda d\lambda . \tag{3.120}$$

The integral presented in (3.120) is extremely important. It provides a means to calculate the *free energy difference* between the two thermodynamic systems (or

states) represented by the Hamiltonians $H_A^N(\vec{\Gamma}^N)$ and $H_B^N(\vec{\Gamma}^N)$. This is the subject of chapter 10.

3.19.3 Pressure Calculations

This application of the partition function depends on the standard thermodynamic relationship between the pressure and the volume derivative of the free energy:

$$P=-\left(\frac{\partial F}{\partial V}\right)_T. \tag{3.121}$$

Using the fundamental relation (3.98) we have

$$P=\frac{-1}{Q^N(V,T)}\int\left(\frac{\partial H^N(\vec{\Gamma}^N)}{\partial V}\right)_T \exp(-\beta H^N(\vec{\Gamma}^N))d\vec{\Gamma}^N, \tag{3.122}$$

and so we arrive at the important relationship

$$P=-\left\langle\left(\frac{\partial H^N(\vec{\Gamma}^N)}{\partial V}\right)_T\right\rangle. \tag{3.123}$$

From this we see that the pressure is the ensemble average of the derivative of the system Hamiltonian with respect to the volume. This is a useful formula for calculating the pressure in a wide variety of systems. The subject is explored in greater detail in chapter 7.

3.19.4 The Bias Potential

Ensemble averages in the form (3.91) can be easily modified by the addition and subtraction of an arbitrary potential $\Phi_b(\Gamma^N)$, which is commonly called a bias potential, to give a form which is useful in several ways. Thus (3.91) becomes

$$\langle A\rangle=\frac{\int A(\vec{\Gamma}^N)\exp(-\beta(H^N(\vec{\Gamma}^N)+\Phi_b(\vec{\Gamma}^N)-\Phi_b(\vec{\Gamma}^N)))d\vec{\Gamma}^N}{\int \exp(-\beta(H^N(\vec{\Gamma}^N)+\Phi_b(\vec{\Gamma}^N)-\Phi_b(\vec{\Gamma}^N)))d\vec{\Gamma}^N}. \tag{3.124}$$

The exponents may be separated to give

$$\langle A\rangle=\frac{\int [A(\vec{\Gamma}^N)\exp(\beta\Phi_b(\vec{\Gamma}^N))]\exp(-\beta(H^N(\vec{\Gamma}^N)+\Phi_b(\vec{\Gamma}^N)))d\vec{\Gamma}^N}{\int [\exp(\beta\Phi_b(\vec{\Gamma}^N))]\exp(-\beta(H^N(\vec{\Gamma}^N)+\Phi_b(\vec{\Gamma}^N)))d\vec{\Gamma}^N}. \tag{3.125}$$

Noting the terms in square brackets, this expression may readily be written as the ratio of two ensemble averages

$$\langle A\rangle=\frac{\langle A(\vec{\Gamma}^N)\exp(\beta\Phi_b(\vec{\Gamma}^N))\rangle_b}{\langle\exp(\beta\Phi_b(\vec{\Gamma}^N))\rangle_b} \tag{3.126}$$

This formula shows that it is possible to obtain the average of some property A from a simulation in which the Hamiltonian $H^N(\vec{\Gamma}^N)$ is augmented by a bias potential $\Phi_b(\vec{\Gamma}^N)$ and yet obtain the result pertaining to the original Hamiltonian, provided that the quantities in the angled brackets are used to compute the average. (The suffix $\langle\rangle_b$ in (3.126) indicates that the ensemble averages are computed in the biased system.)

The value of this formulation is that there are often occasions when the property A is associated with configurations that are rare. For example the activation energy of a chemical process is achieved only by rare configurations of atoms, so the attempt to obtain an accurate estimate of the energy depends on these configurations being encountered an impractically large number of times in a simulation. This situation can be much improved if a bias potential can be used to encourage the system to spend a greater proportion of time in the vicinity of the important configurations. Important molecular dynamics methods that exploit a bias potential in this way are *hyperdynamics* and *metadynamics*, which are described in chapter 11.

3.19.5 Thermodynamic Perturbation

Here we consider the case where a base Hamiltonian $H_0^N(\vec{\Gamma}^N)$ is augmented by the addition of another Hamiltonian $H_1^N(\vec{\Gamma}^N)$ which can be regarded as a small perturbation on the system. In which case we assume that the phase space explored by a simulation of the extended system closely matches that of the base system. This is equivalent to saying we could use the configurations generated in the base simulation to provide reasonably accurate estimates of the properties of the extended system. This is done as follows

Let

$$H_\lambda^N(\vec{\Gamma}^N)=H_0^N(\vec{\Gamma}^N)+\lambda H_1^N(\vec{\Gamma}^N) \tag{3.127}$$

Where $H_0^N(\vec{\Gamma}^N)\gg H_1^N(\vec{\Gamma}^N)$ and $0\leqslant\lambda\leqslant 1$ is a "switching" factor that "turns on" the perturbing Hamiltonian H_1 in a continuous manner. Then for some property A we

have from (3.72)

$$\langle A\rangle_\lambda = \frac{\int A(\vec{\Gamma}^N)\exp(-\beta[H_0^N(\vec{\Gamma}^N)+\lambda H_1^N(\vec{\Gamma}^N)])\,d\vec{\Gamma}^N}{\int \exp(-\beta[H_0^N(\vec{\Gamma}^N)+\lambda H_1^N(\vec{\Gamma}^N)])\,d\vec{\Gamma}^N}. \tag{3.128}$$

which is the ensemble average for the extended system – hence the suffix λ on the angled bracket. (Note that in what follows, for reasons of clarity and economy, we shall cease writing the argument $(\vec{\Gamma}^N)$ associated with the Hamiltonians and the property *A*.)

Now, differentiating (3.128) with respect to λ we obtain

$$\begin{aligned}\frac{\partial\langle A\rangle_\lambda}{\partial\lambda} = &\frac{-\beta}{Q_\lambda^N}\int AH_1^N\exp(-\beta[H_0^N+\lambda H_1^N])\,d\vec{\Gamma}^N + \\ &\frac{\beta}{Q_\lambda^N}\langle A\rangle_\lambda\int H_1^N\exp(-\beta[H_0^N+\lambda H_1^N])\,d\vec{\Gamma}^N,\end{aligned} \tag{3.129}$$

where for brevity we have used $Q_\lambda^N \equiv Q_\lambda^N(\vec{\Gamma}^N)$ as the partition function for the system including the perturbing Hamiltonian. This result can be expressed more simply as

$$\frac{1}{\beta}\frac{\partial\langle A\rangle_\lambda}{\partial\lambda} = -\langle AH_1^N\rangle_\lambda + \langle A\rangle_\lambda\langle H_1^N\rangle_\lambda. \tag{3.130}$$

In the same way we can proceed to the second (and higher derivatives when required), so

$$\frac{1}{\beta^2}\frac{\partial^2\langle A\rangle_\lambda}{\partial\lambda^2} = \langle A(H_1^N)^2\rangle_\lambda - 2\langle AH_1^N\rangle_\lambda\langle H_1^N\rangle_\lambda + 2\langle A\rangle_\lambda\langle H_1^N\rangle_\lambda^2 - \langle A\rangle_\lambda\langle(H_1^N)^2\rangle_\lambda. \tag{3.131}$$

We now consider the Taylor expansion of $\langle A\rangle_\lambda$ in terms of the parameter λ, which is (to second order)

$$\langle A\rangle_\lambda = \langle A\rangle_{(\lambda=0)} + \lambda\left(\frac{\partial\langle A\rangle_\lambda}{\partial\lambda}\right)_{(\lambda=0)} + \frac{\lambda^2}{2}\left(\frac{\partial^2\langle A\rangle_\lambda}{\partial\lambda^2}\right)_{(\lambda=0)} + O(\lambda^3). \tag{3.132}$$

We are assuming that the derivatives associated with terms of order λ^3 and higher are negligible. Substituting into this the derivatives obtained above gives

$$\begin{aligned}\langle A\rangle_\lambda \approx \langle A\rangle_0 - \lambda\beta(\langle AH_1^N\rangle_0 - \langle A\rangle_0\langle H_1^N\rangle_0) + \\ \frac{\lambda^2\beta^2}{2}\left(\langle A(H_1^N)^2\rangle_0 - 2\langle AH_1^N\rangle_0\langle H_1^N\rangle_0 + 2\langle A\rangle_0\langle H_1^N\rangle_0^2 - \langle A\rangle_0\langle(H_1^N)^2\rangle_0\right).\end{aligned} \tag{3.133}$$

Now setting $\lambda=1$ means $\langle A\rangle_\lambda \rightarrow \langle A\rangle_1$ which is the property A evaluated in the perturbed ensemble. However all the terms on the right are evaluated in the base ensemble and provided we can use configurations from that ensemble to calculate H_1^N which is the energy of the perturbation, we can obtain an estimate of $\langle A\rangle_1$. Sometimes, if the perturbation is extremely small, only the term to order λ is used. This expansion can be used to prove the important fluctuation-dissipation theorem (see section 3.23).

3.20 The Liouville Theorem

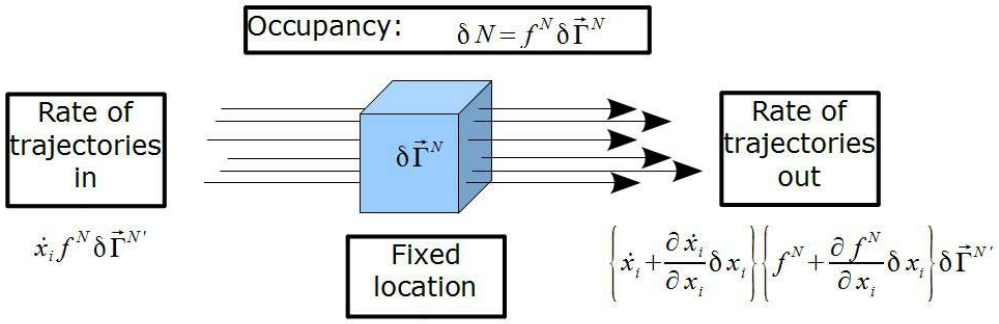

Figure 3.14: The Liouville Theorem

In section 3.16 and figure 3.13 we suggested that the ensemble can be represented as a cluster of points in phase space and the trajectory of each point through phase space represents the time evolution of the ensemble. Furthermore, the density of points in phase space is a realisation of the probability density of the ensemble, which we described in section 3.17 . We will now discuss the collective motion of the ensemble as if it was a fluid flowing in phase space. This will allow us to treat time dependent properties of the ensemble as a manifestation of flow of probability density rather than the result of the mechanics of individual molecules and so provide new perspective on the subject. This brings us to the Liouville theorem.

Figure 3.14 shows a small volume of phase space $\delta\vec{\Gamma}^N$, positioned at coordinate x_i, through which a number of trajectories of the ensemble flow. At a given instant of time t the number of phase space points (or members of the ensemble) δN inside this volume is given by $f^N \delta\vec{\Gamma}^N$. Considering the y-z faces of the volume, the rate of phase points flowing into the volume through one y-z face is $\dot{x}_i f^N \delta\vec{\Gamma}^{N'}$ (where $N'=N-1$ represents all coordinates except x_i) and the rate of phase points leaving through the opposite face is $\left\{\dot{x}_i+\frac{\partial \dot{x}_i}{\partial x_i}\delta x_i\right\}\left\{f^N+\frac{\partial f^N}{\partial x_i}\delta x_i\right\}\delta\vec{\Gamma}^{N'}$ which allows for any change in $\dot{x}_i$ and f^N that occurs inside the volume. The difference between the rate of inflowing and outflowing points give the rate of change of the

occupancy, thus:

$$\frac{d(\delta N)}{dt}=-\left\{f\frac{\partial \dot{x}_i}{\partial x_i}+\dot{x}_i\frac{\partial f^N}{\partial x_i}\right\}\delta\vec{\Gamma}^N, \tag{3.134}$$

(where $\delta\vec{\Gamma}^N=\delta\vec{\Gamma}^{N'}\delta x_i$). A similar expression applies to the momentum coordinates:

$$\frac{d(\delta N)}{dt}=-\left\{f\frac{\partial \dot{p}_i}{\partial p_i}+\dot{p}_i\frac{\partial f^N}{\partial p_i}\right\}\delta\vec{\Gamma}^N. \tag{3.135}$$

So generally, for all phase space coordinates we have

$$\left(\frac{\partial f^N}{\partial t}\right)_{\vec{\Gamma}^N}=-\sum_{i=1}^{3N}\left\{f^N\left(\frac{\partial \dot{x}_i}{\partial x_i}+\frac{\partial \dot{p}_i}{\partial p_i}\right)+\left(\dot{x}_i\frac{\partial f^N}{\partial x_i}+\dot{p}_i\frac{\partial f^N}{\partial p_i}\right)\right\}, \tag{3.136}$$

where we have used the equivalence

$$\left(\frac{\partial f^N}{\partial t}\right)_{\vec{\Gamma}^N}\delta\vec{\Gamma}^N\equiv\sum_i^N\frac{d(\delta N)}{dt}, \tag{3.137}$$

which applies because the rate of change of the occupancy of the volume $\delta\vec{\Gamma}^N$ is equivalent to the rate of change of probability density at the location $\vec{\Gamma}^N$. This is a partial derivative, as it evaluated at a fixed point in phase space.

From Hamilton's equations of motion (2.17) we have that

$$\dot{x}_i=\frac{\partial H^N}{\partial p_i}\quad\text{and}\quad\dot{p}_i=-\frac{\partial H^N}{\partial x_i} \tag{3.138}$$

so the first term right of (3.136) becomes zero, leaving

$$\left(\frac{\partial f^N}{\partial t}\right)_{\vec{\Gamma}^N}=-\sum_{i=1}^{3N}\left(\frac{\partial H^N}{\partial p_i}\frac{\partial f^N}{\partial x_i}-\frac{\partial H^N}{\partial x_i}\frac{\partial f^N}{\partial p_i}\right). \tag{3.139}$$

This is the famous Liouville equation, which plays a major role in modern statistical mechanics (see section 3.22).

If we consider the *total* time derivative of the probability density function $f^N(\vec{\Gamma}^N;t)$ we have

$$\frac{df^N}{dt}=\left(\frac{\partial f^N}{\partial t}\right)_{\vec{\Gamma}^N}+\sum_{i=1}^{3N}\left(\dot{x}\frac{\partial f^N}{\partial x_i}+\dot{p}_i\frac{\partial f^N}{\partial p_i}\right). \tag{3.140}$$

From Liouville's equation (3.139) we easily see that

$$\frac{df^N}{dt}=0. \tag{3.141}$$

This theorem can be interpreted in a number of ways. Firstly it shows that the probability density function is constant in time (for a conservative Hamiltonian system). Secondly it shows that the density of phase points around any flowing point in phase space is constant (i.e. the probability density flows like an incompressible fluid) Thirdly, it shows that at any fixed point in phase space, the rate of change of probability density is due entirely to the flow in and out of the infinitesimal volume surrounding that point and is not due to expansion or compression of the fluid. Equation (3.141) therefore represents an important restriction on the dynamics of the probability density.

3.21 The Formal Solution of Liouville's Equation

We define an operator[10] $i\hat{L}$ in the following manner

$$i\hat{L}=\sum_{j=1}^{3N}\left(\frac{\partial H^N}{\partial p_j}\frac{\partial}{\partial x_j}-\frac{\partial H^N}{\partial x_j}\frac{\partial}{\partial p_j}\right) \tag{3.142}$$

In which case Liouville's equation (3.139) can be written as

$$\frac{\partial f^N}{\partial t}=-i\hat{L}f^N \tag{3.143}$$

which has the formal solution

$$f^N(\vec{\Gamma}^N;t)=\exp(-it\hat{L})f^N(\vec{\Gamma}^N;0). \tag{3.144}$$

The term $\exp(-it\hat{L})$ is called a *propagator* since it propagates the probability density $f^N(\vec{\Gamma}^N;0)$ at time 0 to $f^N(\vec{\Gamma}^N;t)$ at time $t.$ The meaning of the exponential is defined by the Maclaurin series

10 The inclusion of the imaginary factor i in these formulas is merely a convention to link all propagators, classical and quantum.

$$\exp(-it\,\hat{L}) = 1 - it\,\hat{L} + \frac{1}{2}(it\,\hat{L})^2 - \frac{1}{3!}(it\,\hat{L})^3 + \cdots etc \tag{3.145}$$

in which terms like $(it\,\hat{L})^n$ mean the successive application of the operator n times. This formal solution turns out to have real application in the development of molecular dynamics integration algorithms, as will be seen in chapter 4.

The propagator may also be used to advance system properties in time. So if a property is defined as $A(\vec{r}^N, \vec{p}^N)$ then by the normal rules of differentiation we have

$$\frac{dA}{dt} = \sum_{j=1}^{3N} \left(\dot{x}_j \frac{\partial A}{\partial x_j} + \dot{p}_j \frac{\partial A}{\partial p_j} \right), \tag{3.146}$$

which with Hamilton's equations of motion (2.17) becomes

$$\frac{dA}{dt} = \sum_{j=1}^{3N} \left(\frac{\partial H^N}{\partial p_j} \frac{\partial A}{\partial x_j} - \frac{\partial H^N}{\partial x_j} \frac{\partial A}{\partial p_j} \right), \tag{3.147}$$

from which we obtain

$$\frac{dA}{dt} = i\,\hat{L}\,A, \tag{3.148}$$

for which the formal solution is

$$A(\vec{r}^N, \vec{p}^N; t) = \exp(it\,\hat{L})\,A(\vec{r}^N, \vec{p}^N; 0). \tag{3.149}$$

3.22 The Liouville Equation in Statistical Mechanics

The Liouville equation is the starting point for many research areas in statistical mechanics, some of which are shown in figure 3.15. Firstly we mention the development of the Liouville operator $i\,\hat{L}$ as a means of generating numerical integration algorithms (see chapter 4). Next it plays a role in the development of response theory and the calculation of transport properties by the methods of Non-equilibrium Molecular Dynamics (NEMD) [12]. It also provides a foundation for the Boltzmann equation ([1] pp. 13-19), which is the basis for the Lattice Boltzmann simulation method which, as its name suggests, is a lattice based method for modelling continuum fluids [13]. Similarly it can be used to derive the Fokker-Planck equation, which provides the theoretical basis for the mesoscale method known as dissipative Particle Dynamics (DPD) [14]. Lastly we mention its role underpinning the Vlasov equation ([1] p. 17), which has applications in modelling plasma dynamics.

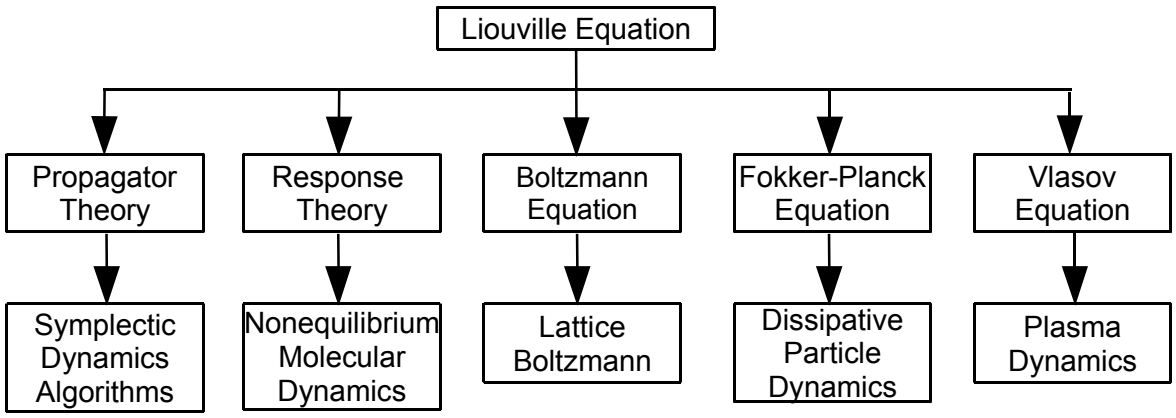

Figure 3.15. The Liouville Equation in Statistical Mechanics

3.23 The Fluctuation-Dissipation Theorem

This important theory has applications in the calculation of transport properties, stochastic processes and spectroscopy. There are many processes in chemical physics in which a sudden, but small, perturbation is made to the system, often by a field of some kind, followed by the system relaxing to the equilibrium state. In order to describe processes of this sort a basic assumption is made, due to Onsager, that the relaxation process is the same as that which occurs following a normal thermodynamic fluctuation. Such a relaxation in the equilibrium system is of course described by the appropriate correlation function. For a given property $A(t),$ written as a function of time $t,$ Onsager's hypothesis is expressed by the relations:

$$\frac{\Delta A_{(\exp)}(t)}{\Delta A_{(\exp)}(0)}=\frac{\left(A_{(\exp)}(t)-\bar{A}_{(\exp)}\right)}{\left(A_{(\exp)}(0)-\bar{A}_{(\exp)}\right)}=\frac{\langle \delta A(t)\delta A(0)\rangle}{\langle \delta A^2\rangle}=\frac{C_A(t)}{C_A(0)}, \tag{3.150}$$

in which $A_{(\exp)}$ is the experimental value of property $A,$ which corresponds to an ensemble average value. $\bar{A}_{(\exp)}$ is the equilibrium value of A to which $A_{(\exp)}(t)$ relaxes at long time and $C_A(t)$ is the autocorrelation function of the property A and $C_A(0)$ is its value at zero time. This relationship is formally proved by the theorem below (see also [15], chapter 8), which is called the *fluctuation-dissipation* theorem.

The theory is developed from the experiment which is shown in figure 3.16. The initial system is governed by a Hamiltonian function H_S which contains a small perturbation $H_1,$ which is generated by a field F that couples to the system. Thus

$$H_1=-AF \tag{3.151}$$

where the property A appears as a coupling parameter.

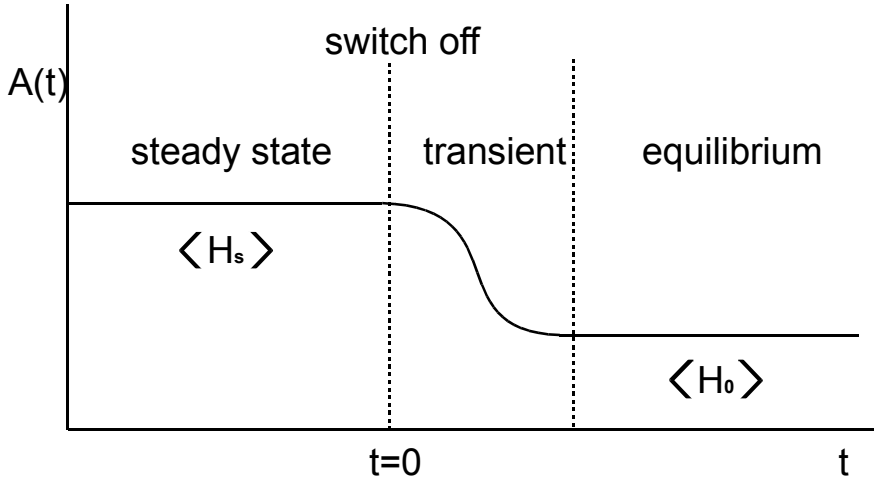

Figure 3.16: Perturbed System Relaxing to Equilibrium

The Hamiltonian for the perturbed system is given by

$$H_S = H_0 + H_1 = H_0 - AF. \tag{3.152}$$

Where H_0 is the base Hamiltonian of the system – without perturbations of any kind. In the experiment, the Hamiltonian H_S is applied for a period of time until the system achieves a *steady state*, which is characterised by the system properties holding unchanging values. Then at a time designated as $t=0,$ the perturbation is switched off and the system relaxes to the equilibrium state governed by the base Hamiltonian $H_0.$

At the time $t=0$ the experimental value $A_{(\exp)}(0)$ is given by the ensemble average $\langle A(0)\rangle_S$ which is calculated over the ensemble of $H_S.$ Similarly the relaxing value of $A(t)$ is given by $\langle A(t)\rangle_S,$ which is the average value of $A(t)$ in the ensemble of $H_S.$

We now employ the perturbation theory outlined in section 3.19.5 and write

$$\langle A(t)\rangle_S = \langle A\rangle_0 - \beta\left(\langle A(t)H_1\rangle_0 - \langle A\rangle_0\langle H_1\rangle_0\right) + O\left((\beta H_1)^2\right). \tag{3.153}$$

The perturbation H_1 is defined at time $t=0,$ so we have $H_1 = -A(0)F,$ which means we can rearrange (3.153) to

$$\langle A(t)\rangle_S - \langle A\rangle_0 = \beta F\left(\langle A(t)A(0)\rangle_0 - \langle A(0)A(0)\rangle_0\right) + O(F^2), \tag{3.154}$$

which is equivalent to

$$\Delta A_{(\exp)}(t)=\beta F\langle \delta A(t)\delta A(0)\rangle_0+O(F^2) \tag{3.155}$$

Likewise we can also write

$$\Delta A_{(\exp)}(0)=\beta F\langle \delta A(0)\delta A(0)\rangle_0+O(F^2). \tag{3.156}$$

Taking the ratio of (3.155) and (3.156) gives the result

$$\frac{\Delta A_{(\exp)}(t)}{\Delta A_{(\exp)}(0)}=\frac{\langle \delta A(t)\delta A(0)\rangle}{\langle \delta A^2\rangle}=\frac{C(t)}{C(0)}, \tag{3.157}$$

which confirms Onsager's hypothesis. The key result is that the response function for a sudden, small perturbation is the appropriate correlation function.

Chapter 4

Integration Algorithms

4.1 Introduction

At the heart of any molecular dynamics program is the integration algorithm by means of which the equations of motion for the molecular system are integrated to obtain the time evolution of the system. As was described in chapter 1, the integration algorithms, at least for systems with analytical or continuous potentials[11], are discrete algorithms that advance the system through a series of short time steps (Δt). A typical time step is of order $1 \sim 10$ fs and a complete simulation requires many time steps (N_T), of order 10^4 to 10^9. The physical time period (T) that this corresponds to is of course $T = N_T \Delta t$ which turns out to be extremely short on a human time scale but is often extremely long in computation time.

In this chapter we will begin with algorithms for translational motion and progress to rotational motion in the later sections. We will conclude with a look at algorithms for generating particular ensembles.

The basic idea of a *translational* integration algorithm is presented in figure 4.1. An atom at position $\vec{r}^n$ at the $n'th$ time step has a velocity $\vec{v}^n$ and a force $\vec{f}^n$. In the

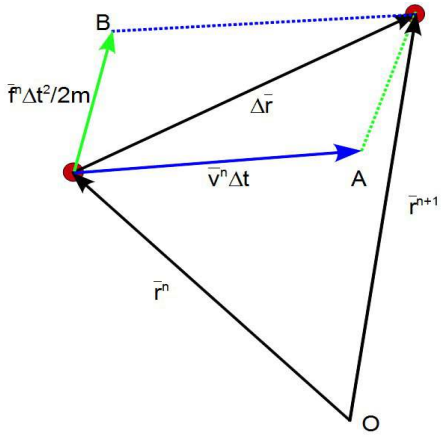

Figure 4.1. The Essentials of Numerical Integration

time interval Δt the velocity alone, if constant, would move the atom to the position A in the figure, which corresponds to a vector displacement of $\vec{v}^n \Delta t$. Likewise, the force alone, if constant, would move the atom to position B, a vector displacement of $\vec{f}^n \Delta t^2 / 2m$ (according to Newton's second law), where m is the atomic mass. Therefore as a first approximation, the actual displacement of the atom $(\Delta \vec{r})$ is the

11 We will not be including algorithms for discontinuous potentials or "hard sphere"- like models in this book.

vector sum of these two displacements. The new position of the atom is thus

$$\begin{aligned}\vec{r}^{\,n+1} &= \vec{r}^{\,n} + \Delta\vec{r} \\ &= \vec{r}^{\,n} + \vec{v}^{\,n}\Delta t + \vec{f}^{\,n}\Delta t^2/2\mathrm{m}\end{aligned} \tag{4.1}$$

and, by similar reasoning, the new velocity is

$$\vec{v}^{\,n+1} = \vec{v}^{\,n} + \vec{f}^{\,n}\Delta t/m. \tag{4.2}$$

At the new position the forces acting on the atom may be recomputed and another step in time made. Repetition of this N_T times comprises the basic integration algorithm. This would be acceptable if the values of $\vec{v}^{\,n}$ and $\vec{f}^{\,n}$ changed negligibly over the interval Δt but, unfortunately, this is not usually the case. So the immediate question is: how good is this as an approximation? Analysing this issue and compensating for the inadequacies found is the basis for constructing every integration algorithm.

In molecular dynamics the most successful algorithms are the so-called Verlet algorithms, of which there are three variants: the original Verlet algorithm [3], the leapfrog algorithm [16] and the velocity Verlet algorithm [17]. In truth these algorithms are not much more advanced than the crude algorithm just described, but they have proven to be highly effective and surprisingly versatile. The reasons for these statements will become apparent later in this chapter. Of course, alternative algorithms exist, but use of these appears to have fallen away as the Verlet forms have risen in popularity. The only "rival" method we shall mention is the Gear predictor-corrector [18] which still finds use in non-equilibrium molecular dynamics applications. It is, in fact, a more accurate method than the Verlet variants, but is less stable over a long time scale, for reasons we will discover.

Most, if not all, integration algorithms derive from the Taylor expansion. If we represent the trajectory of an atom i by the time dependent position vector $\vec{r}_i(t)$, we can write an expansion for this in terms of the time interval Δt as

$$\vec{r}_i(t+\Delta t) = \vec{r}_i(t) + \dot{\vec{r}}_i(t)\Delta t + \ddot{\vec{r}}_i(t)\frac{\Delta t^2}{2!} + \dddot{\vec{r}}_i(t)\frac{\Delta t^3}{3!} + \ddddot{\vec{r}}_i(t)\frac{\Delta t^4}{4!} + O(\Delta t^5), \tag{4.3}$$

where $O(\Delta t^5)$ indicates the order of error caused by truncation after the Δt^4 term and the number of dots above each vector symbolises the degree of differentiation of $\vec{r}_i$ with respect to time. Since the time t evolves discretely in the overall numerical scheme, we can write $t = n\Delta t$ where n is some integer (indicating the $n'th$ time step) and re-write (4.3) as

$$\vec{r}_i^{\,n+1} = \vec{r}_i^{\,n} + \dot{\vec{r}}_i^{\,n}\Delta t + \ddot{\vec{r}}_i^{\,n}\frac{\Delta t^2}{2!} + \dddot{\vec{r}}_i^{\,n}\frac{\Delta t^3}{3!} + \ddddot{\vec{r}}_i^{\,n}\frac{\Delta t^4}{4!} + O(\Delta t^5). \tag{4.4}$$

Thus, if we knew all the time derivatives of $\vec{r}_i$ at the $n'th$ time step, we could in principle calculate exactly the position of the atom at the $n+1'th$ time step. Of course this is not possible, since there is an infinity of such derivatives, but if we truncate the series at some order in $\Delta t,$ we at least have some idea of the accuracy of the approximation we are making – in this example there is an error of order $O(\Delta t^5).$

Assuming we make such a truncation, we then need to calculate the time derivatives of $\vec{r}_i$ to the required order. In general we do not have a means to obtain these directly, so they are often estimated using knowledge of the previous positions of the atom along the trajectory (i.e. using the finite differences in $\vec{r}_i$ from step to step). It follows that, beyond the initial truncation of the Taylor expansion, the accuracy of the algorithm depends on the approximations made in estimating the higher derivatives. The differences between algorithms are largely concerned with how this is done.

Approximating the time derivatives by finite differences is clearly not a self starting approach, as we do not know the atom positions for the first few time steps, but if these can be obtained somehow, then the algorithm can evolve on its own from then onwards. Fortunately, in molecular dynamics we are primarily concerned with the long time evolution of the system, so it is permissible to set the higher derivatives – those beyond the second derivative – to zero to get the algorithm started. The first derivative may be equated with the initial velocity and the second with the initial acceleration (obtained from the force) to permit the algorithm to start. Once the trajectory has progressed some distance, the initial errors arising from this choice steadily diminish in statistical significance, though not in the accuracy of the trajectory.

We might well ask at this stage what properties of an integration algorithm make it most useful? If we assume we compare algorithms using the *same time step* Δt in the trial simulations, the following issues can be meaningfully compared.

- **Stability.** The integration algorithm must be computationally stable. Simulations are often required to run for a long time. If the program "crashes" easily for reasons unrelated to the physics, it is a frustrating business to have to reset the simulation running again, probably with the suspicion of further crashes to follow. We should also consider stability in the thermodynamic sense. Every simulation corresponds to a particular state point in the thermodynamics of the system and algorithms that tend to drift away from the required state point due to some inherent instability are unacceptable.

- **Accuracy.** Clearly we would like the algorithm to follow the "true" trajectories of the constituent atoms as this would provide the most faithful representation of the physics. However, a distinction needs to be drawn between accuracy in the computed trajectories and accuracy in the calculated properties. Algorithms that possess greater accuracy in the computed trajectories, may also be subject to systematic drift, which renders the calculated properties less reliable. So in the context of molecular dynamics it is an absence of drift in the computed properties that is arguably the better determinant of usefulness than accuracy

in trajectory.

- **Speed.** Which means *computational speed*. We require that the simulation explore as much as possible of the trajectory in a finite simulation to guarantee the results are statistically significant. So between two stable and equally accurate algorithms (for the given time step) the one which requires less time to compute the required number of time steps is the one preferred. Practically this boils down to the relative amounts of computational arithmetic the algorithms undertake in computing each time step.

- **Memory.** Different algorithms usually have different data storage requirements and it follows that the one which requires least computer memory is preferred. One is tempted to say that this matters less these days, given the enormous dynamic memories possessed by modern computers, but unfortunately the modern trend is to simulate ever larger systems, and this requirement continues to place limitations on what can be done with a given algorithm.

- **Time Reversibility.** The laws of physics are time reversible, which means that replacing time t by -t in the equations of motion, should cause the atomic trajectories to reverse exactly along the same path. In practise this may not be realisable due to the extreme sensitivity of the dynamics of many particle systems to tiny numerical differences, but in principle, and for a short time at least, this should be the case. However, many algorithms do not have this property, so it is not possible even in principle to reverse the trajectories. It matters because fundamental properties of the dynamical system, such as conservation laws, cannot be guaranteed if the integrator does not have this property.

- **Simplecticness.** Symplectic algorithms possess in-built long term stability. The property[12] can be related to the time reversibility of the Liouville equation [19], which guarantees the conservation of the Boltzmann distribution function and, if this is conserved, then it follows that the properties that are calculated from it, such as the system energy, will also be conserved. Symplectic integrators solve exactly the dynamics of a discrete Hamiltonian that is a close approximation to the true Hamiltonian of the system, so the properties of the solution never deviate far from those of the real system. It is not a property possessed by all integration algorithms but it is clearly highly desirable. Remarkably this is something the Verlet algorithms do possess, which goes a long way towards explaining their success.

In comparing algorithms using the above criteria, one should not forget the possibility that a fast algorithm with low accuracy may, with a reduction in time step, become sufficiently accurate to compete with a slower, more accurate algorithm using the larger time step. The key issue is the computational time required to explore the accessible volume of phase space, not the number of time steps required to do it.

12 Mathematically, symplecticness is the property that the product $d\vec{p} \times d\vec{r}$ is a conserved quantity.

4.2 The Verlet Algorithms

In this section we will derive the various forms of the Verlet algorithm. We will do this in a manner which makes clear how the order of accuracy is determined. The reader is advised to keep an eye on the truncation terms $O(\Delta t^n)$ and the effect of multiplying or dividing this by Δt in places.

We will start with the original algorithm used by Verlet [3]. This is obtained from the Taylor expansion. As we have seen above, the position of an atom at time step $n+1$ is given by equation (4.4). In a similar manner, we may construct an expansion representing a step backwards in time to time step $n-1$ as

$$\vec{r}_i^{\,n-1} = \vec{r}_i^{\,n} - \dot{\vec{r}}_i^{\,n}\Delta t + \ddot{\vec{r}}_i^{\,n}\frac{\Delta t^2}{2!} - \dddot{\vec{r}}_i^{\,n}\frac{\Delta t^3}{3!} + \ddddot{\vec{r}}_i^{\,n}\frac{\Delta t^4}{4!} + O(\Delta t^5). \tag{4.5}$$

Adding equations (4.4) and (4.5) together gives

$$\vec{r}_i^{\,n+1} + \vec{r}_i^{\,n-1} = 2\vec{r}_i^{\,n} + \ddot{\vec{r}}_i^{\,n}\Delta t^2 + 2\ddddot{\vec{r}}_i^{\,n}\frac{\Delta t^4}{4!} + O(\Delta t^6). \tag{4.6}$$

Truncating the right side of (4.6) to $O(\Delta t^4)$ and rearranging gives

$$\vec{r}_i^{\,n+1} = 2\vec{r}_i^{\,n} - \vec{r}_i^{\,n-1} + \ddot{\vec{r}}_i^{\,n}\Delta t^2 + O(\Delta t^4). \tag{4.7}$$

We now recognise the acceleration $\ddot{\vec{r}}_i$ and use Newton's second law relating acceleration and force to write (4.7) as

$$\vec{r}_i^{\,n+1} = 2\vec{r}_i^{\,n} - \vec{r}_i^{\,n-1} + \frac{1}{m_i}\vec{f}_i^{\,n}\Delta t^2 + O(\Delta t^4), \tag{4.8}$$

where m_i is the atomic mass and $\vec{f}_i^{\,n}$ is the force acting on atom i and is a function of the atomic positions at the $n'th$ time step i.e. $\vec{f}_i^{\,n} \equiv \vec{f}_i(\{\vec{r}_i^{\,n}\})$. This is the Verlet *position integrator* and to use it we need to know the atomic positions at time steps n and $n-1$ and the atomic force at time step n. It is evidently accurate to the fourth order in Δt. To obtain the corresponding Verlet *velocity integrator* we *subtract* equation (4.5) from (4.4) to obtain

$$\vec{r}_i^{\,n+1} - \vec{r}_i^{\,n-1} = 2\dot{\vec{r}}_i^{\,n}\Delta t + O(\Delta t^3), \tag{4.9}$$

where we have truncated the terms above $O(\Delta t^3)$. Dividing through by $2\Delta t$ leads to a definition of velocity $\vec{v}_i^{\,n}$ as

$$\vec{v}_i^n = \frac{1}{2\Delta t}(\vec{r}_i^{n+1} - \vec{r}_i^{n-1}) = \dot{\vec{r}}_i^n + O(\Delta t^2), \tag{4.10}$$

which we see approximates the true velocity $\dot{\vec{r}}_i^n$ to an accuracy of $O(\Delta t^2)$. Note that the calculation of $\vec{v}_i^n$ takes place *after* the position $\vec{r}_i^{n+1}$ has been determined. (A caution is necessary here – when calculating the difference between two position vectors, as in (4.10), it is the *minimum image* difference that should be taken. This point is easy to overlook!)

To start a simulation using the Verlet integrators, we assign each atom an initial position $\vec{r}_i^1$ and a *half-step* velocity $\vec{v}_i^{-1/2}$ (sampled from a Boltzmann distribution of velocities) and generate the required atomic position $\vec{r}_i^0$ using the equation

$$\vec{r}_i^0 = \vec{r}_i^1 - \vec{v}_i^{-1/2}\Delta t. \tag{4.11}$$

From which point on the algorithm is self propagating. At each time step a new position $\vec{r}_i^{n+1}$ is calculated using (4.8), followed by the calculation of $\vec{v}_i^n$ using (4.10). During the integration $\vec{r}_i^{n-1}$ and $\vec{r}_i^n$ are stored in temporary variables, so that the $\vec{r}_i^n$ can overwrite $\vec{r}_i^{n-1}$ and the result $\vec{r}_i^{n+1}$ can overwrite $\vec{r}_i^n$.

The algorithm is simple and quick, if not particularly accurate, and is very stable. It is clearly time reversible and is symplectic (provided we pair $\vec{r}_i^{n+1}$ with $\vec{v}_i^{n+1}$ and not with $\vec{v}_i^n$). The storage requirement is modest; it requires storage for vectors $\vec{r}_i^n$, $\vec{r}_i^{n-1}$, $\vec{v}_i^n$ and $\vec{f}_i^n$, which is twelve real numbers per atom.

The main problem with the algorithm is that, at the start of each time step, the atomic velocity is a full step behind the atomic position. This is awkward if we need to perform temperature scaling of the simulation and require the temperature at the current time step (which depends on $\vec{v}_i^n$ through the kinetic energy,) *before* the integration of position takes place. At this point it is necessary to resort to iterative schemes, which slow the integration down and rob the algorithm of its symplecticness and time reversibility.

Simple as it is, the original Verlet algorithm can be improved. One improved form is the leapfrog algorithm [16], which is obtained as follows. Firstly we rewrite Verlet's position integrator (4.8) as

$$\vec{r}_i^{n+1} - \vec{r}_i^n = \vec{r}_i^n - \vec{r}_i^{n-1} + \frac{1}{m_i}\vec{f}_i^n \Delta t^2 + O(\Delta t^4), \tag{4.12}$$

then dividing throughout by Δt we obtain the leapfrog *velocity integrator* as

$$\vec{v}_i^{\,n+1/2} = \vec{v}_i^{\,n-1/2} + \frac{1}{m_i}\vec{f}_i^{\,n}\Delta t + O(\Delta t^3), \tag{4.13}$$

where we have defined

$$\begin{aligned} \vec{v}_i^{\,n+1/2} &= (\vec{r}_i^{\,n+1} - \vec{r}_i^{\,n})/\Delta t, \\ \vec{v}_i^{\,n-1/2} &= (\vec{r}_i^{\,n} - \vec{r}_i^{\,n-1})/\Delta t. \end{aligned} \tag{4.14}$$

to represent the *half time step velocities*. We note in passing that integration of the half time step velocities (4.13) is accurate to order $O(\Delta t^3)$ which is better than the full time step velocities obtained from Verlet's velocity integrator (4.10), with order $O(\Delta t^2)$. To obtain the leapfrog position integrator we rewrite (4.12) as

$$\begin{aligned} \vec{r}_i^{\,n+1} &= \vec{r}_i^{\,n} + (\vec{r}_i^{\,n} - \vec{r}_i^{\,n-1}) + \frac{1}{m_i}\vec{f}_i^{\,n}\Delta t^2 + O(\Delta t^4), \\ &= \vec{r}_i^{\,n} + \left(\vec{v}_i^{\,n-1/2} + \frac{1}{m_i}\vec{f}_i^{\,n}\Delta t\right)\Delta t + O(\Delta t^4), \end{aligned} \tag{4.15}$$

where we have used the definition of the half step velocity $\vec{v}_i^{\,n-1/2}$ from (4.14). With the aid of (4.13) equation (4.15) then becomes

$$\vec{r}_i^{\,n+1} = \vec{r}_i^{\,n} + \vec{v}_i^{\,n+1/2}\Delta t + O(\Delta t^4). \tag{4.16}$$

This is the leapfrog *position integrator*, which evidently has the same order of accuracy as the original Verlet integrator from which it derives.

The leapfrog algorithm possesses much the same properties as the original Verlet method, with regard to time reversibility, but requires less storage. The vectors needed are $\vec{r}_i^{\,n}$, $\vec{v}_i^{\,n-1/2}$ and $\vec{f}_i^{\,n}$, which amounts to nine real numbers per atom. Given the additional bonus of the more accurate velocities the leapfrog algorithm is generally more favoured than Verlet's original. Note however it has the same problems regarding thermostatted dynamics as the original method, since the full step velocities are not available for the integration process, though this is mitigated a little by the more accurate velocities and shorter time extrapolation required.

To start the integration it is only necessary to supply the initial atomic positions and half step velocities, similar to the original Verlet scheme. However, in this case the first integration step is operationally identical to all subsequent steps, so requires no special procedures. Each step begins with the velocity integration (4.13) to obtain $\vec{v}_i^{\,n+1/2}$, which overwrites $\vec{v}_i^{\,n-1/2}$, followed by the position integration (4.16) to obtain $\vec{r}_i^{\,n+1}$, which overwrites $\vec{r}_i^{\,n}$.

For the purpose of calculating the system temperature at each full time step, the full

time step velocities can be recovered by re-writing (4.10) as follows.

$$\begin{aligned}\vec{v}_i^{\,n} &= \frac{1}{2\,\Delta t}\left(\vec{r}_i^{\,n+1} - \vec{r}_i^{\,n} + \vec{r}_i^{\,n} - \vec{r}_i^{\,n-1}\right) + O\left(\Delta t^2\right), \\ &= \frac{1}{2}\left(\vec{v}_i^{\,n+1/2} + \vec{v}_i^{\,n-1/2}\right) + O\left(\Delta t^2\right).\end{aligned} \tag{4.17}$$

The final incarnation of the Verlet algorithm we describe is the velocity Verlet algorithm [17]. The main advantage of this variant is that it generates the atomic velocities and associated positions together at the full time step and so is more amenable to thermostatting without compromising other desirable properties in the scheme.

We begin by re-deriving equation (4.10) in the same way as before, but this time to a higher order in precision, to obtain

$$\vec{v}_i^{\,n} = \dot{\vec{r}}_i^{\,n} + \dddot{\vec{r}}_i^{\,n}\,\Delta\frac{t^2}{3!} + O\left(\Delta t^4\right). \tag{4.18}$$

We also re-arrange (4.4) to give

$$\vec{r}_i^{\,n+1} - \vec{r}_i^{\,n} = \Delta t\left(\dot{\vec{r}}_i^{\,n} + \dddot{\vec{r}}_i^{\,n}\frac{\Delta t^2}{3!}\right) + \ddot{\vec{r}}_i^{\,n}\frac{\Delta t^2}{2!} + O\left(\Delta t^4\right), \tag{4.19}$$

which, after inserting (4.18) and dividing throughout by Δt becomes

$$\frac{1}{\Delta t}\left(\vec{r}_i^{\,n+1} - \vec{r}_i^{\,n}\right) = \vec{v}_i^{\,n} + \ddot{\vec{r}}_i^{\,n}\frac{\Delta t}{2!} + O\left(\Delta t^3\right), \tag{4.20}$$

and from the definitions (4.14) and Newton's law this gives

$$\vec{v}_i^{\,n+1/2} = \vec{v}_i^{\,n} + \vec{f}_i^{\,n}\frac{\Delta t}{2\,m_i} + O\left(\Delta t^3\right). \tag{4.21}$$

This is the velocity Verlet *first velocity integrator*. Next, by rearranging (4.20) and inserting Newton's law again, we get

$$\vec{r}_i^{\,n+1} = \vec{r}_i^{\,n} + \vec{v}_i^{\,n}\,\Delta t + \vec{f}_i^{\,n}\frac{\Delta t^2}{2\,m_i} + O\left(\Delta t^4\right), \tag{4.22}$$

we see easily from (4.21) that this is equivalent to

$$\vec{r}_i^{\,n+1} = \vec{r}_i^{\,n} + \vec{v}_i^{\,n+1/2}\,\Delta t + O(\Delta t^4). \tag{4.23}$$

This is the velocity Verlet *position integrator.*

To determine the velocity at time step $n+1$ we adapt equation (4.17) and write

$$\vec{v}_i^{\,n+1} = \frac{1}{2}(\vec{v}_i^{\,n+3/2} + \vec{v}_i^{\,n+1/2}) + O(\Delta t^2) \tag{4.24}$$

where the half step velocity $\vec{v}_i^{\,n+3/2}$ is, according to the leapfrog velocity integrator (4.13), given by

$$\vec{v}_i^{\,n+3/2} = \vec{v}_i^{\,n+1/2} + \vec{f}_i^{\,n+1}\frac{\Delta t}{m_i} + O(\Delta t^3). \tag{4.25}$$

Substituting this into (4.24) this gives the velocity Verlet *second velocity integrator*:

$$\vec{v}_i^{\,n+1} = \vec{v}_i^{\,n+1/2} + \vec{f}_i^{\,n+1}\frac{\Delta t}{2m_i} + O(\Delta t^2), \tag{4.26}$$

where the larger order of error $O(\Delta t^2)$ in (4.25) takes precedent over the error $O(\Delta t^3)$ in (4.25). This explains why the error in (4.26) is different from (4.21), even though the equations look much the same.

Thus the complete velocity Verlet algorithm generates atomic positions and velocities at the same time step $n+1,$ so thermostatting does not require iterative schemes in this case. The algorithm also has the desirable features of time reversibility and symplecticness and has the same order of accuracy as the original Verlet and leapfrog methods. Its storage requirement is the same as the leapfrog algorithm – the vectors $\vec{r}_i^{\,n},$ $\vec{v}_i^{\,n}$ and $\vec{f}_i^{\,n},$ which is nine real numbers per atom.

Starting the algorithm is simple, only the initial positions $\vec{r}_i^{\,n},$ velocities $\vec{v}_i^{\,n}$ and forces $\vec{f}_i^{\,n}$ are required. Integration at each time step proceeds with the first velocity integrator (4.21) to obtain the half step velocity $\vec{v}_i^{\,n+1/2},$ which overwrites $\vec{v}_i^{\,n}.$ Next the position integrator (4.23) is applied to obtain the new position $\vec{r}_i^{\,n+1}$ which overwrites $\vec{r}_i^{\,n}.$ Lastly, after the calculation of new forces $\vec{f}_i^{\,n+1}$ the second velocity integrator (4.26) is used to calculate $\vec{v}_i^{\,n+1},$ overwriting $\vec{v}_i^{\,n+1/2}$ in the process.

The fact that the velocity Verlet algorithm requires two velocity integrations per step means that it is not as efficient computationally as the leapfrog algorithm. However, the possibility of thermostatting the algorithm without resorting to iteration schemes makes it an excellent choice for simulating isothermal and isobaric systems.

We should note that the form of the velocity Verlet algorithm given here can be written in another way, that is often found in the literature:

$$\begin{aligned}\vec{r}_i^{n+1} &= \vec{r}_i^n + \vec{v}_i^n \Delta t + \vec{f}_i^n \frac{\Delta t^2}{2m_i} + O(\Delta t^4) \\ \vec{v}_i^{n+1} &= \vec{v}_i^n + (\vec{f}_i^n + \vec{f}_i^{n+1}) \frac{\Delta t}{2m_i} + O(\Delta t^2)\end{aligned} \tag{4.27}$$

While this form is entirely equivalent to that given above, it is less convenient because the velocity integrator requires storage of both the force vectors $\vec{f}_i^n$ and $\vec{f}_i^{n+1}$, whereas in the scheme outlined above the new force vectors simply overwrite the old.

To conclude this section we note that all the variants of the Verlet algorithm generate the positions $\vec{r}_i^n$ with an accuracy $O(\Delta t^4)$ and the velocities $\vec{v}_i^n$ with an accuracy $O(\Delta t^2)$. This is not surprising, given that they are all derivatives of the original Verlet form. They are entirely equivalent algorithms. Thus all three generate the same trajectories, within numerical accuracy (small differences due to the finite representation of real numbers on the computer and the effects of numerical round-off will cause them to diverge eventually). The main differences between them are their operational characteristics.

We have stated that the Verlet algorithms are symplectic. In the next section we will prove this is the case.

4.3 The Symplecticness of the Verlet Algorithms

The symplectic nature of the Verlet algorithm can be shown by deriving it from the Liouville equation (see section 3.20), as was shown by Tuckerman et al. [20]. The key is to develop an integration algorithm that preserves the time-reversibility of the Liouville equation and so ensures the constancy of the Boltzmann probability distribution function. It turns out that the simplest such algorithm is the velocity Verlet. We will outline the proof here.

We start by recalling from section 3.21 that the Liouville equation can be formally written as

$$\frac{\partial f^N(\vec{\Gamma}^N, t)}{\partial t} = -i\hat{L} f^N(\vec{\Gamma}^n, t), \tag{4.28}$$

where $f^N(\vec{\Gamma}^N; t)$ is the probability density and $i\hat{L}$ is the Liouville operator

$$i\hat{L} = \sum_{j=1}^{3N} \left(\dot{x}_j \frac{\partial}{\partial x_j} + \dot{p}_j \frac{\partial}{\partial p_j} \right), \tag{4.29}$$

where we have used Hamilton's equations of motion (2.17) to convert equation (3.142) into the form (4.29). Liouville's equation has the formal solution

$$f^N(\vec{\Gamma}^N;t)=\exp(-it\,\hat{L})\,f^N(\vec{\Gamma}^N;0), \tag{4.30}$$

where $\exp(-it\,\hat{L})$ is the so-called *propagator*, since it propagates the starting probability density $f^N(\vec{\Gamma}^N;0)$ forwards in time.

This all assumes time is a continuum, but in numerical integration, it proceeds in discrete time steps, so $t=n\Delta t,$ where n is some integer. Putting this into the propagator and using de Moivre's theorem gives the discrete Liouville propagator

$$\exp(-it\,\hat{L})=\exp(-i\,n\Delta t\,\hat{L})=\left[\exp(-i\,\Delta t\,\hat{L})\right]^n. \tag{4.31}$$

We now split the operator $i\,\hat{L}$ into two sub-operators $i\,\hat{L}_1$ and $i\,\hat{L}_2$ i.e.

$$i\,\hat{L}=i\,\hat{L}_1+i\,\hat{L}_2 \tag{4.32}$$

where

$$\begin{aligned} i\,\hat{L}_1&=\sum_{j=1}^{3N}\left(\dot{x}_j\frac{\partial}{\partial x_j}\right),\\ i\,\hat{L}_2&=\sum_{j=1}^{3N}\left(\dot{p}_j\frac{\partial}{\partial p_j}\right). \end{aligned} \tag{4.33}$$

Splitting the operator in this way potentially provides a means to separate out the propagation of the momenta p_i from that of the positions $x_i.$ However the separation is complicated by the fact that the operators $i\,\hat{L}_1$ and $i\,\hat{L}_2$ are mathematically *non-commutative* - meaning that applying operator $i\,\hat{L}_1$ before operator $i\,\hat{L}_2$ does not give the same result as applying $i\,\hat{L}_2$ before $i\,\hat{L}_1.$ Thus a solution to the discrete Liouville equation, based in this simple split, is not unique.

However a unique solution is possible if we split the operators in a way that is time-reversible, then they will function in the same way if the order of the operators is inverted. The time-reversibility of the operator guarantees to preserve the constancy of probability distribution function. This means the integration algorithm we are deriving will be symplectic.

The correct way to split the operator $i\,\hat{L}$ to preserve time reversibility is provided by Trotter's Theorem [20], according to which we have

$$\left[\exp(-i\Delta t\{\hat{L}_1+\hat{L}_2\})\right]^n=\left[\exp(-i\Delta t\,\hat{L}_2/2)\exp(-i\Delta t\,\hat{L}_1)\exp(-i\Delta t\,\hat{L}_2/2)\right]^n+O(\Delta t^3). \quad (4.34)$$

We interpret this as follows: we should apply operator $i\hat{L}_2$ first for *half a time step* then apply operator $i\hat{L}_1$ for a *whole time step* and finally operator $i\hat{L}_2$ again for a *half time step*.

We can now write the solution to the Liouville equation (4.30) in discrete form:

$$f^N(\vec{\Gamma}^N,n\Delta t)=\left[\exp(-i\Delta t\,\hat{L}_2/2)\exp(-i\Delta t\,\hat{L}_1)\exp(-i\Delta t\,\hat{L}_2/2)\right]^n f^N(\vec{\Gamma}^N,0), \quad (4.35)$$

Then by using the expansion of the propagators in equation (3.145) truncated to first order, we can write

$$f^N(\vec{\Gamma}^N,n\Delta t)=\left[\sum_{j=1}^{3N}\left\{1+\frac{\Delta t}{2}\dot{p}_j\frac{\partial}{\partial p_j}\right\}\cdot\sum_{j=1}^{3N}\left\{1+\Delta t\,\dot{r}_j\frac{\partial}{\partial r_j}\right\}\cdot\sum_{j=1}^{3N}\left\{1+\frac{\Delta t}{2}\dot{p}_j\frac{\partial}{\partial p_j}\right\}\right]^n f^N(\vec{\Gamma}^N,0). \quad (4.36)$$

Now, for a general function $F(x)$ we have from the Taylor expansion

$$F(x+\delta x)\approx\left\{1+\delta x\frac{\partial}{\partial x}\right\}F(x)+O(\delta x^2), \quad (4.37)$$

so if $\delta x=\Delta t\,\dot{x}$ we also have

$$F(x+\dot{x}\Delta t)\approx\left\{1+\dot{x}\Delta t\frac{\partial}{\partial x}\right\}F(x)+O(\Delta t^2), \quad (4.38)$$

From (4.38) and (4.36) it follows that

$$\begin{aligned} f_x(x;\Delta t)&=f_x(x+\dot{x}\Delta t;0),\\ f_p(p;\Delta t/2)&=f_p(p+\dot{p}\Delta t/2;0). \end{aligned} \quad (4.39)$$

where f_x and f_p are the position and momentum components of the probability density. Equations (4.39) show that propagating a probability density $f_x(x;0)$ one time step to $f_x(x;\Delta t),$ is the same as translating the variable x to $x+\dot{x}\Delta t$ and calculating $f_x(x+\dot{x}\Delta t;0)$ (i.e. at time zero), with same principle holding for the momentum probability density $f_p.$ This is consistent with the idea that the probability density function is fixed in phase space and solving the Liouville equation returns the value of the probability density at each point in phase space visited by the evolving phase space vector $\Gamma^N(t).$

In our representation the Liouville equation is of course discretized, as is the phase space path, which we represent as $\Gamma^N(n\Delta t)$. Then according to equation (4.36) the discretized path is given by

$$\Gamma^N(n\Delta t)\approx\left[\left(\forall\, j: p_j\to p_j+\dot{p}_j\Delta t/2\right)\left(\forall\, j: x_j\to x_j+\dot{x}_j\Delta t\right)\left(\forall\, j: p_j\to p_j+\dot{p}_j\Delta t\right)\right]^n, \tag{4.40}$$

which shows that for all atoms j the path is obtained by first applying the momentum update $p_j\to p_j+\dot{p}_j\Delta t/2$ (a half time step), followed by a position update $x_j\to x_j+\dot{x}_j\Delta t$ (a full time step) and lastly another momentum update $p_j\to p_j+\dot{p}_j\Delta t/2$ (a half time step). The procedure is precisely the same as the velocity Verlet algorithm. On this basis we may say that the velocity Verlet is a symplectic algorithm and, by virtue of the equivalence of all the Verlet forms, this is true of them all[13].

Incidentally this derivation of the velocity Verlet algorithm suggests that there is yet another possible form of Verlet algorithm – the *position Verlet*. This is obtained by choosing the alternative Trotter expansion sequence: $\hat{L}_1/2:\hat{L}_2:\hat{L}_1/2$ in place of $\hat{L}_2/2:\hat{L}_1:\hat{L}_2/2,$ as was chosen in equation (4.34). We will not develop this idea here as the resulting algorithm is rarely, if ever, used, though it is a viable alternative.

The Trotter expansion of the Liouville propagator represents a powerful tool for the development of symplectic algorithms and we will use it again later in this chapter to derive thermostatted and barostatted algorithms corresponding to the NVT and NPT ensembles.

4.4 The Gear Predictor-Corrector Algorithm

$$\begin{aligned}
\vec{r}_i^{\,n+1}&\approx\vec{r}_i^{\,n}+\dot{\vec{r}}_i^{\,n}\Delta t+\ddot{\vec{r}}_i^{\,n}\frac{\Delta t^2}{2!}+\dddot{\vec{r}}_i^{\,n}\frac{\Delta t^3}{3!}+\ddddot{\vec{r}}_i^{\,n}\frac{\Delta t^4}{4!}+\dddddot{\vec{r}}_i^{\,n}\frac{\Delta t^5}{5!}+\ddddddot{\vec{r}}_i^{\,n}\frac{\Delta t^6}{6!},\\
\dot{\vec{r}}_i^{\,n+1}&\approx\dot{\vec{r}}_i^{\,n}+\ddot{\vec{r}}_i^{\,n}\Delta t+\dddot{\vec{r}}_i^{\,n}\frac{\Delta t^2}{2!}+\ddddot{\vec{r}}_i^{\,n}\frac{\Delta t^3}{3!}+\dddddot{\vec{r}}_i^{\,n}\frac{\Delta t^4}{4!}+\ddddddot{\vec{r}}_i^{\,n}\frac{\Delta t^5}{5!},\\
\ddot{\vec{r}}_i^{\,n+1}&\approx\ddot{\vec{r}}_i^{\,n}+\dddot{\vec{r}}_i^{\,n}\Delta t+\ddddot{\vec{r}}_i^{\,n}\frac{\Delta t^2}{2!}+\dddddot{\vec{r}}_i^{\,n}\frac{\Delta t^3}{3!}+\ddddddot{\vec{r}}_i^{\,n}\frac{\Delta t^4}{4!},\\
\dddot{\vec{r}}_i^{\,n+1}&\approx\dddot{\vec{r}}_i^{\,n}+\ddddot{\vec{r}}_i^{\,n}\Delta t+\dddddot{\vec{r}}_i^{\,n}\frac{\Delta t^2}{2!}+\ddddddot{\vec{r}}_i^{\,n}\frac{\Delta t^3}{3!},\\
\ddddot{\vec{r}}_i^{\,n+1}&\approx\ddddot{\vec{r}}_i^{\,n}+\dddddot{\vec{r}}_i^{\,n}\Delta t+\ddddddot{\vec{r}}_i^{\,n}\frac{\Delta t^2}{2!},\\
\dddddot{\vec{r}}_i^{\,n+1}&\approx\dddddot{\vec{r}}_i^{\,n}+\ddddddot{\vec{r}}_i^{\,n}\Delta t,\\
\ddddddot{\vec{r}}_i^{\,n+1}&\approx\ddddddot{\vec{r}}_i^{\,n}.
\end{aligned} \tag{4.41}$$

The Gear predictor-corrector algorithm is an accurate, high-order algorithm which, though not symplectic, is nevertheless much used in molecular dynamics.

13 In the case of leapfrog, this assumes we are considering the full, not half-step velocity.

We begin once again with the Taylor expansion in the form given in equation (4.4). However we define the expansion not only of the position vector $\vec{r}_i,$ but of the time derivatives of $\vec{r}_i$ up to some order $M.$ So for $M=6,$ for example, we have the equations presented in (4.41).

Mercifully, with a change of variables all the above can be condensed into a simpler representation. We first define vectors $\vec{R}_i^{k,n}$ as follows[14]

$$\vec{R}_i^{k,n}=\frac{\Delta t^k}{k!}\frac{d^k\vec{r}_i^{\,n}}{d\,t^k} \tag{4.42}$$

where k ranges from 0 to $M.$ Then *all* the above equations can be written as

$$\vec{R}_i^{k,n+1\prime}=\sum_{j=k}^{M}\frac{j!}{k!(j-k)!}\vec{R}_i^{j,n}, \tag{4.43}$$

where we have used the dash $(\prime)$ on the left of (4.43) to indicate that this is not the final value $\vec{R}_i^{k,n+1}$ at the $(n+1)'th$ time step, but only a first (predicted) estimate. Improved values are obtained from the knowledge that the acceleration term $\vec{R}_i^{2,n},$ at the predicted position $\vec{R}_i^{0,n},$ is known exactly from Newton's force law. This allows the calculation of a correction vector $\vec{R}_i^c$ which is given by

$$\vec{R}_i^c=\frac{\Delta t^2}{2m_i}\vec{f}_i(\{\vec{R}_i^{0,n+1\prime}\})-\vec{R}_i^{2,n+1\prime}. \tag{4.44}$$

Using this correction vector, the vectors $\vec{R}_i^k$ may be improved upon using the correction step:

$$\vec{R}_i^{k,n+1\prime\prime}=\vec{R}_i^{k,n+1\prime}+\beta_k\vec{R}_i^c \tag{4.45}$$

where the coefficients β_k are "magic numbers" derived through an optimisation intended to maximise the overall accuracy of the correction. Of course, once the more accurate vectors have been obtained, the correction step may be repeated several times to improve accuracy further. However in molecular dynamics, each cycle of the correction means another (expensive) calculation of the forces, so in practice just one correction step is usually applied. The sequence for each time step is therefore firstly to perform the predictor step (4.43), then to calculate the forces at the new atomic positions and apply the corrector step (4.45). It is normal practice in molecular dynamics to calculate the atomic forces once only per time step.

14 The indices k,n associated with vectors $\vec{R}_i^{k,n}$ are not to be confused with exponents.

The Gear predictor-corrector algorithm remains popular, particularly in applications of non-equilibrium molecular dynamics, where the production of accurate velocities at each time step is considered an advantage. The algorithm is clearly more accurate than the Verlet variants and can be made more so by working with even higher orders of M. It may reasonably be expected that larger time steps Δt can be used than with Verlet, which is another advantage. The disadvantages however are several. There is potentially a much higher storage requirement than for Verlet – three real numbers per atom for each order of k. Also the algorithm is not symplectic and is known to drift from the required thermodynamic state point and must be assisted by a thermostat. Lastly the algorithm is not self starting since the initial values for the higher order derivatives are needed, at least in principle. In practice most users are content to set all derivatives above the second to zero at the start.

4.5 SHAKE : The Dynamics of Rigid Bonds

The SHAKE algorithm is one of the most important algorithms in molecular dynamics [6], [21]. It is concerned with the dynamics of constrained systems, in which some particular property (usually a part of the molecular structure) is held constant as the dynamics proceeds. The algorithm can take many forms, depending on what kind of constraint is required, but in this section we will describe the most common constraint: the rigid bond with a fixed bond length.

It is pertinent first to ask why fixing bond lengths is necessary, since it is easy enough to define a bond as (say) a simple harmonic spring and handle the dynamics of the system in the usual way. There are however some basic objections to doing this. To start with, bond force constants are often very large (the bonds are said to be "stiff"), which leads to very high vibration frequencies. In consequence the time step required for the simulation is very short, which means that the evolution of the system through phase space is slow and the simulation is therefore computationally expensive. There is however an even more insidious problem, concerning the statistical mechanics of such systems.

Stiff bonds do not exchange energy very efficiently with the slower modes of motion in the system. In cases where the vibrational frequencies occupy a separate (i.e. non-overlapping) part of the vibrational spectrum from the rest of the system, the energy exchange can be negligible. This means that it is *practically impossible* to establish a proper equilibrium of energy between the bonds and the rest of the system and this invalidates the equipartition principle, one of the most basic assumptions of statistical thermodynamics. In practical terms this also means that the system temperature is not properly defined. Fixing the bond lengths does away with these problems altogether. This is hardly a loss, since the detailed dynamics of vibrating bonds is not generally as important as conformational dynamics, where changes in dihedral angles are of primary interest.

We will present the SHAKE algorithm for rigid bonds in an intuitive way, but it should be noted that the method can be founded more rigorously using Lagrangian dynamics [6]. The algorithm is first presented in a form specifically for the original Verlet algorithm as represented by the equations (4.8) and (4.10).

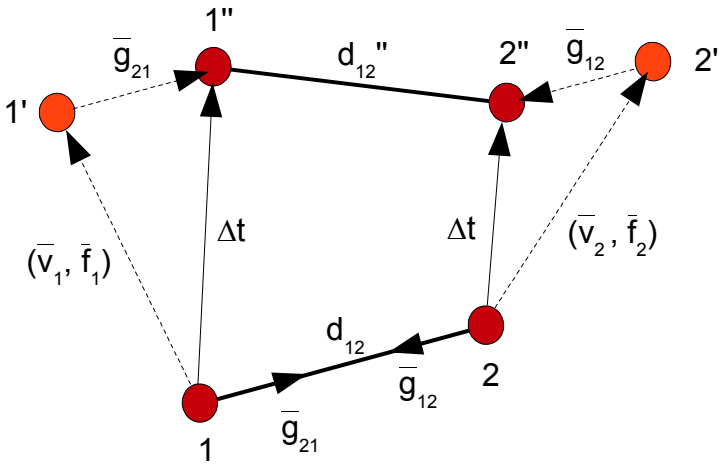

Figure 4.2. The SHAKE Algorithm

In figure 4.2 we present the SHAKE algorithm for constraining the bond in a diatomic molecule. At some instant in time a diatomic molecule, with atoms labelled 1 and 2 and bond length $d_{12}=|\vec{d}_{12}|,$ experiences a force $\vec{f}_1$ on atom 1 and a force $\vec{f}_2$ on atom 2. These atoms are currently moving with velocities $\vec{v}_1$ and $\vec{v}_2.$ If the correct dynamics are operating these atoms would, as a result of these forces and velocities, move to positions 1'' and 2'' in a time step $\Delta t,$ such that the length of the bond between them (d_{12}) is conserved, which in figure 4.2 means that $d_{12}=d_{12}$''[15]. However, for this to be possible, we must include the constraint forces due to the rigid bond that resist any tendency to extend or contract the bond. These forces are shown as $\vec{g}_{21}$ and $\vec{g}_{12}$ acting on atoms 1 and 2 respectively. Unfortunately, we do not immediately know the magnitude of these forces, but we do know that they act along the bond vector $\vec{d}_{12}$ and that, by Newton's third law relating action and reaction: $\vec{g}_{12}+\vec{g}_{21}=0,$ so that the forces are equal and opposite.

It might be thought that such conditions would allow us to calculate the constraint forces immediately by, for example, resolving the forces $\vec{f}_1$ and $\vec{f}_2$ into components parallel and perpendicular to $\vec{d}_{12}$ which would allow the constraint force to be determined as the negative of the parallel force. This however is not a complete description, because there are additional centrifugal forces acting on the rotating bond that also have to be accounted for. But while there are ways to obtain this also, we do not choose to proceed in this way. The problem is, the Verlet (or any other) algorithm is not infinitely precise, so integrating the equations of motion for the individual atoms will only provide the *approximate* atomic positions after the interval Δt and repeating the calculation for many time steps leads to a gradual divergence of the bond length as the time increases. What is needed is a way of guaranteeing the bond

15 Specifically, we seek to conserve the square of the bond length $d_{12}^2=\vec{d}_{12}\cdot\vec{d}_{12}.$

length has the required value (within a finite error that does not grow with time) *before* proceeding with the following time step. This is what the SHAKE algorithm provides.

4.5.1 SHAKE and the Original Verlet Algorithm

If we consider the motion of the atoms *without* the constraint forces in the time step Δt, the atoms 1 and 2 they would in fact proceed to positions 1' and 2' in figure 4.2, under the influence of velocities $\vec{v}_1$ and $\vec{v}_2$ and forces $\vec{f}_1$ and $\vec{f}_2$ respectively. The constraint forces $\vec{g}_{21}$ and $\vec{g}_{12}$ are therefore merely *corrections* to the *unconstrained* motion of the atoms. This suggests a way to determine them.

The procedure is first to integrate the atomic positions assuming the constraint force is absent, to obtain the unconstrained atomic positions. We can then use the unconstrained positions to provide an estimate for the constraint force that would have to be applied to maintain the required bond length. This estimated constraint force, in turn, provides a better estimate of the constrained atomic positions, from which a first correction to the constraint force can be calculated. An iterative procedure is therefore possible that can satisfy the constraint to any required accuracy. The detailed algorithm is therefore as follows.

If we assume the constraint forces are known, the Verlet position integrator (4.8) would give the positions of the atoms as

$$\begin{aligned}\vec{r}_1^{\,n+1} &= 2\vec{r}_1^{\,n} - \vec{r}_1^{\,n-1} + \frac{1}{m_1}(\vec{f}_1^{\,n} + \vec{g}_{21}^{\,n})\Delta t^2, \\ \vec{r}_2^{\,n+1} &= 2\vec{r}_2^{\,n} - \vec{r}_2^{\,n-1} + \frac{1}{m_2}(\vec{f}_2^{\,n} + \vec{g}_{12}^{\,n})\Delta t^2.\end{aligned} \tag{4.46}$$

But, in the absence of the constraint forces we have

$$\begin{aligned}\tilde{\vec{r}}_1^{\,n+1} &= 2\vec{r}_1^{\,n} - \vec{r}_1^{\,n-1} + \frac{1}{m_1}\vec{f}_1^{\,n}\Delta t^2, \\ \tilde{\vec{r}}_2^{\,n+1} &= 2\vec{r}_2^{\,n} - \vec{r}_2^{\,n-1} + \frac{1}{m_2}\vec{f}_2^{\,n}\Delta t^2,\end{aligned} \tag{4.47}$$

where the tilde (~) over the vectors on the left indicate that these are unconstrained positions. Using (4.47) we can write (4.46) as

$$\begin{aligned}\vec{r}_1^{\,n+1} &= \tilde{\vec{r}}_1^{\,n+1} + \frac{1}{m_1}\vec{g}_{21}\Delta t^2, \\ \vec{r}_2^{\,n+1} &= \tilde{\vec{r}}_2^{\,n+1} + \frac{1}{m_2}\vec{g}_{12}\Delta t^2.\end{aligned} \tag{4.48}$$

We now note the following definitions

$$\begin{aligned}\vec{d}_{12}^{\,n+1}&=\vec{r}_2^{\,n+1}-\vec{r}_1^{\,n+1},\\ \tilde{\vec{d}}_{12}^{\,n+1}&=\tilde{\vec{r}}_2^{\,n+1}-\tilde{\vec{r}}_1^{\,n+1},\\ \vec{g}_{12}^{\,n}&=g_{12}^{\,n}\vec{d}_{12}^{\,n}=-\vec{g}_{21}^{\,n}.\end{aligned} \tag{4.49}$$

The last of these stems from the fact that the constraint force $\vec{g}_{12}$ must act along the vector $\vec{d}_{12}$. Subtracting the first equation of (4.48) from the second and using the definitions (4.49) leads to

$$\vec{d}_{12}^{\,n+1}=\tilde{\vec{d}}_{12}^{\,n+1}+\frac{g_{12}^{\,n}}{\mu_{12}}\vec{d}_{12}^{\,n}\Delta t^2, \tag{4.50}$$

in which the *reduced mass* μ_{12} is defined as

$$\frac{1}{\mu_{12}}=\left(\frac{1}{m_1}+\frac{1}{m_2}\right). \tag{4.51}$$

Squaring both sides of (4.50) gives

$$(d_{12}^{\,n+1})^2=(\tilde{d}_{12}^{\,n+1})^2+2\frac{g_{12}^{\,n}}{\mu_{12}}\vec{d}_{12}^{\,n}\cdot\tilde{\vec{d}}_{12}^{\,n+1}\Delta t^2+\frac{(g_{12}^{\,n})^2}{\mu_{12}^2}(d_{12}^{\,n})^2\Delta t^4, \tag{4.52}$$

and of course we require that

$$(d_{12}^{\,n})^2=(d_{12}^{\,n+1})^2=d_{12}^2. \tag{4.53}$$

With the specified values given in (4.53) equation (4.52) becomes a quadratic in the one unknown parameter: $g_{12}^{\,n}$, which may, in principle, be solved using Newton-Raphson iteration. The final value may then be inserted into equations (4.48) to obtain the positions $\vec{r}_1^{\,n+1}$ and $\vec{r}_2^{\,n+1}$ which *exactly* satisfy the constraint condition. However, the last term on the right of (4.52) is of order $O(\Delta t^4)$ and is therefore negligible. So it is usually dropped and, with rearrangement, equation (4.52) then becomes

$$g_{12}^{\,n}\approx-\frac{\mu_{12}\left\{(\tilde{d}_{12}^{\,n+1})^2-(d_{12})^2\right\}}{2\vec{d}_{12}^{\,n}\cdot\tilde{\vec{d}}_{12}^{\,n+1}\Delta t^2}. \tag{4.54}$$

The value of g_{12} obtained from this may be used to approximate the constraint forces in equations (4.46) to get a better estimate of $\tilde{\vec{r}}_1^{\,n+1}$ and $\tilde{\vec{r}}_2^{\,n+1}$. This allows

(4.54) to be solved again to obtain *additional corrections* to g_{12}. Further iteration will improve the accuracy even further.

In general however, we are often concerned with polyatomic molecules and not just diatomics. In which case we may need to satisfy many bond constraints simultaneously. The many constraint conditions are often *antagonistic* in nature, which means satisfying one constraint may perturb other solved constraints that happen to involve the one of the two bond atoms. The solution is to cycle over the different constraints several times, solving (4.54) iteratively for each constraint in every cycle. This strategy will eventually satisfy all constraints to the desired accuracy. Afterwards, the velocities of the atoms are obtained in the same way as the original Verlet scheme, i.e. using equation (4.10). This iterative approach to satisfying all the constraints is undoubtedly the origin of the name "SHAKE".

It is important to note that the SHAKE algorithm also makes a contribution to the system virial, which is used to calculate the pressure (see chapter 7). The contribution takes the form of a sum over all the rigid bonds:

$$\Psi_{SHAKE} = -\sum_{bonds} \vec{g}_{ij} \cdot \vec{d}_{ij}, \tag{4.55}$$

where atoms i and j belong to a particular rigid bond. This expression is identical to that for any pair force. However care must be taken in calculating the constraint force $\vec{g}_{ij}$. It is *not* based on the final value of g_{ij} that emerges from the last cycle of equation (4.54) since this is a correction that approaches zero as the iteration cycle proceeds. Rather it is the *accumulated sum* of g_{ij} over all iterations for a given bond that represents the full constraint force.

4.5.2 SHAKE and the Verlet Leapfrog Algorithm

Casting the SHAKE algorithm into a form suitable for the leapfrog scheme presents no great difficulty.

With full knowledge of the constraint forces, the half step velocities are updated using the equations

$$\begin{aligned} \vec{v}_1^{\,n+1/2} &= \vec{v}_1^{\,n-1/2} + \frac{1}{m_1}(\vec{f}_1^{\,n} + \vec{g}_{21}^{\,n})\Delta t, \\ \vec{v}_2^{\,n+1/2} &= \vec{v}_2^{\,n-1/2} + \frac{1}{m_2}(\vec{f}_2^{\,n} + \vec{g}_{12}^{\,n})\Delta t, \end{aligned} \tag{4.56}$$

and the positions are then updated using

$$\begin{aligned} \vec{r}_1^{\,n+1} &= \vec{r}_1^{\,n} + \vec{v}_1^{\,n+1/2}\Delta t, \\ \vec{r}_2^{\,n+1} &= \vec{r}_2^{\,n} + \vec{v}_2^{\,n+1/2}\Delta t. \end{aligned} \tag{4.57}$$

We can use (4.56) to expand (4.57) as

$$\begin{aligned}\vec{r}_1^{\,n+1} &= \vec{r}_1^{\,n} + \left(\vec{v}_1^{\,n-1/2} + \frac{1}{m_1}(\vec{f}_1^{\,n} + \vec{g}_{21}^{\,n})\Delta t \right)\Delta t\,, \\ \vec{r}_2^{\,n+1} &= \vec{r}_2^{\,n} + \left(\vec{v}_2^{\,n-1/2} + \frac{1}{m_2}(\vec{f}_2^{\,n} + \vec{g}_{12}^{\,n})\Delta t \right)\Delta t.\end{aligned} \tag{4.58}$$

In the absence of the constraint forces we can write (4.56) as

$$\begin{aligned}\tilde{\vec{v}}_1^{\,n+1/2} &= \vec{v}_1^{\,n-1/2} + \frac{1}{m_1}\vec{f}_1^{\,n}\Delta t\,, \\ \tilde{\vec{v}}_2^{\,n+1/2} &= \vec{v}_2^{\,n-1/2} + \frac{1}{m_2}\vec{f}_2^{\,n}\Delta t.\end{aligned} \tag{4.59}$$

which we can use to rewrite (4.58) in the form

$$\begin{aligned}\vec{r}_1^{\,n+1} &= \vec{r}_1^{\,n} + \tilde{\vec{v}}_1^{\,n+1/2}\Delta t + \frac{1}{m_1}\vec{g}_{21}^{\,n}\Delta t^2\,, \\ \vec{r}_2^{\,n+1} &= \vec{r}_2^{\,n} + \tilde{\vec{v}}_2^{\,n+1/2}\Delta t + \frac{1}{m_2}\vec{g}_{12}^{\,n}\Delta t^2.\end{aligned} \tag{4.60}$$

In the absence of constraint forces we may write (4.57) as

$$\begin{aligned}\tilde{\vec{r}}_1^{\,n+1} &= \vec{r}_1^{\,n} + \tilde{\vec{v}}_1^{\,n+1/2}\Delta t\,, \\ \tilde{\vec{r}}_2^{\,n+1} &= \vec{r}_2^{\,n} + \tilde{\vec{v}}_2^{\,n+1/2}\Delta t\,,\end{aligned} \tag{4.61}$$

so (4.60) can now be written as

$$\begin{aligned}\vec{r}_1^{\,n+1} &= \tilde{\vec{r}}_1^{\,n+1} + \frac{1}{m_1}\vec{g}_{21}\Delta t^2\,, \\ \vec{r}_2^{\,n+1} &= \tilde{\vec{r}}_2^{\,n+1} + \frac{1}{m_2}\vec{g}_{12}\Delta t^2.\end{aligned} \tag{4.62}$$

This is the same pair of equations as (4.48) for the original SHAKE algorithm. So we may follow exactly the same procedure to arrive once again at equation (4.54) for estimating the scalar factor g_{12}. This value may then be used in equations (4.56) to obtain better estimates of $\vec{v}_1^{\,n+1/2}$ and $\vec{v}_2^{\,n+1/2}$, which in turn provide better estimates of $\vec{r}_1^{\,n+1}$ and $\vec{r}_2^{\,n+1}$ via equations (4.57) and so on to further corrections of g_{12} using (4.54). In this way an iterative scheme unfolds and on satisfying the bond constraints to within the required tolerance the last obtained values for atomic positions and half step velocities represent the solution. As with the original SHAKE scheme, the iterated

values of g_{12} are summed to obtain the full constraint force needed for the virial calculation (4.55).

4.5.3 SHAKE and the Velocity Verlet Algorithm

The SHAKE adaptation of the velocity Verlet algorithm was devised by Andersen [22]. It differs from the original and leapfrog versions in that it requires an additional constraint, characterised by a force $\vec{h}_{12}$, which again acts along the bond vector, and ensures that the velocities of the atoms with respect to the bond centre of mass are always perpendicular to the bond vector. This is consistent with the idea that atoms in rotating rigid bonds cannot be moving towards, or away from, the centre of rotation. The actual constraint condition is therefore

$$\vec{v}_{12}\cdot\vec{d}_{12}=0\,, \tag{4.63}$$

and force $\vec{h}_{12}$ must be constructed to ensure this condition. The additional constraint gives the algorithm a distinct name of its own: RATTLE.

The derivation of the algorithm is as follows. If the constraint forces $\vec{g}_{12}$ and $\vec{h}_{12}$ were known beforehand, the velocity Verlet algorithm updating the velocities and positions of bonded atoms 1 and 2 could be written as

$$\begin{aligned}\vec{v}_1^{\,n+1/2}&=\vec{v}_1^{\,n}+(\vec{f}_1^{\,n}+\vec{g}_{21}^{\,n})\frac{\Delta t}{2m_1},\\ \vec{v}_2^{\,n+1/2}&=\vec{v}_2^{\,n}+(\vec{f}_2^{\,n}+\vec{g}_{12}^{\,n})\frac{\Delta t}{2m_2},\end{aligned} \tag{4.64}$$

$$\begin{aligned}\vec{r}_1^{\,n+1}&=\vec{r}_1^{\,n}+\vec{v}_1^{\,n+1/2}\Delta t\,,\\ \vec{r}_2^{\,n+1}&=\vec{r}_2^{\,n}+\vec{v}_2^{\,n+1/2}\Delta t\,,\end{aligned} \tag{4.65}$$

$$\begin{aligned}\vec{v}_1^{\,n+1}&=\vec{v}_1^{\,n+1/2}+(\vec{f}_1^{\,n+1}+\vec{h}_{21}^{\,n+1})\frac{\Delta t}{2m_1},\\ \vec{v}_2^{\,n+1}&=\vec{v}_2^{\,n+1/2}+(\vec{f}_2^{\,n+1}+\vec{h}_{12}^{\,n+1})\frac{\Delta t}{2m_2}.\end{aligned} \tag{4.66}$$

As was done for the leapfrog case above, the equations (4.64) may be substituted into (4.65) to give

$$\vec{r}_1^{\,n+1}=\vec{r}_1^{\,n}+(\vec{v}_1^{\,n}+(\vec{f}_1^{\,n}+\vec{g}_{21}^{\,n})\frac{\Delta t}{2m_1})\Delta t,$$
$$\vec{r}_2^{\,n+1}=\vec{r}_2^{\,n}+(\vec{v}_1^{\,n}+(\vec{f}_2^{\,n}+\vec{g}_{12}^{\,n})\frac{\Delta t}{2m_1})\Delta t. \tag{4.67}$$

In the absence of the constraint force $\vec{g}_{12}$ the velocity update equations (4.64) would be

$$\tilde{\vec{v}}_1^{\,n+1/2}=\vec{v}_1^{\,n}+\frac{1}{2m_1}\vec{f}_1^{\,n}\Delta t,$$
$$\tilde{\vec{v}}_2^{\,n+1/2}=\vec{v}_2^{\,n}+\frac{1}{2m_2}\vec{f}_2^{\,n}\Delta t, \tag{4.68}$$

so that (4.67) can now be written as

$$\vec{r}_1^{\,n+1}=\vec{r}_1^{\,n}+\tilde{\vec{v}}_1^{\,n+1/2}\Delta t+\frac{1}{2m_1}\vec{g}_{21}^{\,n}\Delta t^2,$$
$$\vec{r}_2^{\,n+1}=\vec{r}_2^{\,n}+\tilde{\vec{v}}_2^{\,n+1/2}\Delta t+\frac{1}{2m_2}\vec{g}_{12}^{\,n}\Delta t^2. \tag{4.69}$$

In the absence of the constraint force (4.65) can be written as

$$\tilde{\vec{r}}_1^{\,n+1}=\vec{r}_1^{\,n}+\tilde{\vec{v}}_1^{\,n+1/2}\Delta t,$$
$$\tilde{\vec{r}}_2^{\,n+1}=\vec{r}_2^{\,n}+\tilde{\vec{v}}_2^{\,n+1/2}\Delta t, \tag{4.70}$$

so that (4.69) can be simplified to

$$\vec{r}_1^{\,n+1}=\tilde{\vec{r}}_1^{\,n+1}+\frac{1}{2m_1}\vec{g}_{21}^{\,n}\Delta t^2,$$
$$\vec{r}_2^{\,n+1}=\tilde{\vec{r}}_2^{\,n+1}+\frac{1}{2m_2}\vec{g}_{12}^{\,n}\Delta t^2. \tag{4.71}$$

Now, in the same way as for the original Verlet SHAKE derivation, we may proceed toward the constraint equation. The procedure is almost identical and in this case we arrive at

$$g_{12}^{n}\approx-\frac{\mu_{12}\left\{(\tilde{d}_{12}^{\,n+1})^2-(d_{12})^2\right\}}{\vec{d}_{12}^{\,n}\cdot\tilde{\vec{d}}_{12}^{\,n+1}\Delta t^2}, \tag{4.72}$$

which closely resembles equation (4.54) and is used in the same way, but notice that the correction is exactly *twice* that given by (4.54). This fact does not matter much for the conservation of the bond length, since it is guaranteed by iterating to

convergence, but it matters very much in calculating the constraint virial (4.55).

But we are not done yet. We need to calculate the constraint force $\vec{h}_{12}$. For that we rewrite equations (4.66) for the unconstrained case:

$$\begin{aligned}\tilde{\vec{v}}_1^{n+1} &= \vec{v}_1^{n+1/2} + \vec{f}_1^{n+1}\frac{\Delta t}{2m_1},\\ \tilde{\vec{v}}_2^{n+1} &= \vec{v}_2^{n+1/2} + \vec{f}_2^{n+1}\frac{\Delta t}{2m_2},\end{aligned} \tag{4.73}$$

which, on substitution back into (4.66) gives

$$\begin{aligned}\vec{v}_1^{n+1} &= \tilde{\vec{v}}_1^{n+1} + \vec{h}_{21}^{n+1}\frac{\Delta t}{2m_1},\\ \vec{v}_2^{n+1} &= \tilde{\vec{v}}_2^{n+1} + \vec{h}_{12}^{n+1}\frac{\Delta t}{2m_2}.\end{aligned} \tag{4.74}$$

and hence, on subtraction of the first equation of (4.74) from the second, we have

$$\vec{v}_{12}^{n+1} = \tilde{\vec{v}}_{12}^{n+1} + \vec{h}_{12}^{n+1}\frac{\Delta t}{2\mu_{12}}. \tag{4.75}$$

and taking the scalar product with $\vec{d}_{12}^{n+1} = \vec{r}_2^{n+1} - \vec{r}_1^{n+1}$ gives

$$\vec{v}_{12}^{n+1}\cdot\vec{d}_{12}^{n+1} = \tilde{\vec{v}}_{12}^{n+1}\cdot\vec{d}_{12}^{n+1} + h_{12}^{n+1}d_{12}^2\frac{\Delta t}{2\mu_{12}}, \tag{4.76}$$

where we have used the identity

$$\vec{h}_{12}^{n+1} = h_{12}^{n+1}\vec{d}_{12}^{n+1} = -\vec{h}_{21}^{n+1} \tag{4.77}$$

and, of course, the left hand side of (4.76) is zero by requirement, so rearranging gives

$$h_{12}^{n+1} = -\frac{2\mu_{12}\tilde{\vec{v}}_{12}^{n+1}\cdot\vec{d}_{12}^{n+1}}{d_{12}^2\Delta t}. \tag{4.78}$$

Once again, the value of h_{12}^{n+1} from (4.78) can be used in (4.66) to improve the estimates of $\vec{v}_1^{n+1}$ and $\vec{v}_2^{n+1}$, and the difference vector between these becomes the new $\tilde{\vec{v}}_{12}^{n+1}$. And so on through several cycles of improvement of h_{12}^{n+1} to convergence within a chosen tolerance. In systems with polyatomic molecules repeated iteration

over successive bonds is performed until all of them have converged.

4.5.4 Further Comments on SHAKE

In the above derivations we have not mentioned accuracy. This is because the iteration cycles in SHAKE and RATTLE have their own order of error that is set by the iteration tolerances, and these could be less or more than the errors in the Verlet integrators. Clearly, if the constraints are satisfied to a greater accuracy than that implicit in the Verlet integrators, then we might expect the accuracy of the atomic trajectories to be similar to, though not better than, the Verlet schemes.

Another issue is the matter of symplecticness. It is thought that, because of the iteration cycles in the SHAKE algorithms, symplecticness cannot be guaranteed. The time reversibility property has been lost. Nevertheless, if the constraints are iterated to a high degree of accuracy, we might expect that simplecticness is retained to some degree, and the trajectories will be to some degree reversible. Certainly, the SHAKE algorithms are very stable when used properly, but in practice the iteration tolerances are not always very exacting (tolerances are of the order of 10^{-4} in the bond length are often used) so symplecticness on this account seems unlikely. Also, it should be noted that the original SHAKE algorithm and the leapfrog version place no constraints on the velocities, which means that these versions cannot be symplectic by definition. It is only the RATTLE algorithm that constrains the velocities in the required way, so that near-symplectic behaviour is possible if the constraints are solved to sufficient accuracy. Presumably iterating to the machine accuracy would make the algorithm fully symplectic, but this is computationally expensive.

There are also some practical issues for rigid bonds that are worthy of mention. The first is that each bond constraint removes one degree of freedom from the system. This must be factored in when calculating the system temperature. A system of N atoms with a stationary centre of mass and N_c constraints has $3(N-1)-N_c$ degrees of freedom and thus the temperature is calculated from

$$T=\left\langle \frac{1}{(3(N-1)-N_c)}\sum_{i=1}^{N} m_i v_i^2 \right\rangle. \tag{4.79}$$

It also follows from this that specifying redundant constraints i.e. constraints that fix distances already directly or indirectly constrained, will lead to an error in the calculation of the temperature. This danger is particularly pernicious in cases where bond constraints are used to construct a molecule that is formally a rigid body. The best advice is not to bother and use rigid body dynamics instead.

It is also worth noting that a common error in defining rigid bonds is failure to specify the molecular geometry with sufficient accuracy for a given SHAKE bond tolerance. Sometimes, particularly with rigid structures (again!) the constraints cannot *in principle* be satisfied because the specified bond lengths do not actually fit into the structure within the specified tolerance. It can be hard to spot this in practice.

Lastly, note that rigid bonds cannot be used to keep ring structures like benzene flat. No amount of cross linking of the atoms will stop significant out-of-plane displacement of the atoms. Furthermore the use of multiple cross-links often has one or more of the deleterious effects mentioned in the paragraphs above. Such inappropriate use of constraints is probably the source of the legend that they do not work with ring molecules. They do work if sufficient care is taken, for example controlling the "flatness" of the ring by appropriate dihedral potentials and the overall shape by appropriate bond angle potentials.

4.6 Algorithms for Rigid Molecules

In this section we assume the molecules are completely rigid entities. (We leave aside, for now, the possibility of linking rigid bodies in to larger molecules which possess a degree of flexibility. This is dealt with in section 4.7.) We start by summarising the results of chapter 2, sections 2.6 and 2.12 .

The translational equation of motion of a rigid molecule is simply

$$M\,\ddot{\vec{R}} = \vec{F} \tag{4.80}$$

where M is the molecular mass (assumed to be positioned at $\vec{R}$, the centre of mass) at and $\vec{F}$ is the molecular force. These are respectively given by appropriate sums of atomic masses m_i and forces $\vec{f}_i$, as in

$$M = \sum_{i=1}^{N_a} m_i \qquad \text{and} \qquad \vec{F} = \sum_{i=1}^{N_a} \vec{f}_i, \tag{4.81}$$

where N_a is the number of atoms in the molecule. These equations are identical in kind to the translational equations of motion for individual atoms and we may use any of the Verlet algorithms to obtain a numerical integration.

The rotational equations of motion of a rigid molecule are based on the relation between the rate of change of angular momentum, $\vec{J}$, and the molecular torque, $\vec{\tau}$, :

$$\dot{\vec{J}} = \vec{\tau}, \tag{4.82}$$

with

$$\vec{J} = \boldsymbol{I}\,\vec{\omega}. \tag{4.83}$$

The matrix $\boldsymbol{I}$ and the vector $\vec{\omega}$ are the moment of inertia tensor and the angular velocity respectively. $\boldsymbol{I}$ and $\vec{\tau}$ are defined by the relations

$$I^{\alpha\beta}=\sum_{i=1}^{N_a} m_i\left(\delta_{\alpha\beta}d_i^2-d_i^{\alpha}d_i^{\beta}\right) \qquad \text{and} \qquad \vec{\tau}=\sum_{i=1}^{N_a}\vec{\tau}_i=\sum_{i=1}^{N_a}\vec{d}_i\times\vec{f}_i, \tag{4.84}$$

where m_i is the mass of the $i'th$ atom, $\vec{d}_i$ is its displacement from the molecular centre of mass at $\vec{R}$, and $\vec{f}_i$ is the net force on the atom. In the case of a linear molecule $\boldsymbol{I}$ becomes a scalar quantity I, which is given by

$$I=\sum_{i=1}^{N_a} m_i d_i^2. \tag{4.85}$$

Combining (4.82) and (4.83) leads to the key equation of rotational motion:

$$\dot{\vec{\omega}}=\boldsymbol{I}^{-1}\left(\vec{\tau}-\dot{\boldsymbol{I}}\,\vec{\omega}\right) \tag{4.86}$$

Explicit formulae for the time derivative $\dot{\boldsymbol{I}}$ are given in equations (4.87) which are taken from equations (2.53).

$$\begin{aligned}
&\dot{I}^{xy}=\dot{I}^{yx}=\omega^z\left(I^{xx}-I^{yy}\right)-\omega^x I^{xz}+\omega^y I^{yz},\\
&\dot{I}^{yz}=\dot{I}^{zy}=\omega^x\left(I^{yy}-I^{zz}\right)-\omega^y I^{yx}+\omega^z I^{zx}, \qquad \text{(a)}\\
&\dot{I}^{zx}=\dot{I}^{xz}=\omega^y\left(I^{zz}-I^{xx}\right)-\omega^z I^{zy}+\omega^x I^{xy},\\
&\dot{I}^{xx}=2\left(\omega^y I^{xz}-\omega^z I^{xy}\right),\\
&\dot{I}^{yy}=2\left(\omega^z I^{yx}-\omega^x I^{yz}\right), \qquad \text{(b)}\\
&\dot{I}^{zz}=2\left(\omega^x I^{zy}-\omega^y I^{zx}\right).
\end{aligned} \tag{4.87}$$

In the case of linear molecules (4.86) becomes

$$\dot{\vec{\omega}}=I^{-1}\vec{\tau}. \tag{4.88}$$

The scalar moment of inertia is constant, so $\dot{I}=0.$ Thus, once the inverse $\boldsymbol{I}^{-1}$ or I^{-1} is calculated, the above equations may be used in a numerical integration to update the angular velocity $\vec{\omega}$.

Updating the molecular orientation requires exploiting one of the following equations for the general or linear molecule respectively

$$\begin{aligned}
\dot{\vec{d}}_i&=\vec{\omega}\times\vec{d}_i,\\
\dot{\vec{e}}_i&=\vec{\omega}\times\vec{e}_i,
\end{aligned} \tag{4.89}$$

in which $\vec{e}$ is the unit vector defining the direction of the linear molecule axis. As was

mentioned in chapter 2, sections 2.6 and 2.12 , these equations are not complete from a dynamical perspective and so do not offer a means to update the molecular orientation on their own. However we may easily construct viable schemes that are based on them, as we show below in section 4.6.1.2 .

4.6.1 Direct Numerical Integration of the Rigid Body Equations of Motion

We now outline numerical schemes for integrating the equations of motion for a rigid body in the *velocity Verlet* and *leapfrog* forms. The schemes are direct in the sense that it performs the integration entirely in the laboratory frame of reference. No reference to a local molecular frame is made. Both linear and general rigid bodies are dealt with.

4.6.1.1 Velocity Verlet:

At time step $n,$ we assume the atomic properties $m_i, \vec{r}_i^{\,n}, \vec{v}_i^{\,n}$ and $\vec{f}_i^{\,n}$ are known for all the atoms $(i=1,\ldots,N_a)$ in the molecule. The molecular properties $M, \vec{R}^n, \vec{V}^n, \vec{J}^n, \boldsymbol{I}^n(I^n), \quad \dot{\boldsymbol{I}}^n, \vec{\omega}^n, \vec{F}^n, \vec{\tau}^n, \vec{e}, \vec{d}_i^{\,n},$ may be calculated from the atomic data on-the-fly, or carried as additional simulation variables to save time. The integration is as follows, the accuracy of each stage in the scheme is given in terms of an order in the time step, $\Delta t,$ and follow from the parent velocity Verlet scheme.

a. $\vec{V}^{n+1/2}=\vec{V}^n+\vec{F}^n\Delta t/2\mathrm{M}+O(\Delta t^3)$

b. $\vec{J}^{n+1/2}=\vec{J}^n+\vec{\tau}^n\Delta t/2+O(\Delta t^3)$

c. $\vec{R}^{n+1}=\vec{R}^n+\vec{V}^{n+1/2}\Delta t+O(\Delta t^4)$

d. $\vec{\omega}^{n+1/2}=\vec{\omega}^n+(\boldsymbol{I}^n)^{-1}(\vec{\tau}^n-\dot{\boldsymbol{I}}^n\vec{\omega}^n)\Delta t/2+O(\Delta t^3)$ [general molecule]
$\vec{\omega}^{n+1/2}=\vec{\omega}^n+\vec{\tau}^n\Delta t/2\mathrm{I}+O(\Delta t^3)$ [linear molecule]

e. $\vec{d}_i^{\,n+1}=\vec{d}_i^{\,n}+(\vec{\omega}^{n+1/2}\times\vec{d}^{\,n})\Delta t+\vec{C}_{rot}+O(\Delta t^4)$ [general molecule, $(i=1,\ldots,N_a)$]
$\vec{e}^{\,n+1}=\vec{e}^{\,n}+(\vec{\omega}^{n+1/2}\times\vec{e}^{\,n})\Delta t+\vec{C}_{rot}+O(\Delta t^4)$ [linear molecule]

f. $\vec{r}_i^{\,n+1}=\vec{R}^{n+1}+\vec{d}_i^{\,n+1}+O(\Delta t^4)$ [general molecule $(i=1,\ldots,N_a)$]
$\vec{r}_i^{\,n+1}=\vec{R}^{n+1}+d_i^{n+1}\vec{e}^{\,n+1}+O(\Delta t^4)$ [linear molecule $(i=1,\ldots,N_a)$]

g. Calculate $\vec{F}^{n+1}, \vec{\tau}^{n+1}, \boldsymbol{I}^{n+1}.$

h. $\vec{V}^{n+1}=\vec{V}^{n+1/2}+\vec{F}^{n+1}\Delta t/2\mathrm{M}+O(\Delta t^2)$

i. $\vec{J}^{n+1}=\vec{J}^{n+1/2}+\vec{\tau}^{n+1}\Delta t/2+O(\Delta t^2)$

j. $\vec{\omega}^{n+1}=(\boldsymbol{I}^{n+1})^{-1}\vec{J}^{n+1}+O(\Delta t^2)$ [general molecule]
$\vec{\omega}^{n+1}=\vec{J}^{n+1}/I+O(\Delta t^2)$ [linear molecule]

k. $\vec{v}_i^{n+1} = \vec{V}^{n+1} + \vec{\omega}^{n+1} \times \vec{d}_i^{n+1} + O(\Delta t^2).$ $(i = 1, \ldots, N_a)$

In stage e. above the vector quantity $\vec{C}_{rot}$ represents a correction which ensures the integration preserves the norm of the vector $\vec{d}_i$ or $\vec{e}$, as required. We describe how this correction is determined in the following section.

4.6.1.2 Integrating equations containing the form $\vec{\omega} \times \vec{d}_i$

As was remarked in sections 2.6 and 2.12 , the equations presented in (4.89) do not fully express the dynamics of the rotating vector $\vec{d}_i$ or $\vec{e}$. In particular they do not account for the centripetal forces that constrain the vector length. To have any hope of retaining the structure of the rigid molecules in the course of the simulation, such effects must be incorporated. In the scheme outlined above, this applies to stage e.

In the velocity Verlet scheme a suitable algorithm for calculating the motion of the vector $\vec{d}_i$ is

$$\vec{d}_i^{n+1} = \vec{d}_i^n + \Delta t \, \dot{\vec{d}}_i^n + \frac{\Delta t^2}{2} \ddot{\vec{d}}_i^n + O(\Delta t^4), \tag{4.90}$$

in which the vector $\dot{\vec{d}}_i^n$ is distinct from the true time derivative $\dot{\vec{d}}_i$ in being the *approximate* velocity of $\vec{d}_i^n$, accurate to order $O(\Delta t^2)$. Equation (4.90) should be compared with equation (4.27), which has the same order of error $(O(\Delta t^4))$ despite the error of $O(\Delta t^2)$ in $\vec{v}_i^n$. Using (4.89) we can now re-write this as

$$\vec{d}_i^{n+1} = \vec{d}_i^n + \Delta t \, \vec{\omega}^n \times \vec{d}_i^n + \frac{\Delta t^2}{2} \frac{d}{dt} \vec{\omega}^n \times \vec{d}_i^n + O(\Delta t^4), \tag{4.91}$$

in which we understand $\vec{\omega}^n$ is accurate to order $O(\Delta t^2)$ only, since we are using $\dot{\vec{d}}_i^n$ in place of $\dot{\vec{d}}_i^n$. Completing the differentiation in (4.91) leads to

$$\vec{d}_i^{n+1} = \vec{d}_i^n + \Delta t \, \vec{\omega}^n \times \vec{d}_i^n + \frac{\Delta t^2}{2} \left(\dot{\vec{\omega}}^n \times \vec{d}_i^n + \vec{\omega}^n \times \dot{\vec{d}}_i^n \right) + O(\Delta t^4). \tag{4.92}$$

Replacing $\dot{\vec{d}}_i^n$ once again and rearranging gives

$$\vec{d}_i^{n+1} = \vec{d}_i^n + \Delta t \left(\vec{\omega}^n + \frac{\Delta t}{2} \dot{\vec{\omega}}^n \right) \times \vec{d}_i^n + \frac{\Delta t^2}{2} \vec{\omega}^n \times \left(\vec{\omega}^n \times \vec{d}_i^n \right) + O(\Delta t^4). \tag{4.93}$$

We may now define $\vec{\omega}^{n+1/2}$ as

$$\omega^{n+1/2} = \omega^n + \frac{\Delta t}{2} \dot{\vec{\omega}}^n + O(\Delta t^3), \tag{4.94}$$

which is consistent with the accuracy of the velocity Verlet scheme. Using equation (4.86) this may be written as

$$\vec{\omega}^{n+1/2}=\vec{\omega}^{n}+(\boldsymbol{I}^{n})^{-1}(\vec{\tau}^{n}-\dot{\boldsymbol{I}}^{n}\vec{\omega}^{n})\Delta t/2+O(\Delta t^{3}) \tag{4.95}$$

which represents stage d. in the above scheme. Equation (4.93) can now be written as

$$\vec{d}_i^{\,n+1}=\vec{d}_i^{\,n}+\Delta t\,\vec{\omega}^{n+1/2}\times\vec{d}_i^{\,n}+\frac{\Delta t^2}{2}\vec{\omega}^{n}\times\left(\vec{\omega}^{n}\times\vec{d}_i^{\,n}\right)+O(\Delta t^4), \tag{4.96}$$

which represents stage e. in the integration scheme. It follows that

$$\vec{C}_{rot}=\frac{\Delta t^2}{2}\vec{\omega}^{n}\times\left(\vec{\omega}^{n}\times\vec{d}_i^{\,n}\right)=\frac{\Delta t^2}{2}\left((\vec{\omega}^{n}\cdot\vec{d}_i^{\,n})\vec{\omega}^{n}-(\vec{\omega}^{n}\cdot\vec{\omega}^{n})\vec{d}_i^{\,n}\right). \tag{4.97}$$

The form of $\vec{C}_{rot}$ is consistent with it being due to centripetal forces, since, if weighted by mass m_i, it becomes exactly the centripetal force of a mass m_i at location $\vec{d}_i^{\,n}$ when spun around the axis of $\vec{\omega}^{n}$. Equation (4.96) is thus the *full* form of the equation for stage e. of the scheme.

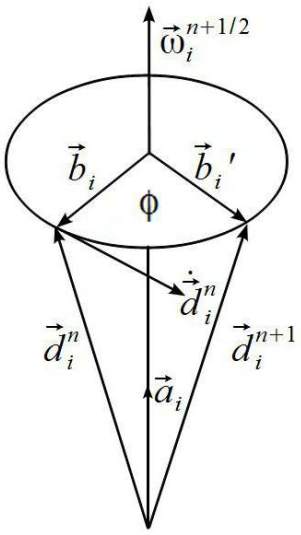

Figure 4.3: Rotation of vector $\vec{d}_i^{\,n}$ around the vector $\vec{\omega}^{n+1/2}$

Unfortunately, integration of (4.96) is accurate only to order $O(\Delta t^4)$ which means that, despite the the inclusion of the centripetal force, it cannot guarantee the conservation of the magnitude of vector $\vec{d}_i$ nor the structure of the rigid molecule in the long term.

The problem is shown in figure 4.3, where the vector $\vec{d}_i^{\,n}$ is rotated about the angular velocity vector $\vec{\omega}^{n+1/2}$ through an angle $\phi=\omega^{n+1/2}\Delta t$ in a time Δt to become $\vec{d}_i^{\,n+1}$. It is required that the rotation conserves the length of $\vec{d}_i$. However, the integration

of (4.96) does not guarantee motion along the shown circular path.

We can fix the problem with equation (4.96) as follows. We first define two auxiliary vectors $\vec{a}_i$ and $\vec{b}_i$ as follows:

$$\vec{a}_i=(\omega^{\vec{n+1/2}}\cdot\vec{d}_i^{\,n})\,\omega^{\vec{n+1/2}}/|\omega^{n+1/2}|^2, \tag{4.98}$$

where $\vec{a}_i$ is aligned along the vector $\vec{\omega}^{n+1/2}$ by design and

$$\vec{b}_i=\vec{d}_i^{\,n}-\vec{a}_i. \tag{4.99}$$

Thus vectors $\vec{a}_i,\vec{b}_i$ and $\vec{d}_i^{\,n}$ complete a right-angled triangle, with $\vec{b}_i$ in the plane of the rotation circle shown in figure 4.3. Rotation of $\vec{b}_i$ to $\vec{b}_i'$ through the angle ϕ where

$$\phi=\Delta t\,\omega^{n+1/2}=\Delta t\,\frac{|\dot{\vec{d}}_i^{\,n}|}{|b_i|} \tag{4.100}$$

defines the required change $\vec{d}_i^{\,n}\rightarrow\vec{d}_i^{\,n+1}$ over the time interval $\Delta t.$ Since $\vec{b}_i$ and $\dot{\vec{d}}_i^{\,n}$ are orthogonal vectors by definition, basic geometry shows that

$$\vec{b}_i'=\vec{b}_i\cos(\phi)+\Delta t\,\dot{\vec{d}}_i^{\,n}\sin(\phi)/\phi. \tag{4.101}$$

From this we obtain the result of the rotation as

$$\vec{d}_i^{\,n+1}=\vec{b}_i'+\vec{a}_i \tag{4.102}$$

This approach guarantees the conservation of the magnitude of the rotating vector and the centripetal forces are automatically allowed for. Equation (4.102) replaces equation (4.96) which determines stage e. of the scheme.

In equation (4.101) some caution is clearly required when ϕ is small, since then the ratio $\sin(\phi)/\phi$ needs to be handled properly in the limit $\phi\rightarrow 0.$ It is therefore recommended that the following expansion is used for this circumstance:

$$\frac{\sin(\phi)}{\phi}=1-\frac{\phi^2}{6}\left(1-\frac{\phi^2}{20}\left(1-\frac{\phi^2}{42}\right)\right), \tag{4.103}$$

which has an error $O(\phi^8)$ and is therefore highly accurate for ϕ less than $\sim 1^o.$

Adaptation for the integration of $\vec{e}^{\,n}$ in the case of linear molecules is straightforward.

4.6.1.3 Leapfrog Verlet:

At time step n, the atomic properties $m_i, \vec{r}_i^{\,n}, \vec{v}_i^{\,n-1/2}$ and $\vec{f}_i^{\,n}$ are assumed known for all the atoms $(i=1,\ldots,N_a)$ in the molecule. The molecular properties $M, \vec{R}^n, \vec{V}^{n-1/2}, \vec{J}^{n-1/2}, \quad \boldsymbol{I}^n(I^n), \dot{\boldsymbol{I}}^n, \vec{\omega}^{n-1/2}, \vec{F}^n, \vec{\tau}^n, \vec{e}, \vec{d}_i^{\,n},$ can mostly be calculated on-the-fly from the atomic data, but variables $\vec{J}^{n-1/2}$ and $\vec{\omega}^{n-1/2},$ which are defined at half time steps must be carried as additional simulation variables. These variables also need to be initialised, which is a matter of backward integration from the time step $(n=0)$ (at which all other variables are initialised) as follows:

$$\begin{aligned}
\vec{J}^{-1/2} &= \vec{J}^0 - \vec{\tau}^0 \Delta t/2 \\
\vec{\omega}^{-1/2} &= \vec{\omega}^0 - (\boldsymbol{I}^0)^{-1}(\vec{\tau}^0 - \dot{\boldsymbol{I}}^0 \vec{\omega}^0)\Delta t/2 \\
\vec{V}^{-1/2} &= M^{-1}\sum_{i=1}^{N_a} m_i \vec{v}_i^{\,-1/2}.
\end{aligned} \tag{4.104}$$

From these starting conditions the integration may continue as follows:

a. $\vec{V}^{n+1/2} = \vec{V}^{n-1/2} + \vec{F}^n \Delta t/M + O(\Delta t^3)$

b. $\vec{J}^{n+1/2} = \vec{J}^{n-1/2} + \vec{\tau}^n \Delta t + O(\Delta t^3)$

c. $\vec{R}^{n+1} = \vec{R}^n + \vec{V}^{n+1/2}\Delta t + O(\Delta t^4)$

d. $\vec{\omega}^n = (\boldsymbol{I}^n)^{-1}(\vec{J}^{n+1/2} + \vec{J}^{n-1/2})/2 + O(\Delta t^2)$ [general molecule]
$\vec{\omega}^n = (\vec{J}^{n+1/2} + \vec{J}^{n-1/2})/2I + O(\Delta t^2)$ [linear molecule]

e. Calculate $\dot{\boldsymbol{I}}^n$ from $\boldsymbol{I}^n$ and $\vec{\omega}^n$ [general molecule]

f. $\vec{\omega}^{n+1/2} = \vec{\omega}^{n-1/2} + (\boldsymbol{I}^n)^{-1}(\vec{\tau}^n - \dot{\boldsymbol{I}}^n \vec{\omega}^n)\Delta t + O(\Delta t^3)$ [general molecule]
$\vec{\omega}^{n+1/2} = \vec{\omega}^{n-1/2} + \vec{\tau}^n \Delta t/I + O(\Delta t^3)$ [linear molecule]

g. $\vec{d}_i^{\,n+1} = \vec{d}_i^{\,n} + (\vec{\omega}^{n+1/2} \times \vec{d}^{\,n})\Delta t + \vec{C}_{rot} + O(\Delta t^4)$ [general molecule, $(i=1,\ldots,N_a)$]
$\vec{e}^{\,n+1} = \vec{e}^{\,n} + (\vec{\omega}^{n+1/2} \times \vec{e}^{\,n})\Delta t + \vec{C}_{rot} + O(\Delta t^4)$ [linear molecule]

h. $\vec{r}_i^{\,n+1} = \vec{R}^{n+1} + \vec{d}_i^{\,n+1} + O(\Delta t^4)$ $(i=1,\ldots,N_a)$

i. $\vec{v}_i^{\,n+1} = \vec{V}^{n+1} + \vec{\omega}^{n+1} \times \vec{d}_i^{\,n+1} + O(\Delta t^2).$ $(i=1,\ldots,N_a)$

Once again the stage g requires the amendment outlined in section 4.6.1.2 to ensure the molecule retains structural integrity. The order of accuracy in each of these equations is, once again, set by the underlying leapfrog scheme.

4.6.2 Numerical Integration of Euler's Rotational Equations of Motion

Euler's rotational equations of motion were described in chapter 2, section 2.9 . Equations (2.72) describes the rate of change of angular velocity $\vec{\omega}^p$, in the local frame of the molecule (which is also the principal frame), while equation (2.79) describes rate of change of the orientation of the local frame (in terms of the Euler angles) in the laboratory frame. This coupled set of equations may (in principle) be solved numerically in the following manner:

1. At a given time t, construct the rotation matrix $\boldsymbol{R}$ from the Euler angles (ϕ, θ, ψ) using equation (2.65),

2. Transform the moment of inertia tensor, torque and angular velocity vectors to the local frame, i.e. $\boldsymbol{I}^p = \tilde{\boldsymbol{R}}\boldsymbol{I}\boldsymbol{R}$, $\vec{\tau}^p = \tilde{\boldsymbol{R}}\vec{\tau}$ and $\vec{\omega}^p = \tilde{\boldsymbol{R}}\vec{\omega}$.

3. Update the angular velocity from time t to time $t+\Delta t$ using Euler's equation (2.72), which becomes:

$$\begin{aligned}
\omega^{p^x}(t+\Delta t) &= \omega^{p^x}(t) + \Delta t\left(\tau^{p^x} + \omega^{p^y}\omega^{p^z}\left(I^{p^{yy}} - I^{p^{zz}}\right)\right)/I^{p^{xx}}, \\
\omega^{p^y}(t+\Delta t) &= \omega^{p^y}(t) + \Delta t\left(\tau^{p^y} + \omega^{p^z}\omega^{p^x}\left(I^{p^{zz}} - I^{p^{xx}}\right)\right)/I^{p^{yy}}, \\
\omega^{p^z}(t+\Delta t) &= \omega^{p^z}(t) + \Delta t\left(\tau^{p^z} + \omega^{p^x}\omega^{p^y}\left(I^{p^{xx}} - I^{p^{yy}}\right)\right)/I^{p^{zz}},
\end{aligned} \tag{4.105}$$

4. Update the Euler angles from t to $t+\Delta t$ using equation (2.79) in the form:

$$\begin{aligned}
\phi(t+\Delta t) &= \phi(t) + \Delta t\left(\omega^{p^x}\frac{\sin\psi}{\sin\theta} + \omega^{p^y}\frac{\cos\psi}{\sin\theta}\right), \\
\theta(t+\Delta t) &= \theta(t) + \Delta t\left(\omega^{p^x}\cos\psi - \omega^{p^y}\sin\psi\right), \\
\psi(t+\Delta t) &= \psi(t) + \Delta t\left(-\omega^{p^x}\frac{\sin\psi}{\tan\theta} - \omega^{p^y}\frac{\cos\psi}{\tan\theta} + \omega^{p^z}\right).
\end{aligned} \tag{4.106}$$

Unfortunately, as was noted in chapter 2, section 2.9 , this scheme will become unstable whenever θ approaches zero in the equations (4.106) – something we can not prevent from happening in the normal evolution of the system. For this reason we will not persist with this approach and instead consider the alternatives that define the orientation using quaternions. The original scheme for this was devised by Evans and Murad [7] using the Gear predictor-corrector [18] scheme. In keeping with the content of this book, we shall describe algorithms using the Verlet integrators.

4.6.3 Fincham's (Leapfrog) Implicit Quaternion Algorithm

Fincham's rotational algorithm is, in essence, the leapfrog form of Evans and Murad's scheme which used the Gear predictor corrector. The key variables are the angular velocity in the local or principal frame $\vec{\omega}^p$, the quaternion parameters $\breve{q} = (q_0, q_1, q_2, q_3)$, defined in equation (2.80), and the quaternion matrix $\boldsymbol{Q}$, defined in equation (2.83). We also require the rotation matrix $\boldsymbol{R}$, which is obtained from the quaternion parameters using equation (2.82). This is needed to locate the atoms of

the molecule in space relative to its centre of mass.

Since this is a leapfrog algorithm, the angular velocities are defined at the half time step and the quaternions at the full time step, so each time step of the algorithm starts with the known quantities $\vec{\omega}_i^{p\,n-1/2}$ and $\breve{q}_i^n$, where index i refers to the $i'th$ molecule. Once the torque $\vec{\tau}_i^{p\,n}$ has been calculated in the local frame, the angular velocity may be updated using Euler's equation (2.72):

$$\begin{aligned}
\omega_i^{px\,n+1/2} &= \omega_i^{px\,n-1/2} + \Delta t\left(\tau_i^{px\,n} + \omega_i^{py\,n}\omega_i^{pz\,n}\left(I_i^{p}{}^{yy} - I_i^{p}{}^{zz}\right)\right)/I_i^{p}{}^{xx}, \\
\omega_i^{py\,n+1/2} &= \omega_i^{py\,n-1/2} + \Delta t\left(\tau_i^{py\,n} + \omega_i^{pz\,n}\omega_i^{px\,n}\left(I_i^{p}{}^{zz} - I_i^{p}{}^{xx}\right)\right)/I_i^{p}{}^{yy}, \qquad +O\left(\Delta t^3\right) \\
\omega_i^{pz\,n+1/2} &= \omega_i^{pz\,n-1/2} + \Delta t\left(\tau_i^{pz\,n} + \omega_i^{px\,n}\omega_i^{py\,n}\left(I_i^{p}{}^{xx} - I_i^{p}{}^{yy}\right)\right)/I_i^{p}{}^{zz}.
\end{aligned} \tag{4.107}$$

There is an immediate problem with these equations: the angular velocities $\vec{\omega}_i^{p\,n}$ are not known at the start of the time step; they can only be calculated once $\vec{\omega}_i^{p\,n+1/2}$ has been obtained *vis*:

$$\vec{\omega}_i^{p\,n} = \left(\vec{\omega}_i^{p\,n+1/2} + \vec{\omega}_i^{p\,n-1/2}\right)/2. \tag{4.108}$$

The solution to this dilemma is to use iteration. Equation (4.107) is first solved using $\vec{\omega}_i^{p\,n-1/2}$ in place of $\vec{\omega}_i^{p\,n}$ and this allows a *first estimate* of $\vec{\omega}_i^{p\,n}$ using (4.108). The first estimate can then be used again in (4.107) and (4.108) to obtain an improved estimate. Repeated iteration of (4.107) and (4.108) with successive improvements of $\vec{\omega}_i^{p\,n}$ will converge on the required value of $\vec{\omega}_i^{p\,n+1/2}$. Only on convergence is the error quoted in equation truly $O\left(\Delta t^3\right)$. It is the need for such iteration that makes this an *implicit* scheme.

The converged value of $\vec{\omega}_i^{p\,n+1/2}$ can then be used to update the quaternions using equation (2.83):

$$\breve{q}_i^{n+1} = \breve{q}_i^n + \frac{\Delta t}{2}\boldsymbol{Q}_i^{n+1/2}\tilde{\vec{\omega}}_i^{p\,n+1/2} + O\left(\Delta t^4\right). \tag{4.109}$$

Once again we have a problem in that $\boldsymbol{Q}_i^{n+1/2}$ is not known at this stage because $\boldsymbol{Q}_i$, as shown by equation (2.83) is a function of $\breve{q}_i$, which is defined only at the full time step (i.e. $\boldsymbol{Q}_i^n \equiv \boldsymbol{Q}_i\left(q_{0\mathrm{i}}^n, q_{1\mathrm{i}}^n, q_{2\mathrm{i}}^n, q_{3\mathrm{i}}^n\right)$. However $\boldsymbol{Q}_i^{n+1/2}$ can be obtained once $\breve{q}_i^{n+1}$ is known:

$$\boldsymbol{Q}_i^{n+1/2} = \left(\boldsymbol{Q}_i^{n+1}\left(\breve{q}_i^{n+1}\right) + \boldsymbol{Q}_i^n\left(\breve{q}_i^n\right)\right)/2. \tag{4.110}$$

It therefore follows that $\boldsymbol{Q}_i^{n+1/2}$ can be obtained by iteration, starting from equation (4.109) with $\boldsymbol{Q}_i^n$ in place of $\boldsymbol{Q}_i^{n+1/2}$ as the starting value. A few iterations suffice to

obtain $\boldsymbol{Q}_i^{n+1/2}$ (and hence $\breve{q}_i^{n+1}$) to an accurate enough value.

To complete the algorithm, there is a further detail we need to attend to: the quaternions must always obey the *normalisation condition* given in equation (2.81). To ensure this, we must *impose* normalisation at *each iteration* of (4.109). This amounts to calculating the quantity:

$$|\breve{q}_i^{n+1}| = \sqrt{(q_{0\mathrm{i}}^{n+1})^2 + (q_{1\mathrm{i}}^{n+1})^2 + (q_{2\mathrm{i}}^{n+1})^2 + (q_{3\mathrm{i}}^{n+1})^2} \tag{4.111}$$

and then renormalising $\breve{q}_i^{n+1}$ as follows:

$$\breve{q}_i^{n+1} \leftarrow \breve{q}_i^{n+1} / |\breve{q}_i^{n+1}|. \tag{4.112}$$

It only remains to compute the rotation matrix $\boldsymbol{R}_i$ from equation (2.82) to re-orientate the molecule correctly in space at time step $n+1,$ which amounts to applying equation (2.64) to all local atomic coordinates.

Fincham's implicit quaternion algorithm is widely used and it performs reasonably well. However it is not as robust as might be expected for a leapfrog scheme, in particular when larger time steps are used. A possible explanation of this is the need for iteration, but it may also be because the imposition of the normalisation (4.112) is too *ad hoc* (a criticism that can also be levelled at the scheme of Evans and Murad). Algorithms developed later, which handle the normalisation more considerately, appear to be more robust and permit the use of larger time steps.

Examples of such schemes are those devised by Sonnenschein [23] and Powles *et al.* [24] which make use of second derivatives of the quaternions. We shall show how these ideas may be incorporated into the leapfrog and velocity Verlet schemes below, but first we derive the second order equations needed.

4.6.4 Rotational Motion Using Second Derivatives of the Quaternions

We begin with the quaternion equation (2.83) presented in chapter 2 and differentiate with respect to time to give

$$\ddot{\breve{q}} = \frac{1}{2}\dot{\boldsymbol{Q}}\breve{\omega}^p + \frac{1}{2}\boldsymbol{Q}\dot{\breve{\omega}}^p, \tag{4.113}$$

in which

$$\dot{\boldsymbol{Q}} = \begin{bmatrix} \dot{q}_0 & -\dot{q}_1 & -\dot{q}_2 & -\dot{q}_3 \\ \dot{q}_1 & \dot{q}_0 & -\dot{q}_3 & \dot{q}_2 \\ \dot{q}_2 & \dot{q}_3 & \dot{q}_0 & -\dot{q}_1 \\ \dot{q}_3 & -\dot{q}_2 & \dot{q}_1 & \dot{q}_0 \end{bmatrix}, \tag{4.114}$$

and

$$\dot{\breve{\omega}}^p = \begin{bmatrix} 0 \\ \dot{\omega}^{p^x} \\ \dot{\omega}^{p^y} \\ \dot{\omega}^{p^z} \end{bmatrix} = \begin{bmatrix} 0 \\ \left(\tau^{p^x} + \omega^{p^y}\omega^{p^z}\left(I^{p^{yy}} - I^{p^{zz}}\right)\right)/I^{p^{xx}} \\ \left(\tau^{p^y} + \omega^{p^z}\omega^{p^x}\left(I^{p^{zz}} - I^{p^{xx}}\right)\right)/I^{p^{zz}} \\ \left(\tau^{p^z} + \omega^{p^x}\omega^{p^y}\left(I^{p^{xx}} - I^{p^{yy}}\right)\right)/I^{p^{zz}} \end{bmatrix}. \tag{4.115}$$

Equation (4.115) derives directly from Euler's equation (2.72).

Inserting equation (2.84) into the first term on the right of (4.113) leads to

$$\frac{1}{2}\dot{\boldsymbol{Q}}\breve{\omega}^p = \dot{\boldsymbol{Q}}\tilde{\boldsymbol{Q}}\breve{\dot{q}} = -\boldsymbol{Q}\dot{\tilde{\boldsymbol{Q}}}\breve{\dot{q}}, \tag{4.116}$$

in which we have used the time derivative of the identity $\boldsymbol{Q}\tilde{\boldsymbol{Q}} = \mathbf{1}$ to obtain the final expression right of (4.116). The product $\dot{\tilde{\boldsymbol{Q}}}\breve{\dot{q}}$ is easily shown to be

$$\dot{\tilde{\boldsymbol{Q}}}\breve{\dot{q}} = \begin{bmatrix} \dot{q}^2 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{4.117}$$

With the results (4.115) to (4.117) equation (4.113) can be transformed into

$$\begin{bmatrix} \ddot{q}_0 \\ \ddot{q}_1 \\ \ddot{q}_2 \\ \ddot{q}_3 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & q_0 \end{bmatrix}\begin{bmatrix} -2\dot{q}^2 \\ \left(\tau^{p^x} + \omega^{p^y}\omega^{p^z}\left(I^{p^{yy}} - I^{p^{zz}}\right)\right)/I^{p^{xx}} \\ \left(\tau^{p^y} + \omega^{p^z}\omega^{p^x}\left(I^{p^{zz}} - I^{p^{xx}}\right)\right)/I^{p^{zz}} \\ \left(\tau^{p^z} + \omega^{p^x}\omega^{p^y}\left(I^{p^{xx}} - I^{p^{yy}}\right)\right)/I^{p^{zz}} \end{bmatrix}. \tag{4.118}$$

The second order quaternion equation (4.118) has been shown by Sonnenschein to be more stable than original scheme by Evans and Murad while conserving the normalisation of the quaternions [23]. These properties make it a more attractive option than either the Fincham or Evans and Murad scheme. Armed with this equation, we can now present the Verlet integration algorithms for quaternions as follows.

4.6.5 The Leapfrog Scheme with Quaternion Second Derivatives

At the start of the $n'th$ time step, we assume $\breve{q}_i^n$, $\dot{\breve{q}}_i^{n-1/2}$ and $\vec{\tau}_i^n$ are known for each molecule i. We also assume that the principal moments of inertia $I_i^{p^{xx}}$, $I_i^{p^{yy}}$, $I_i^{p^{zz}}$, have been calculated previously. The scheme involves some iteration, as is apparent in what follows:

1. Construct rotation matrix $\boldsymbol{R}_i^n$ from $\breve{q}_i^n$ using (2.82) and use it to transform $\vec{\tau}_i^n$ to $\vec{\tau}_i^{p^n}$. Estimate a *first approximation* of $\vec{\omega}_i^{p^n}$ using (2.84) with $\tilde{\boldsymbol{Q}}_i^n$ and $\dot{\breve{q}}_i^{n-1/2}$. Use these results to calculate $\ddot{\breve{q}}_i^n$ at time step n from equation (4.118).

2. Update the quaternion velocity $\dot{\breve{q}}_i^{n-1/2}$ to step $n+1/2$:
$$\dot{\breve{q}}_i^{n+1/2} = \dot{\breve{q}}_i^{n-1/2} + \Delta t\, \ddot{\breve{q}}_i^n + O(\Delta t^3),$$

3. Re-calculate $\dot{\breve{q}}_i^n$ using
$$\dot{\breve{q}}_i^n = \frac{1}{2}\left(\dot{\breve{q}}_i^{n+1/2} + \dot{\breve{q}}_i^{n-1/2}\right),$$

4. Use $\dot{\breve{q}}_i^n$ from step 3. with $\tilde{\boldsymbol{Q}}_i^n$ in (2.84) to obtain an improved estimate of $\vec{\omega}_i^{p^n}$ and recalculate $\ddot{\breve{q}}_i^n$ using equation (4.118).

5. Repeat steps 2-4 until $\vec{\omega}_i^{p^n}$ and $\dot{\breve{q}}_i^{n+1/2}$ converge.

6. Update $\breve{q}_i^n$ to $\breve{q}_i^{n+1}$:
$$\breve{q}_i^{n+1} = \breve{q}_i^n + \Delta t\, \dot{\breve{q}}_i^{n+1/2} + O(\Delta t^4),$$

7. Calculate $\boldsymbol{R}_i^{n+1}$ from $\breve{q}_i^{n+1}$ using equation (2.82) and rotate the molecule to the correct orientation at step $n+1$ using equation (2.64).

8. Repeat steps 1-7 for subsequent time steps.

Note that only when the convergence of the iteration in this scheme is complete, will the order of errors indicated in the equations be true. Note also that the iterations do not require re-calculation of the torque $\vec{\tau}_i^{p^n}$, so are not expensive to compute.

4.6.6 The Velocity Verlet Scheme with Quaternion Second Derivatives

At the start of the $n'th$ time step, we assume $\breve{q}_i^n$, $\dot{\breve{q}}_i^n$ and $\vec{\tau}_i^n$ together with $I_i^{p^{xx}}$, $I_i^{p^{yy}}$, $I_i^{p^{zz}}$, are known for each molecule i. The scheme is then:

1. Construct rotation matrix $\boldsymbol{R}_i^n$ from $\breve{q}_i^n$ using (2.82) and use it to transform $\vec{\tau}_i^n$ to $\vec{\tau}_i^{p^n}$. Construct $\vec{\omega}_i^{p^n}$ using (2.84). Use these results to calculate $\ddot{\breve{q}}_i^n$ at time step n using equation (4.118).

2. Update the quaternion velocity to step $(n+1/2)$:
$$\dot{\breve{q}}_i^{n+1/2} = \dot{\breve{q}}_i^n + \frac{\Delta t}{2}\ddot{\breve{q}}_i^n + O(\Delta t^3)$$

3. Update the quaternions to step $n+1$:

$$\breve{q}_i^{n+1} = \breve{q}_i^n + \Delta t\, \dot{\breve{q}}_i^{n+1/2} + O(\Delta t^4)$$

4. Use $\breve{q}_i^{n+1}$ to construct $\boldsymbol{R}_i^{n+1}$ and re-orientate the molecules for step $n+1$ using equation (2.64). Then calculate $\vec{\tau}_i^{p\,n+1}$.

5. The vector $\vec{\omega}^{p\,n+1}$ is unknown at this stage, but we can obtain $\breve{q}_i^{n+1/2}$ from the average of $\breve{q}_i^n$ and $\breve{q}_i^{n+1}$ to construct $\tilde{\boldsymbol{Q}}_i^{n+1/2}$, which can be used with $\dot{\breve{q}}_i^{n+1/2}$ to obtain $\vec{\omega}^{p\,n+1/2}$ from (2.84). This is used as a *first approximation* to $\vec{\omega}^{p\,n+1}$.

6. Using $\vec{\omega}^{p\,n+1}$ in equation (4.118), calculate $\ddot{\breve{q}}_i^{n+1}$ and then update the quaternion velocity to step $n+1$:
$$\dot{\breve{q}}_i^{n+1} = \dot{\breve{q}}_i^{n+1/2} + \frac{\Delta t}{2}\ddot{\breve{q}}_i^{n+1} + O(\Delta t^2)$$

7. Using $\breve{q}_i^{n+1}$ and $\dot{\breve{q}}_i^{n+1}$, $\vec{\omega}_i^{p\,n+1}$ can again be obtained from equation (2.84) and then returned to step 6. to calculate $\dot{\breve{q}}_i^{n+1}$ more accurately.

8. Steps 6. and 7. may be iterated until $\dot{\breve{q}}_i^{n+1}$ and $\vec{\omega}_i^{p\,n}$ are accurately converged.

9. Repeat steps 1.-8. for subsequent time steps.

As before, only when the convergence is complete, will the order of errors indicated in the scheme be true. The algorithm should be symplectic if convergence is accurate. As with leapfrog the iteration in the scheme does not require re-calculation of the torque $\vec{\tau}_i^{p\,n+1}$, so is not computationally expensive.

4.6.7 The NOSQUISH Algorithm

The NOSQUISH algorithm, due to Miller *et al.* [25], is yet another rigid body rotational algorithm based on quaternions, but unlike those described above it does not require iteration and is fully symplectic. It therefore gives rise to a very stable integration scheme which we give in outline below. The details are fully described in the original paper [25].

We begin by defining quaternion momenta:

$$\breve{p} = \begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \end{bmatrix} = 2 \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & q_0 \end{bmatrix} \begin{bmatrix} 0 \\ I^{p\,xx}\omega^{p\,x} \\ I^{p\,yy}\omega^{p\,y} \\ I^{p\,zz}\omega^{p\,z} \end{bmatrix} \tag{4.119}$$

and quaternion torques:

$$\breve{\tau}=\begin{bmatrix}\tau_0\\ \tau_1\\ \tau_2\\ \tau_3\end{bmatrix}=2\begin{bmatrix}q_0 & -q_1 & -q_2 & -q_3\\ q_1 & q_0 & -q_3 & q_2\\ q_2 & q_3 & q_0 & -q_1\\ q_3 & -q_2 & q_1 & q_0\end{bmatrix}\begin{bmatrix}0\\ \tau^{p\,x}\\ \tau^{p\,y}\\ \tau^{p\,z}\end{bmatrix}. \tag{4.120}$$

These, together with the quaternions, are the main dynamical variables of the NOSQUISH scheme. The system Hamiltonian for a rigid molecule is given as

$$H(\breve{q},\breve{p})=\sum_{k=0}^{4}h_k(\breve{q},\breve{p}). \tag{4.121}$$

where each $h_k(\breve{q},\breve{p})$ is a sub-Hamiltonian:

$$\begin{aligned}&\text{(kinetic)} && h_k(\breve{q},\breve{p})=\frac{1}{8I_k^p}[\breve{p}\cdot\hat{P}_k\breve{q}] \text{ with } (k=0,\ldots,3),\\ &\text{(potential)} && h_4(\breve{q},\breve{p})=\Phi(\breve{q})\end{aligned} \tag{4.122}$$

and $\hat{P}_k$ is a *permutation operator* defined by the operations:

$$\begin{aligned}\hat{P}_0\breve{q}&=(q_0,q_1,q_2,q_3),\\ \hat{P}_1\breve{q}&=(-q_1,q_0,q_3,-q_2),\\ \hat{P}_2\breve{q}&=(-q_2,-q_3,q_0,q_1),\\ \hat{P}_3\breve{q}&=(-q_3,q_2,-q_1,q_0),\end{aligned} \tag{4.123}$$

which can be seen to generate the columns of the matrix $\boldsymbol{Q}$ defined in equation (2.83). The quantities given as $I_k^p,$ with $(k=1,\ldots,3),$ are the appropriate elements of a 4×4 diagonal matrix which is the analogue of the moment of inertia tensor $(\boldsymbol{I}^p)$ in the local frame i.e. $I^{p1}=I^{pxx}$, $I^{p2}=I^{pyy}$, $I^{p3}=I^{pzz}$. Also I^{p0} is an arbitrary non-zero constant required to complete the formal structure of the equations. The equations of motion for this Hamiltonian, derived in the usual way using equations (2.17), conserve the normalization of the quaternions and zero as the first element of $\breve{\omega}^p$.

The integration algorithm for NOSQUISH is derived via the scheme outlined in section 4.3 , involving the discretization of the Liouville equation and employing the Trotter expansion of the Liouville operator. Under this procedure the Liouville propagator for one time step can be written as

$$e^{i\Delta t\hat{L}}=e^{i\Delta t/2\hat{L}_4}\left[e^{i\delta t/2\hat{L}_3}e^{i\delta t/2\hat{L}_2}e^{i\delta t\hat{L}_1}e^{i\delta t/2\hat{L}_2}e^{i\delta t/2\hat{L}_3}\right]^{m_{rot}}e^{i\Delta t/2\hat{L}_4}+O(\Delta t^3) \tag{4.124}$$

in which

$$i\hat{L}_k = \breve{\nabla}_p h_k(\breve{q},\breve{p})\cdot\breve{\nabla}_q - \breve{\nabla}_q h_k(\breve{q},\breve{p})\cdot\breve{\nabla}_p \tag{4.125}$$

and $\delta t = \Delta t / m_{rot}$, with m_{rot} as an integer that determines the degree of splitting of the standard time step Δt into sub-steps for an accurate and economical integration of the "free rotation". Typically $m_{rot} = 10.$ (It should be noted that in this scheme $h_0(\breve{q},\breve{p}) \equiv 0$ and $e^{it\hat{L}_0} \equiv 1.$) The operators $e^{i\Delta t/2\hat{L}_k}$, with $(k=1,\ldots,3),$ have the following effect:

$$\begin{aligned} e^{i\delta t\hat{L}_k}\breve{q} &= \cos(\zeta_k \delta t)\breve{q} + \sin(\zeta_k \delta t)\hat{P}_k\breve{q}, \\ e^{i\delta t\hat{L}_k}\breve{p} &= \cos(\zeta_k \delta t)\breve{p} + \sin(\zeta_k \delta t)\hat{P}_k\breve{p}, \end{aligned} \tag{4.126}$$

where

$$\zeta_k = \frac{1}{4\mathrm{I}^{p_k}}\vec{p}\cdot\hat{P}_k\vec{q}. \tag{4.127}$$

The operator $e^{i\Delta t/2\hat{L}_4}$ updates the quaternion momentum by one half time step as follows

$$e^{i\Delta t/2\hat{L}_4}\breve{p} = \breve{p} + \frac{\Delta t}{2}\breve{\tau}. \tag{4.128}$$

With these equations we may now write out the NOSQUISH scheme, which bears a strong resemblance to a velocity Verlet scheme. At the start of a time step, $n,$ we assume that the principal moments of inertia $I^{p_{xx}}, I^{p_{yy}}, I^{p_{zz}},$ are available and that $\breve{q}_i^n$, $\vec{\omega}_i^{p^n}$ and $\vec{\tau}_i^n$ are known for each molecule $i.$ The sub-step time, $\delta t,$ is set as $\delta t = \Delta t / m_{rot}.$ The scheme is then:

1. Calculate the quaternion momenta, $\breve{p}_i^n$ and torque $\breve{\tau}_i^n,$ from equations (4.119) and (4.120) respectively.

2. Update the quaternion momenta to step $(n+1/2)$:
$$\breve{p}_i^{n+1/2} = \breve{p}_i^n + \frac{\Delta t}{2}\breve{\tau}_i^n$$

3. Perform free rotation operations (operators $\hat{L}_1, \hat{L}_2, \hat{L}_3$) in steps of δt :

   a) Calculate $\zeta_3 = \vec{p}_i\cdot\hat{P}_3\vec{q}_i / 4\mathrm{I}^{p^3},$
   b) Update $\breve{q}_i = \cos(\zeta_3\delta t/2)\breve{q}_i + \sin(\zeta_3\delta t/2)\hat{P}_3\breve{q}_i,$
   c) Update $\breve{p}_i = \cos(\zeta_3\delta t/2)\breve{p}_i + \sin(\zeta_3\delta t/2)\hat{P}_3\breve{p}_i,$
   d) Calculate $\zeta_2 = \vec{p}_i\cdot\hat{P}_2\vec{q}_i / 4\mathrm{I}^{p^2},$

e) Update $\breve{q}_i = \cos(\zeta_2 \delta t/2)\breve{q}_i + \sin(\zeta_2 \delta t/2)\hat{P}_2 \breve{q}_i$,
f) Update $\breve{p}_i = \cos(\zeta_2 \delta t/2)\breve{p}_i + \sin(\zeta_2 \delta t/2)\hat{P}_2 \breve{p}_i$,
g) Calculate $\zeta_1 = \vec{p}_i \cdot \hat{P}_1 \vec{q}_i / 4\mathrm{I}^{p^1}$,
h) Update $\breve{q}_i = \cos(\zeta_1 \delta t)\breve{q}_i + \sin(\zeta_1 \delta t)\hat{P}_1 \breve{q}_i$,
i) Update $\breve{p}_i = \cos(\zeta_1 \delta t)\breve{p}_i + \sin(\zeta_1 \delta t)\hat{P}_1 \breve{p}_i$,
j) Calculate $\zeta_2 = \vec{p}_i \cdot \hat{P}_2 \vec{q}_i / 4\mathrm{I}^{p^2}$,
k) Update $\breve{q}_i = \cos(\zeta_2 \delta t/2)\breve{q}_i + \sin(\zeta_2 \delta t/2)\hat{P}_2 \breve{q}_i$,
l) Update $\breve{p}_i = \cos(\zeta_2 \delta t/2)\breve{p}_i + \sin(\zeta_2 \delta t/2)\hat{P}_2 \breve{p}_i$,
m) Calculate $\zeta_3 = \vec{p}_i \cdot \hat{P}_3 \vec{q}_i / 4\mathrm{I}^{p^3}$,
n) Update $\breve{q}_i = \cos(\zeta_3 \delta t/2)\breve{q}_i + \sin(\zeta_3 \delta t/2)\hat{P}_3 \breve{q}_i$,
o) Update $\breve{p}_i = \cos(\zeta_3 \delta t/2)\breve{p}_i + \sin(\zeta_3 \delta t/2)\hat{P}_3 \breve{p}_i$,
p) Repeat steps a) to o) m_{rot} times.

The final output from these operations are to be regarded as $\breve{q}_i^{n+1}$, which are complete, and $\breve{p}_i^{n+1/2}$, which require further integration.

4. Use $\breve{q}_i^{n+1}$ to construct $\boldsymbol{R}_i^{n+1}$ and re-orientate the molecules for step $(n+1)$ using equation (2.64). Then calculate $\vec{\tau}_i^{p^{n+1}}$.

5. Update $\breve{p}_i^{n+1/2}$ to step $(n+1)$:

$$\breve{p}_i^{n+1} = \breve{p}_i^{n+1/2} + \frac{\Delta t}{2}\breve{\tau}_i^{n+1}$$

Step 5 completes the algorithm. In practice this algorithm proves to be both stable and accurate, as expected for a symplectic algorithm.

4.6.8 Direct Integration of the Rotational Matrix

In section 4.6.1 we described the direct integration of the rotational equations of motions for a molecule that was composed of a rigid framework of atoms. Here we describe a variant of that algorithm that is also suitable for molecules that are based on the Gay-Berne [26] and Gaussian Density [27] models[16], which often have ellipsoidal shapes. Such molecules are characterised by the total mass, M, and the moment of inertia tensor, $\boldsymbol{I}$, which has components that vary with time when described from the laboratory reference frame. The molecule's centre of mass has a position $\vec{R}$ in space and a translational velocity $\vec{V}$. The molecule also rotates about its centre of mass with an angular momentum $\vec{J}$ and angular velocity $\vec{\omega}$. The equations of motion for such a body (and the important variables) were given in section 4.6 , (equations (4.80) to (4.89)). Here we focus on the computational algorithm.

In this algorithm is described in the laboratory frame of reference. The laboratory frame is defined by mutually orthogonal unit vectors $\vec{e}_1, \vec{e}_2, \vec{e}_3$, fixed in space. A general vector, $\vec{v}$, with components (v_1, v_2, v_3) is therefore written as

16 See sections 5.4.3 and 5.4.4 for more information on these models.

$$\vec{v}=v_1\vec{e}_1+v_2\vec{e}_2+v_3\vec{e}_3. \tag{4.129}$$

The local frame of a rotating body is also based on a set mutually orthogonal unit vectors, $\vec{e}_1', \vec{e}_2', \vec{e}_3',$ and is chosen so that the moment of inertia tensor in this frame is a diagonal matrix.

In the local frame the vector, $\vec{v},$ has components (v_1', v_2', v_3') and can be written as

$$\vec{v}=v_1'\vec{e}_1'+v_2'\vec{e}_2'+v_3'\vec{e}_3'. \tag{4.130}$$

This is the same vector as that given in (4.129), but expressed in a different reference frame. The relationship between the two representations is defined by the rotation matrix, $\boldsymbol{R},$ such that

$$\text{(a)}\quad \vec{e}_i=\sum_{j=1}^{3} R_{ij}\vec{e}_j' \quad \text{and} \quad \text{(b)}\quad \vec{e}_i'=\sum_{j=1}^{3} R_{ji}\vec{e}_j \tag{4.131}$$

and

$$\text{(a)}\quad v_i=\sum_{j=1}^{3} R_{ij}v_j' \quad \text{and} \quad \text{(b)}\quad v_i'=\sum_{j=1}^{3} R_{ji}v_j. \tag{4.132}$$

It is useful to note that, since

$$\vec{e}_1=\begin{bmatrix}1\\0\\0\end{bmatrix}, \quad \vec{e}_2=\begin{bmatrix}0\\1\\0\end{bmatrix}, \quad \text{and} \quad \vec{e}_3=\begin{bmatrix}0\\0\\1\end{bmatrix}, \tag{4.133}$$

we can write directly from equation of (4.131) (b)

$$\vec{e}_1'=\begin{bmatrix}R_{11}\\R_{21}\\R_{31}\end{bmatrix}, \quad \vec{e}_2'=\begin{bmatrix}R_{12}\\R_{22}\\R_{32}\end{bmatrix}, \quad \text{and} \quad \vec{e}_3'=\begin{bmatrix}R_{13}\\R_{23}\\R_{33}\end{bmatrix}. \tag{4.134}$$

So the rotation matrix may be written in the abbreviated form

$$\boldsymbol{R}=\left[\vec{e}_1', \vec{e}_2', \vec{e}_3'\right], \tag{4.135}$$

in which the vectors form columns in the matrix. The relationship (4.135) is central to

the algorithm. In essence the algorithm integrates the motion of the vectors $\vec{e}_1\,',\ \vec{e}_2\,',\ \vec{e}_3\,',$ which from (4.135) is equivalent to updating the rotation matrix. This is what we mean by the direct integration of the rotation matrix.

The mass M and the diagonal moment of inertia tensor $\boldsymbol{I}^p$ (the *principal* tensor) are known constants. At the start of each time step it is assumed that $\vec{R}^n,\ \vec{V}^n,\ \vec{F}^n,\ \boldsymbol{R}^n,\ \vec{J}^n$ and $\vec{\tau}^n$ are known. The matrix $\boldsymbol{R}$ (as defined by its use in equations (4.131) and (4.132)) can be used to construct the instantaneous moment of inertia tensor, $\boldsymbol{I}$, in the laboratory frame:

$$\boldsymbol{I} = \boldsymbol{R}\,\boldsymbol{I}^p\,\tilde{\boldsymbol{R}}. \tag{4.136}$$

The angular velocity at any time is given by

$$\vec{\omega} = \boldsymbol{I}^{-1}\vec{J}. \tag{4.137}$$

The inverse matrix $\boldsymbol{I}^{-1}$ is easily obtained from

$$\boldsymbol{I}^{-1} = \boldsymbol{R}\left(\boldsymbol{I}^p\right)^{-1}\tilde{\boldsymbol{R}}. \tag{4.138}$$

The algorithm presented in the following scheme is couched in the velocity Verlet form for both translational and rotational motion.

1. Start step n with $\vec{R}^n,\ \vec{V}^n,\ \vec{F}^n,\ \vec{J}^n,\ \vec{\tau}^n,\ \boldsymbol{R}^n \equiv \{\vec{e}_i^{\,n}\,'\}.$
2. Calculate: $\boldsymbol{I}^n = \boldsymbol{R}^n\boldsymbol{I}^p\,\tilde{\boldsymbol{R}}^n$ and $\left(\boldsymbol{I}^n\right)^{-1} = \boldsymbol{R}^n\left(\boldsymbol{I}^p\right)^{-1}\tilde{\boldsymbol{R}}^n.$
3. Calculate: $\vec{\omega}^n = \left(\boldsymbol{I}^n\right)^{-1}\vec{J}^n.$
4. Calculate: $\dot{\boldsymbol{I}}^n,$ using $\boldsymbol{I}^n,\ \vec{\omega}^n.$ (Equation (4.87))
5. Update: $\vec{V}^{n+1/2} \leftarrow \vec{V}^n + \dfrac{\Delta t}{2M}\vec{F}^n.$
6. Update: $\vec{J}^{n+1/2} \leftarrow \vec{J}^n + \dfrac{\Delta t}{2}\vec{\tau}^n.$
7. Update: $\vec{\omega}^{n+1/2} \leftarrow \vec{\omega}^n + \dfrac{\Delta t}{2}\left(\boldsymbol{I}^n\right)^{-1}\left(\vec{\tau}^n - \dot{\boldsymbol{I}}^n\vec{\omega}^n\right).$
8. Update: $\vec{R}^{n+1} \leftarrow \vec{R}^n + \Delta t\,\vec{V}^{n+1/2}.$
9. Update: $\vec{e}_i^{\,n+1}\,' \leftarrow \vec{e}_i^{\,n}\,' + \Delta t\,\vec{\omega}^{n+1/2} \times \vec{e}_i^{\,n}\,',\quad for\ all\ \{\vec{e}_i^{\,n}\,'.\}$ (See comments below!)
10. Calculate: $\boldsymbol{R}^{n+1} = \left[\vec{e}_1^{\,n+1}\,',\ \vec{e}_2^{\,n+1}\,',\ \vec{e}_3^{\,n+1}\,'\right].$
11. Calculate $\vec{F}^{n+1}$ and $\vec{\tau}^{n+1}.$

12. Update: $\vec{V}^{n+1} \leftarrow \vec{V}^{n+1/2} + \frac{\Delta t}{2M}\vec{F}^{n+1}$.

13. Update: $\vec{J}^{n+1} \leftarrow \vec{J}^{n+1/2} + \frac{\Delta t}{2}\vec{\tau}^{n+1}$.

As was described in section 4.6.1 , step 9 in this algorithm requires a more sophisticated approach than the direct integration shown, but it can be handled in precisely the same way as described in section 4.6.1.2 . We shall not reiterate the approach here, but simply note that the vectors being rotated are the set $\{\vec{e}_1', \vec{e}_2', \vec{e}_3'\}$ defining the orientation of the local molecular axes in the laboratory frame and that step 9 must preserve their unit magnitude. The scheme presented in section 4.6.1.2 achieves precisely this.

The scheme above works just as well with rigid non-linear molecules composed of point atoms in a rigid framework. However, it appears to present a problem for linear molecules. Steps 3 and 7 of the scheme require the inverse of matrix $\boldsymbol{I}$, but if one of the principal moments of inertia of the molecule is zero, as it is for linear molecules, then formally the inverse $\boldsymbol{I}^{-1}$ does not exist. Fortunately it is possible to patch the algorithm with a simple change to accommodate linear molecules, as explained below.

We start by considering the case of a uni-axial ellipsoid of the Gay-Berne or Gaussian kind. These are characterised by having two equal moments of inertia, $I^p_{xx} = I^p_{yy}$, (which follows from molecular symmetry) and one unique moment I^p_{zz}. All three moments are non-zero in this case. Next we write the angular equation of motion (4.86) in the form first shown in equation (2.51)

$$\vec{\tau} = \boldsymbol{I}\dot{\vec{\omega}} + \dot{\boldsymbol{I}}\vec{\omega}, \tag{4.139}$$

where, in the laboratory frame, $\boldsymbol{I}$ is not generally diagonal. We now multiply (4.139) by the rotation matrix $\boldsymbol{R}$, which is able to diagonalise $\boldsymbol{I}$ at a fixed point in time, and obtain

$$\tilde{\boldsymbol{R}}\vec{\tau} = \tilde{\boldsymbol{R}}\boldsymbol{I}\boldsymbol{R}\tilde{\boldsymbol{R}}\dot{\vec{\omega}} + \tilde{\boldsymbol{R}}\dot{\boldsymbol{I}}\boldsymbol{R}\tilde{\boldsymbol{R}}\vec{\omega}, \tag{4.140}$$

where we have exploited the relation $\boldsymbol{R}\tilde{\boldsymbol{R}} = \boldsymbol{1}$, where $\boldsymbol{1}$ is the identity matrix. This equation can now be written as

$$\vec{\tau}^p = \boldsymbol{I}^p\dot{\vec{\omega}}^p + \tilde{\boldsymbol{R}}\dot{\boldsymbol{I}}\boldsymbol{R}\vec{\omega}^p, \tag{4.141}$$

where the superscript p indicates the principal frame of reference. (Equation (4.141), as was shown in section 2.9 , is another form of Euler's rotational equations of motion.) Thus we see that equation (4.139) and (4.141) are physically equivalent and differ only in the choice of reference frame. Equation (4.139) is couched in the laboratory frame, while (4.141) is in the principal frame, which is a stationary frame in which the moment of inertia tensor is instantaneously diagonal. Euler's equations thus only hold at one particular instant, though we can always find a different principal

frame for any other instant.

Because $I^p_{xx} = I^p_{yy}$, the third of Euler's equations (2.72) can be written as

$$\tau^p_z = I^p_{zz} \dot{\omega}^p_z . \tag{4.142}$$

This equation is solely responsible for rotation around the (principal) z-axis and is independent of the mechanics of rotation about the x- and y-axes. Furthermore, since an ellipsoidal molecule is rotationally symmetric about the z-axis, the torque component τ^p_z derived from interaction with other molecules must be zero. It follows from (4.142) that $\dot{\omega}^p_z$ must also be zero.

Thus we cannot expect ω^p_z ever to change, since there is nothing to drive such change in *any* principal frame of reference. For this reason we may set $\omega^p_z = 0$ once and for all time, since it is dynamically redundant. Setting $\omega^p_z = 0$ (and ignoring for a moment that $\tau^p_z = \dot{\omega}^p_z = 0$) means Euler's equations (2.72) are now reduced to

$$\begin{aligned} \tau^p_x &= I^p_{xx} \dot{\omega}^p_x , \\ \tau^p_y &= I^p_{yy} \dot{\omega}^p_y , \\ \tau^p_z &= I^p_{zz} \dot{\omega}^p_z . \end{aligned} \tag{4.143}$$

The third equation of (4.143) is of course redundant since the dynamics of the molecule are governed *entirely* by the first two equations, which are not in any way coupled to the third. Nevertheless the algorithm presented above is entirely suitable for simulating this system, provided the simulation starts (and thus continues) with the condition $\omega^p_z = 0.$ It works in this case because, despite the redundancy in the set of equations (4.143), the inverse $\boldsymbol{I}^{-1}$ of the moment of inertia tensor $\boldsymbol{I}$ can always be obtained.

At this point we note that the first two equations of (4.143) are the same as those for a linear molecule consisting of point particles. In this case the third equation does not exist, but as we have noted, it is also redundant for ellipsoidal molecules. It follows that all that is required to adapt this algorithm to linear molecules, is to set I^p_{zz} to a convenient finite value and ensure that the simulation starts with $\omega^p = 0.$ The two systems, linear and ellipsoidal molecules, are dynamically isomorphic under these circumstances.

In practice it is best to initialise a simulation with every molecule defined in its principal frame so that the occurrence of a zero moment of inertia can be identified and reset to a finite value, while the corresponding component of the angular momentum is set to zero. The entire system can then be recast back to the laboratory frame. Note that subsequent values of the moment of inertia tensor $\boldsymbol{I}$ and its inverse are calculated from $\boldsymbol{I}^p$ using equations (4.136) and (4.138). It is not necessary to "carry" current values of $\boldsymbol{I}$ or $\boldsymbol{I}^{-1}$ throughout the simulation.

Some precautions are necessary. It should be noted that linearity implies the molecule has lost one rotational degree of freedom and this must be remembered when converting the kinetic energy into temperature. (This also applies to uni-axial ellipsoids of the Gay-Berne or Gaussian kind.) Also, some caution is necessary when deciding the number of degrees of freedom of a molecule based on inspecting the principal moments of inertia alone. It may be accidentally true (or nearly true) that two or more moments are equal, without rendering a degree of freedom redundant.

4.7 Algorithms for Flexible Molecules with Rigid Groups of Atoms

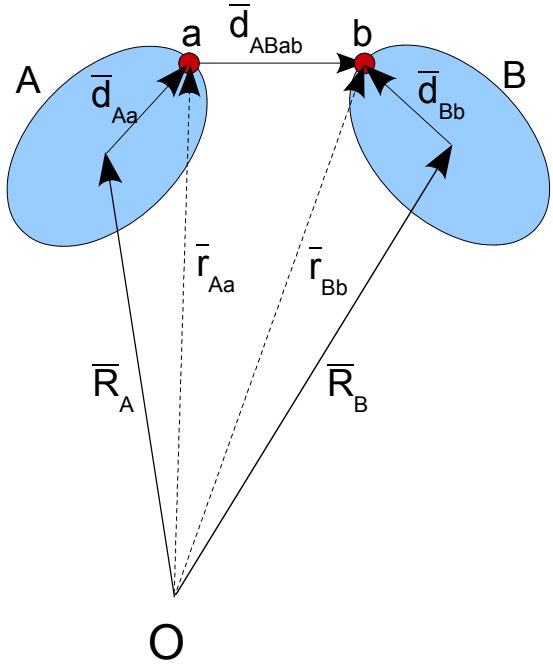

Figure 4.4. The Geometry of the QSHAKE Algorithm

Molecules that can be described as dynamically rigid bodies are many, but there also exists a large number of molecules that are best described as linked rigid bodies. Many of these are polymers of some kind, either synthetic or natural, which have substituent groups of atoms which strongly hold their structure while the rest of the molecule undergoes conformational change. Examples of such groups include aromatic rings, sulphate or phosphate groups, the internal dynamics of which are problematical or merely uninformative and are best treated as rigid.

If, in a given molecular model, the rigid groups are linked to the rest of the molecule by extensible bonds, then it immediately apparent that the forces arising from the flexible link can be regarded as external to the rigid group, which means the dynamics of the group can be handled by the algorithms given above for independent rigid molecules. Linked rigid body models with extensible bond links are viable provided that the bonds are not too stiff – an attribute that implies they would best be modelled by rigid bonds. Otherwise, there is no new difficulty in handling such

molecules beyond what has been described so far.

If, however, the bonds linking rigid groups to the molecule are to be regarded as inextensible, a new approach is required to combine the dynamics of the rigid group with the constraint dynamics of the rigid bonds. In such a model it should be understood that while the bond may be rigid, the bond angles associated with the bond generally are not. There is thus a fair degree of flexibility in the link, which includes rotation of the rigid group about the bond. The algorithm that handles such molecules is known as QSHAKE [28] which we will describe in both the leapfrog and velocity Verlet versions.

First some preliminaries, in figure 4.4 we show two rigid groups, A and B, connected via a bond vector $\vec{d}_{ABab}$, which links the atoms a and b belonging to groups A and B respectively. Vectors $\vec{R}_A$ and $\vec{R}_B$ locate the centres of mass of the two groups and vectors $\vec{d}_{Aa}$ and $\vec{d}_{Bb}$ locate the atoms a and b within their respective groups. Vectors $\vec{r}_{Aa}$ and $\vec{r}_{Bb}$ locate these atoms in space. Thus we have:

$$\begin{aligned}\vec{r}_{Aa} &= \vec{R}_A + \vec{d}_{Aa}, \\ \vec{r}_{Bb} &= \vec{R}_B + \vec{d}_{Bb},\end{aligned} \tag{4.144}$$

and it follows that

$$\vec{d}_{ABab} = \vec{r}_{Bb} - \vec{r}_{Aa}. \tag{4.145}$$

In any dynamical treatment of the molecular motion the distance $d_{ABab} = |\vec{d}_{ABab}|$ must remain constant and we must devise an algorithm that guarantees this while determining the translational and rotational dynamics of the rigid groups.

In the SHAKE algorithm described in section 4.5 , the constraint is maintained by the introduction of a constraint force $\vec{g}_{ab}$ which is applied along the bond vector $\vec{d}_{ABab}$ i.e. $\vec{g}_{ab} = g_{ab}\vec{d}_{ABab}$, where g_{ab} is an undetermined scalar quantity, which is obtained by an iterative procedure. We show how this is achieved in QSHAKE in the following sections.

4.7.1 QSHAKE in the Leapfrog Form

In the leapfrog algorithm, at a time step n, the positions of the rigid units and the linked atoms are given as the vectors $\vec{R}_A, \vec{R}_B, \vec{d}_{Aa}$, and $\vec{d}_{Bb}$. We also have the velocities of the rigid unit centres of mass at the half time step, $\vec{V}_A^{n-1/2}$ and $\vec{V}_B^{n-1/2}$. Depending on which approach we wish to take in solving the rotational motion, we may also need to know the moment of inertia tensors $\boldsymbol{I}_A^n$ and $\boldsymbol{I}_B^n$ (and perhaps their time derivatives) and the rotational matrices $\boldsymbol{R}_A^n$ and $\boldsymbol{R}_B^n$. Discounting the constraint

force, there are net forces[17] of $\vec{F}_A^n$ and $\vec{F}_B^n$ acting on groups A and B respectively. If the constraint force $\vec{g}_{ab}$ were known, the leapfrog integration of the translational motion of the centres of mass of the rigid groups would be

$$\begin{aligned}
\vec{V}_A^{n+1/2} &= \vec{V}_A^{n-1/2} + \frac{\Delta t}{M_A}(\vec{F}_A^n + \vec{g}_{ba}^n), \\
\vec{V}_B^{n+1/2} &= \vec{V}_B^{n-1/2} + \frac{\Delta t}{M_B}(\vec{F}_B^n + \vec{g}_{ab}^n), \\
\vec{R}_A^{n+1} &= \vec{R}_A^n + \Delta t\, \vec{V}_A^{n+1/2}, \\
\vec{R}_B^{n+1} &= \vec{R}_B^n + \Delta t\, \vec{V}_B^{n+1/2},
\end{aligned} \tag{4.146}$$

where M_A and M_B are the masses of the rigid groups. Following the procedure outlined in section 4.5.2 this set of equations can be reduced (*c.f.* Equation (4.62)) to

$$\begin{aligned}
\vec{R}_A^{n+1} &= \tilde{R}_A^{n+1} + \frac{\Delta t^2}{M_A}\vec{g}_{ba}^n, \\
\vec{R}_B^{n+1} &= \tilde{R}_B^{n+1} + \frac{\Delta t^2}{M_B}\vec{g}_{ab}^n,
\end{aligned} \tag{4.147}$$

Where the tilde (~) symbolises a vector obtained by integrating the equations of motion (4.146) in the absence of the constraint force $\vec{g}_{ab}$.

We now need to consider rotational motion. The constraint force also exerts a torque on rigid bodies A and B additional to the normal torques $\vec{\tau}_A^n$ and $\vec{\tau}_B^n$ arising from molecular forces. Incorporating this effect into the leapfrog scheme outlined in section 4.6.1.3 leads to

$$\begin{aligned}
\vec{\omega}_A^{n+1/2} &= \vec{\omega}_A^{n-1/2} + (\boldsymbol{I}_A^n)^{-1}(\vec{\tau}_A^n - \dot{\boldsymbol{I}}_A^n \vec{\omega}_A^n + \vec{d}_{Aa}^n \times \vec{g}_{ba}^n)\Delta t, \\
\vec{\omega}_B^{n+1/2} &= \vec{\omega}_B^{n-1/2} + (\boldsymbol{I}_B^n)^{-1}(\vec{\tau}_B^n - \dot{\boldsymbol{I}}_B^n \vec{\omega}_B^n + \vec{d}_{Bb}^n \times \vec{g}_{ab}^n)\Delta t, \\
\vec{d}_{Aa}^{n+1} &= \vec{d}_{Aa}^n + (\vec{\omega}_A^{n+1/2} \times \vec{d}_{Aa}^n)\Delta t, \\
\vec{d}_{Bb}^{n+1} &= \vec{d}_{Bb}^n + (\vec{\omega}_B^{n+1/2} \times \vec{d}_{Bb}^n)\Delta t,
\end{aligned} \tag{4.148}$$

where $\vec{\omega}_A$, $\vec{\omega}_B$ *etc.* are the angular velocities of the rigid groups. As with translational motion, these equations can be reduced to

$$\begin{aligned}
\vec{d}_{Aa}^{n+1} &= \tilde{d}_{Aa}^{n+1} + \Delta t^2 (\boldsymbol{I}_A^n)^{-1}(\vec{d}_{Aa}^n \times \vec{g}_{ba}^n) \times \vec{d}_{Aa}^n, \\
\vec{d}_{Bb}^{n+1} &= \tilde{d}_{Bb}^{n+1} + \Delta t^2 (\boldsymbol{I}_B^n)^{-1}(\vec{d}_{Bb}^n \times \vec{g}_{ab}^n) \times \vec{d}_{Bb}^n.
\end{aligned} \tag{4.149}$$

Once again the tilde (~) has been used to indicate vectors obtained from equations

17 These forces arise from the normal inter-atomic forces, summed to give a total force, though forces between atoms in the same rigid group are not normally included.

(4.148) in the absence of the constraint force.

Substituting the equations (4.147) and (4.149) into (4.144) gives

$$\begin{aligned}\vec{r}_{Aa}^{\,n+1} &= \tilde{R}_A^{n+1} + \tilde{d}_{Aa}^{\,n+1} - \Delta t^2\, g_{ab}^n \vec{\Theta}_A^n,\\ \vec{r}_{Bb}^{\,n+1} &= \tilde{R}_B^{n+1} + \tilde{d}_{Bb}^{\,n+1} + \Delta t^2\, g_{ab}^n \vec{\Theta}_B^n,\end{aligned} \tag{4.150}$$

where

$$\begin{aligned}\vec{\Theta}_A^n &= \frac{\vec{d}_{ab}^{\,n}}{M_A} + (\boldsymbol{I}_A^n)^{-1}(\vec{d}_{Aa}^{\,n} \times \vec{d}_{ab}^{\,n}) \times \vec{d}_{Aa}^{\,n},\\ \vec{\Theta}_B^n &= \frac{\vec{d}_{ab}^{\,n}}{M_B} + (\boldsymbol{I}_B^n)^{-1}(\vec{d}_{Bb}^{\,n} \times \vec{d}_{ab}^{\,n}) \times \vec{d}_{Bb}^{\,n},\end{aligned} \tag{4.151}$$

and we have used the identity

$$\vec{g}_{ab}^{\,n} = -\vec{g}_{ba}^{\,n} = g_{ab}^n \vec{d}_{ab}^{\,n}, \tag{4.152}$$

which ensures that the constraint force acts along the bond vector $\vec{d}_{ab}^{\,n}$.

From (4.145) and (4.150) it is now apparent that

$$\vec{d}_{ABab}^{\,n+1} = \tilde{d}_{ABab}^{\,n+1} + \Delta t^2 g_{ab}^n (\vec{\Theta}_B^n + \vec{\Theta}_A^n). \tag{4.153}$$

Squaring both sides of (4.153) and neglecting terms of order higher than Δt^2 gives after rearrangement

$$g_{ab}^n \approx \frac{|\vec{d}_{ABab}^{\,n+1}|^2 - |\tilde{d}_{ABab}^{\,n+1}|^2}{2\Delta t^2\, \tilde{d}_{ABab}^{\,n+1} \cdot (\vec{\Theta}_B^n + \vec{\Theta}_A^n)}. \tag{4.154}$$

In this equation $|\vec{d}_{ABab}^{\,n+1}|^2$ is required to be a constant. Equation (4.154) may be solved to obtain a first approximation to g_{ab}^n, which may be used in equations (4.146) and (4.148) to obtain better estimates of $\tilde{R}_A^{n+1}$, $\tilde{R}_B^{n+1}$, $\tilde{d}_{Aa}^{\,n+1}$ and $\tilde{d}_{Bb}^{\,n+1}$, which in turn may be used to obtain an additional *correction to* g_{ab}^n through equation (4.154) again. Repeated cycling using improved values for g_{ab}^n will obtain convergence. At this point the integration of the equations of motion over time step n are complete and the full constraint force (used also in the calculation of the virial) is the sum of all the correction contributions calculated.

It is worth noting that, in this scheme, it does not matter how the vectors $\tilde{d}_{Aa}^{\,n+1}$ and $\tilde{d}_{Bb}^{\,n+1}$ are obtained for equation (4.149). In this description a direct integration

scheme (section 4.6.1.3) has been used, but one employing Quaternions (such as those in 4.6.3 and 4.6.5) would be equally acceptable.

Provided the iterations are carries out to sufficient accuracy, the leapfrog QSHAKE is expected to have a similar accuracy to the leapfrog SHAKE algorithm, but like that algorithm it cannot be symplectic as there is no constraint placed on the velocities.

4.7.2 QSHAKE in the Velocity Verlet Form

At the start of the n'th time step the vectors $\vec{R}_A^n$, $\vec{R}_B^n$, $\vec{V}_A^n$ and $\vec{V}_B^n$, are known, as are $\vec{d}_{Aa}^n$, $\vec{d}_{Bb}^n$, $\vec{\omega}_A^n$ and $\vec{\omega}_B^n$. Once again, depending on our choice of rotational integration algorithm, we may also need to know the moment of inertia tensors $\boldsymbol{I}_A^n$ and $\boldsymbol{I}_B^n$ (and perhaps their time derivatives) and the rotational matrices $\boldsymbol{R}_A^n$ and $\boldsymbol{R}_B^n$. Also, acting on groups A and B respectively, are the net forces $\vec{F}_A^n$ and $\vec{F}_B^n$ (which do not include the constraint force $\vec{g}_{ab}$, which is unknown).

The first stage of velocity Verlet integration of the translational motion of the centres of mass of the rigid groups in the presence of the constraint force is

$$\begin{aligned}
\vec{V}_A^{n+1/2} &= \vec{V}_A^n + \frac{\Delta t}{2M_A}\left(\vec{F}_A^n + \vec{g}_{ba}^n\right), \\
\vec{V}_B^{n+1/2} &= \vec{V}_B^n + \frac{\Delta t}{2M_B}\left(\vec{F}_B^n + \vec{g}_{ab}^n\right), \\
\vec{R}_A^{n+1} &= \vec{R}_A^n + \Delta t\, \vec{V}_A^{n+1/2}, \\
\vec{R}_B^{n+1} &= \vec{R}_B^n + \Delta t\, \vec{V}_B^{n+1/2},
\end{aligned} \tag{4.155}$$

and the rotational motion in the first stage is given by

$$\begin{aligned}
\vec{\omega}_A^{n+1/2} &= \vec{\omega}_A^n + \left(\boldsymbol{I}_A^n\right)^{-1}\left(\vec{\tau}_A^n - \dot{\boldsymbol{I}}_A^n \vec{\omega}_A^n + \vec{d}_{Aa}^n \times \vec{g}_{ba}^n\right)\Delta t/2, \\
\vec{\omega}_B^{n+1/2} &= \vec{\omega}_B^n + \left(\boldsymbol{I}_B^n\right)^{-1}\left(\vec{\tau}_B^n - \dot{\boldsymbol{I}}_B^n \vec{\omega}_B^n + \vec{d}_{Bb}^n \times \vec{g}_{ab}^n\right)\Delta t/2, \\
\vec{d}_{Aa}^{n+1} &= \vec{d}_{Aa}^n + \left(\vec{\omega}_A^{n+1/2} \times \vec{d}_{Aa}^n\right)\Delta t, \\
\vec{d}_{Bb}^{n+1} &= \vec{d}_{Bb}^n + \left(\vec{\omega}_B^{n+1/2} \times \vec{d}_{Bb}^n\right)\Delta t.
\end{aligned} \tag{4.156}$$

These sets of equations may be reduced by separating out the contributions of terms involving the constraint force to give

$$\begin{aligned}
\vec{R}_A^{n+1} &= \tilde{R}_A^{n+1} + \frac{\Delta t^2}{2M_A}\vec{g}_{ba}^n, \\
\vec{R}_B^{n+1} &= \tilde{R}_B^{n+1} + \frac{\Delta t^2}{2M_B}\vec{g}_{ab}^n,
\end{aligned} \tag{4.157}$$

and

$$\begin{aligned}\vec{d}_{Aa}^{\,n+1}&=\tilde{d}_{Aa}^{\,n+1}+\Delta t^2(\boldsymbol{I}_A^n)^{-1}(\vec{d}_{Aa}^{\,n}\times\vec{g}_{ba}^{\,n})\times\vec{d}_{Aa}^{\,n}/2,\\ \vec{d}_{Bb}^{\,n+1}&=\tilde{d}_{Bb}^{\,n+1}+\Delta t^2(\boldsymbol{I}_B^n)^{-1}(\vec{d}_{Bb}^{\,n}\times\vec{g}_{ab}^{\,n})\times\vec{d}_{Bb}^{\,n}/2.\end{aligned} \tag{4.158}$$

In equations (4.157) and (4.158) the tilde (~) indicates the vectors obtained by integrating the equations (4.155) and (4.156) in the absence of the constraint force $\vec{g}_{ab}^{\,n}$.

Substituting the equations (4.157) and (4.158) into (4.144) gives

$$\begin{aligned}\vec{r}_{Aa}^{\,n+1}&=\tilde{R}_A^{n+1}+\tilde{d}_{Aa}^{\,n+1}-\Delta t^2\, g_{ab}^n\vec{\Theta}_A^n/2,\\ \vec{r}_{Bb}^{\,n+1}&=\tilde{R}_B^{n+1}+\tilde{d}_{Bb}^{\,n+1}+\Delta t^2\, g_{ab}^n\vec{\Theta}_B^n/2,\end{aligned} \tag{4.159}$$

where

$$\begin{aligned}\vec{\Theta}_A^n&=\frac{\vec{d}_{ab}^{\,n}}{M_A}+(\boldsymbol{I}_A^n)^{-1}(\vec{d}_{Aa}^{\,n}\times\vec{d}_{ab}^{\,n})\times\vec{d}_{Aa}^{\,n},\\ \vec{\Theta}_B^n&=\frac{\vec{d}_{ab}^{\,n}}{M_B}+(\boldsymbol{I}_B^n)^{-1}(\vec{d}_{Bb}^{\,n}\times\vec{d}_{ab}^{\,n})\times\vec{d}_{Bb}^{\,n},\end{aligned} \tag{4.160}$$

and we have used the identity

$$\vec{g}_{ab}^{\,n}=-\vec{g}_{ba}^{\,n}=g_{ab}^n\vec{d}_{ab}^{\,n}, \tag{4.161}$$

which ensures that the constraint force acts along the bond vector $\vec{d}_{ab}^{\,n}$. Taking the difference of the two equations (4.159) we can now write

$$\vec{d}_{ABab}^{\,n+1}=\tilde{d}_{ABab}^{\,n+1}+\Delta t^2\, g_{ab}^n(\vec{\Theta}_B^n+\vec{\Theta}_A^n)/2. \tag{4.162}$$

Squaring this equation and neglecting terms of order higher than Δt^2 leads to

$$g_{ab}^n\approx\frac{|\vec{d}_{ABab}^{\,n+1}|^2-|\tilde{d}_{ABab}^{\,n+1}|^2}{\Delta t^2\,\tilde{d}_{ABab}^{\,n+1}\cdot(\vec{\Theta}_B^n+\vec{\Theta}_A^n)}. \tag{4.163}$$

(It should be noted that this equation differs from the QSHAKE leapfrog form (4.154) by a factor of two.) Here again, as with the leapfrog variant, the value of g_{ab}^n obtained may be substituted back into the starting equations (4.155) and the process repeated. Solving (4.163) again gives a further correction to the constraint force, which may be recycled yet again. Repeated iteration will converge the calculation of

g_{ab}^{n} and provide the full-step positions and half-step velocities to complete the first stage of the QSHAKE velocity Verlet scheme.

The second stage of velocity Verlet is to advance the velocities from the half to the full time step. Once again a constraint force, $\vec{h}_{ab}$, must be applied along the bond, $\vec{d}_{ABab}$, this time to ensure linked atoms a and b, on rigid groups A and B respectively, have no velocity relative to each other along the rigid bond direction. In the presence of this constraint force it can be shown that the velocity update equations are

$$\begin{aligned}
\vec{V}_A^{n+1} &= \tilde{V}_A^{n+1} + \frac{\Delta t}{2M_A}\vec{h}_{ba}^{n+1}, \\
\vec{V}_B^{n+1} &= \tilde{V}_B^{n+1} + \frac{\Delta t}{2M_B}\vec{h}_{ab}^{n+1}, \\
\vec{\omega}_A^{n+1} &= \tilde{\omega}_A^{n+1} + \frac{\Delta t}{2}\left(\boldsymbol{I}_A^{n+1}\right)^{-1}\vec{d}_{Aa}^{n+1}\times\vec{h}_{ba}^{n}, \\
\vec{\omega}_B^{n+1} &= \tilde{\omega}_B^{n+1} + \frac{\Delta t}{2}\left(\boldsymbol{I}_B^{n+1}\right)^{-1}\vec{d}_{Bb}^{n+1}\times\vec{h}_{ab}^{n},
\end{aligned} \tag{4.164}$$

where the tilde (~) indicates the velocities obtained in the absence of the constraint force $\vec{h}_{ab}$.

The velocities of the linked atoms a and b are

$$\begin{aligned}
\vec{v}_{Aa}^{n+1} &= \vec{V}_A^{n+1} + \vec{\omega}_A^{n+1}\times\vec{d}_{Aa}^{n+1}, \\
\vec{v}_{Bb}^{n+1} &= \vec{V}_B^{n+1} + \vec{\omega}_B^{n+1}\times\vec{d}_{Bb}^{n+1}.
\end{aligned} \tag{4.165}$$

Substitution of equations (4.164) into (4.165) above leads to

$$\begin{aligned}
\vec{v}_{Aa}^{n+1} &= \tilde{v}_{Aa}^{n+1} - \frac{\Delta t}{2}h_{ab}^{n+1}\vec{\Xi}_A^{n+1}, \\
\vec{v}_{Bb}^{n+1} &= \tilde{v}_{Bb}^{n+1} + \frac{\Delta t}{2}h_{ab}^{n+1}\vec{\Xi}_B^{n+1},
\end{aligned} \tag{4.166}$$

where

$$\begin{aligned}
\vec{\Xi}_A^{n+1} &= \frac{\vec{d}_{ab}^{n+1}}{M_A} + \left(\boldsymbol{I}_A^{n+1}\right)^{-1}\left(\vec{d}_{Aa}^{n+1}\times\vec{d}_{ab}^{n+1}\right)\times\vec{d}_{Aa}^{n+1}, \\
\vec{\Xi}_B^{n+1} &= \frac{\vec{d}_{ab}^{n+1}}{M_B} + \left(\boldsymbol{I}_B^{n+1}\right)^{-1}\left(\vec{d}_{Bb}^{n+1}\times\vec{d}_{ab}^{n+1}\right)\times\vec{d}_{Bb}^{n+1},
\end{aligned} \tag{4.167}$$

and we have used the identity

$$\vec{h}_{ab}^{n+1} = -\vec{h}_{ba}^{n+1} = h_{ab}^{n+1}\vec{d}_{ab}^{n+1}. \tag{4.168}$$

The constraint condition requires that

$$\vec{d}_{ab}^{n+1} \cdot (\vec{v}_{Bb}^{n+1} - \vec{v}_{Aa}^{n+1}) = 0. \tag{4.169}$$

Substituting (4.166) and (4.167) into this leads to

$$h_{ab}^{n+1} = -\frac{2\vec{d}_{ab}^{n+1} \cdot (\vec{v}_{Bb}^{n+1} - \vec{v}_{Aa}^{n+1})}{\Delta t \vec{d}_{ab}^{n+1} \cdot (\vec{\Xi}_{B}^{n+1} + \vec{\Xi}_{A}^{n+1})} \tag{4.170}$$

which is an equation which again may be solved iteratively, cycling further corrections to $\vec{h}_{ab}^{n+1}$ back through equations (4.164) onwards until convergence and the final values of the full-step velocities. The full constraint force (used also in the calculation of the virial) is the sum of all the correction contributions calculated. This completes the QSHAKE velocity Verlet scheme for one integration time step.

As with the QSHAKE leapfrog scheme, it does not matter how the rotational motion is handled. The direct integration scheme (section 4.6.1.3) or the Quaternion schemes given in sections 4.6.6 and 4.6.7 are equally acceptable. Also the scheme should be as accurate as the original velocity Verlet provided the iterations are carried out to sufficient accuracy. Unlike the QSHAKE leapfrog algorithm however it can be considered symplectic (if iterated to sufficient accuracy) as there is an explicit constraint placed on the velocities.

4.8 Algorithms for Various Thermodynamic Ensembles

All the algorithms described so far generate the thermodynamic ensemble that is frequently referred to as the *microcanonical* or NVE ensemble (section 3.18.1), which conserves particle number (N), system volume (V) and energy (E). Such a system exists in isolation and does not exchange particles or energy with the outside world. This is suitable for many practical simulations but it is thermodynamically restrictive. It is difficult, for example, to generate systems with a required volume (or equivalently density), pressure and temperature except through *ad hoc* procedures, which lack validity. For most laboratory experiments, these are the variables that define the state of the system and it is inconvenient not to be able to set them at the outset of a simulation study.

More commonly used experimentally is the *canonical* or NVT ensemble, which conserves N, V and temperature (T) rather than energy. The temperature is maintained through the exchange of energy (in the form of heat) with the environment surrounding the system. The external environment is thus a heat-bath that supplies or receives heat energy from the simulated system. In the equilibrated

system the energy E is not an absolute constant (as in NVE), but fluctuates about a mean value like any other thermodynamic variable. Thus the canonical ensemble is appropriate for experiments that are conduced at a constant volume with temperature determined by an external heat-bath. Also commonly used is the isothermal-isobaric or NPT ensemble, which has the canonical ensemble's ability to exchange heat with its surrounds, but also allows density and volume to change in response to the pressure. The ensemble is appropriate for experiments where the system temperature and pressure are determined by an external heat bath and an external pressure – circumstances appropriate to many chemical and biological experiments.

In molecular dynamics simulations of the NVE ensemble, the particle number and volume are rigorously fixed from the outset and the energy becomes fixed once the simulation leaves the equilibration period and *ad hoc* scaling of the system temperature ceases. The energy is henceforth fixed because the Hamiltonian of the system is conserved. In the canonical ensemble it is temperature rather than energy that we wish to control, while in the isothermal-isobaric ensemble we wish to control temperature and pressure rather than energy and volume. The simulations have to be constructed to meet these requirements.

There are many approaches that have been taken to generate various ensembles in molecular dynamics and this has led to a large number of alternative methods appearing in the literature. Some are simply *ad hoc* and involve adjusting the system temperature or volume in a rather direct manor to push the system towards the required temperature or pressure. These are unsatisfactory on a number of levels. Not only are the natural dynamics of the system perturbed but the fluctuations in the thermodynamic variables are no longer properly represented by a Boltzmann distribution, which undermines the theoretical basis of the statistical thermodynamics on which molecular dynamics rests. Examples of these approaches are the Berendsen thermostat and barostat [29]. This deficiency does not mean these methods are not widely used, nor that they generate unusable results.

Other approaches have involved the careful amendment of the equations of motion of the system to incorporate the effects of a thermostat (to control temperature) or barostat (to control pressure) and then proving that the modified equations properly generate the required ensemble. These intuitive approaches are sometimes incomplete and may not reproduce all the expected aspects of the ensemble. For example, the generated configurations of the atoms may be appropriate for the required ensemble but the distribution of momenta may not, as is the case for the Evans thermostat [12]. Nevertheless these methods usually have a strong theoretical foundation and for this reason are to be preferred over the *ad hoc* approaches.

Arguably the most complete approach to simulating ensembles is to start with an extended Hamiltonian. The extension in this case is to include a thermostat or barostat which provides extra degrees of freedom through which temperature or pressure are coupled to the atomic positions and momenta of the system and function as an external agent (like a heat bath), adjusting these properties as the system evolves. The constructed Hamiltonian can be used to derive suitable equations of motion and also provide a direct route to the system thermodynamics. Often the equations of motion are the same as those obtained by less formal approaches, but it is not always the case that intuitively amended equations can be shown to derive from

an associated Hamiltonian. In which case the full power of the Hamiltonian approach is lost, in particular the opportunity to generate symplectic algorithms using the methods outlined in section 4.3 .

In the following sections we describe some of the more commonly used ensemble methods.

4.8.1 The Berendsen Thermostat and Barostat

The Berendsen thermostat and barostat [29] are *ad hoc* methods which can be easily incorporated into any of the forms of the Verlet algorithm. Their main advantages are their conceptual simplicity (which is the major part of their appeal) and their computational robustness. Being ad hoc however, they are not able to generate proper thermodynamic ensembles, nor can they be part of a symplectic algorithm. Also, when using the thermostat, it is necessary to be aware of a destructive (but avoidable) phenomenon known as the "flying ice cube" [30] which we shall describe below.

The Berendsen thermostat $\chi(t)$ is a function that steers the system temperature T incrementally towards the required temperature T_o and is defined by the formula

$$\chi(t)=\left[1-\frac{\Delta t}{\tau_T}\left(1-\frac{T_o}{T(t)}\right)\right]^{1/2} \tag{4.171}$$

where Δt is the integration time step, $T(t)$ is the instantaneous temperature and τ_T is a user defined parameter regarded as a *relaxation time* controlling the rate at which the system is steered to the required temperature. Larger values of τ_T mean that the thermostat makes smaller adjustments to the temperature and this way the user gains fine control of the thermostatting process. In the leapfrog algorithm, the thermostat is applied as a velocity scaling factor as follows

$$\begin{aligned}
&\tilde{v}_i^{n+1/2}=\vec{v}_i^{n-1/2}+\frac{\Delta t}{m_i}\vec{f}_i^{\,n},\\
&\tilde{v}_i^{n}=(\tilde{v}_i^{n+1/2}+\vec{v}_i^{n-1/2})/2,\\
&\chi^n=[1-\Delta t(1-T_o/T^n)/\tau_T]^{1/2},\\
&\vec{v}_i^{n+1/2}=\chi^n\tilde{v}_i^{n+1/2},\\
&\vec{r}_i^{n+1}=\vec{r}_i^{n}+\Delta t\,\vec{v}_i^{n+1/2}.
\end{aligned} \tag{4.172}$$

The instantaneous temperature T^n needed to calculate χ^n (both at time step *n*) is obtained from the velocities $\tilde{v}_i^n$. (Note the use of the tilde (~) to indicate vectors that do not have their final values.) If the system contains rigid molecules, we must include the rotational motion in the calculation of T^n and then χ^n may be used to scale the angular velocities also.

The corresponding velocity Verlet form is as follows

$$
\begin{aligned}
\vec{v}_i^{\,n+1/2} &= \vec{v}_i^{\,n} + \frac{\Delta t}{2m_i}\vec{f}_i^{\,n}, \\
\vec{r}_i^{\,n+1} &= \vec{r}_i^{\,n} + \Delta t\,\vec{v}_i^{\,n+1/2}, \\
\tilde{v}_i^{\,n+1} &= \vec{v}_i^{\,n+1/2} + \frac{\Delta t}{2m_i}\vec{f}_i^{\,n+1}, \\
\chi^{n+1} &= \left[1-\Delta t\left(1-T_o/T^{n+1}\right)/\tau_T\right]^{1/2}, \\
\vec{v}_i^{\,n+1} &= \chi^n \tilde{v}_i^{\,n+1}.
\end{aligned}
\tag{4.173}
$$

The intermediate velocity $\tilde{v}_i^{\,n+1}$ is required to calculate the temperature T^{n+1} which is needed to obtain χ^{n+1}. Once again, if rotational motion of rigid bodies occurs in the system, the rotational kinetic energy must be factored into the calculation of T^{n+1} and the resulting χ^{n+1} used to scale the angular velocity.

The "flying ice cube" syndrome mentioned above occurs because the Berendsen thermostat has a tendency to dampen high frequency motion more than low frequency. This, coupled with the fact that the centre of mass motion of the system, is only nominally zero within numerical accuracy, means that over the course of a long simulation, the kinetic energy is extracted from the high frequency modes and dumped into the centre of mass motion. The system thus freezes as it acquires a large translational velocity. This can be avoided if the centre of mass motion is periodically re-zeroed and the atomic velocities re-assigned from a Gaussian distribution. This fix is acceptable if static properties of the system are being calculated, but is not safe if time dependent properties are required.

The Berendsen barostat $\eta(t)$ takes the form

$$
\eta(t) = \left[1-\beta\frac{\Delta t}{\tau_P}\left(P_o - P(t)\right)\right]^{1/3}, \tag{4.174}
$$

in which τ_P is a user defined relaxation time, P_o is the required (target) pressure and $P(t)$ the instantaneous pressure. The quantity β is the system isothermal compressibility which, if unknown, may be replaced by the isothermal compressibility of water as it is not thermodynamically significant in this context. The leapfrog integrator for this barostat, including the thermostat $\chi(t)$ necessary for the NPT system is as follows.

$$
\begin{aligned}
\tilde{v}_i^{\,n+1/2} &= \vec{v}_i^{\,n-1/2} + \frac{\Delta t}{m_i}\vec{f}_i^{\,n}, \\
\tilde{v}_i^{\,n} &= \left(\tilde{v}_i^{\,n+1/2} + \vec{v}_i^{\,n-1/2}\right)/2, \\
\chi^n &= \left[1-\Delta t\left(1-T_o/T^n\right)/\tau_T\right]^{1/2}, \\
\vec{v}_i^{\,n+1/2} &= \chi^n \tilde{v}_i^{\,n+1/2}, \\
\vec{v}_i^{\,n} &= \left(\vec{v}_i^{\,n+1/2} + \vec{v}_i^{\,n-1/2}\right)/2, \\
\eta^n &= \left[1-\beta\,\Delta t\left(P_o - P^n\right)/\tau_P\right]^{1/3}, \\
\vec{r}_i^{\,n+1} &= \eta^n \vec{r}_i^{\,n} + \Delta t\,\vec{v}_i^{\,n+1/2}.
\end{aligned}
\tag{4.175}
$$

The intermediate velocity $\tilde{v}_i^n$ (line 2) is required to calculate the temperature T^n and hence χ^n. The rescaled velocity $\vec{v}_i^n$ (line 5) is needed to calculate the pressure P^n and hence η^n. The velocity Verlet form is

$$\begin{aligned}
&\eta^n = [1 - \Delta t (P_o - P^n)/\tau_P]^{1/3}, \\
&\vec{v}_i^{n+1/2} = \vec{v}_i^n + \frac{\Delta t}{2m_i} \vec{f}_i^n, \\
&\vec{r}_i^{n+1} = \eta^n \vec{r}_i^n + \Delta t \, \vec{v}_i^{n+1/2}, \\
&\tilde{v}_i^{n+1} = \vec{v}_i^{n+1/2} + \frac{\Delta t}{2m_i} \vec{f}_i^{n+1}, \\
&\chi^{n+1} = [1 - \Delta t (1 - T_o / T^{n+1})/\tau_T]^{1/2}, \\
&\vec{v}_i^{n+1} = \chi^{n+1} \tilde{v}_i^{n+1}.
\end{aligned} \tag{4.176}$$

As has been remarked above, these algorithms do not reproduce the true canonical or isothermal isobaric ensembles. At best they sample configurations that are in the vicinity of configurations of the true ensemble. In the context of obtaining average thermodynamic properties over instantaneous values (rather than from the fluctuations) this probably doesn't matter much. But the lack of an underpinning in statistical mechanics terms must be considered a weakness. Where these algorithms do have an advantage is in driving a system from one state point (T_1, P_1) to a different state point (T_2, P_2) (a circumstance that occurs every time a new simulation is started). The stability of the algorithms is very helpful when making such gross changes. It is always possible to adopt a more refined algorithm when the required state point has been reached.

4.8.2 The Andersen Thermostat and Barostat

Andersen devised the earliest schemes for NVT and NPT molecular dynamics [31]. The canonical ensemble was generated by selecting atoms at infrequent intervals and reassigning their velocities according to the Boltzmann distribution, which is thus effectively a stochastic collision. The extension to the isothermal isobaric ensemble was achieved through a novel volume scaling technique in which the scaling parameter was dynamically coupled to the system pressure.

To implement the thermostat the user decides on the system temperature, T, and a *mean collision rate*, ν, at which the atom velocities are to be reset. Thus the probability that an atom experiences a stochastic collision in the time interval δt is given by $\nu \delta t$. The constant ν is small, so collisions are infrequent on the molecular scale and the process follows a Poisson distribution. The rate ν governs the probability distribution of collision times, $P(t)$, which is

$$P(t) = \nu e^{-\nu t}. \tag{4.177}$$

Therefore $P(t)\delta t$ is the probability that the time interval between collisions is

between t and $t+\delta t.$ Andersen [31] recommends a value of ν of the order

$$\nu \approx \kappa V^{1/3} / k_B N, \tag{4.178}$$

where κ is the thermal conductivity, V the system volume and N the number of atoms in the system.

In practice the dynamics of the atoms follow the usual algorithm (in our case one of the Verlet schemes) except that, at each time step and for each atom, a random number is generated between 0 and 1 and compared with $\nu \Delta t,$ and if it is smaller, the atomic velocity is reassigned from the standard Gaussian velocity distribution at the required temperature $T.$ The algorithm is simple to implement and Anderson showed that system averages calculated from simulations employing this scheme are consistent with the canonical ensemble. It is not clear however, as to what extent time dependence (as given by correlation functions) is correctly reproduced, but since the stochastic episodes experienced by each atom are infrequent, it is expected that time dependence is reasonably well described.

Andersen's barostat for controlling the pressure is based on the *scaled* Hamiltonian

$$H(\{\vec{s}_i\}, \{\vec{\pi}_i\}, V, P_o) = \frac{1}{2V^{2/3}} \sum_{i=1}^{N} \frac{\pi_i^2}{m_i} + \Phi(\{V^{1/3}\vec{s}_i\}) + \frac{\Pi^2}{2M} + P_o V \tag{4.179}$$

in which $\{\vec{s}_i\}$ is the scaled counterpart to $\{\vec{r}_i\}$ and $\{\vec{\pi}_i\}$ is the momentum conjugate to $\{\vec{s}_i\}.$ P_o is the (fixed) thermodynamic pressure, V the volume, Π is the momentum conjugate to the volume and M is a *pseudo-mass* introduced to allow the volume to behave dynamically (it does not have conventional mass units, as a dimensional analysis shows). Φ represents the system potential, which depends on the inter-atomic interactions. The relationship between $\{\vec{s}_i\}$ and $\{\vec{r}_i\}$ reveals the volume scaling central to this scheme:

$$\vec{r}_i = V^{1/3} \vec{s}_i. \tag{4.180}$$

The vectors $\{\vec{s}_i\}$ are thus *scaled vectors*, scaling with the width of the simulation cell, which is $V^{1/3}.$ The scaled momentum $\{\vec{\pi}_i\}$ is related to the conventional momentum $\{\vec{p}_i\}$ via the equation

$$\vec{p}_i = V^{-1/3} \vec{\pi}_i. \tag{4.181}$$

The Hamiltonian (4.179) can be seen to be an *extended* Hamiltonian, having the additional degree of freedom provided by the volume. It is an early example of such an extension applied to molecular dynamics, though in this case the coupling of the volume to the atomic coordinates through a scaling relation is a novel feature.

The equations of motion for a system based on the Hamiltonian (4.179) are obtained in the usual way (see chapter 2, section 2.5). They are presented here in their simplest form

$$\begin{aligned}
\dot{\vec{\pi}}_i &= V^{1/3}\vec{f}_i(\{\vec{r}_i\}),\\
\dot{\vec{s}}_i &= \vec{\pi}_i/m_i V^{2/3},\\
\dot{\Pi} &= P(t)-P_o,\\
\dot{V} &= \Pi/M.
\end{aligned} \tag{4.182}$$

In the first of these equations $\vec{f}_i(\{\vec{r}_i\})$ is the conventional force, written as a function of $\{\vec{r}_i\}$. Also $P(t)$ is the instantaneous pressure, calculated from the Clausius equation appropriate for this system

$$P(t)=\frac{1}{3V}\left(\frac{1}{V^{2/3}}\sum_{i=1}^{N}\frac{\pi_i^2}{m_i}+\sum_{i=1}^{N}\vec{f}_i\cdot\vec{r}_i\right). \tag{4.183}$$

(Note. The last term on the right of (4.183) should be recast into a form suitable for periodic systems, see chapter 7.)

Assuming our simulation program works with conventional variables $\{\vec{r}_i\}$ and $\{\vec{p}_i\}$, the known quantities at the start of each time step in the leapfrog scheme are $\vec{r}_i^{\,n}$, $\vec{p}_i^{\,n-1/2}$, P^n and V^n. An additional variable must be created for Π and this is carried to the $n'th$ time step as $\Pi^{n-1/2}$. The first requirement is to convert $\vec{r}_i^{\,n}$ and $\vec{p}_i^{\,n-1/2}$ into the corresponding $\vec{s}_i^{\,n}$ and $\vec{\pi}_i^{\,n-1/2}$. This is accomplished using

$$\begin{aligned}
\vec{s}_i^{\,n} &= (V^n)^{-1/3}\vec{r}_i^{\,n},\\
V^{n-1/2} &= V^n-\Delta t\,\Pi^{n-1/2}/2M,\\
\vec{\pi}_i^{\,n-1/2} &= (V^{n-1/2})^{1/3}\vec{p}_i^{\,n-1/2}.
\end{aligned} \tag{4.184}$$

Integration then proceeds as follows

$$\begin{aligned}
\Pi^{n+1/2} &= \Pi^{n-1/2}+\Delta t\,(P^n-P_o),\\
\vec{\pi}_i^{\,n+1/2} &= \vec{\pi}_i^{\,n-1/2}+\Delta t\,(V^n)^{1/3}\vec{f}_i^{\,n},\\
\vec{s}_i^{\,n+1} &= \vec{s}_i^{\,n}+\Delta t\,\vec{\pi}_i^{\,n+1/2}/m_i V^{2/3},\\
V^{n+1} &= V^n+\Delta t\,\Pi^{n+1/2}/M.
\end{aligned} \tag{4.185}$$

The restoration of the original coordinates is accomplished using

$$\begin{aligned}
\vec{r}_i^{\,n+1} &= \left(V^{n+1}\right)^{1/3}\vec{s}_i^{\,n+1},\\
V^{n+1/2} &= V^{n+1} - \Delta t\,\Pi^{n+1/2}/2\mathrm{M},\\
\vec{p}_i^{\,n+1/2} &= \left(V^{n+1/2}\right)^{-1/3}\vec{\pi}_i^{\,n+1/2}.
\end{aligned} \tag{4.186}$$

The corresponding scheme using the velocity Verlet algorithm starts each time step with the known values $\vec{r}_i^{\,n}$, $\vec{p}_i^{\,n}$, P^n, V^n and Π^n. The conversion of the variables and integration is then

$$\begin{aligned}
\vec{\pi}_i^{\,n} &= \left(V^n\right)^{1/3}\vec{p}_i^{\,n},\\
\vec{s}_i^{\,n} &= \left(V^n\right)^{-1/3}\vec{r}_i^{\,n}.
\end{aligned} \tag{4.187}$$

$$\begin{aligned}
\Pi^{n+1/2} &= \Pi^n + \Delta t\left(P^n - P_o\right)/2,\\
\vec{\pi}_i^{\,n+1/2} &= \vec{\pi}_i^{\,n} + \Delta t\left(V^n\right)^{1/3}\vec{f}_i^{\,n}/2,\\
\vec{s}_i^{\,n+1} &= \vec{s}_i^{\,n} + \Delta t\left(V^n\right)^{-2/3}\vec{\pi}_i^{\,n+1/2}/2\,m_i,\\
V^{n+1} &= V^n + \Delta t\,\Pi^{n+1/2}/M,\\
\vec{\pi}_i^{\,n+1} &= \vec{\pi}_i^{\,n+1/2} + \Delta t\left(V^{n+1}\right)^{1/3}\vec{f}_i^{\,n+1}/2,\\
\Pi^{n+1} &= \Pi^{n+1/2} + \Delta t\left(P^{n+1} - P_o\right)/2.
\end{aligned} \tag{4.188}$$

$$\begin{aligned}
\vec{p}_i^{\,n+1} &= \left(V^{n+1}\right)^{-1/3}\vec{\pi}_i^{\,n+1},\\
\vec{r}_i^{\,n+1} &= \left(V^{n+1}\right)^{1/3}\vec{s}_i^{\,n+1}.
\end{aligned} \tag{4.189}$$

A word needs to be said about the parameter M, the pseudo-mass that allows the volume to be driven dynamically. Andersen showed that this parameter has no bearing on the thermodynamic averages obtained. The same averages should result whatever value of M is adopted. However it is clear that it affects the rate at which the volume fluctuates and this does have an impact on the time dependent properties of the system. Andersen recommended that M should be chosen so that the time scale of the fluctuations in V should be approximately equal to $V^{1/3}$ divided by the velocity of sound in the system.

Clearly both Andersen's thermostat and barostat may be combined in the same simulation to obtain the isothermal-isobaric ensemble. This entails a slight modification of the thermostat however. The Boltzmann distribution for the particle momentum is proportional to $\exp\left(-\pi_i^2/2\,m_i V^{2/3} k_B T\right)$ for the NPT ensemble and not $\exp\left(-p_i^2/2\,m_i k_B T\right)$ as it is for the NVT ensemble. This means that selecting the new momenta after a collision episode is different in these two cases i.e. new momenta are selected from different Gaussian distributions.

4.8.3 The Nosé-Hoover Thermostat

The algorithm devised by Nosé [32] was the first instance of a molecular dynamics algorithm that dynamically generated the canonical ensemble. Unfortunately the algorithm required a variable scaling of time and was therefore somewhat awkward to

use in practice. Nevertheless its impact on the theoretical development of molecular dynamics was huge. Subsequently Hoover [33] cast the algorithm into a form not requiring time scaling and the result is known as the Nosé-Hoover algorithm.

The equations of motion for this algorithm are

$$\begin{aligned} \dot{\vec{r}}_i &= \vec{p}_i / m_i, \\ \dot{\vec{p}}_i &= \vec{f}_i(\{\vec{r}_i\}) - \zeta \vec{p}_i, \\ \dot{\zeta} &= \frac{1}{Q}\left[\sum_{i=1}^{N} p_i^2 / m_i - N_f k_B T_0\right]. \end{aligned} \tag{4.190}$$

Vectors $\vec{r}_i$ and $\vec{p}_i$ are the usual position and momentum of the atoms. T_0 is the target system temperature and k_B Boltzmann's constant. The variable ζ is the *thermostat* variable and Q the fixed "mass" associated with its dynamics. N_f is the number of degrees of freedom in the system. This is an example of an *extended system* in which an additional time dependent variable (ζ) is introduced, effectively to function as a *heat bath* controlling the temperature. Note however that these equations are *not* derived from a Hamiltonian. The first of the above equations is the usual expression relating velocity and momentum. The second however, is a modified form of Newton's second law of motion, in which the atomic momentum is coupled to the thermostat variable, which provides a means by which the thermostat affects the momentum via a feedback mechanism. The third equation is the equation of motion of the thermostat itself, which is driven by the deviation of the system kinetic energy from that associated with the target temperature.

The algorithm for the Nosé-Hoover thermostat is best suited to a velocity Verlet form, which does not require iteration and is presented in (4.191). In contrast the leapfrog variant requires an iteratative scheme because the atomic velocities are not known at time step $n.$

The initial value for the thermostat variable ζ can be set to zero. Its derivative $\dot{\zeta}$ should fluctuate about a zero average in the equilibrium system. $Q,$ the "mass" of the thermostat, is an adjustable parameter. If it is too large then the thermostat responds sluggishly and the atomic trajectories resemble the microcanonical system. Too small a value and the fluctuations become extremely rapid, which may raise stability issues. However the main concern with the algorithm (and indeed with many similar algorithms) is that in "stiff" systems (i.e. those with large force constants) the algorithm does not sample phase space adequately, no matter what value of Q is chosen. It was to remedy this deficiency that Nosé-Hoover chains were introduced (see section 4.8.7).

The Nosé-Hoover thermostat can be used in conjunction with a barostat in the same manner as the Andersen barostat in section 4.8.2 See also the following sections on various isothermal-isobaric algorithms.

$$
\begin{aligned}
\zeta^{n+1/4} &= \zeta^{n} + \frac{\Delta t}{4Q}\left[\sum_{i=1}^{N} m_i (v_i^n)^2 - N_f k_B T_0\right], \\
\tilde{v}_i^{n+1/2} &= \vec{v}_i^{\,n} - \frac{\Delta t}{2}\zeta^{n+1/4}\vec{v}_i^{\,n}, \\
\zeta^{n+1/2} &= \zeta^{n+1/4} + \frac{\Delta t}{4Q}\left[\sum_{i=1}^{N} m_i (\tilde{v}_i^{n+1/2})^2 - N_f k_B T_0\right], \\
\vec{v}_i^{\,n+1/2} &= \tilde{v}_i^{n+1/2} + \frac{\Delta t}{2m_i}\vec{f}_i^{\,n}, \\
\vec{r}_i^{\,n+1} &= \vec{r}_i^{\,n} + \Delta t\, \vec{v}_i^{\,n+1/2}, \\
\tilde{v}_i^{n+1} &= \vec{v}_i^{\,n+1/2} + \frac{\Delta t}{2m_i}\vec{f}_i^{\,n+1}, \\
\zeta^{n+3/4} &= \zeta^{n+1/2} + \frac{\Delta t}{4Q}\left[\sum_{i=1}^{N} m_i (\tilde{v}_i^{n+1})^2 - N_f k_B T_0\right], \\
\vec{v}_i^{\,n+1} &= \tilde{v}_i^{n+1} - \frac{\Delta t}{2}\zeta^{n+3/4}\tilde{v}_i^{n+1}, \\
\zeta^{n+1} &= \zeta^{n+3/4} + \frac{\Delta t}{4Q}\left[\sum_{i=1}^{N} m_i (v_i^{n+1})^2 - N_f k_B T_0\right].
\end{aligned}
\tag{4.191}
$$

4.8.4 The Melchionna-Ciccotti-Holian Isothermal-Isobaric Algorithm

The MCH algorithm [34] couples the Nosé-Hoover thermostat with a barostat to generate the isothermal-isobaric ensemble. The equations of motion are as follows:

$$
\begin{aligned}
\dot{\vec{r}}_i &= \frac{\vec{p}_i}{m_i} + \eta(\vec{r}_i - \vec{R}_0), \\
\dot{\vec{p}}_i &= \vec{f}_i(\{\vec{r}_i\}) - [\zeta + \eta]\vec{p}_i, \\
\dot{\zeta} &= \frac{N_f k_B}{Q}(T - T_0) + \frac{1}{Q}(W\eta^2 - k_B T_0), \\
\dot{\eta} &= \frac{3}{W}V(P - P_0) - \zeta\eta, \\
\dot{V} &= 3\eta V.
\end{aligned}
\tag{4.192}
$$

In these equations ζ and η represent the time dependent thermostat and barostat variables respectively. Variables P and T are the *instantaneous* values of the pressure and temperature, calculated in the usual way, and P_0 and T_0 are their target equilibrium values. The constant N_f defines the number of degrees of freedom in the system. The first equation provides for the integration of the atomic position $\vec{r}_i$ and, additional to the momentum contribution $\vec{p}_i$ is a term which scales the atomic position with respect to the centre of the system mass at $\vec{R}_0$, by the

barostat variable η. The second equation provides for the integration of the momentum $\vec{p}_i$ and not only includes the expected force contribution $\vec{f}_i$, but also a momentum dependent term which is scaled by both the thermostat and barostat variables ζ and η. These two equations demonstrate how the thermostat and barostat influence the dynamics of the atoms to generate configurations from the isothermal-isobaric ensemble.

The third and fourth equations represent the equations of motion of the thermostat and barostat respectively. The thermostat equation is "driven" by the expected difference between the instantaneous and target temperatures T and T_0, as was seen in the Nosé-Hoover algorithm. However, it also contains a second term derived from the kinetic energy of the barostat, which also has an effect on the thermalisation. In the fourth equation the barostat is driven by the difference between pressures P and P_0, which is intended to hold the pressure P close to the target P_0. This equation also contains a second term, which couples the thermostat and barostat variables and helps to prevent drift in the system state point. The last equation describes the time dependence of the system volume V.

The constants Q and W represent the "mass" parameters for the thermostat and barostat respectively. These are usually expressed in terms of *relaxation times* τ_T and τ_P through the formulae:

$$\begin{aligned} Q &= N_f k_B \tau_T^2, \\ W &= N_f k_B \tau_P^2. \end{aligned} \tag{4.193}$$

These relaxation times loosely represent the time scales on which the temperature and pressure fluctuations take place and are user defined properties. Theoretically they do not affect the equilibrium values of the temperature and pressure, but they do affect the rate of achievement of equilibrium and the sampling interval to be used in the statistical analysis of the results.

The equations of motion, though *not* Hamiltonian, possess a convenient conserved quantity, which is:

$$H_{MCH} = K + \Phi + P_0 V + \frac{1}{2} Q \zeta^2 + \frac{1}{2} W \eta^2 + \int_0^t \left(\frac{Q}{\tau_T^2} \zeta + k_B T_0 \right) dt. \tag{4.194}$$

As with the Nosé-Hoover algorithm the MCH algorithm is best couched in the form of the velocity Verlet integrator, which is presented in scheme (4.195).

$$
\begin{aligned}
\zeta^{n+1/4} &= \zeta^{n} + \frac{\Delta t}{4Q}\left\{ N_f k_B (T^n - T_0) + \left(W(\eta^n)^2 - k_B T_0\right)\right\},\\
\eta^{n+1/4} &= \eta^{n} + \frac{\Delta t}{4}\left\{ \frac{3}{W} V^n (P^n - P_0) - \zeta^{n+1/4}\eta^n, \right\},\\
\tilde{v}_i^{n+1/2} &= \vec{v}_i^{n} - \frac{\Delta t}{2}\left(\zeta^{n+1/4} + \eta^{n+1/4}\right)\vec{v}_i^{n},\\
\eta^{n+1/2} &= \eta^{n+1/4} + \frac{\Delta t}{4}\left\{ \frac{3}{W} V^n (P^{n+1/2} - P_0) - \zeta^{n+1/4}\eta^{n+1/4} \right\},\\
\zeta^{n+1/2} &= \zeta^{n+1/4} + \frac{\Delta t}{4Q}\left\{ N_f k_B (T^{n+1/2} - T_0) + \left(W(\eta^{n+1/2})^2 - k_B T_0\right)\right\},\\
\vec{v}_i^{n+1/2} &= \tilde{v}_i^{n+1/2} + \frac{\Delta t}{2m_i}\vec{f}_i^{\,n},\\
\vec{r}_i^{n+1} &= \vec{r}_i^{n} + \Delta t\left\{ \vec{v}_i^{n+1/2} + \eta^{n+1/2}(\vec{r}_i^{n} - \vec{R}_0)\right\},\\
V^{n+1} &= V^n \exp\left(3\Delta t\,\eta^{n+1/2}\right),\\
\tilde{v}_i^{n+1} &= \vec{v}_i^{n+1/2} + \frac{\Delta t}{2m_i}\vec{f}_i^{\,n+1},\\
\zeta^{n+3/4} &= \zeta^{n+1/2} + \frac{\Delta t}{4Q}\left\{ N_f k_B (T^{n+1} - T_0) + \left(W(\eta^{n+1/2})^2 - k_B T_0\right)\right\},\\
\eta^{n+3/4} &= \eta^{n+1/2} + \frac{\Delta t}{4}\left\{ \frac{3}{W} V^{n+1} (P^{n+1} - P_0) - \zeta^{n+3/4}\eta^{n+1/2} \right\},\\
v_i^{n+1} &= \tilde{v}_i^{n+1} - \frac{\Delta t}{2}\left(\zeta^{n+3/4} + \eta^{n+3/4}\right)\tilde{v}_i^{n+1},\\
\eta^{n+1} &= \eta^{n+3/4} + \frac{\Delta t}{4}\left\{ \frac{3}{W} V^{n+1} (P^{n+1} - P_0) \right\} - \zeta^{n+3/4}\eta^{n+3/4},\\
\zeta^{n+1} &= \zeta^{n+3/4} + \frac{\Delta t}{4Q}\left\{ N_f k_B (T^{n+1} - T_0) + \left(W(\eta^{n+1})^2 - k_B T_0\right)\right\}.
\end{aligned}
\tag{4.195}
$$

The pressure P and temperature T appearing in this scheme should always be based on the most recent values of underlying variables. The initial values of ζ and η may be set to zero. At equilibrium the derivatives $\dot{\zeta}$ and $\dot{\eta}$ should fluctuate about a zero average. The algorithm performs well provided the system force constants are not too stiff, in which case it can be improved by using Nosé-Hoover chains (see section 4.8.7).

4.8.5 The Martyna-Tobias-Klein Isothermal-Isobaric Algorithm

The MTK algorithm [35] is an alternative to the MCH algorithm which, the authors argue, gives a more accurate representation of the isothermal-isobaric ensemble. The MTK equations of motion, which have the same conserved property (4.194) as the MCH algorithm (4.192), are

$$
\begin{aligned}
&\dot{\vec{r}}_i = \frac{\vec{p}_i}{m_i} + \eta \vec{r}_i, \\
&\dot{\vec{p}}_i = \vec{f}_i(\{\vec{r}_i\}) - \left(1 + \frac{3}{N_f}\right)\eta \vec{p}_i - \zeta \vec{p}_i, \\
&\dot{\zeta} = \frac{N_f k_B}{Q}(T - T_0) + \frac{1}{Q}(W\eta^2 - k_B T_0), \\
&\dot{\eta} = \frac{3}{W} V(P - P_0) + \frac{3}{W} k_B T - \zeta\eta, \\
&\dot{V} = 3\eta V.
\end{aligned}
\tag{4.196}
$$

The variables in these equations have the same meaning as in the MCH algorithm (4.192), and masses Q and W are defined by equations (4.193). The equations may be integrated in much the same way as for MCH. This is presented in scheme (4.197).

$$
\begin{aligned}
&\zeta^{n+1/4} = \zeta^n + \frac{\Delta t}{4Q}\left\{N_f k_B(T^n - T_0) + (W(\eta^n)^2 - k_B T_0)\right\}, \\
&\eta^{n+1/4} = \eta^n + \frac{\Delta t}{4}\left\{\frac{3}{W} V^n(P^n - P_0) + \frac{3}{W} k_B T^n - \zeta^{n+1/4}\eta^n\right\}, \\
&\tilde{v}_i^{n+1/2} = \vec{v}_i^n - \frac{\Delta t}{2}\left\{\zeta^{n+1/4} + \left(1 + \frac{3}{N_f}\right)\eta^{n+1/4}\right\}\vec{v}_i^n, \\
&\eta^{n+1/2} = \eta^{n+1/4} + \frac{\Delta t}{4}\left\{\frac{3}{W} V^n(P^{n+1/2} - P_0) + \frac{3}{W} k_B T^{n+1/2} - \zeta^{n+1/4}\eta^{n+1/4}\right\}, \\
&\zeta^{n+1/2} = \zeta^{n+1/4} + \frac{\Delta t}{4Q}\left\{N_f k_B(T^{n+1/2} - T_0) + (W(\eta^{n+1/2})^2 - k_B T_0)\right\}, \\
&\vec{v}_i^{n+1/2} = \tilde{v}_i^{n+1/2} + \frac{\Delta t}{2m_i}\vec{f}_i^n, \\
&\vec{r}_i^{n+1} = \vec{r}_i^n + \Delta t\left(\vec{v}_i^{n+1/2} + \eta^{n+1/2}\vec{r}_i^n\right), \\
&V^{n+1} = V^n \exp(3\Delta t\,\eta^{n+1/2}), \\
&\tilde{v}_i^{n+1} = \vec{v}_i^{n+1/2} + \frac{\Delta t}{2m_i}\vec{f}_i^{n+1}, \\
&\zeta^{n+3/4} = \zeta^{n+1/2} + \frac{\Delta t}{4Q}\left\{N_f k_B(T^{n+1} - T_0) + (W(\eta^{n+1/2})^2 - k_B T_0)\right\}, \\
&\eta^{n+3/4} = \eta^{n+1/2} + \frac{\Delta t}{4}\left\{\frac{3}{W} V^{n+1}(P^{n+1} - P_0) + \frac{3}{W} k_B T^{n+1} - \zeta^{n+3/4}\eta^{n+1/2}\right\}, \\
&\vec{v}_i^{n+1} = \tilde{v}_i^{n+1} - \frac{\Delta t}{2}\left\{\zeta^{n+3/4} + \left(1 + \frac{3}{N_f}\right)\eta^{n+3/4}\right\}\tilde{v}_i^{n+1}, \\
&\eta^{n+1} = \eta^{n+3/4} + \frac{\Delta t}{4}\left\{\frac{3}{W} V^{n+1}(P^{n+1} - P_0) + \frac{3}{W} k_B T^{n+1} - \zeta^{n+3/4}\eta^{n+3/4}\right\}, \\
&\zeta^{n+1} = \zeta^{n+3/4} + \frac{\Delta t}{4Q}\left\{N_f k_B(T^{n+1} - T_0) + (W(\eta^{n+1})^2 - k_B T_0)\right\}.
\end{aligned}
\tag{4.197}
$$

As with the MCH algorithm, the initial values of ζ and η may be set to zero and at equilibrium the derivatives $\dot{\zeta}$ and $\dot{\eta}$ should fluctuate about a zero average. The pressure P and temperature T appearing in this scheme should always be based on the most recent values of underlying variables. The scheme can also be used in conjunction with a Nose-Hoover thermostat chain (see section 4.8.7).

4.8.6 The Liouville Operator and Algorithm Structure

The structures of the algorithms presented in (4.191), (4.195) and (4.197) deserve comment, since at first sight they appear arbitrary and perhaps unnecessarily complicated. On a practical level, it is not necessary to know the details of the derivation of these algorithms to apply them correctly, but it is insightful to know why they take the forms they do. Their structure has been laid out according to the principles outlined in section 4.3 , which exploits the properties of the Liouville Equation to construct symplectic algorithms. In section 4.3 we showed how the decomposition of the Liouville propagator into its discrete counterpart (equation (4.31)) and the subsequent split of the Liouville operator into two basic operators, $i\hat{L}_1$ and $i\hat{L}_2$ (equations (4.33)) provides a template for the velocity Verlet algorithm, which is an appropriate integrator for the NVE system. A similar procedure can be applied to other ensemble methods to obtain integrators that have symplectic characteristics. We shall demonstrate this with the Nosé-Hoover case.

To start we re-write the Nosè-Hoover equations of motion (4.190) in a slightly different way:

$$\begin{aligned}
\dot{\vec{r}}_i &= \vec{p}_i/m_i, \\
\dot{\vec{p}}_i &= \vec{f}_i(\{\vec{r}_i\}) - \dot{\xi}\vec{p}_i, \\
\dot{\xi} &= p_\xi/Q, \\
\dot{p}_\xi &= \left[\sum_{i=1}^{N} p_i^2/m_i - N_f k_B T_0\right],
\end{aligned} \tag{4.198}$$

in which we have replaced the variable ζ by $\dot{\xi}$, the time derivative of a variable ξ, for which the momentum is given as p_ξ. Using these variables allows us to write the equations of motion for the thermostat in the same way as the atomic motion so as to exploit the Liouville equation approach of section 4.3. The Liouville operators associated with these equations can immediately be written (*c.f.* equations (4.33)):

$$\begin{aligned}
i\hat{L}_1 &= \sum_{j=1}^{3N}\left(\frac{p_j}{m_j}\frac{\partial}{\partial x_j}\right), \quad i\hat{L}_2 = \sum_{j=1}^{3N}\left(f_j\frac{\partial}{\partial p_j}\right), \\
i\hat{L}_3 &= -\sum_{j=1}^{3N}\left(\dot{\xi}\,p_j\frac{\partial}{\partial p_j}\right), \\
i\hat{L}_4 &= \dot{\xi}\frac{\partial}{\partial \xi}, \qquad i\hat{L}_5 = \dot{p}_\xi\frac{\partial}{\partial p_\xi}.
\end{aligned} \tag{4.199}$$

Evidently, as previously, $i\hat{L}_1$ and $i\hat{L}_2$ are the propagators of position and momentum (or velocity), while $i\hat{L}_3$ applies the thermostat to the atomic momenta, and finally $i\hat{L}_4$ and $i\hat{L}_5$ update the thermostat variable ξ and its momentum p_ξ. If we temporarily consolidate $i\hat{L}_3$ to $i\hat{L}_5$ into a "thermostat" propagator $i\hat{L}_T = i\hat{L}_3 + i\hat{L}_4 + i\hat{L}_5$, we can write the full Liouville propagator in a form resembling equation (4.34) i.e.

$$\begin{aligned}\left[\exp\left(-i\Delta t\{\hat{L}_1+\hat{L}_2+\hat{L}_T\}\right)\right]^n = \Big[&\exp\left(-i\Delta t\,\hat{L}_T/2\right)\\ &\exp\left(-i\Delta t\,\hat{L}_2/2\right)\exp\left(-i\Delta t\,\hat{L}_1\right)\exp\left(-i\Delta t\,\hat{L}_2/2\right)\\ &\exp\left(-i\Delta t\,\hat{L}_T/2\right)\Big]^n + O(\Delta t^3).\end{aligned} \tag{4.200}$$

The central three propagators on the left follow the same pattern as the velocity Verlet given in section 4.3. This means the standard velocity Verlet scheme occupies the central portion of the final integration scheme. This is bracketed either side by the thermostat propagator $i\hat{L}_T$ acting for half a time step. If this in turn is split into its components, $i\hat{L}_3$ to $i\hat{L}_5$, then following from the Trotter expansion (4.34)we can write

$$\begin{aligned}\exp\left(-i\Delta t\,\hat{L}_T/2\right) = &\exp\left(-i\Delta t\,\hat{L}_5/4\right)\exp\left(-i\Delta t\,\hat{L}_4/4\right)\\ &\exp\left(-i\Delta t\,\hat{L}_3/2\right)\\ &\exp\left(-i\Delta t\,\hat{L}_4/4\right)\exp\left(-i\Delta t\,\hat{L}_5/4\right),\end{aligned} \tag{4.201}$$

and putting this back into (4.200) gives:

$$\begin{aligned}\left[\exp\left(-i\Delta t\{\hat{L}_1+\hat{L}_2+\hat{L}_T\}\right)\right]^n = \Big[&\exp\left(-i\Delta t\,\hat{L}_5/4\right)\exp\left(-i\Delta t\,\hat{L}_4/4\right)\\ &\exp\left(-i\Delta t\,\hat{L}_3/2\right)\\ &\exp\left(-i\Delta t\,\hat{L}_4/4\right)\exp\left(-i\Delta t\,\hat{L}_5/4\right)\\ &\exp\left(-i\Delta t\,\hat{L}_2/2\right)\exp\left(-i\Delta t\,\hat{L}_1\right)\exp\left(-i\Delta t\,\hat{L}_2/2\right)\\ &\exp\left(-i\Delta t\,\hat{L}_5/4\right)\exp\left(-i\Delta t\,\hat{L}_4/4\right)\\ &\exp\left(-i\Delta t\,\hat{L}_3/2\right)\\ &\exp\left(-i\Delta t\,\hat{L}_4/4\right)\exp\left(-i\Delta t\,\hat{L}_5/4\right)\Big]^n + O(\Delta t^3).\end{aligned} \tag{4.202}$$

This result maps out the full scheme for the Nosè-Hoover thermostat algorithm in the velocity Verlet framework. It is presented in (4.203) as an alternative to (4.191).

There are some things to note about the algorithm (4.203). Firstly, the integration of ξ, which is the rôle of propagator $i\hat{L}_4$, is carried out twice only, and not four times as implied by the expansion (4.202). This is because $i\hat{L}_4$ and $i\hat{L}_3$ are *commuting* operators and so the two quarter-step integrations of $i\hat{L}_4$ either side of $i\hat{L}_3$ can be combined into one half-step integration. Secondly, removal of the operator $i\hat{L}_4$ from

the scheme altogether renders the scheme equivalent to (4.191). The value of ξ is, in any case, only required to determine the conserved variable, which though useful for other reasons, is not required for integrating the motion of the atoms.

This approach can be (and has been) used to derive symplectic integrators for all the ensemble algorithms described here. The procedure is always essentially the same, except that an extra set of integrators is required for the barostat and for the coupling of the thermostat and barostat for the isothermal-isobaric ensemble. There is also the scaling of the atomic positions to consider and this is performed at the same location in the algorithm as the position update. In other words, the position scaling is included as a modification to operator $i\hat{L}_1$. This can be seen in the MCH and MTK algorithms presented in equations (4.195) and (4.197).

$$
\begin{aligned}
p_\xi^{n+1/4} &= p_\xi^n + \frac{\Delta t}{4}\left[\sum_{i=1}^{N} m_i (v_i^n)^2 - N_f k_B T_0\right], \\
\xi^{n+1/2} &= \xi^n + \frac{\Delta t}{2Q} p_\xi^{n+1/4}, \\
\tilde{v}_i^{n+1/2} &= \vec{v}_i^n - \frac{\Delta t}{2Q} p_\xi^{n+1/4} \vec{v}_i^n, \\
p_\xi^{n+1/2} &= p_\xi^{n+1/4} + \frac{\Delta t}{4}\left[\sum_{i=1}^{N} m_i (\tilde{v}_i^{n+1/2})^2 - N_f k_B T_0\right], \\
\vec{v}_i^{n+1/2} &= \tilde{v}_i^{n+1/2} + \frac{\Delta t}{2 m_i} \vec{f}_i^n, \\
\vec{r}_i^{n+1} &= \vec{r}_i^n + \Delta t\, \vec{v}_i^{n+1/2}, \\
\tilde{v}_i^{n+1} &= \vec{v}_i^{n+1/2} + \frac{\Delta t}{2 m_i} \vec{f}_i^{n+1}, \\
p_\xi^{n+3/4} &= p_\xi^{n+1/2} + \frac{\Delta t}{4}\left[\sum_{i=1}^{N} m_i (\tilde{v}_i^{n+1})^2 - N_f k_B T_0\right], \\
\xi^{n+1} &= \xi^{n+1/2} + \frac{\Delta t}{2Q} p_\xi^{n+3/4}, \\
\vec{v}_i^{n+1} &= \tilde{v}_i^{n+1} - \frac{\Delta t}{2Q} p_\xi^{n+3/4} \tilde{v}_i^{n+1}, \\
p_\xi^{n+1} &= p_\xi^{n+3/4} + \frac{\Delta t}{4}\left[\sum_{i=1}^{N} m_i (v_i^{n+1})^2 - N_f k_B T_0\right].
\end{aligned}
\tag{4.203}
$$

4.8.7 Nosé-Hoover Thermostat Chains

Algorithms with thermostats derived from the Nosé-Hoover form are generally well behaved and do a good job creating canonical and isothermal-isobaric ensembles. However there are occasions when it is found that the distribution of energies of some of the degrees of freedom are not properly described by the Boltzmann distribution, which reflects that the system is not properly ergodic. This can happen for example when stiff or harmonic potentials are used to describe chemical bonds. A cure for this

problem is to use a chain of Nosé-Hoover thermostats [36].

The idea is a simple one: if every degree of freedom of a system is supposed to sample from a Boltzmann distribution, then it must be true for the thermostat and barostat as well. If these are not properly sampling from this distribution, it implies that the system is not properly ergodic. So an additional thermostat can, in principle, be introduced for the original thermostat and barostat, to help ensure that it does behave properly. These additional thermostats can be thermostatted in turn, which leads inevitably to a chain of thermostats of arbitrary order. The computational expense, in time and storage, of such chains need not be great (unless every individual degree of freedom of the system is independently thermostatted) and the quality of the ensemble is significantly improved.

To create a chain it is necessary to write a set of coupled equations to a chosen order M :

$$\begin{aligned}
\dot{\zeta}_1 &= \frac{N_f k_B}{Q_1}\left(T - T_0\right) - \zeta_1 \zeta_2 , \\
\dot{\zeta}_i &= \frac{1}{Q_i}\left\{Q_{i-1}\zeta_{i-1}^2 - k_B T_0\right\} - \zeta_i \zeta_{i+1} , \quad \left(i = 2, \ldots, M-1\right) \\
\dot{\zeta}_M &= \frac{1}{Q_M}\left\{Q_{M-1}\zeta_{M-1}^2 - k_B T_0\right\} .
\end{aligned} \tag{4.204}$$

The first of these equations is for the original Nosé-Hoover thermostat in the absence of a barostat, though note that this has an additonal coupling term, $\left(-\zeta_1 \zeta_2\right),$ linking the thermostat to the chain of higher order thermostats. The remaining equations are for all the other thermostats to order $M.$ This chain of thermostats is coupled to the atomic motions via the familiar equations:

$$\begin{aligned}
\dot{\vec{r}}_i &= \vec{p}_i / m_i , \\
\dot{\vec{p}}_i &= \vec{f}_i\left(\{\vec{r}_i\}\right) - \zeta_1 \vec{p}_i .
\end{aligned} \tag{4.205}$$

The structure of an algorithm for applying such a chain of thermostats in the velocity Verlet scheme is based on the Liouville equation approach outlined in sections 4.3 and 4.8.6 . This gives the following form:

1. Integrate thermostats over 1/4 time step
2. Apply thermostat ζ_1 to atomic velocities (over half time step)
3. Integrate thermostats over 1/4 time step
4. Integrate atomic velocities over 1/2 time step
5. Integrate atomic positions over 1 time step
6. Integrate atomic velocities over 1/2 time step
7. Integrate thermostats over 1/4 time step
8. Apply thermostat ζ_1 to atomic velocities (over half time step)
9. Integrate thermostats over 1/4 time step

Steps 4 - 6 in this scheme represent the standard velocity Verlet algorithm. Also it is apparent that steps 1 – 3 and 7 – 9 are equivalent, so it is sensible to write one computational routine for both. A scheme for such a routine is presented in (4.206). It should be applied both before and after the usual velocity Verlet scheme. Note that in this scheme the index n can mean either a full or a half time step, depending on its application before or after the velocity Verlet scheme.

$$
\begin{aligned}
&\zeta_M^{n+1/4} = \zeta_M^n + \frac{\Delta t}{4Q_M}\left(Q_{M-1}(\zeta_{M-1}^n)^2 - k_B T_0\right),\\
&\zeta_{M-1}^{n+1/8} = \zeta_{M-1}^n - \frac{\Delta t}{8}\zeta_{M-1}^n \zeta_M^{n+1/4},\\
&\tilde{\zeta}_{M-1}^{n+1/4} = \zeta_{M-1}^{n+1/8} + \frac{\Delta t}{4Q_{M-1}}\left(Q_{M-2}(\zeta_{M-2}^n)^2 - k_B T_0\right),\\
&\zeta_{M-1}^{n+1/4} = \tilde{\zeta}_{M-1}^{n+1/4} - \frac{\Delta t}{8}\tilde{\zeta}_{M-1}^{n+1/4}\zeta_M^{n+1/4},\\
&\qquad\qquad \cdots \ etc.\\
&\zeta_1^{n+1/8} = \zeta_1^n - \frac{\Delta t}{8}\zeta_1^n \zeta_2^{n+1/4},\\
&\tilde{\zeta}_1^{n+1/4} = \zeta_1^{n+1/8} + \frac{\Delta t\, N_f k_B}{4Q_1}\left(T^n - T_0\right),\\
&\zeta_1^{n+1/4} = \tilde{\zeta}_1^{n+1/4} - \frac{\Delta t}{8}\tilde{\zeta}_1^{n+1/4}\zeta_2^{n+1/4},\\
&\vec{v}_i^{\,n+1/2} = \vec{v}_i^{\,n} - \frac{\Delta t}{2}\zeta_1^{n+1/4}\vec{v}_i^{\,n},\\
&\zeta_1^{n+3/8} = \zeta_1^{n+1/4} - \frac{\Delta t}{8}\zeta_1^{n+1/4}\zeta_2^{n+1/4},\\
&\tilde{\zeta}_1^{n+1/2} = \zeta_1^{n+3/8} + \frac{\Delta t\, N_f k_B}{4Q_1}\left(\tilde{T}^{n+1/2} - T_0\right),\\
&\zeta_1^{n+1/2} = \tilde{\zeta}_1^{n+1/2} - \frac{\Delta t}{8}\tilde{\zeta}_1^{n+1/2}\zeta_2^{n+1/4},\\
&\qquad\qquad \cdots \ etc.\\
&\zeta_{M-1}^{n+3/8} = \zeta_{M-1}^{n+1/4} - \frac{\Delta t}{8}\zeta_{M-1}^{n+1/4}\zeta_M^{n+1/4},\\
&\tilde{\zeta}_{M-1}^{n+1/2} = \zeta_{M-1}^{n+3/8} + \frac{\Delta t}{4Q_{M-1}}\left(Q_{M-2}(\zeta_{M-2}^{n+1/2})^2 - k_B T_0\right),\\
&\zeta_{M-1}^{n+1/2} = \tilde{\zeta}_{M-1}^{n+1/2} - \frac{\Delta t}{8}\tilde{\zeta}_{M-1}^{n+1/2}\zeta_M^{n+1/4},\\
&\zeta_M^{n+1/2} = \zeta_M^{n+1/4} + \frac{\Delta t}{4Q_M}\left(Q_{M-1}(\zeta_{M-1}^{n+1/2})^2 - k_B T_0\right).
\end{aligned}
\tag{4.206}
$$

There are some additional points to make about the scheme presented in (4.206). Firstly, as in the Nosè-Hoover, MCH and MTK schemes presented previously, the thermostat variables are based on ζ_i rather than ξ_i, where $\zeta_i = \dot{\xi}_i$, (see section 4.8.6). The variables ξ_i are required to calculate the conserved variable (*c.f.* equation (4.194)), but are not needed for integrating the atomic equations of motion.

The second point is that scheme (4.206) does not include any barostats and their associated thermostats. However, extension of the scheme to include them using the approaches described above is straightforward (if tedious). Finally, it should be remembered that the scheme described assumes a half-step integration. It is in fact possible to subdivide the half-step into (say) n_c sub-steps and apply the scheme repeatedly n_c times for successive time intervals of duration $\Delta t/2n_c$. This leads to a more accurate and stable scheme. However, this is not thought necessary for most cases. A thorough discussion of all these points is provided in the article by Martyna *et al.* [37].

4.8.8 The "Gentle" Thermostat

This thermostat algorithm provides an alternative to Nosè-Hoover chains. Devised by Leimkuhler *et al.* [38], it couples the Nosè-Hoover thermostat to a Langevin or stochastic heat bath. The thermostat is therefore a kind of Brownian particle in the extended system. The equations of motion have the form:

$$\begin{aligned}
&\dot{\vec{r}}_i = \vec{p}_i/m_i, \\
&\dot{\vec{p}}_i = \vec{f}_i - \zeta \vec{p}_i, \\
&d\zeta = \frac{1}{Q}\left[\sum_{i=1}^{N} p_i^2/m_i - N_f k_B T_0\right]dt - \frac{Q\sigma^2}{2k_B T_0}\zeta dt + \sigma\, dW.
\end{aligned} \tag{4.207}$$

The resemblance of these equations to the Nosè-Hoover equations (4.190) is clear. The third equation differs only in the inclusion of a dissipation term (second right) and a stochastic term (third right). The parameter σ determines the magnitude of the impulse change in ζ and dW represents a random number on the interval [-1,1]. The coefficient of ζdt in the dissipation term has the form required to satisfy the fluctuation-dissipation theorem, which means that, at equilibrium, ζ will properly sample from the Boltzmann distribution and the associated temperature will be the target temperature T_0. The integration scheme for this thermostat has the form

$$\begin{aligned}
&\tilde{p}_i^{\,n+1/2} = \vec{p}_i^{\,n} + \frac{\Delta t}{2}\vec{f}_i^{\,n}, \\
&\vec{r}_i^{\,n+1/2} = \vec{r}_i^{\,n} + \frac{\Delta t}{2m_i}\tilde{p}_i^{\,n+1/2}, \\
&\vec{p}_i^{\,n+1/2} = \tilde{p}_i^{\,n+1/2}\exp\left(-\frac{\Delta t}{2}\zeta^n\right), \\
&\zeta^{n+1} = \zeta^n + \frac{\Delta t\, N_f k_B}{Q}(T - T_0) + \sigma\sqrt{\Delta t}\, W - \frac{\Delta t\,\sigma^2}{4Q}(\zeta^{n+1} + \zeta^n), \\
&\tilde{p}_i^{\,n+1} = \vec{p}_i^{\,n+1/2}\exp\left(-\frac{\Delta t}{2}\zeta^{n+1}\right), \\
&\vec{r}_i^{\,n+1} = \vec{r}_i^{\,n+1/2} + \frac{\Delta t}{2m_i}\tilde{p}_i^{\,n+1}, \\
&\vec{p}_i^{\,n+1} = \tilde{p}_i^{\,n+1} + \frac{\Delta t}{2}\vec{f}_i^{\,n+1}.
\end{aligned} \tag{4.208}$$

The fourth equation above is implicit in nature and therefore iterative, but it is linear in ζ and converges quickly with low computational cost. The rate at which these equations converge to the canonical ensemble is determined by the parameter σ. Too large a value will affect the trajectories of the atoms and introduce greater uncertainty in the determination of time dependent properties, hence the concept of a "gentle" thermostat, with relatively small values of this parameter, to retain these properties. Leimkuhler *et al.* have shown that this simple algorithm is as effective as the Nosè-Hoover chains in maintaining a canonical ensemble and rather simpler to implement. But note that there is no conserved variable for this scheme, only a conservation of the system energy within the fluctuations of the thermostat.

4.8.9 Verifying the Canonical Ensemble

Given the efforts made in the algorithms described above to establish the canonical ensemble and the hidden possibility of failure it would be helpful to know that this objective has been achieved in a molecular dynamics simulation. As a bare minimum, the simulation should achieve and maintain the target temperature T_0, but this alone does not guarantee a proper Boltzmann distribution of the atomic and thermostat velocities.

One thing that can be done during the simulation is to sample the velocities and show that the resulting distributions are Boltzmann-like. A plot of the simulated and theoretical distributions on the same graph is particularly revealing. For atomic velocities, the Cartesian components of each atom, scaled by the square root of its mass, can be sampled since this is expected to produce a Gaussian distribution:

$$P\left(u_i^x\right) du_i^x = \left(\frac{1}{2\pi k_B T_0}\right)^{1/2} \exp\left(-\frac{\left(u_i^x\right)^2}{2 k_B T_0}\right) du_i^x, \tag{4.209}$$

where

$$u_i^x = m_i^{1/2} v_i^x. \tag{4.210}$$

The final distribution is obtained as an average over all N atoms in the system and all M sampled time points taken during the simulation. If all three dimensions of the system are equivalent, the average may be taken over all coordinates as well. Similarly, for the thermostat variable we have

$$P(\zeta)\, d\zeta = \left(\frac{Q}{2\pi k_B T_0}\right)^{1/2} \exp\left(-\frac{Q\zeta^2}{2 k_B T_0}\right) d\zeta. \tag{4.211}$$

(It is hopefully unnecessary to remark that (4.209) and (4.211) sample both positive and negative values of the argument and that the expected Gaussian plot is an even function.) Incidentally, if Nosè-Hoover chains are in use, (4.211) can be also adapted

for the higher thermostats.

Unfortunately, this approach is revealing only after the simulation has been completed and an approach that shows up problems earlier than this is desirable. For this purpose, the determination of the moments of the Gaussian function can be useful [4]. Gaussian moments, M_u^n, are defined in the following way (for a one dimensional Gaussian that is even about the origin, which is all we need here):

$$M_u^n = \left(\frac{1}{2\pi\sigma^2}\right)^{1/2} \int_{-\infty}^{\infty} u^n \exp\left(-\frac{u^2}{2\sigma^2}\right) d\,u, \tag{4.212}$$

where n is the *order* of the moment and u is the variable defining the Gaussian. σ^2 is the variance of the Gaussian function. In the case of (4.209) we have $\sigma^2 = k_B T_0$ and for (4.211) we have $\sigma^2 = k_B T_0 / Q.$ In the context of statistical mechanics we might also note that

$$M_u^n \equiv \langle u^n \rangle, \tag{4.213}$$

which is the ensemble average of u^n.

Since the Gaussian is an even function, it is readily apparent that $M_u^n = 0$ whenever n is odd. For n even (i.e. $n = 2\mathrm{m}$) we have

$$M_u^{2\mathrm{m}} = \sigma^{2m}(2m-1)!!, \tag{4.214}$$

where the double factorial (!!) indicates a product of all odd integers up to the value of the number within the brackets. Examples of explicit moments are as follows:

$$\begin{aligned} \langle u^2 \rangle &= \sigma^2, \\ \langle u^4 \rangle &= 3\sigma^4, \\ \langle u^6 \rangle &= 15\sigma^6, \\ \langle u^8 \rangle &= 105\sigma^8, \\ \langle u^{10} \rangle &= 945\sigma^{10}. \end{aligned} \tag{4.215}$$

Now, as we have noted above σ^2 is defined by the temperature and is a known quantity for the simulated system. So the calculation of $\langle u^2 \rangle,$ where u is an atomic velocity component or a thermostat variable, should return the value of the target system temperature. The inability of a simulation to maintain this quantity should immediately indicate (serious!) problems.

But what of the higher moments? For these it is convenient to define another

parameter, g_u^m :

$$g_u^m = \frac{\langle u^{2m} \rangle}{\langle u^2 \rangle^m (2m-1)!!} - 1. \tag{4.216}$$

It is evident from this definition that when the distribution function is truly Gaussian, then g_u^m should be zero, no matter what the value of m. It is therefore proposed that, during the course of a simulation, running averages of the quantities $\langle u^2 \rangle$ and g_u^m, up to some convenient value of m should be maintained, where u is any of the components of the atomic velocity (defined as $m_i^{1/2} v_i^x$ *etc*) or a thermostat variable ζ. The failure of any of these quantities to acquire and hold the expected value can be taken as an indication that the Boltzmann distribution is not being properly sampled.

4.8.10 Comments on Rotational Motion and Thermostats and Barostats

The preceding discussion of algorithms for canonical and isothermal and isobaric ensembles made no mention of the effect these have on rotational motion. We address this question here.

Firstly, since rotational motion clearly contributes to the kinetic energy of the system, it follows that it must also be thermostatted in the same way as the translational motion. There is no difficulty here, the rotational kinetic energy must be added to the translational and the system temperature calculated from this – allowing for the additional degrees of freedom associated with the rotational motion. Beyond that, there is no further change necessary to adapt the integration schemes presented.

Secondly, the effect of a barostat is to adjust the system volume to enable equilibration to the desired pressure. The scaling of the volume has no direct bearing or impact on the rotational motion and consequently no modifications are necessary to the integration algorithms on that account. There are however some issues concerning the calculation of the pressure in rigid molecule systems. For example the rotational kinetic energy does not contribute to the system pressure and there are corrections to the virial arising from the molecular rigidity. These matters are dealt with in some detail in Chapter 7.

Chapter 5

The Calculation of Forces and Torques

5.1 Introduction

At the heart of every molecular dynamics simulation is the calculation of forces on the system's constituent atoms. It is fair to say that in most circumstances this represents the major computational cost of a simulation. As was explained in chapter 1, the forces acting in a simulation are, for the most part, derived from empirical potentials, which have a wide variety of forms that accord with chemical intuition about how atoms interact with each other. We shall see numerous examples of empirical potentials in this chapter. These mostly have simple analytical forms, but some have forms which defy simple analytical expression and are (or can be presented) in tabular form.

Empirical potentials are generally categorised as intra-molecular or inter-molecular. The former refers to potentials that define the structure and internal dynamics of molecules and the latter define interactions between atoms that are not directly connected through the molecular structure. The definitions overlap somewhat, as the same empirical potentials can sometimes be used in both contexts. Inter-molecular interactions are often referred to as non-bonded interactions. They can be further categorised as either short-ranged or long-ranged. Short-ranged potentials decay in strength rapidly and exert their influence over distances that are only a few multiples of the nearest-neighbour inter-atomic distances. Long-ranged potentials are usually electrostatic in nature and can be practically infinite in range. They require special treatment and for this reason are dealt with separately in chapter 6.

As we saw in chapters 1 and 2, the interactions occurring between atoms in a molecular system are described by the system configuration energy function, which we shall write as $\Phi(\vec{r}^N)$. It is a function of the coordinates of every atom, $(1,\ldots,N)$, in the system and contains all the empirical potential functions required to specify each kind of interaction. The net force acting on any one of the atoms, $\vec{f}_j$, is then expressed as a derivative of the form

$$\vec{f}_j = -\frac{\partial}{\partial \vec{r}_j}\Phi(\vec{r}^N), \tag{5.1}$$

or equivalently

$$f_j^\alpha = -\frac{\partial}{\partial r_j^\alpha}\Phi(\vec{r}^N), \qquad \alpha = x, y, z \tag{5.2}$$

which refers to the individual Cartesian components of the force. Such expressions are

easy to write, but hide the sometimes great complexity of the system configuration energy and the often horrendous task of performing the differentiation. It is the task of this chapter to break this problem down into manageable pieces. A typical, but not exceptional, configuration energy function is presented in equation (5.3), which contains bonds, bond angles, dihedral angles, van der Waals pair interactions and electrostatic interactions. The details will be specified later. This would be suitable for simulating biological systems (for example). In addition we may require so-called three- and four- body interactions (typically required for amorphous systems) , density dependent potentials (for metals and covalent systems) and restraint bonds (to constrain large structural changes). We shall deal with these possibilities in what follows.

$$\begin{aligned}\Phi(\vec{r}^N) = &\sum_{b=1}^{N_{bonds}} \phi_b^{bond}(r_{jk}) + \sum_{a=1}^{N_{angles}} \phi_a^{angle}(\theta_{jkl}) + \sum_{d=1}^{N_{diheds}} \phi_d^{dihed}(\phi_{jklm}) \\ &+ \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \phi_{ij}^{vdw}(\vec{r}_i, \vec{r}_j) + \frac{1}{4\pi\epsilon_0} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \frac{q_i q_j}{r_{ij}}\end{aligned} \tag{5.3}$$

We should also note in passing that the calculation of forces in a molecular dynamics simulation is intimately related to the calculation of the pressure by means of the virial (see section 3.5.3 in chapter 3). However we shall not dwell on this topic here, but deal with it separately in chapter 7 of this book.

5.2 Intra-molecular Forces

We shall deal with the various types of intra-molecular interactions in turn, starting with the simplest, which are the bond forces and progressing to the most difficult, which are the dihedral and inversion potentials.

5.2.1 Bond Forces

The contribution of chemical bonds to the system configuration energy can be written as

$$\Phi(\vec{r}^N) = \sum_{b=1}^{N_{bonds}} \phi_b^{bond}(r_{jk}), \tag{5.4}$$

where ϕ_b^{bond} represents the analytical expression for the $b'th$ bond, r_{jk} is the distance between the two atoms, j and k, specific to bond with index b, i.e.

$$r_{jk} = |\vec{r}_{jk}| \quad \text{with} \quad \vec{r}_{jk} = \vec{r}_k - \vec{r}_j \tag{5.5}$$

and n_{bonds} is the number of bonds in the system as a whole. (Note that a given atom may participate in more than one bond, which the form (5.4) allows for.) Formally, the force acting on an arbitrary atom n due to the system configuration energy (5.4) is

$$\begin{aligned}\vec{f}_n^{bond} &= -\frac{\partial}{\partial \vec{r}_n}\sum_{b=1}^{N_{bonds}} \phi_b^{bond}(r_{jk}),\\ &= -\sum_{b=1}^{N_{bonds}} \frac{\partial \phi_b^{bond}(r_{jk})}{\partial r_{jk}}\frac{\partial r_{jk}}{\partial \vec{r}_n}.\end{aligned} \tag{5.6}$$

We may write (5.6) as

$$\vec{f}_n^{bond} = -\sum_{b=1}^{N_{bonds}} \phi'^{bond}_b(r_{jk})(\delta_{nk}-\delta_{nj})\frac{\vec{r}_{jk}}{r_{jk}}, \tag{5.7}$$

where we have used the chain rule and the Kronecker delta function[18], which is defined as

$$\delta_{ij} = \begin{cases} 0 & if \quad i \neq j, \\ 1 & if \quad i = j, \end{cases} \tag{5.8}$$

to indicate that the derivatives appearing in the brackets can only be non-zero if n is one of the atoms, j or k, associated with the bond b. $\phi'^{bond}_b(r_{jk})$ is the derivative of $\phi_b^{bond}(r_{jk})$ with respect to the argument r_{jk}. Note that (5.7) is valid for all bond potentials, though the form of $\phi'(r_{jk})$ may of course be different in each case. For each bond in (5.7) we can split off two cases: when $n=j$ or $n=k$, which separates the *contribution* of the bond b to the force on atoms j and k and we write this as follows:

$$\begin{aligned}\vec{f}_j^{bond} &\leftarrow \phi'^{bond}_b(r_{jk})\frac{\vec{r}_{jk}}{r_{jk}},\\ \vec{f}_k^{bond} &\leftarrow -\phi'^{bond}_b(r_{jk})\frac{\vec{r}_{jk}}{r_{jk}},\end{aligned} \tag{5.9}$$

Equations (5.9) show that the forces on the atoms j and k forming the bond are equal and opposite, as we should expect from Newton's third law.

Bonds within molecules are generally specified by simple functions and depend only on the distance between the two bonded atoms. We present two example empirical potentials that are typical of the wide range of possible bond potentials. The most simple form in common use is the harmonic function

$$\phi^{harm}(r_{jk}) = \frac{\kappa}{2}(r_{jk}-r_0)^2, \tag{5.10}$$

where κ is the so-called force constant and r_0 is the equilibrium bond length. Another form is the anharmonic Morse potential

$$\phi^{morse}(r_{jk}) = D_e(\{1-\exp[-\alpha(r_{jk}-r_0)]\}^2-1), \tag{5.11}$$

18 For more information on the Kronecker delta function, see Appendix 3.

where D_e is the Morse bond energy and r_0 is the equilibrium bond length. Parameter α controls the stiffness of the potential function for a given bond energy. For these potentials we have

$$\phi'^{harm}(r_{jk}) = \kappa(r_{jk} - r_0) \quad \text{(Harmonic)} \tag{5.12}$$

and

$$\phi'^{morse}(r_{jk}) = 2\alpha D_e[1 - \exp\{-\alpha(r_{jk} - r_0)\}]\exp\{-\alpha(r_{jk} - r_0)\}. \quad \text{(Morse)} \tag{5.13}$$

from which the bond forces may be readily calculated using (5.9). The same basic approach applies to all bond potentials.

We conclude by noting that restraint bonds, such as are used in bio-molecular simulations are essentially bonds of the kind described here. However these bonds are artificial. They do not correspond to classical chemical bonds and often have large equilibrium distances by normal standards. They are used to restrain the geometric fluctuation of the molecular structure, often in the hope that the result will conform to known large-scale molecular behaviour. Mathematically they are no different from what we have discussed already.

5.2.2 Bond Angle Forces

Here we are referring to the angle defined by two bonds, the so-called valence angle. The contribution to the system configuration energy made by such angles is

$$\Phi(\{\vec{r}_i\}) = \sum_{a=1}^{N_{angles}} \phi_a^{angle}(\theta_{jkl}), \tag{5.14}$$

where ϕ_a^{angle} is the angle potential for the bond angle indexed a and n_{angles} is the number of bond angles in the whole system. θ_{jkl} is the *bond angle* defined by the two bonds between atoms j and k and between k and l, as in figure 5.1.

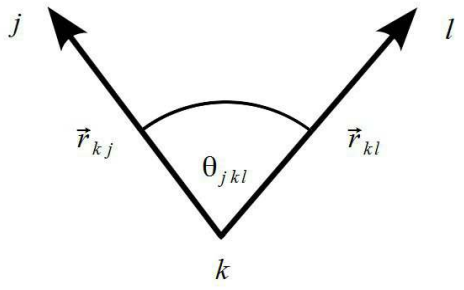

Figure 5.1: The Bond Angle θ_{jkl}

The atoms j, k and l are specific to the bond angle with the index a. The bonds are defined by the vectors $\vec{r}_{kj}$ and $\vec{r}_{kl}$, in the same way as equation (5.5), which

means the angle θ_{jkl} can be determined from the scalar product of these vectors i.e.

$$\theta_{jkl} = \arccos\{u_{jkl}\} \qquad \text{with} \qquad u_{jkl} = \left\{ \frac{\vec{r}_{kj} \cdot \vec{r}_{kl}}{r_{kj} r_{kl}} \right\}. \tag{5.15}$$

The force acting on an arbitrary atom n due to the angle potentials is given by

$$\begin{aligned} \vec{f}_n^{angle} &= -\frac{\partial}{\partial \vec{r}_n} \sum_{a=1}^{N_{angles}} \phi_a^{angle}(\theta_{jkl}), \\ &= -\sum_{a=1}^{N_{angles}} \frac{\partial \phi_a^{angle}(\theta_{jkl})}{\partial \theta_{jkl}} \frac{\partial \theta_{jkl}}{\partial u_{jkl}} \frac{\partial u_{jkl}}{\partial \vec{r}_n}, \\ &= \sum_{a=1}^{N_{angles}} \phi'^{angle}_a(\theta_{jkl}) \frac{1}{\sin\theta_{jkl}} \frac{\partial u_{jkl}}{\partial \vec{r}_n}, \end{aligned} \tag{5.16}$$

where we have used the chain rule to obtain the second line. The function $\phi'^{angle}_a(\theta_{jkl})$ is the derivative of $\phi_a^{angle}(\theta_{jkl})$ with respect to θ_{jkl}. Obviously, the third derivative of equation (5.16) can only return a non-zero result if u_{jkl} contains a dependence on the vector $\vec{r}_n$. This can be seen in the following expansion, which is obtained by differentiating (5.15).

$$\begin{aligned} \frac{\partial u_{jkl}}{\partial \vec{r}_n} &= (\delta_{nj} - \delta_{nk}) \frac{\vec{r}_{kl}}{r_{kj} r_{kl}} + (\delta_{nl} - \delta_{nk}) \frac{\vec{r}_{kj}}{r_{kj} r_{kl}} \\ &\quad - \cos\theta_{jkl} \left\{ (\delta_{nj} - \delta_{nk}) \frac{\vec{r}_{kj}}{r_{kj}^2} + (\delta_{nl} - \delta_{nk}) \frac{\vec{r}_{kl}}{r_{kl}^2} \right\}. \end{aligned} \tag{5.17}$$

Inserting (5.17) into (5.16) gives

$$\begin{aligned} \vec{f}_n^{angle} = \sum_{a=1}^{N_{angles}} \frac{\phi'^{angle}_a(\theta_{jkl})}{\sin\theta_{jkl}} \Bigg[& (\delta_{nj} - \delta_{nk}) \frac{\vec{r}_{kl}}{r_{kj} r_{kl}} + (\delta_{nl} - \delta_{nk}) \frac{\vec{r}_{kj}}{r_{kj} r_{kl}} \\ & - \cos\theta_{jkl} \left\{ (\delta_{nj} - \delta_{nk}) \frac{\vec{r}_{kj}}{r_{kj}^2} + (\delta_{nl} - \delta_{nk}) \frac{\vec{r}_{kl}}{r_{kl}^2} \right\} \Bigg]. \end{aligned} \tag{5.18}$$

This is the most general formula for the bond angle forces. It differs for different bond angle potentials only in so far as the term $\phi'^{angle}_a(\theta_{jkl})$ may be different in each case. For each bond angle a, we may now split off the contribution to the atomic force on each of the atoms defining the angle, which are j, k and l.

$$\begin{aligned} \vec{f}_k^{angle} &\leftarrow -\frac{\phi'^{angle}_a(\theta_{jkl})}{\sin\theta_{jkl}} \left[\frac{\vec{r}_{kl}}{r_{kj} r_{kl}} + \frac{\vec{r}_{kj}}{r_{kj} r_{kl}} - \cos\theta_{jkl} \left\{ \frac{\vec{r}_{kj}}{r_{kj}^2} + \frac{\vec{r}_{kl}}{r_{kl}^2} \right\} \right], \\ \vec{f}_j^{angle} &\leftarrow \frac{\phi'^{angle}_a(\theta_{jkl})}{\sin\theta_{jkl}} \left[\frac{\vec{r}_{kl}}{r_{kj} r_{kl}} - \cos\theta_{jkl} \left\{ \frac{\vec{r}_{kj}}{r_{kj}^2} \right\} \right], \end{aligned} \tag{5.19}$$

$$\vec{f}_l^{angle} \leftarrow \frac{\phi'^{angle}_a(\theta_{jkl})}{\sin\theta_{jkl}}\left[\frac{\vec{r}_{kj}}{r_{kj}r_{kl}} - \cos\theta_{jkl}\left\{\frac{\vec{r}_{kl}}{r_{kl}^2}\right\}\right].$$

These formulae show that the net force on atom k is the sum of the forces acting on atoms j and l, with the vector direction reversed. Thus the net sum of all atomic forces in (5.19) is zero, which underlines the general fact that the sum of all the intra-molecular forces must always be zero and cannot generate motion of the molecule as a whole.

An unfortunate property of (5.19) derives from the presence of the term $\sin\theta_{jkl}$ in the denominator. This inevitably means that when $\theta_{jkl}=0^\circ$ or $\theta_{jkl}=180^\circ$ the force is likely to become indeterminate. It is therefore unwise to define bond angles where the equilibrium angle is close to either 0° or 180° in case this unfortunate circumstance arises. But it can also arise if the bond angle potential is weak and the angle fluctuates widely. However, it can be shown that the terms within the square brackets right of (5.19) are zero when $\theta_{jkl}=0^\circ$ or $\theta_{jkl}=180^\circ$, which suggests that setting the force to zero when close to the problem angles may be a sensible option.

An example of bond angle potential is the harmonic function:

$$\phi^{angle}(\theta_{jkl}) = \frac{\kappa}{2}(\theta_{jkl}-\theta_0)^2 \quad \text{with} \quad \phi'^{angle}(\theta_{jkl}) = \kappa(\theta_{jkl}-\theta_0), \tag{5.20}$$

where κ is a force constant and θ_0 is the equilibrium bond angle. This simple form is common. Another is the harmonic cosine function:

$$\phi^{angle}(\theta_{jkl}) = \frac{\kappa}{2}(\cos\theta_{jkl}-\cos\theta_0)^2 \quad \text{with} \quad \phi'^{angle}(\theta_{jkl}) = -\kappa(\cos\theta_{jkl}-\cos\theta_0)\sin\theta_{jkl}. \tag{5.21}$$

Notice that the derivative of the harmonic cosine function contains $\sin\theta_{jkl}$, which cancels the $\sin\theta_{jkl}$ in the denominator of (5.19). If this cancellation is done the force for this potential function is well-behaved when $\theta_{jkl}=180^\circ$. For this reason many angle potentials are based on cosine expressions.

5.2.3 Dihedral Angle Forces

Dihedral angles are defined by the locations of four atoms, j, k, l and m, which are connected via three bond vectors, $\vec{r}_{jk}$, $\vec{r}_{kl}$, and $\vec{r}_{lm}$, in the manner shown in figure 5.2. Collectively all dihedrals define a system configuration energy of the form

$$\Phi(\vec{r}^N) = \sum_{d=1}^{N_{diheds}} \phi_d^{dihed}(\phi_{jklm}), \tag{5.22}$$

where ϕ_d^{dihed} is one of N_{diheds} dihedral angles defined for the system. Atoms j, k, l and m, are specific to the dihedral indexed as d. The angle ϕ_{jklm} is the dihedral angle, which is defined as the angle between the plane containing atoms j, k and l, and the plane containing atoms k, l and m, as in figure 5.2. Formally this is

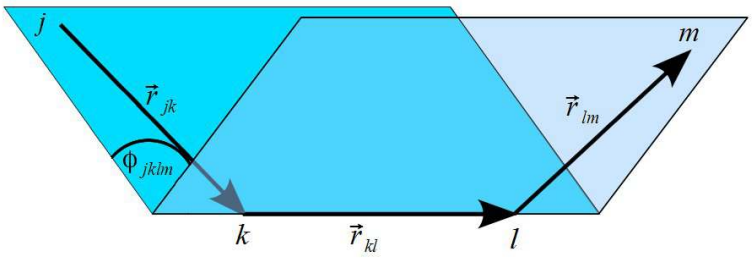

Figure 5.2: The Dihedral Angle ϕ_{jklm}

obtained from the scalar product of vectors perpendicular to these two planes i.e.

$$\phi_{jklm} = \arccos\{u_{jklm}\}, \tag{5.23}$$

with

$$u_{jklm} = \vec{e}_{jkl} \cdot \vec{e}_{klm} \tag{5.24}$$

and *unit* vectors $\vec{e}_{jkl}$ and $\vec{e}_{klm}$ defined as

$$\vec{e}_{jkl} = \frac{\vec{r}_{jk} \times \vec{r}_{kl}}{|\vec{r}_{jk} \times \vec{r}_{kl}|} \quad \text{and} \quad \vec{e}_{klm} = \frac{\vec{r}_{kl} \times \vec{r}_{lm}}{|\vec{r}_{kl} \times \vec{r}_{lm}|}, \tag{5.25}$$

Following our standard procedures, the force on an arbitrary atom n due to the configuration energy (5.22) is

$$\begin{aligned} \vec{f}_n^{dihed} &= -\frac{\partial}{\partial \vec{r}_n} \sum_{d=1}^{N_{diheds}} \phi_d^{dihed}(\phi_{jklm}), \\ &= -\sum_{d=1}^{N_{diheds}} \frac{\partial \phi_d^{dihed}(\phi_{jklm})}{\partial \phi_{jklm}} \frac{\partial \phi_{jklm}}{\partial u_{jklm}} \frac{\partial u_{jklm}}{\partial \vec{r}_n}, \\ &= \sum_{d=1}^{N_{diheds}} \phi'^{dihed}_d(\phi_{jklm}) \frac{1}{\sin \phi_{jklm}} \frac{\partial u_{jklm}}{\partial \vec{r}_n}. \end{aligned} \tag{5.26}$$

Equation (5.26) follows from the chain rule. The function $\phi'^{dihed}_d(\phi_{jklm})$ is the first derivative of $\phi_d^{dihed}(\phi_{jklm})$ with respect to the argument ϕ_{jklm}.

The derivative of u_{jklm} with respect to $\vec{r}_n$ can be written as

$$\frac{\partial u_{jklm}}{\partial \vec{r}_n} = \frac{\{\vec{e}_{klm} - \cos\phi_{jklm} \vec{e}_{jkl}\}}{|\vec{r}_{jk} \times \vec{r}_{kl}|} \cdot \frac{\partial(\vec{r}_{jk} \times \vec{r}_{kl})}{\partial \vec{r}_n} + \frac{\{\vec{e}_{jkl} - \cos\phi_{jklm} \vec{e}_{klm}\}}{|\vec{r}_{kl} \times \vec{r}_{lm}|} \cdot \frac{\partial(\vec{r}_{kl} \times \vec{r}_{lm})}{\partial \vec{r}_n}. \tag{5.27}$$

Once again we note that not every derivative survives; one of the atoms j, k, l or m, must be the atom n. The derivatives of the vector products appearing in (5.27) are tedious to derive, though this need only be done once, as it is the same for all dihedral potentials. For an arbitrary vector $\vec{a}$, it can be shown that

$$\vec{a}\cdot\frac{\partial(\vec{r}_{jk}\times\vec{r}_{kl})}{\partial\vec{r}_n}=\vec{a}\times[(\delta_{ln}-\delta_{kn})\vec{r}_{jk}-(\delta_{kn}-\delta_{jn})\vec{r}_{kl}],$$
$$\vec{a}\cdot\frac{\partial(\vec{r}_{kl}\times\vec{r}_{lm})}{\partial\vec{r}_n}=\vec{a}\times[(\delta_{mn}-\delta_{ln})\vec{r}_{kl}-(\delta_{ln}-\delta_{kn})\vec{r}_{lm}]. \tag{5.28}$$

Hence the force on atom n can be written as

$$\vec{f}_n^{dihed}=\sum_{d=1}^{N_{diheds}}\frac{\phi'^{angle}_d(\phi_{jklm})}{\sin\phi_{jklm}}\left(\frac{\{\vec{e}_{klm}-\cos\phi_{jklm}\vec{e}_{jkl}\}}{|\vec{r}_{jk}\times\vec{r}_{kl}|}\times[(\delta_{ln}-\delta_{kn})\vec{r}_{jk}-(\delta_{kn}-\delta_{jn})\vec{r}_{kl}]\right.$$
$$\left.+\frac{\{\vec{e}_{jkl}-\cos\phi_{jklm}\vec{e}_{klm}\}}{|\vec{r}_{kl}\times\vec{r}_{lm}|}\times[(\delta_{mn}-\delta_{ln})\vec{r}_{kl}-(\delta_{ln}-\delta_{kn})\vec{r}_{lm}]\right). \tag{5.29}$$

This is the general result. We may now consider the specific cases where atom n is one of j, k, l or m, and obtain the contributions to the forces on these atoms due to one dihedral potential $\phi_d^{dihed}(\phi_{jklm})$ in the form (5.30).

$$\vec{f}_j^{dihed}\leftarrow\frac{\phi'^{angle}_d(\phi_{jklm})}{\sin\phi_{jklm}}\left(\frac{\{\vec{e}_{klm}-\cos\phi_{jklm}\vec{e}_{jkl}\}}{|\vec{r}_{jk}\times\vec{r}_{kl}|}\times\vec{r}_{kl}\right),$$
$$\vec{f}_k^{dihed}\leftarrow-\frac{\phi'^{angle}_d(\phi_{jklm})}{\sin\phi_{jklm}}\left(\frac{\{\vec{e}_{klm}-\cos\phi_{jklm}\vec{e}_{jkl}\}}{|\vec{r}_{jk}\times\vec{r}_{kl}|}\times(\vec{r}_{jk}+\vec{r}_{kl})-\frac{\{\vec{e}_{jkl}-\cos\phi_{jklm}\vec{e}_{klm}\}}{|\vec{r}_{kl}\times\vec{r}_{lm}|}\times\vec{r}_{lm}\right),$$
$$\vec{f}_l^{dihed}\leftarrow-\frac{\phi'^{angle}_d(\phi_{jklm})}{\sin\phi_{jklm}}\left(\frac{\{\vec{e}_{jkl}-\cos\phi_{jklm}\vec{e}_{klm}\}}{|\vec{r}_{kl}\times\vec{r}_{lm}|}\times(\vec{r}_{kl}+\vec{r}_{lm})-\frac{\{\vec{e}_{klm}-\cos\phi_{jklm}\vec{e}_{jkl}\}}{|\vec{r}_{jk}\times\vec{r}_{kl}|}\times\vec{r}_{jk}\right), \tag{5.30}$$
$$\vec{f}_m^{dihed}\leftarrow\frac{\phi'^{angle}_d(\phi_{jklm})}{\sin\phi_{jklm}}\left(\frac{\{\vec{e}_{jkl}-\cos\phi_{jklm}\vec{e}_{klm}\}}{|\vec{r}_{kl}\times\vec{r}_{lm}|}\times\vec{r}_{kl}\right).$$

The sum of these forces for all the atoms involved in the dihedral is zero, which means the dihedral angle forces do not produce a net force on the molecule.

It should also be noted that, just as with the bond angle forces, the presence of the $\sin\phi_{jklm}$ function in the denominator of (5.30) can lead to indeterminate forces when the dihedral angle is 0° or 180°, which unfortunately is a likely possibility for dihedral angles. This can be alleviated if the derivative $\phi'^{dihed}_d(\phi_{jklm})$ contains the same sine function and can cancel the denominator. In which case the forces sensibly go to zero at the problem angles, since the terms in the large brackets right of (5.30) tend to zero as these angles are approached. If this cancellation cannot be done, it is a sensible alternative option to set the forces to zero close the problem angles.

A typical dihedral angle potential is the triple cosine:

$$\phi^{dihed}(\phi_{jklm})=A_1(1+\cos\phi_{jklm})+A_2(1-\cos2\phi_{jklm})+A_3(1+\cos3\phi_{jklm}), \tag{5.31}$$

where A_1, A_2 and A_3 are constants. The derivative is

$$\phi'^{dihed}(\phi_{jklm}) = \{-A_1 + 4A_2\cos\phi_{jklm} - 3A_3(4\cos^2\phi_{jklm} - 1)\}\sin\phi_{jklm}, \quad (5.32)$$

which is a form that cancels the $\sin\phi_{jklm}$ in the denominator of (5.30) and so is well-behaved at $\phi_{jklm} = 0^\circ$ and $\phi_{jklm} = 180^\circ$. Forms like this are used to define multiple energy barriers to free rotation about the dihedral angle – a property common to many covalent molecules.

A particularly important dihedral potential is the Ryckaert-Bellemans potential [21] for hydrocarbon chains:

$$\phi^{dihed}(\phi_{jklm}) = A\left(a_0 + \sum_{i=1}^{5} a_i \cos^i \phi_{jklm}\right), \quad (5.33)$$

where A and a_0 to a_5 are constants. The derivative is

$$\phi'^{dihed}(\phi_{jklm}) = -A\sin\phi_{jklm}\left(\sum_{i=1}^{5} i\, a_i \cos^{i-1}\phi_{jklm}\right), \quad (5.34)$$

which is another well-behaved function for simulation purposes.

Incidentally, the constants appearing in the above potential functions can be obtained from experimental studies of rotation barriers or from first-principle quantum chemistry calculations.

It is worth noting that dihedral angle potentials are often defined with an additional potential term that depends on the distance between the atoms identified as j and m in figure 5.2 (the so-called 1-4 term). Such an addition is intended to modify or augment the non-bonding interaction between these atoms. This possibility is clearly not dealt with in the above treatment. However it suffices to note that the 1-4 term can be handled as a "bond" between these atoms and dealt with as described in section 5.2.1 .

5.2.4 Inversion Angle Forces

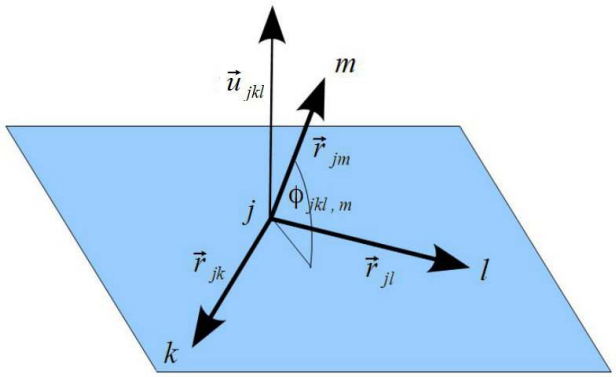

Figure 5.3: The Inversion Angle $\phi_{jkl,m}$

Like dihedral angles, inversions are defined by the locations of four atoms, j, k, l and m, but in this case the atoms are connected via three bond vectors, $\vec{r}_{jk}$, $\vec{r}_{jl}$, and $\vec{r}_{jm}$, as shown in figure 5.3. The purpose of the inversion potential is to hold each of the four atoms in a particular position relative to the other three and is often used to hold atoms in a plane. The inversions define a system configuration energy of the form

$$\Phi(\vec{r}^{N}) = \sum_{d=1}^{N_{invers}} \phi_d^{invers}(\phi_{jkl,m}, \phi_{jlm,k}, \phi_{jmk,l}), \tag{5.35}$$

where ϕ_d^{invers} is one of the N_{invers} inversion angles in the system. Atoms j, k, l and m, are specific to the inversion indexed as d. Angles $\phi_{jkl,m}$, $\phi_{jlm,k}$ and $\phi_{jmk,l}$ are the inversion angles. As shown in figure 5.3 the inversion angle $\phi_{jkl,m}$ is defined as the angle the bond vector $\vec{r}_{jm}$ makes with the plane containing atoms j, k and l. Angles $\phi_{jlm,k}$ and $\phi_{jmk,l}$ are defined similarly. It will be apparent that the inversion angles are defined according to which three atoms are chosen to describe the plane and which atom is deemed to be out of the plane. One atom, in this case j, is common to all the bonds and is therefore always located in the defining plane. A single specification of an inversion potential therefore defines all three inversion angles at once.

The inversion angles are obtained from the bond vectors as follows:

$$\phi_{jkl,m} = \arcsin\{u_{jkl,m}\}, \qquad \phi_{jlm,k} = \arcsin\{u_{jlm,k}\}, \qquad \phi_{jmk,l} = \arcsin\{u_{jmk,l}\}, \tag{5.36}$$

where

$$u_{jkl,m} = \vec{u}_{jkl} \cdot \vec{e}_{jm}, \qquad u_{jlm,k} = \vec{u}_{jlm} \cdot \vec{e}_{jk}, \qquad u_{jmk,l} = \vec{u}_{jmk} \cdot \vec{e}_{jl}, \tag{5.37}$$

and

$$\vec{u}_{jkl} = \frac{\vec{r}_{jk} \times \vec{r}_{jl}}{|\vec{r}_{jk} \times \vec{r}_{jl}|}, \qquad \vec{u}_{jlm} = \frac{\vec{r}_{jl} \times \vec{r}_{jm}}{|\vec{r}_{jl} \times \vec{r}_{jm}|}, \qquad \vec{u}_{jmk} = \frac{\vec{r}_{jm} \times \vec{r}_{jk}}{|\vec{r}_{jm} \times \vec{r}_{jk}|} \tag{5.38}$$

and

$$\vec{e}_{jm} = \frac{\vec{r}_{jm}}{|\vec{r}_{jm}|}, \qquad \vec{e}_{jk} = \frac{\vec{r}_{jk}}{|\vec{r}_{jk}|}, \qquad \vec{e}_{jm} = \frac{\vec{r}_{jl}}{|\vec{r}_{jl}|}. \tag{5.39}$$

It will be seen from these formulae and figure 5.3, that $\vec{u}_{jkl}$ for example, is a unit vector perpendicular to the plane of the atoms j, k and l and $\vec{e}_{jm}$ is a unit vector directed along the bond $\vec{r}_{jm}$. In this example atom m is therefore deemed to be out of the plane of atoms j, k and l.

The force on an arbitrary atom, n, arising from the configuration energy (5.35) is

$$\begin{aligned}\vec{f}_n^{invers} &= -\frac{\partial}{\partial \vec{r}_n}\sum_{d=1}^{N_{invers}} \phi_d^{invers}\left(\phi_{jkl,m},\phi_{jlm,k},\phi_{jmk,l}\right),\\ &= -\sum_{d=1}^{N_{invers}}\left(\frac{\partial \phi_d^{invers}}{\partial \phi_{jkl,m}}\frac{\partial \phi_{jkl,m}}{\partial u_{jkl,m}}\frac{\partial u_{jkl,m}}{\partial \vec{r}_n}+\frac{\partial \phi_d^{invers}}{\partial \phi_{jlm,k}}\frac{\partial \phi_{jlm,k}}{\partial u_{jlm,k}}\frac{\partial u_{jlm,k}}{\partial \vec{r}_n}+\frac{\partial \phi_d^{invers}}{\partial \phi_{jmk,l}}\frac{\partial \phi_{jmk,l}}{\partial u_{jmk,l}}\frac{\partial u_{jmk,l}}{\partial \vec{r}_n}\right)\\ &= -\sum_{d=1}^{N_{invers}}\left(\frac{\phi'^{invers}_d\left(\phi_{jkl,m}\right)}{\cos\phi_{jkl,m}}\frac{\partial u_{jkl,m}}{\partial \vec{r}_n}+\frac{\phi'^{invers}_d\left(\phi_{jlm,k}\right)}{\cos\phi_{jlm,k}}\frac{\partial u_{jlm,k}}{\partial \vec{r}_n}+\frac{\phi'^{invers}_d\left(\phi_{jmk,l}\right)}{\cos\phi_{jmk,l}}\frac{\partial u_{jmk,l}}{\partial \vec{r}_n}\right).\end{aligned} \tag{5.40}$$

Equation (5.40) follows from the chain rule and the function $\phi'^{invers}_d(\phi_{jkl,m})$ is the first derivative of $\phi_d^{invers}(\phi_{jkl,m})$ with respect to angle $\phi_{jkl,m}$ and so on. The derivatives of $u_{jkl,m}$, $u_{jlm,k}$, and $u_{jmk,l}$ with respect to $\vec{r}_n$ are

$$\begin{aligned}\frac{\partial u_{jkl,m}}{\partial \vec{r}_n} &= \frac{\left\{\vec{e}_{jm}-\sin\phi_{jkl,m}\vec{u}_{jkl}\right\}}{|\vec{r}_{jk}\times\vec{r}_{jl}|}\cdot\frac{\partial(\vec{r}_{jk}\times\vec{r}_{jl})}{\partial \vec{r}_n}+\frac{\left\{\vec{u}_{jkl}-\sin\phi_{jkl,m}\vec{e}_{jm}\right\}}{|\vec{r}_{jm}|}\cdot\frac{\partial \vec{r}_{jm}}{\partial \vec{r}_n},\\ \frac{\partial u_{jlm,k}}{\partial \vec{r}_n} &= \frac{\left\{\vec{e}_{jk}-\sin\phi_{jlm,k}\vec{u}_{jlm}\right\}}{|\vec{r}_{jl}\times\vec{r}_{jm}|}\cdot\frac{\partial(\vec{r}_{jl}\times\vec{r}_{jm})}{\partial \vec{r}_n}+\frac{\left\{\vec{u}_{jlm}-\sin\phi_{jlm,k}\vec{e}_{jk}\right\}}{|\vec{r}_{jk}|}\cdot\frac{\partial \vec{r}_{jk}}{\partial \vec{r}_n},\\ \frac{\partial u_{jmk,l}}{\partial \vec{r}_n} &= \frac{\left\{\vec{e}_{jl}-\sin\phi_{jmk,l}\vec{u}_{jmk}\right\}}{|\vec{r}_{jm}\times\vec{r}_{jk}|}\cdot\frac{\partial(\vec{r}_{jm}\times\vec{r}_{jk})}{\partial \vec{r}_n}+\frac{\left\{\vec{u}_{jmk}-\sin\phi_{jmk,l}\vec{e}_{jl}\right\}}{|\vec{r}_{jl}|}\cdot\frac{\partial \vec{r}_{jl}}{\partial \vec{r}_n}.\end{aligned} \tag{5.41}$$

In order for the derivative to be non-zero n must be one of the atoms j, k, l or m. To proceed further we require the derivatives on the right. For an arbitrary vector $\vec{a}$ it can be shown that

$$\begin{aligned}\vec{a}\cdot\frac{\partial(\vec{r}_{jk}\times\vec{r}_{jl})}{\partial \vec{r}_n} &= \vec{a}\times[(\delta_{ln}-\delta_{jn})\vec{r}_{jk}-(\delta_{kn}-\delta_{jn})\vec{r}_{jl}], \qquad \vec{a}\cdot\frac{\partial \vec{r}_{jm}}{\partial \vec{r}_n} = (\delta_{mn}-\delta_{jn})\vec{a},\\ \vec{a}\cdot\frac{\partial(\vec{r}_{jl}\times\vec{r}_{jm})}{\partial \vec{r}_n} &= \vec{a}\times[(\delta_{mn}-\delta_{jn})\vec{r}_{jl}-(\delta_{ln}-\delta_{jn})\vec{r}_{jm}], \qquad \vec{a}\cdot\frac{\partial \vec{r}_{jk}}{\partial \vec{r}_n} = (\delta_{kn}-\delta_{jn})\vec{a},\\ \vec{a}\cdot\frac{\partial(\vec{r}_{jm}\times\vec{r}_{jk})}{\partial r_n} &= \vec{a}\times[(\delta_{kn}-\delta_{jn})\vec{r}_{jm}-(\delta_{mn}-\delta_{jn})\vec{r}_{jk}], \qquad \vec{a}\cdot\frac{\partial \vec{r}_{jl}}{\partial r_n} = (\delta_{ln}-\delta_{jn})\vec{a},\end{aligned} \tag{5.42}$$

The force on atom n can now be written as

$$\begin{aligned}\vec{f}_n^{invers} = -\sum_{d=1}^{N_{invers}}\Bigg(&\frac{\phi'^{invers}_d\left(\phi_{jkl,m}\right)}{\cos\phi_{jkl,m}}\Bigg\{\frac{\left\{\vec{e}_{jm}-\sin\phi_{jkl,m}\vec{u}_{jkl}\right\}}{|\vec{r}_{jk}\times\vec{r}_{jl}|}\times[(\delta_{ln}-\delta_{jn})\vec{r}_{jk}-(\delta_{kn}-\delta_{jn})\vec{r}_{jl}]\\ &+(\delta_{mn}-\delta_{jn})\frac{\left\{\vec{u}_{jkl}-\sin\phi_{jkl,m}\vec{e}_{jm}\right\}}{|\vec{r}_{jm}|}\Bigg\}+\\ &\frac{\phi'^{invers}_d\left(\phi_{jlm,k}\right)}{\cos\phi_{jlm,k}}\Bigg\{\frac{\left\{\vec{e}_{jk}-\sin\phi_{jlm,k}\vec{u}_{jlm}\right\}}{|\vec{r}_{jl}\times\vec{r}_{jm}|}\times[(\delta_{mn}-\delta_{jn})\vec{r}_{jl}-(\delta_{ln}-\delta_{jn})\vec{r}_{jm}]\\ &+(\delta_{kn}-\delta_{jn})\frac{\left\{\vec{u}_{jlm}-\sin\phi_{jlm,k}\vec{e}_{jk}\right\}}{|\vec{r}_{jk}|}\Bigg\}+\end{aligned} \tag{5.43}$$

$$\frac{\phi'^{invers}_d(\phi_{jmk,l})}{\cos\phi_{jmk,l}}\left\{\frac{\{\vec{e}_{jl}-\sin\phi_{jmk,l}\vec{u}_{jmk}\}}{|\vec{r}_{jm}\times\vec{r}_{jk}|}\times[(\delta_{kn}-\delta_{jn})\vec{r}_{jm}-(\delta_{mn}-\delta_{jn})\vec{r}_{jk}]\right.$$
$$\left.+(\delta_{ln}-\delta_{jn})\frac{\{\vec{u}_{jmk}-\sin\phi_{jmk,l}\vec{e}_{jl}\}}{|\vec{r}_{jl}|}\right\}\Bigg)$$

This general result can now be applied to specific cases where atom n is one of k, l or m, to obtain the contributions added to the forces on these atoms due to one inversion potential $\phi^{invers}_d(\phi_{jkl,m},\phi_{jlm,k},\phi_{jmk,l})$:

$$\begin{aligned}
\vec{f}^{invers}_k \leftarrow & \frac{\phi'^{invers}_d(\phi_{jkl,m})}{\cos\phi_{jkl,m}}\left(\frac{\{\vec{e}_{jm}-\sin\phi_{jkl,m}\vec{u}_{jkl}\}}{|\vec{r}_{jk}\times\vec{r}_{jl}|}\times\vec{r}_{jl}\right)-\frac{\phi'^{invers}_d(\phi_{jlm,k})}{\cos\phi_{jlm,k}}\left(\frac{\{\vec{u}_{jlm}-\sin\phi_{jlm,k}\vec{e}_{jk}\}}{|\vec{r}_{jk}|}\right)\\
& -\frac{\phi'^{invers}_d(\phi_{jmk,l})}{\cos\phi_{jmk,l}}\left(\frac{\{\vec{e}_{jl}-\sin\phi_{jmk,l}\vec{u}_{jmk}\}}{|\vec{r}_{jm}\times\vec{r}_{jk}|}\times\vec{r}_{jm}\right),\\
\vec{f}^{invers}_l \leftarrow & \frac{\phi'^{invers}_d(\phi_{jlm,k})}{\cos\phi_{jlm,k}}\left(\frac{\{\vec{e}_{jk}-\sin\phi_{jlm,k}\vec{u}_{jlm}\}}{|\vec{r}_{jl}\times\vec{r}_{jm}|}\times\vec{r}_{jm}\right)-\frac{\phi'^{invers}_d(\phi_{jmk,l})}{\cos\phi_{jmk,l}}\left(\frac{\{\vec{u}_{jmk}-\sin\phi_{jmk,l}\vec{e}_{jl}\}}{|\vec{r}_{jl}|}\right)\\
& -\frac{\phi'^{invers}_d(\phi_{jkl,m})}{\cos\phi_{jkl,m}}\left(\frac{\{\vec{e}_{jm}-\sin\phi_{jkl,m}\vec{u}_{jkl}\}}{|\vec{r}_{jk}\times\vec{r}_{jl}|}\times\vec{r}_{jk}\right),\\
\vec{f}^{invers}_m \leftarrow & \frac{\phi'^{invers}_d(\phi_{jmk,l})}{\cos\phi_{jmk,l}}\left(\frac{\{\vec{e}_{jl}-\sin\phi_{jmk,l}\vec{u}_{jmk}\}}{|\vec{r}_{jm}\times\vec{r}_{jk}|}\times\vec{r}_{jk}\right)-\frac{\phi'^{invers}_d(\phi_{jkl,m})}{\cos\phi_{jkl,m}}\left(\frac{\{\vec{u}_{jkl}-\sin\phi_{jkl,m}\vec{e}_{jm}\}}{|\vec{r}_{jm}|}\right)\\
& -\frac{\phi'^{invers}_d(\phi_{jlm,k})}{\cos\phi_{jlm,k}}\left(\frac{\{\vec{e}_{jk}-\sin\phi_{jlm,k}\vec{u}_{jlm}\}}{|\vec{r}_{jl}\times\vec{r}_{jm}|}\times\vec{r}_{jl}\right),
\end{aligned}\qquad (5.44)$$

It is easy to show that the force contribution to atom j is the negative sum of all the contributions shown in (5.44) and that the net force on all four atoms j, k, l and m is zero, as expected. We also note that, as we saw for bond angles and dihedrals there are certain angles for which these formulae are problematic - in this case, when $\phi_{jkl,m}$ is $+90^\circ$ or -90°. These cases should be catered for in a simulation. It is also at these angles that the terms within the large brackets in (5.44) become zero, so setting the forces to zero at these extremes is an option.

Example forms for the inversion potential include the harmonic and harmonic cosine form described by equations (5.20) and (5.21) in section 5.2.2 . However inversion potentials can also be used to enforce a planar structure on a group of atoms and the form for this is

$$\phi^{invers}_d(\phi_{jkl,m},\phi_{jlm,k},\phi_{jmk,l})=A(3-\cos\phi_{jkl,m}-\cos\phi_{jlm,k}-\cos\phi_{jmk,l}),\qquad (5.45)$$

for which the derivatives are

$$\phi'^{invers}_d(\phi_{jkl,m})=A\sin\phi_{jkl,m},\qquad \phi'^{invers}_d(\phi_{jlm,k,m})=A\sin\phi_{jlm,k},\qquad \phi'^{invers}_d(\phi_{jmk,l})=A\sin\phi \qquad (5.46)$$

from which it is evident that the force restoring atom m to the plane defined by

atoms j, k, l is zero when $\phi_{jkl,m}=0^\circ$ and maximum when $\phi_{jkl,m}=90^\circ,$ as required.

5.2.5 The "Calcite" Potential

Restraining a group of atoms to a plane is a common requirement in molecular dynamics, where planar chemical groups like $[NO_3]^{-1}$ and $[CO_3]^{-2}$ abound. So it is no surprise that alternatives to the inversion potential exist. The Calcite potential [39] (which clearly has applications to other groups) is an example of this. It is interesting because it contains no angular dependence; the central atom of the carbonate group is restrained in the plane of the oxygen atoms by a distance dependent potential.

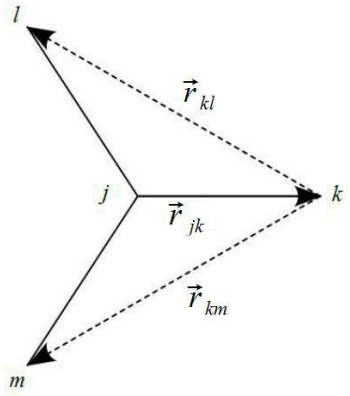

Figure 5.4: The Calcite Potential Vectors

The system configuration energy takes the form

$$\Phi(\vec{r}^{\,N})=\sum_{d=1}^{N_{calcite}}\phi_d^{calcite}(u_{jklm}), \tag{5.47}$$

where $\phi_d^{calcite}$ is one of the $N_{calcite}$ calcite structures in the system. Atoms j, k, l and m, are specific to the calcite structure indexed as $d.$

The function $\phi_d^{calcite}(u_{jklm})$ has the form

$$\phi_d^{calcite}=Au_{jklm}^2+Bu_{jklm}^4, \tag{5.48}$$

where u_{jklm} is the displacement of atom j above or below the plane defined by atoms k, l, m and is given by

$$u_{jklm}=\frac{\vec{r}_{jk}\cdot\vec{r}_{kl}\times\vec{r}_{km}}{|\vec{r}_{kl}\times\vec{r}_{km}|}. \tag{5.49}$$

The vectors $\vec{r}_{jk}$, $\vec{r}_{kl}$ and $\vec{r}_{km}$ are shown in figure 5.4. The vector product $\vec{r}_{kl}\times\vec{r}_{km}$ defines a vector perpendicular to the plane of the atoms k, l, m and is converted to a unit vector by division by $|\vec{r}_{kl}\times\vec{r}_{km}|.$ Its scalar product with $\vec{r}_{jk}$ defines the displacement of the atom j with respect to the plane.

The force on an arbitrary atom, n, is therefore

$$\begin{aligned}\vec{f}_n^{calcite} &= -\frac{\partial}{\partial \vec{r}_n}\sum_{d=1}^{N_{calcite}} \phi_d^{calcite}(u_{jklm}),\\ &= -\sum_{d=1}^{N_{calcite}} \frac{\partial \phi_d^{calcite}(u_{jklm})}{\partial u_{jklm}}\frac{\partial u_{jklm}}{\partial \vec{r}_n},\\ &= -\sum_{d=1}^{N_{calcite}} \phi'^{calcite}_d(u_{jklm})\frac{\partial u_{jklm}}{\partial \vec{r}_n}.\end{aligned} \tag{5.50}$$

In this equation $\phi'^{calcite}_d(u_{jklm})$ is the derivative of $\phi_d^{calcite}(u_{jklm})$ with respect to u_{jklm}. The derivative of u_{jklm} with respect to $\vec{r}_n$ is

$$\frac{\partial u_{jklm}}{\partial \vec{r}_n} = \frac{\vec{r}_{kl}\times\vec{r}_{km}}{|\vec{r}_{kl}\times\vec{r}_{km}|}.\frac{\partial \vec{r}_{jk}}{\partial \vec{r}_n} + \left\{\frac{\vec{r}_{jk}}{|\vec{r}_{kl}\times\vec{r}_{km}|} - u_{jklm}\frac{\vec{r}_{kl}\times\vec{r}_{km}}{|\vec{r}_{kl}\times\vec{r}_{km}|^2}\right\}.\frac{\partial(\vec{r}_{kl}\times\vec{r}_{km})}{\partial \vec{r}_n}, \tag{5.51}$$

and to obtain the partial derivatives appearing on the right of (5.51) we note that, for an arbitrary vector $\vec{a}$ it can be shown that

$$\begin{aligned}\vec{a}.\frac{\partial \vec{r}_{jk}}{\partial \vec{r}_n} &= (\delta_{kn}-\delta_{jn})\vec{a},\\ \vec{a}.\frac{\partial(\vec{r}_{kl}\times\vec{r}_{km})}{\partial \vec{r}_n} &= \vec{a}\times[(\delta_{mn}-\delta_{kn})\vec{r}_{kl}-(\delta_{ln}-\delta_{kn})\vec{r}_{km}],\end{aligned} \tag{5.52}$$

The force on atom n can then be written as

$$\begin{aligned}\vec{f}_n^{calcite} = -\sum_{d=1}^{N_{calcite}} \phi'^{calcite}_d(u_{jklm})\Bigg((\delta_{kn}-\delta_{jn})\frac{\vec{r}_{kl}\times\vec{r}_{km}}{|\vec{r}_{kl}\times\vec{r}_{km}|} + \\ \left\{\frac{\vec{r}_{jk}}{|\vec{r}_{kl}\times\vec{r}_{km}|} - u_{jklm}\frac{\vec{r}_{kl}\times\vec{r}_{km}}{|\vec{r}_{kl}\times\vec{r}_{km}|^2}\right\}\times[(\delta_{mn}-\delta_{kn})\vec{r}_{kl}-(\delta_{ln}-\delta_{kn})\vec{r}_{km}]\Bigg).\end{aligned} \tag{5.53}$$

We can now consider specific cases where atom n is j, k, l or m, to get the contributions to the forces on these atoms due to one calcite potential $\phi_d^{calcite}(u_{jklm})$:

$$\begin{aligned}\vec{f}_j^{calcite} &\leftarrow \phi'^{calcite}_d(u_{jklm})\left(\frac{\vec{r}_{kl}\times\vec{r}_{km}}{|\vec{r}_{kl}\times\vec{r}_{km}|}\right),\\ \vec{f}_k^{calcite} &\leftarrow -\phi'^{calcite}_d(u_{jklm})\left(\frac{\vec{r}_{kl}\times\vec{r}_{km}}{|\vec{r}_{kl}\times\vec{r}_{km}|} - \left\{\frac{\vec{r}_{jk}}{|\vec{r}_{kl}\times\vec{r}_{km}|} - u_{jklm}\frac{\vec{r}_{kl}\times\vec{r}_{km}}{|\vec{r}_{kl}\times\vec{r}_{km}|^2}\right\}\times[\vec{r}_{kl}-\vec{r}_{km}]\right),\\ \vec{f}_l^{calcite} &\leftarrow \phi'^{calcite}_d(u_{jklm})\left(\frac{\vec{r}_{jk}}{|\vec{r}_{kl}\times\vec{r}_{km}|} - u_{jklm}\frac{\vec{r}_{kl}\times\vec{r}_{km}}{|\vec{r}_{kl}\times\vec{r}_{km}|^2}\right)\times\vec{r}_{km},\\ \vec{f}_m^{calcite} &\leftarrow -\phi'^{calcite}_d(u_{jklm})\left(\frac{\vec{r}_{jk}}{|\vec{r}_{kl}\times\vec{r}_{km}|} - u_{jklm}\frac{\vec{r}_{kl}\times\vec{r}_{km}}{|\vec{r}_{kl}\times\vec{r}_{km}|^2}\right)\times\vec{r}_{kl}.\end{aligned} \tag{5.54}$$

In this case $\phi'^{calcite}_{d}(u_{jklm})$ is always given by

$$\phi'^{calcite}_{d}(u_{jklm}) = 2Au_{jklm} + 4Bu^{3}_{jklm}. \tag{5.55}$$

As usual all the forces of (5.54) sum to zero, which prevents a force accumulating on the molecule as a whole.

5.3 Inter-molecular Forces

We should begin by saying that only short-ranged forces are considered here. Long-ranged forces, which are exclusively electrostatic in origin, are dealt with in chapter 6. Once again, we start here with the simplest two-body potentials and move on to the more complex many-body potentials later. All the forces we shall discuss here are of the non-bonding type, which means that the atoms are free in principle to dissociate and re-associate in the course of a molecular dynamics simulation. They are generally called van der Waals interactions and are usually an order of magnitude weaker than chemical bonds. But we should bear in mind that non-bonding interactions can also occur within molecules, which happens when all the valency issues have been satisfied and any remaining interactions are of the van der Waals kind. In practice these issues are mainly of significance in deciding how best to incorporate the interactions into a simulation scheme. Bonding interactions imply a permanent connection between atoms, with associated book keeping to mange them, but non-bonded interactions imply no permanent connection and interacting atoms have to be found by searching through the system.

5.3.1 Pair Forces

Pair forces are ubiquitous in molecular dynamics to such an extent that a molecular dynamics simulation cannot really be imagined without them. Though these interactions are considered short ranged, they exert a significant influence on the dynamics of the system over length scales equivalent to several times the mean inter-atomic distance. The Lennard-Jones interaction described in chapter 1, equation (1.2), is the archetypical example, but there are many others.

The contribution of the non-bonding pair interactions to the system configuration energy is

$$\Phi(\vec{r}^{N}) = \sum_{i=1}^{N-1} \sum_{j>i}^{N} \phi_{ij}^{vdw}(r_{ij}) \tag{5.56}$$

where $\phi_{ij}^{vdw}(r_{ij})$ is the potential energy function specific to the interaction between atoms i and j. (It is quite common for this function to be the same for many pairs of atoms, depending on the prevalence of various atom types.) The functions are dependent only on the scalar distance r_{ij} between the atoms, where

$$r_{ij} \equiv |\vec{r}_{ij}| = |\vec{r}_{j} - \vec{r}_{i}|. \tag{5.57}$$

In (5.56), N is the number of atoms in the system. The expression on the right of (5.56) includes a double sum, which reflects the possibility that every atom in the system may interact with every other atom. However, very often this possibility is prevented by the application of a cut-off criterion (see section 1.2.4) such that any pair of atoms separated by a distance greater than a cut-off, r_{cut}, is assumed, *a priori*, to be zero (i.e. $\phi_{ij}(r_{ij})=0$ if $r_{ij}>r_{cut}$). This is a consequence of the fact that all realistic potentials decay in strength as r_{ij} increases and thus, it is hoped, become negligible at sufficient range. This practice can have a significant effect on the cost of computing (5.56) as it permits the use of various time-saving techniques that avoid unnecessary calculation (see chapter 8). As written in (5.56), the computational cost of calculating $\Phi(\{\vec{r}_i\})$ evidently scales as $O(N^2),$ but if a cut-off is applied to exclude a fraction of the pair interactions, then the cost is going to be less severe. It is even possible to reduce the scaling dependence to $\sim O(N),$ which has a major impact on the viability of large scale molecular dynamics simulations (see chapter 8, section 8.4).

It is usual when applying a cut-off condition to incorporate some form of *long range correction* to the pair force energy and the virial contribution that arises from it. This was dealt with in section 3.11 of chapter 3. Note also the structure of the two sums over i and j in (5.56); the condition $(j>i)$ avoids the recurrence of any pair (i,j) in the double sum, and thus prevents double counting of the interactions.

From (5.56) using the chain rule the force on an arbitrary atom $n,$ can be obtained as

$$\begin{aligned}\vec{f}_n^{vdw} &= -\frac{\partial}{\partial \vec{r}_n}\sum_{i=1}^{N-1}\sum_{j>i}^{N}\phi_{ij}^{vdw}(r_{ij}),\\ &= -\sum_{i=1}^{N-1}\sum_{j>i}^{N}\frac{\partial \phi_{ij}^{vdw}(r_{ij})}{\partial r_{ij}}\frac{\partial r_{ij}}{\partial \vec{r}_n},\\ &= -\sum_{i=1}^{N-1}\sum_{j>i}^{N}\phi'^{vdw}_{ij}(r_{ij})\frac{\vec{r}_{ij}}{r_{ij}}(\delta_{jn}-\delta_{in}).\end{aligned} \tag{5.58}$$

Now considering the cases where either i or j can be $n,$ leads to the formula for the full force on the atom n :

$$\vec{f}_n^{vdw} = \sum_{j>n}^{N}\frac{\phi'^{vdw}_{nj}(r_{nj})}{r_{nj}}\vec{r}_{nj} - \sum_{i=1}^{i<n}\frac{\phi'^{vdw}_{in}(r_{in})}{r_{in}}\vec{r}_{in}, \tag{5.59}$$

which , since $\vec{r}_{nj}=-\vec{r}_{jn},$ is the same as

$$\vec{f}_n^{vdw} = -\sum_{i\neq n}^{N}\frac{\phi'^{vdw}_{in}(r_{in})}{r_{in}}\vec{r}_{in}. \tag{5.60}$$

Thus we see that the full force on atom n is the sum of the pair forces exerted by all the other atoms in the system, assuming no cut-off was applied. In practical simulations, there would be fewer than the $N-1$ interactions indicated by (5.60) that would require calculation.

Equation (5.60) can also be written as

$$\vec{f}_n^{vdw} = \sum_{i \neq n}^{N} \vec{f}_{in}^{vdw}. \tag{5.61}$$

Where $\vec{f}_{in}^{vdw}$ is a *pair force*, corresponding to the force atom i exerts on atom n. It is defined by the formula

$$\vec{f}_{in}^{vdw} = -\frac{\phi'^{vdw}_{in}(r_{in})}{r_{in}} \vec{r}_{in}. \tag{5.62}$$

As expected, the force atom i exerts on atom n is equal and opposite to the force atoms n exerts on i i.e.

$$\vec{f}_{in} = -\vec{f}_{ni}. \tag{5.63}$$

It follows from this that the sum of all forces acting in the system is zero:

$$\begin{aligned} \sum_{n=1}^{N} \vec{f}_n^{vdw} &= \sum_{n=1}^{N} \sum_{i \neq n}^{N} \vec{f}_{in}^{vdw}, \\ &= \sum_{n=1}^{N-1} \sum_{i>n}^{N} (\vec{f}_{in}^{vdw} + \vec{f}_{ni}^{vdw}), \\ &= \sum_{n=1}^{N-1} \sum_{i>n}^{N} (\vec{f}_{in}^{vdw} - \vec{f}_{in}^{vdw}) = 0. \end{aligned} \tag{5.64}$$

The second line of this derivation comes from the fact that in the double sum there are two pair forces for each pair of indices i and j. In one case the force is $\vec{f}_{in}^{vdw}$ and in the other it is $\vec{f}_{ni}^{vdw}$. The third line follows from (5.63).

We now consider some common forms for $\phi_{ij}^{vdw}(r_{ij})$ (other than the Lennard-Jones form from chapter 1).

i. The **n-m potential** is a more general alternative to the Lennard-Jones potential:

$$\begin{aligned} \phi_{ij}^{vdw}(r_{ij}) &= \frac{\varepsilon_{ij}}{(n-m)} \left(m \left\{ \frac{r_0}{r_{ij}} \right\}^n - n \left\{ \frac{r_0}{r_{ij}} \right\}^m \right), \\ \vec{f}_{ij}^{vdw}(r_{ij}) &= \frac{nm\,\varepsilon_{ij}}{(n-m) r_{ij}^2} \left(\left\{ \frac{r_0}{r_{ij}} \right\}^n - \left\{ \frac{r_0}{r_{ij}} \right\}^m \right) \vec{r}_{ij}, \end{aligned} \tag{5.65}$$

where n, m are integers $(n > m)$ and ε_{ij} is the well depth of the potential. r_0 is the separation in r_{ij} for which the potential minimum occurs.

ii. The ***Buckingham potential*** is normally used in ionic models and has a soft repulsion at short range:

$$\begin{aligned}\phi_{ij}^{vdw}(r_{ij}) &= A_{ij}\exp\left(\frac{-r_{ij}}{\rho_{ij}}\right)-\frac{C_{ij}}{r_{ij}^{6}},\\ \vec{f}_{ij}^{vdw}(r_{ij}) &= \left(\frac{A_{ij}}{\rho_{ij}r_{ij}}\exp\left(\frac{-r_{ij}}{\rho_{ij}}\right)-\frac{6C_{ij}}{r_{ij}^{8}}\right)\vec{r}_{ij},\end{aligned} \tag{5.66}$$

where A_{ij}, ρ_{ij} and C_{ij} are parameters. The first two together determine the softness of the short range repulsion and C_{ij} determines the longer range attraction. An unfortunate property of this potential is that, at very short range, the r_{ij}^{-6} term overcomes the exponential term and the potential function rapidly diverges to $-\infty$. Extreme care is therefore required when using this potential in high energy or high temperature dynamics, where penetration of this region is possible..

iii. The **12-6 potential** is a common potential in bio-simulations. It is equivalent to the Lennard-Jones potential but is parameterised differently.

$$\begin{aligned}\phi_{ij}^{vdw}(r_{ij}) &= \frac{A_{ij}}{r_{ij}^{12}}-\frac{B_{ij}}{r_{ij}^{6}},\\ \vec{f}_{ij}^{vdw}(r_{ij}) &= \left(\frac{12A_{ij}}{r_{ij}^{14}}-\frac{6B_{ij}}{r_{ij}^{8}}\right)\vec{r}_{ij}.\end{aligned} \tag{5.67}$$

The equivalence between this potential and the Lennard-Jones form is through the relations: $A_{ij}=4\varepsilon_{ij}\sigma_{ij}^{12}$ and $B_{ij}=4\varepsilon_{ij}\sigma_{ij}^{6}$.

iv. The **WCA potential** (named after authors Weeks, Chandler and Andersen [40]) is a purely repulsive potential derived from the Lennard-Jones potential by truncation and shifting.

$$\begin{aligned}\phi_{ij}^{vdw}(r_{ij}) &= \begin{cases}4\varepsilon_{ij}\left(\left\{\frac{\sigma_{ij}}{r_{ij}}\right\}^{12}-\left\{\frac{\sigma_{ij}}{r_{ij}}\right\}^{6}\right)+\varepsilon_{ij} & if \quad r_{ij}\le 2^{1/6}\sigma,\\ 0 & if \quad r_{ij}>2^{1/6}\sigma,\end{cases}\\ \vec{f}_{ij}^{vdw}(r_{ij}) &= \begin{cases}\frac{24\varepsilon_{ij}}{r_{ij}^{2}}\left(2\left\{\frac{\sigma_{ij}}{r_{ij}}\right\}^{12}-\left\{\frac{\sigma_{ij}}{r_{ij}}\right\}^{6}\right)\vec{r}_{ij} & if \quad r_{ij}\le 2^{1/6}\sigma,\\ 0 & if \quad r_{ij}>2^{1/6}\sigma.\end{cases}\end{aligned} \tag{5.68}$$

This is a very short-ranged interaction by normal standards and resembles an atomic hard-sphere model. Its main application is in studies of fluid flow.

v. **Tabulated pair potentials** are resorted to when the pair potential is not expressible in any convenient analytical form or is simply not available within

the library of options in an existing computer program. In the latter case it is useful for the program to incorporate tabulated potentials as an optional feature. The tabulation can be made in terms of increments in r_{ij} (i.e. Δr_{ij}) or increments in r_{ij}^2 (i.e. $\Delta(r_{ij}^2)$) and should range beyond the required cut-off distance to permit accurate numerical interpolation at the cut-off. The force may be calculated from the numerical derivative of the data points, but greater accuracy is possible if the force is tabulated separately – which is a practical possibility if the potential function is analytical. A compromise needs to be struck between the amount of data storage (i.e. number of data points), the accuracy of interpolation and the speed of calculating the potential and force.

While the vast majority of molecular dynamics simulations are undertaken with intra-molecular potentials to maintain molecular structure and pair potentials to describe the non-bonding interactions, there are occasions when something more sophisticated is required. This entails incorporating three-body, four-body, or even many-body potentials to achieve the required accuracy. This may be needed to encapsulate a structural detail in the simulated system or, equivalently, to improve its thermodynamic properties. An excellent sample of such an addition is Hydrogen bonding, which is required to give water, in particular, its peculiar structural and thermodynamic characteristics.

5.3.2 Three-Body Forces

The lowest level improvement on pair-force models is to include three-body interactions. For these we write the system configuration energy as

$$\Phi(\vec{r}^N)=\sum_{i=1}^{N-2}\sum_{j>i}^{N-1}\sum_{k>j}^{N}\phi_{ijk}^{3\text{bdy}}(r_{ij},r_{jk},r_{ki},\theta_{kij},\theta_{ijk},\theta_{jki}), \tag{5.69}$$

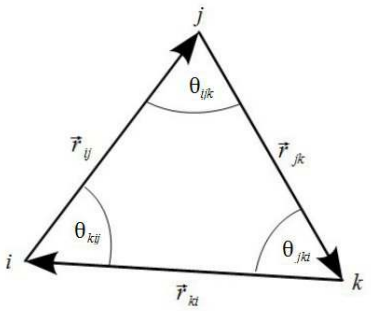

Figure 5.5: Three-Body Interaction Variables

In which the potential function $\phi_{ijk}^{3\text{bdy}}$ depends on the triangular arrangement of the atoms i,j,k as characterised by the pair separations r_{ij},r_{jk},r_{ki} and the angles, $\theta_{kij},\theta_{ijk},\theta_{jki}$, which we define in the following manner (see figure 5.5):

$$\begin{aligned} r_{ij}&=|\vec{r}_j-\vec{r}_i|, & r_{jk}&=|\vec{r}_k-\vec{r}_j|, & r_{ki}&=|\vec{r}_i-\vec{r}_k|,\\ \cos\theta_{kij}&=-\frac{\vec{r}_{ij}\cdot\vec{r}_{ki}}{r_{ij}r_{ki}}, & \cos\theta_{ijk}&=-\frac{\vec{r}_{jk}\cdot\vec{r}_{ij}}{r_{jk}r_{ij}}, & \cos\theta_{jki}&=-\frac{\vec{r}_{ki}\cdot\vec{r}_{jk}}{r_{ki}r_{jk}}. \end{aligned} \tag{5.70}$$

The form of the summation on the right of (5.69) implies it has a cost of order $O(N^3)$ to compute. However, three-body potentials are generally of much shorter range than pair potentials which means a short ranged cut-off can be applied. Thus three-body interactions very often can be made to scale as $\sim O(N),$ though this may still require some intense computing, as the terms are often more complicated than for pair interactions.

It is also worth noting that the structure of the triple sum in (5.69) is designed to ensure that each combination of three different atoms is specified once only. This does not mean that every atom in each triplet is identical or even equivalent. Indeed the actual form of the function $\phi_{ijk}^{3\mathrm{bdy}}$ may not make use all of the variables we have defined in the function arguments. Nevertheless we will assume that, once the triplet of atoms has been identified, the interaction will be handled correctly, using whatever information is necessary to identify the atoms appropriately. This is not difficult in practice but we need to be aware that this is sometimes necessary.

Writing (u_{ijk}) as an abbreviation for $(r_{ij}, r_{jk}, r_{ki}, \theta_{kij}, \theta_{ijk}, \theta_{jki})$ in (5.69) we can obtain the force acting on an arbitrary atom n as follows

$$\begin{aligned}\vec{f}_n^{\,3\mathrm{bdy}} &= -\frac{\partial}{\partial \vec{r}_n}\sum_{i=1}^{N-2}\sum_{j>i}^{N-1}\sum_{k>j}^{N}\phi_{ijk}^{3\mathrm{bdy}}(u_{ijk}),\\ &= -\sum_{i=1}^{N-2}\sum_{j>i}^{N-1}\sum_{k>j}^{N}\left(\frac{\partial\phi_{ijk}^{3\mathrm{bdy}}(u_{ijk})}{\partial r_{ij}}\frac{\partial r_{ij}}{\partial \vec{r}_n}+\frac{\partial\phi_{ijk}^{3\mathrm{bdy}}(u_{ijk})}{\partial r_{jk}}\frac{\partial r_{jk}}{\partial \vec{r}_n}+\right.\\ &\qquad \frac{\partial\phi_{ijk}^{3\mathrm{bdy}}(u_{ijk})}{\partial r_{ki}}\frac{\partial r_{ki}}{\partial \vec{r}_n}+\frac{\partial\phi_{ijk}^{3\mathrm{bdy}}(u_{ijk})}{\partial \theta_{kij}}\frac{\partial \theta_{kij}}{\partial \vec{r}_n}+\\ &\qquad \left.\frac{\partial\phi_{ijk}^{3\mathrm{bdy}}(u_{ijk})}{\partial \theta_{ijk}}\frac{\partial \theta_{ijk}}{\partial \vec{r}_n}+\frac{\partial\phi_{ijk}^{3\mathrm{bdy}}(u_{ijk})}{\partial \theta_{jki}}\frac{\partial \theta_{jki}}{\partial \vec{r}_n}\right),\end{aligned} \tag{5.71}$$

which can be further expanded into

$$\begin{aligned}\vec{f}_n^{\,3\mathrm{bdy}} = -\sum_{i=1}^{N-2}\sum_{j>i}^{N-1}\sum_{k>j}^{N}\Bigg(&\phi'^{3\mathrm{bdy}}_{ijk}(r_{ij})\vec{e}_{ij}(\delta_{jn}-\delta_{in})+\phi'^{3\mathrm{bdy}}_{ijk}(r_{jk})\vec{e}_{jk}(\delta_{kn}-\delta_{jn})+\phi'^{3\mathrm{bdy}}_{ijk}(r_{ki})\vec{e}_{ki}(\delta_{in}-\delta_{kn})\\ &+\frac{\phi'^{3\mathrm{bdy}}_{ijk}(\theta_{kij})}{\sin\theta_{kij}}\left\{\frac{(\vec{e}_{ij}+\cos\theta_{kij}\vec{e}_{ki})}{r_{ki}}(\delta_{in}-\delta_{kn})+\frac{(\vec{e}_{ki}+\cos\theta_{kij}\vec{e}_{ij})}{r_{ij}}(\delta_{jn}-\delta_{in})\right\}\\ &+\frac{\phi'^{3\mathrm{bdy}}_{ijk}(\theta_{ijk})}{\sin\theta_{ijk}}\left\{\frac{(\vec{e}_{jk}+\cos\theta_{ijk}\vec{e}_{ij})}{r_{ij}}(\delta_{jn}-\delta_{in})+\frac{(\vec{e}_{ij}+\cos\theta_{ijk}\vec{e}_{jk})}{r_{jk}}(\delta_{kn}-\delta_{jn})\right\}\\ &+\frac{\phi'^{3\mathrm{bdy}}_{ijk}(\theta_{jki})}{\sin\theta_{jki}}\left\{\frac{(\vec{e}_{ki}+\cos\theta_{jki}\vec{e}_{jk})}{r_{jk}}(\delta_{kn}-\delta_{jn})+\frac{(\vec{e}_{jk}+\cos\theta_{jki}\vec{e}_{ki})}{r_{ki}}(\delta_{in}-\delta_{kn})\right\}\Bigg),\end{aligned} \tag{5.72}$$

in which we have defined the unit vectors

$$\vec{e}_{ij}=\vec{r}_{ij}/r_{ij}, \qquad \vec{e}_{jk}=\vec{r}_{jk}/r_{jk}, \qquad \vec{e}_{ki}=\vec{r}_{ki}/r_{ki}, \tag{5.73}$$

and $\phi'^{3bdy}_{ijk}(x)$ is an abbreviation referring to the derivative of ϕ^{3bdy}_{ijk} with respect to the variable x. This is the most general form for the force. Taking specific cases where n is one of i, j or k gives

$$\begin{aligned}
\vec{f}^{\,3bdy}_{n} = &-\sum_{j=n+1}^{N-1}\sum_{k>j}^{N}\left(\phi'^{3bdy}_{njk}(r_{kn})\vec{e}_{kn} - \phi'^{3bdy}_{njk}(r_{nj})\vec{e}_{nj} - \frac{\phi'^{3bdy}_{njk}(\theta_{njk})}{\sin\theta_{njk}}\left\{\frac{(\vec{e}_{jk}+\cos\theta_{njk}\vec{e}_{nj})}{r_{nj}}\right\}\right.\\
&+\frac{\phi'^{3bdy}_{njk}(\theta_{knj})}{\sin\theta_{knj}}\left\{\frac{(\vec{e}_{nj}+\cos\theta_{knj}\vec{e}_{kn})}{r_{kn}} - \frac{(\vec{e}_{kn}+\cos\theta_{knj}\vec{e}_{nj})}{r_{nj}}\right\}\\
&\left.+\frac{\phi'^{3bdy}_{njk}(\theta_{jkn})}{\sin\theta_{jkn}}\left\{\frac{(\vec{e}_{jk}+\cos\theta_{jkn}\vec{e}_{kn})}{r_{kn}}\right\}\right)\\
&-\sum_{i=1}^{n-1}\sum_{k>n}^{N}\left(\phi'^{3bdy}_{ink}(r_{in})\vec{e}_{in} - \phi'^{3bdy}_{ink}(r_{nk})\vec{e}_{nk} + \frac{\phi'^{3bdy}_{ink}(\theta_{kin})}{\sin\theta_{kin}}\left\{\frac{(\vec{e}_{ki}+\cos\theta_{kin}\vec{e}_{in})}{r_{in}}\right\}\right.\\
&+\frac{\phi'^{3bdy}_{ink}(\theta_{ink})}{\sin\theta_{ink}}\left\{\frac{(\vec{e}_{nk}+\cos\theta_{ink}\vec{e}_{in})}{r_{in}} - \frac{(\vec{e}_{in}+\cos\theta_{ink}\vec{e}_{nk})}{r_{nk}}\right\}\\
&\left.-\frac{\phi'^{3bdy}_{ink}(\theta_{nki})}{\sin\theta_{nki}}\left\{\frac{(\vec{e}_{ki}+\cos\theta_{nki}\vec{e}_{nk})}{r_{nk}}\right\}\right)\\
&-\sum_{i=1}^{n-2}\sum_{j>i}^{n-1}\left(\phi'^{3bdy}_{ijn}(r_{jn})\vec{e}_{jn} - \phi'^{3bdy}_{ijn}(r_{ni})\vec{e}_{ni} - \frac{\phi'^{3bdy}_{ijn}(\theta_{nij})}{\sin\theta_{nij}}\left\{\frac{(\vec{e}_{ij}+\cos\theta_{nij}\vec{e}_{ni})}{r_{ni}}\right\}\right.\\
&+\frac{\phi'^{3bdy}_{ijn}(\theta_{jni})}{\sin\theta_{jni}}\left\{\frac{(\vec{e}_{ni}+\cos\theta_{jni}\vec{e}_{jn})}{r_{jn}} - \frac{(\vec{e}_{jn}+\cos\theta_{jni}\vec{e}_{ni})}{r_{ni}}\right\}\\
&\left.+\frac{\phi'^{3bdy}_{ijn}(\theta_{ijn})}{\sin\theta_{ijn}}\left\{\frac{(\vec{e}_{ij}+\cos\theta_{ijn}\vec{e}_{jn})}{r_{jn}}\right\}\right).
\end{aligned} \tag{5.74}$$

The series of double sums in (5.74) can be reconstructed through a redefinition of index variables to make them consistent across all three sums. This leads to the reduced form:

$$\begin{aligned}
\vec{f}^{\,3bdy}_{n} = &-\sum_{i=1}^{N-1}{}'\sum_{j>i}^{N}{}'\left(\phi'^{3bdy}_{ijn}(r_{jn})\vec{e}_{jn} - \phi'^{3bdy}_{ijn}(r_{ni})\vec{e}_{ni} - \frac{\phi'^{3bdy}_{ijn}(\theta_{nij})}{\sin\theta_{nij}}\left\{\frac{(\vec{e}_{ij}+\cos\theta_{nij}\vec{e}_{ni})}{r_{ni}}\right\}\right.\\
&+\frac{\phi'^{3bdy}_{ijn}(\theta_{jni})}{\sin\theta_{jni}}\left\{\frac{(\vec{e}_{ni}+\cos\theta_{jni}\vec{e}_{jn})}{r_{jn}} - \frac{(\vec{e}_{jn}+\cos\theta_{jni}\vec{e}_{ni})}{r_{ni}}\right\}\\
&\left.+\frac{\phi'^{3bdy}_{ijn}(\theta_{ijn})}{\sin\theta_{ijn}}\left\{\frac{(\vec{e}_{ij}+\cos\theta_{ijn}\vec{e}_{jn})}{r_{jn}}\right\}\right),
\end{aligned} \tag{5.75}$$

in which the dash $(')$ on both the summation symbols signals that the index cannot equal n in either case.

The contribution of a given three-body potential, ϕ^{3bdy}_{ijk}, to the forces on the participating atoms i, j, k are evidently:

$$
\begin{aligned}
\vec{f}_i^{\,3bdy} \leftarrow -\Bigg(& \phi'^{3bdy}_{ijk}(r_{ki})\vec{e}_{ki} - \phi'^{3bdy}_{ijk}(r_{ij})\vec{e}_{ij} - \frac{\phi'^{3bdy}_{ijk}(\theta_{ijk})}{\sin\theta_{ijk}}\left\{\frac{(\vec{e}_{jk} + \cos\theta_{ijk}\vec{e}_{ij})}{r_{ij}}\right\} \\
& + \frac{\phi'^{3bdy}_{ijk}(\theta_{kij})}{\sin\theta_{kij}}\left\{\frac{(\vec{e}_{ij} + \cos\theta_{kij}\vec{e}_{ki})}{r_{ki}} - \frac{(\vec{e}_{ki} + \cos\theta_{kij}\vec{e}_{ij})}{r_{ij}}\right\} + \frac{\phi'^{3bdy}_{ijk}(\theta_{jki})}{\sin\theta_{jki}}\left\{\frac{(\vec{e}_{jk} + \cos\theta_{jki}\vec{e}_{ki})}{r_{ki}}\right\}\Bigg), \\
\vec{f}_j^{\,3bdy} \leftarrow -\Bigg(& \phi'^{3bdy}_{ijk}(r_{ij})\vec{e}_{ij} - \phi'^{3bdy}_{ijk}(r_{jk})\vec{e}_{jk} - \frac{\phi'^{3bdy}_{ijk}(\theta_{jki})}{\sin\theta_{jki}}\left\{\frac{(\vec{e}_{ki} + \cos\theta_{jki}\vec{e}_{jk})}{r_{jk}}\right\} \\
& + \frac{\phi'^{3bdy}_{ijk}(\theta_{ijk})}{\sin\theta_{ijk}}\left\{\frac{(\vec{e}_{jk} + \cos\theta_{ijk}\vec{e}_{ij})}{r_{ij}} - \frac{(\vec{e}_{ij} + \cos\theta_{ijk}\vec{e}_{jk})}{r_{jk}}\right\} + \frac{\phi'^{3bdy}_{ijk}(\theta_{kij})}{\sin\theta_{kij}}\left\{\frac{(\vec{e}_{ki} + \cos\theta_{kij}\vec{e}_{ij})}{r_{ij}}\right\}\Bigg), \\
\vec{f}_k^{\,3bdy} \leftarrow -\Bigg(& \phi'^{3bdy}_{ijk}(r_{jk})\vec{e}_{jk} - \phi'^{3bdy}_{ijk}(r_{ki})\vec{e}_{ki} - \frac{\phi'^{3bdy}_{ijk}(\theta_{kij})}{\sin\theta_{kij}}\left\{\frac{(\vec{e}_{ij} + \cos\theta_{kij}\vec{e}_{ki})}{r_{ki}}\right\} \\
& + \frac{\phi'^{3bdy}_{ijk}(\theta_{jki})}{\sin\theta_{jki}}\left\{\frac{(\vec{e}_{ki} + \cos\theta_{jki}\vec{e}_{jk})}{r_{jk}} - \frac{(\vec{e}_{jk} + \cos\theta_{jki}\vec{e}_{ki})}{r_{ki}}\right\} + \frac{\phi'^{3bdy}_{ijk}(\theta_{ijk})}{\sin\theta_{ijk}}\left\{\frac{(\vec{e}_{ij} + \cos\theta_{ijk}\vec{e}_{jk})}{r_{jk}}\right\}\Bigg).
\end{aligned}
\tag{5.76}
$$

It can be seen from (5.76) that, for a given set of three atoms i, j, k, the sum of all the atomic forces arising from the corresponding three-body potential function, ϕ^{3bdy}_{ijk}, is zero. It follows from this that the net force on the whole system due to all the three-body terms must also be zero. It is also apparent that the three formulae in (5.76) are essentially the same, differing only by a cyclic permutation of the indices. This is because all the atoms are mathematically equivalent in our representation.

The archetypical potential of this kind is the triple-dipole potential due to Axilrod and Teller [41] which is used as a higher order correction to the attractive $(1/r^6)$ tail of the Lennard_Jones potential (the so-called London dispersion term). It has the following form

$$
\phi^{3bdy}_{ijk}(r_{ij}, r_{jk}, r_{ki}, \theta_{kij}, \theta_{ijk}, \theta_{jki}) = E_0\left[\frac{(1 + 3\cos\theta_{kij}\cos\theta_{ijk}\cos\theta_{jki})}{(r_{ij}r_{jk}r_{ki})^3}\right] \tag{5.77}
$$

from which it follows that the derivatives required for the force calculations are

$$
\begin{aligned}
&\phi'^{3bdy}_{ijk}(r_{ij}) = -\frac{3E_0}{r_{ij}}\left[\frac{1 + 3\cos\theta_{kij}\cos\theta_{ijk}\cos\theta_{jki}}{(r_{ij}r_{jk}r_{ki})^3}\right], \quad \phi'^{3bdy}_{ijk}(\theta_{kij}) = -3E_0\left[\frac{\sin\theta_{kij}\cos\theta_{ijk}\cos\theta_{jki}}{(r_{ij}r_{jk}r_{ki})^3}\right], \\
&\phi'^{3bdy}_{ijk}(r_{jk}) = -\frac{3E_0}{r_{jk}}\left[\frac{1 + 3\cos\theta_{kij}\cos\theta_{ijk}\cos\theta_{jki}}{(r_{ij}r_{jk}r_{ki})^3}\right], \quad \phi'^{3bdy}_{ijk}(\theta_{ijk}) = -3E_0\left[\frac{\cos\theta_{kij}\sin\theta_{ijk}\cos\theta_{jki}}{(r_{ij}r_{jk}r_{ki})^3}\right], \\
&\phi'^{3bdy}_{ijk}(r_{ki}) = -\frac{3E_0}{r_{ki}}\left[\frac{1 + 3\cos\theta_{kij}\cos\theta_{ijk}\cos\theta_{jki}}{(r_{ij}r_{jk}r_{ki})^3}\right], \quad \phi'^{3bdy}_{ijk}(\theta_{jki}) = -3E_0\left[\frac{\cos\theta_{kij}\cos\theta_{ijk}\sin\theta_{jki}}{(r_{ij}r_{jk}r_{ki})^3}\right].
\end{aligned}
\tag{5.78}
$$

Another three-body potential worth mentioning is the angular potential often used in ionic models of glassy materials to enforce the arrangement of oxygen atoms around atoms such as silicon or boron. In the formal ionic model these are treated as ionic species e.g. the tetrahedral $[SiO_4]^{4-}$ or the trigonal planar $[BO_3]^{3-}$ units in silicate and borate glasses respectively. Such potentials have forms like

$$\phi_{ijk}^{3bdy}(\theta_{kij}) = \frac{\kappa}{2}(\theta_{kij} - \theta_0)^2, \qquad \phi'^{3bdy}_{ijk}(\theta_{kij}) = \kappa(\theta_{kij} - \theta_0), \tag{5.79}$$

and

$$\phi_{ijk}^{3bdy}(\theta_{kij}) = \frac{\kappa}{2}(\cos\theta_{kij} - \cos\theta_0)^2, \qquad \phi'^{3bdy}_{ijk}(\theta_{kij}) = -\kappa(\cos\theta_{kij} - \cos\theta_0)\sin\theta_{kij}. \tag{5.80}$$

For such potentials the dependence on only one variable (θ_{kij}) greatly reduces the complexity of the force expressions (5.76) to give:

$$\begin{aligned}
\vec{f}_i^{3bdy} &\leftarrow -\frac{\phi'^{3bdy}_{ijk}(\theta_{kij})}{\sin\theta_{kij}}\left\{\frac{(\vec{e}_{ij} + \cos\theta_{kij}\vec{e}_{ki})}{r_{ki}} - \frac{\vec{e}_{ki} + \cos\theta_{kij}\vec{e}_{ij}}{r_{ij}}\right\} \\
\vec{f}_j^{3bdy} &\leftarrow -\frac{\phi'^{3bdy}_{ijk}(\theta_{kij})}{\sin\theta_{kij}}\left\{\frac{\vec{e}_{ki} + \cos\theta_{kij}\vec{e}_{ij}}{r_{ij}}\right\}, \\
\vec{f}_k^{3bdy} &\leftarrow \frac{\phi'^{3bdy}_{ijk}(\theta_{kij})}{\sin\theta_{kij}}\left\{\frac{(\vec{e}_{ij} + \cos\theta_{kij}\vec{e}_{ki})}{r_{ki}}\right\}.
\end{aligned} \tag{5.81}$$

These expressions are very similar to the forms (5.19) for bond angles (allowing for slightly different definitions of the vectors $\vec{r}_{ij}$ *etc*). Note that the force on the "central" atom i is equal to the negative sum of the forces on atoms j and k.

5.3.3 Four-Body Forces

Four-body, non-bonded potentials do not commonly occur in molecular dynamics simulations. One of the few applications that require them is the simulation of systems containing trigonal arrangements of atoms like carbonates $[CO_3]^{2-}$, nitrates $[NO_3]^{-1}$ and borates $[BO_3]^{3-}$, where a four-body potential is necessary to maintain the quasi-planar structure. While it is probably more usual in the case of carbonates and nitrates to model the trigonal entity as a *molecular* species (possibly even rigid) and therefore as an *intra*-molecular potential, the modelling of borate glasses, at least, merits this less restricted approach, since the glass structure is often required to emerge spontaneously during the simulation. For such potentials we write

$$\Phi(\vec{r}^N) = \sum_{i=1}^{N}\sum_{j=1}^{N-2}{}'\sum_{k>j}^{N-1}{}'\sum_{l>k}^{N}{}'\phi_{ijkl}^{4bdy}(\theta_j, \theta_k, \theta_l), \tag{5.82}$$

which is the four-body configuration energy and the potential function ϕ_{ijkl}^{4bdy} depends on the angular arrangement of the atoms j, k, l around the central atom i, as in figure 5.6. The three angles $\theta_j, \theta_k, \theta_l$ are defined by the relations

$$\sin\theta_j = \frac{\vec{r}_{ij}\cdot(\vec{r}_{ik}\times\vec{r}_{il})}{r_{ij}|\vec{r}_{ik}\times\vec{r}_{il}|}, \qquad \sin\theta_k = \frac{\vec{r}_{ik}\cdot(\vec{r}_{il}\times\vec{r}_{ij})}{r_{ik}|\vec{r}_{il}\times\vec{r}_{ij}|}, \qquad \sin\theta_l = \frac{\vec{r}_{il}\cdot(\vec{r}_{ij}\times\vec{r}_{ik})}{r_{il}|\vec{r}_{ij}\times\vec{r}_{ik}|}, \tag{5.83}$$

where the vectors appearing are defined (with reference to figure 5.6) as

$$\vec{r}_{ij} = \vec{r}_j - \vec{r}_i, \qquad \vec{r}_{ik} = \vec{r}_k - \vec{r}_i, \qquad \vec{r}_{il} = \vec{r}_l - \vec{r}_i. \tag{5.84}$$

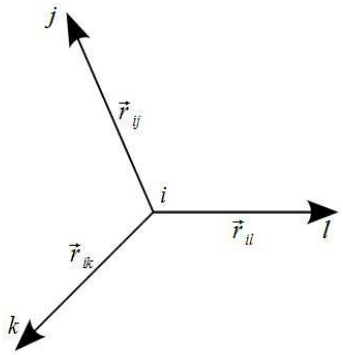

Figure 5.6: The Vectors of the Non-Bonded Planar Potential

Thus the angle θ_j (for example) is the angle between the vector $\vec{r}_{ij}$ and the plane containing the vectors $\vec{r}_{ik}$ and $\vec{r}_{il}$. Clearly three such angles can be constructed from figure 5.6.

The structure of the summations in (5.82) is designed so that the outer sum over atoms indexed i defines the *central* atom of the structure shown in figure 5.6. This atom has a rôle that differs from the others. The other summations run over atoms with the indices j, k, l and the summation is designed to generate combinations of these that occur once only for a given central atom i. None of these three summations can include the index i, which is the meaning of the dash $(')$ appearing against the summation symbol. The overall structure is a sum of order $O(N^4)$, which implies great computational expense. However, the range of these interactions, measured as the distance from the central atom i is normally very short. So it is best to think of the quadruple sum as approximately of order $O(N)$, since the short range implies a search over just the near neighbours of each of the N atoms in the system. The sum limits shown are therefore somewhat excessive, but the redundant contributions can be bypassed by applying the cut off to all the distances r_{ij}, r_{ik} and r_{il} simultaneously. Furthermore, if an atom i is not one of the required central atoms, it can be skipped without such a search.

From (5.82) the force acting on an atom n is given by

$$\begin{aligned}\vec{f}_n^{\,4\mathrm{bdy}} &= -\frac{\partial}{\partial \vec{r}_n}\sum_{i=1}^{N}\sum_{j=1}^{N-2}{}'\sum_{k>j}^{N-1}{}'\sum_{l>k}^{N}{}'\phi_{ijkl}^{4\mathrm{bdy}}(\theta_j,\theta_k,\theta_l),\\ &= -\sum_{i=1}^{N}\sum_{j=1}^{N-2}{}'\sum_{k>j}^{N-1}{}'\sum_{l>k}^{N}{}'\left(\frac{\partial\phi_{ijkl}^{4\mathrm{bdy}}(\theta_j,\theta_k,\theta_l)}{\partial\theta_j}\frac{\partial\theta_j}{\partial\vec{r}_n}\right.\\ &\qquad +\frac{\partial\phi_{ijkl}^{4\mathrm{bdy}}(\theta_j,\theta_k,\theta_l)}{\partial\theta_k}\frac{\partial\theta_k}{\partial\vec{r}_n}\\ &\qquad \left.+\frac{\partial\phi_{ijkl}^{4\mathrm{bdy}}(\theta_j,\theta_k,\theta_l)}{\partial\theta_l}\frac{\partial\theta_l}{\partial\vec{r}_n}\right),\end{aligned} \tag{5.85}$$

which expands into the following form:

$$\begin{aligned}\vec{f}_n^{\,4\mathrm{bdy}} = -\sum_{i=1}^{N}\sum_{j=1}^{N-2}{}'\sum_{k>j}^{N-1}{}'\sum_{l>k}^{N}{}'\Bigg(&\frac{\phi'^{4\mathrm{bdy}}_{ijkl}(\theta_j)}{\cos\theta_j}\left\{\frac{(\vec{e}_{ij}-\sin\theta_j\vec{e}_{ikl})}{|\vec{r}_{ik}\times\vec{r}_{il}|}\times[\vec{r}_{ik}(\delta_{ln}-\delta_{in})-\vec{r}_{il}(\delta_{kn}-\delta_{in})]\right.\\ &\left.+\frac{(\vec{e}_{ikl}-\sin\theta_j\vec{e}_{ij})}{r_{ij}}(\delta_{jn}-\delta_{in})\right\}\\ &+\frac{\phi'^{4\mathrm{bdy}}_{ijkl}(\theta_k)}{\cos\theta_k}\left\{\frac{(\vec{e}_{ik}-\sin\theta_k\vec{e}_{ilj})}{|\vec{r}_{il}\times\vec{r}_{ij}|}\times[\vec{r}_{il}(\delta_{jn}-\delta_{in})-\vec{r}_{ij}(\delta_{ln}-\delta_{in})]\right.\\ &\left.+\frac{(\vec{e}_{ilj}-\sin\theta_k\vec{e}_{ik})}{r_{ik}}(\delta_{kn}-\delta_{in})\right\}\\ &+\frac{\phi'^{4\mathrm{bdy}}_{ijkl}(\theta_l)}{\cos\theta_l}\left\{\frac{(\vec{e}_{il}-\sin\theta_l\vec{e}_{ijk})}{|\vec{r}_{ij}\times\vec{r}_{ik}|}\times[\vec{r}_{ij}(\delta_{kn}-\delta_{in})-\vec{r}_{ik}(\delta_{jn}-\delta_{in})]\right.\\ &\left.+\frac{(\vec{e}_{ijk}-\sin\theta_l\vec{e}_{il})}{r_{il}}(\delta_{ln}-\delta_{in})\right\}\Bigg),\end{aligned} \tag{5.86}$$

in which $\phi'^{4\mathrm{bdy}}_{ijkl}(x)$ is the derivative of $\phi'^{4\mathrm{bdy}}_{ijkl}$ with respect to the argument x. We have also defined the following unit vectors

$$\begin{aligned}\vec{e}_{ij}&=\vec{r}_{ij}/r_{ij}, \qquad \vec{e}_{ikl}=(\vec{r}_{ik}\times\vec{r}_{il})/|\vec{r}_{ik}\times\vec{r}_{il}|,\\ \vec{e}_{ik}&=\vec{r}_{ik}/r_{ik}, \qquad \vec{e}_{ilj}=(\vec{r}_{il}\times\vec{r}_{ij})/|\vec{r}_{il}\times\vec{r}_{ij}|,\\ \vec{e}_{il}&=\vec{r}_{il}/r_{il}, \qquad \vec{e}_{ijk}=(\vec{r}_{ij}\times\vec{r}_{ik})/|\vec{r}_{ij}\times\vec{r}_{ik}|.\end{aligned} \tag{5.87}$$

We now need to be specific about the identity of atom n. We note from the potential energy expression (5.82) central atoms (indexed i) are treated differently from those that are non-central. We treat this case first and first set $n=i$ to obtain:

$$(\vec{f}_n^{\,4\mathrm{bdy}})_c = -\sum_{j=1}^{N-2}{}' \sum_{k>j}^{N-1}{}' \sum_{l>k}^{N}{}' \left(\frac{\phi'^{4\mathrm{bdy}}_{njkl}(\theta_j)}{\cos\theta_j} \left\{ \frac{(\vec{e}_{nj} - \sin\theta_j \vec{e}_{nkl})}{|\vec{r}_{nk} \times \vec{r}_{nl}|} \times [\vec{r}_{nl} - \vec{r}_{nk}] - \frac{(\vec{e}_{nkl} - \sin\theta_j \vec{e}_{nj})}{r_{nj}} \right\} \right.$$
$$+ \frac{\phi'^{4\mathrm{bdy}}_{njkl}(\theta_k)}{\cos\theta_k} \left\{ \frac{(\vec{e}_{nk} - \sin\theta_k \vec{e}_{nlj})}{|\vec{r}_{nl} \times \vec{r}_{nj}|} \times [\vec{r}_{nj} - \vec{r}_{nl}] - \frac{(\vec{e}_{nlj} - \sin\theta_k \vec{e}_{nk})}{r_{nk}} \right\} \qquad (5.88)$$
$$\left. + \frac{\phi'^{4\mathrm{bdy}}_{njkl}(\theta_l)}{\cos\theta_l} \left\{ \frac{(\vec{e}_{nl} - \sin\theta_l \vec{e}_{njk})}{|\vec{r}_{nj} \times \vec{r}_{nk}|} \times [\vec{r}_{nk} - \vec{r}_{nj}] - \frac{(\vec{e}_{njk} - \sin\theta_l \vec{e}_{nl})}{r_{nl}} \right\} \right),$$

In which the dash $(')$ against the summation symbol indicates the value n is forbidden. This formula defines the force on an atom that is central in sense shown in figure 5.6. (Hence the subscript c (meaning *central*) in $(\vec{f}_n^{\,4\mathrm{bdy}})_c$.) Now considering the case where n is one of j, k, or l, leads to the formula:

$$(\vec{f}_n^{\,4\mathrm{bdy}})_p = -\sum_{i\neq n}^{N} \sum_{k>n}^{N-1}{}' \sum_{l>k}^{N}{}' \left(\frac{\phi'^{4\mathrm{bdy}}_{inkl}(\theta_n)}{\cos\theta_n} \left\{ \frac{(\vec{e}_{ikl} - \sin\theta_n \vec{e}_{in})}{r_{in}} \right\} + \frac{\phi'^{4\mathrm{bdy}}_{inkl}(\theta_k)}{\cos\theta_k} \left\{ \frac{(\vec{e}_{ik} - \sin\theta_k \vec{e}_{iln})}{|\vec{r}_{il} \times \vec{r}_{in}|} \times \vec{r}_{il} \right\} \right.$$
$$\left. - \frac{\phi'^{4\mathrm{bdy}}_{inkl}(\theta_l)}{\cos\theta_l} \left\{ \frac{(\vec{e}_{il} - \sin\theta_l \vec{e}_{ink})}{|\vec{r}_{in} \times \vec{r}_{ik}|} \times \vec{r}_{ik} \right\} \right)$$
$$-\sum_{i\neq n}^{N} \sum_{j=1}^{n-1}{}' \sum_{l>n}^{N}{}' \left(\frac{\phi'^{4\mathrm{bdy}}_{ijnl}(\theta_n)}{\cos\theta_n} \left\{ \frac{(\vec{e}_{ilj} - \sin\theta_n \vec{e}_{in})}{r_{in}} \right\} + \frac{\phi'^{4\mathrm{bdy}}_{ijnl}(\theta_l)}{\cos\theta_l} \left\{ \frac{(\vec{e}_{il} - \sin\theta_l \vec{e}_{ijn})}{|\vec{r}_{ij} \times \vec{r}_{in}|} \times \vec{r}_{ij} \right\} \right.$$
$$\left. - \frac{\phi'^{4\mathrm{bdy}}_{ijnl}(\theta_j)}{\cos\theta_j} \left\{ \frac{(\vec{e}_{ij} - \sin\theta_j \vec{e}_{inl})}{|\vec{r}_{in} \times \vec{r}_{il}|} \times \vec{r}_{il} \right\} \right) \qquad (5.89)$$
$$-\sum_{i\neq n}^{N} \sum_{j=1}^{n-2}{}' \sum_{k>j}^{n-1}{}' \left(\frac{\phi'^{4\mathrm{bdy}}_{ijkn}(\theta_n)}{\cos\theta_n} \left\{ \frac{(\vec{e}_{ijk} - \sin\theta_n \vec{e}_{in})}{r_{in}} \right\} + \frac{\phi'^{4\mathrm{bdy}}_{ijkn}(\theta_j)}{\cos\theta_j} \left\{ \frac{(\vec{e}_{ij} - \sin\theta_j \vec{e}_{ikn})}{|\vec{r}_{ik} \times \vec{r}_{in}|} \times \vec{r}_{ik} \right\} \right.$$
$$\left. - \frac{\phi'^{4\mathrm{bdy}}_{ijkn}(\theta_k)}{\cos\theta_k} \left\{ \frac{(\vec{e}_{ik} - \sin\theta_k \vec{e}_{inj})}{|\vec{r}_{in} \times \vec{r}_{ij}|} \times \vec{r}_{ij} \right\} \right),$$

in which the dash $(')$ against a summation indicates that the index value i is forbidden, while the condition $i \neq n$ forbids the value n, but allows all others values in the relevant summations. Equation (5.89) applies whenever n represents a non-central atom. (Hence the subscript p , meaning *peripheral*, in $(\vec{f}_n^{\,4\mathrm{bdy}})_p$.)

Equation (5.89) can be simplified by a re-definition of indices j, k, l to give

$$(\vec{f}_n^{\,4\mathrm{bdy}})_p = -\sum_{i\neq n}^{N} \sum_{j=1}^{N-1}{}'' \sum_{k>j}^{N}{}'' \left(\frac{\phi'^{4\mathrm{bdy}}_{ijkn}(\theta_n)}{\cos\theta_n} \left\{ \frac{(\vec{e}_{ijk} - \sin\theta_n \vec{e}_{in})}{r_{in}} \right\} + \frac{\phi'^{4\mathrm{bdy}}_{ijkn}(\theta_j)}{\cos\theta_j} \left\{ \frac{(\vec{e}_{ij} - \sin\theta_j \vec{e}_{ikn})}{|\vec{r}_{ik} \times \vec{r}_{in}|} \times \vec{r}_{i} \right\} \right.$$
$$\left. - \frac{\phi'^{4\mathrm{bdy}}_{ijkn}(\theta_k)}{\cos\theta_k} \left\{ \frac{(\vec{e}_{ik} - \sin\theta_k \vec{e}_{inj})}{|\vec{r}_{in} \times \vec{r}_{ij}|} \times \vec{r}_{ij} \right\} \right), \qquad (5.90)$$

in which the double dash $('')$ indicates the values n and i are forbidden and the first summation allows all values of i except n.

From (5.90) it can be shown that for a specific four-body interaction the contribution to the forces on peripheral atoms j, k and l are:

$$\begin{aligned}
\vec{f}_j^{\,4\mathrm{bdy}} \leftarrow -\Bigg(& \frac{\phi'^{4\mathrm{bdy}}_{ijkl}(\theta_j)}{\cos\theta_j}\left\{\frac{(\vec{e}_{ikl}-\sin\theta_j\vec{e}_{ij})}{r_{ij}}\right\}+\frac{\phi'^{4\mathrm{bdy}}_{ijkl}(\theta_k)}{\cos\theta_k}\left\{\frac{(\vec{e}_{ik}-\sin\theta_k\vec{e}_{ilj})}{|\vec{r}_{il}\times\vec{r}_{ij}|}\times\vec{r}_{il}\right\} \\
& -\frac{\phi'^{4\mathrm{bdy}}_{ijkl}(\theta_l)}{\cos\theta_l}\left\{\frac{(\vec{e}_{il}-\sin\theta_l\vec{e}_{ijk})}{|\vec{r}_{ij}\times\vec{r}_{ik}|}\times\vec{r}_{ik}\right\}\Bigg), \\
\vec{f}_k^{\,4\mathrm{bdy}} \leftarrow -\Bigg(& \frac{\phi'^{4\mathrm{bdy}}_{ijkl}(\theta_k)}{\cos\theta_k}\left\{\frac{(\vec{e}_{ilj}-\sin\theta_k\vec{e}_{ik})}{r_{ik}}\right\}+\frac{\phi'^{4\mathrm{bdy}}_{ijkl}(\theta_l)}{\cos\theta_l}\left\{\frac{(\vec{e}_{il}-\sin\theta_l\vec{e}_{ijk})}{|\vec{r}_{ij}\times\vec{r}_{ik}|}\times\vec{r}_{ij}\right\} \\
& -\frac{\phi'^{4\mathrm{bdy}}_{ijkl}(\theta_j)}{\cos\theta_j}\left\{\frac{(\vec{e}_{ij}-\sin\theta_j\vec{e}_{ikl})}{|\vec{r}_{ik}\times\vec{r}_{il}|}\times\vec{r}_{il}\right\}\Bigg), \\
\vec{f}_l^{\,4\mathrm{bdy}} \leftarrow -\Bigg(& \frac{\phi'^{4\mathrm{bdy}}_{ijkl}(\theta_l)}{\cos\theta_l}\left\{\frac{(\vec{e}_{ijk}-\sin\theta_l\vec{e}_{il})}{r_{il}}\right\}+\frac{\phi'^{4\mathrm{bdy}}_{ijkl}(\theta_j)}{\cos\theta_j}\left\{\frac{(\vec{e}_{ij}-\sin\theta_j\vec{e}_{ikl})}{|\vec{r}_{ik}\times\vec{r}_{il}|}\times\vec{r}_{ik}\right\} \\
& -\frac{\phi'^{4\mathrm{bdy}}_{ijkl}(\theta_k)}{\cos\theta_k}\left\{\frac{(\vec{e}_{ik}-\sin\theta_k\vec{e}_{ilj})}{|\vec{r}_{il}\times\vec{r}_{ij}|}\times\vec{r}_{ij}\right\}\Bigg).
\end{aligned} \tag{5.91}$$

According to (5.88) the force contribution to a central atom i is

$$\begin{aligned}
\vec{f}_i^{\,4\mathrm{bdy}} \leftarrow -\Bigg(& \frac{\phi'^{4\mathrm{bdy}}_{ijkl}(\theta_j)}{\cos\theta_j}\left\{\frac{(\vec{e}_{ij}-\sin\theta_j\vec{e}_{ikl})}{|\vec{r}_{ik}\times\vec{r}_{il}|}\times[\vec{r}_{il}-\vec{r}_{ik}]-\frac{(\vec{e}_{ikl}-\sin\theta_j\vec{e}_{ij})}{r_{ij}}\right\} \\
& +\frac{\phi'^{4\mathrm{bdy}}_{ijkl}(\theta_k)}{\cos\theta_k}\left\{\frac{(\vec{e}_{ik}-\sin\theta_k\vec{e}_{ilj})}{|\vec{r}_{il}\times\vec{r}_{ij}|}\times[\vec{r}_{ij}-\vec{r}_{il}]-\frac{(\vec{e}_{ilj}-\sin\theta_k\vec{e}_{ik})}{r_{ik}}\right\} \\
& +\frac{\phi'^{4\mathrm{bdy}}_{ijkl}(\theta_l)}{\cos\theta_l}\left\{\frac{(\vec{e}_{il}-\sin\theta_l\vec{e}_{ijk})}{|\vec{r}_{ij}\times\vec{r}_{ik}|}\times[\vec{r}_{ik}-\vec{r}_{ij}]-\frac{(\vec{e}_{ijk}-\sin\theta_l\vec{e}_{il})}{r_{il}}\right\}\Bigg).
\end{aligned} \tag{5.92}$$

Comparison of (5.91) with (5.92) reveals that the sum of the forces from one four-body potential acting on all the atoms i, j, k, l is zero as expected.

It is of course possible for an atom to feature as both a central and a peripheral atom, in which case both sets of formulae (5.91) and (5.92) apply, though not at the same time in a computational scheme running over the central atom i. The point is that (5.92) contains all the quantities that need to be calculated for a given four-body potential. A change of sign and reallocation of these quantities according to (5.91) deals with the peripheral atoms.

An example four-body potential for maintaining the planarity of a four-atom group is

$$\phi^{4\mathrm{bdy}}_{ijkl}(\theta_j,\theta_k,\theta_l)=A(3-\cos\theta_j-\cos\theta_k-\cos\theta_l). \tag{5.93}$$

and the required derivatives are

$$\phi'^{invers}_{d}(\theta_j) = A\sin\theta_j, \qquad \phi'^{invers}_{d}(\theta_k) = A\sin\theta_k, \qquad \phi'^{invers}_{d}(\theta_l) = A\sin\theta_l. \tag{5.94}$$

It should be noted that, since this represents a non-bonded interaction, it is assumed that there are other atom-atom interactions also present to maintain the atomic structure within the plane. Additional three-body angular forces may also be required, as described in section 5.3.2 .

5.3.4 Covalent Forces – The Tersoff Potential

It will be apparent from the previous sections that 3- and 4-body forces are often introduced into the description of non-bonded inter-molecular interactions as a way of incorporating what are essentially covalent (particularly angular) effects into the atomic structure of the system. This is particularly required when modelling amorphous systems like glasses, which are disordered on a larger scale but show a high degree of *local order* around each atom. Amorphous structures often emerge when systems are quenched from a high temperature melt, in which case it is not possible to define a fixed bonding template (i.e. a molecular structure) from the outset. It may also be the case that such systems can show different bonding characteristics depending on density as, for example, in diamond and graphite. This aspect of the potential needs to be included in a way that does not require direct recourse to quantum mechanical treatment.

The Tersoff potential [42] is an important example of this kind of potential. It is a density dependent potential (meaning that it incorporates many-body effects) and exhibits *variable valency* with regard to bonding between atoms. It was designed primarily for elements in group IV of the periodic table, which includes carbon, silicon and germanium. Superficially it resembles a pair potential (to describe covalent bonds) but its parameters include a density dependence which makes it sensitive to the presence of other atoms in the vicinity. From this density dependence arises an angular bonding that helps coordinate the local structure.

Formally the system Tersoff configuration energy is written as

$$\Phi(\vec{r}^N) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j\neq i}^{N}\phi_{ij}^{T}(r_{ij}), \tag{5.95}$$

where $\phi_{ij}^{T}(r_{ij})$ may (loosely) be called the Tersoff *pair potential* which has the form

$$\phi(r_{ij})^{T} = f_C(r_{ij})[f_A(r_{ij}) - \gamma_{ij}(\vec{r}^N) f_B(r_{ij})], \tag{5.96}$$

where

$$f_A(r_{ij}) = A_{ij}\exp(-a_{ij}r_{ij}), \qquad f_B(r_{ij}) = B_{ij}\exp(-b_{ij}r_{ij}), \tag{5.97}$$

are *pair repulsive* and *pair attractive* terms respectively, which are characterised by the parameters $A_{ij},\ B_{ij},\ a_{ij},\ b_{ij}.$ Also, the function

$$f_C(r_{ij}) = \begin{cases} 1 & : r_{ij} \le R_{ij} \\ \frac{1}{2} + \frac{1}{2}\cos[\pi(r_{ij} - R_{ij})/(S_{ij} - R_{ij})] & : R_{ij} < r_{ij} < S_{ij} \\ 0 & : r_{ij} \ge S_{ij} \end{cases} \tag{5.98}$$

is a *truncation function* ensuring that the bond "switches off" smoothly at a range beyond the standard bond length. The parameters R_{ij} and S_{ij} define the range over which the switch off occurs, where $(R_{ij} < S_{ij})$. Lastly the function $\gamma_{ij}(\vec{r}^N)$ represents a *density dependent* parameter which controls the strength of the bond between atoms i and j, together with any directional (bond angle) effects. It has the following form

$$\gamma_{ij}(\vec{r}^N) = \chi_{ij}\left(1 + \beta_i^{\eta_i} L_{ij}^{\eta_i}(\vec{r}^N)\right)^{-1/(2\eta_i)}, \tag{5.99}$$

where $\chi_{ij}, \beta_{ij}, \eta_i$ are parameters and the function $L_{ij}(\vec{r}^N)$ is

$$L_{ij}(\vec{r}^N) = \sum_{k \ne i,\, j}^{N} \omega_{ik} f_C(r_{ik}) g(\theta_{jik}), \tag{5.100}$$

in which $f_C(r_{ik})$ is the truncation function (5.98), ω_{ik} is a parameter and $g(\theta_{jik})$ is an angular function of the positions of the atoms i, j and *k:*

$$g(\theta_{jik}) = 1 + c_i^2/d_i^2 - c_i^2/[d_i^2 + (h_i - \cos\theta_{jik})^2], \tag{5.101}$$

where c_i, d_i and h_i are parameters and θ_{jik} is the angle between the atomic displacement vectors $\vec{r}_{ij}$ and $\vec{r}_{ik}$ i.e.

$$\theta_{jik} = \arccos\left(\frac{\vec{r}_{ij} \cdot \vec{r}_{ik}}{r_{ij} r_{ik}}\right). \tag{5.102}$$

The Tersoff potential may be visualised as follows. A given atom i in close proximity to an atom j (i.e. such that the truncation function $f_C(r_{ij})$ is greater than zero), forms a bond with atom j, the strength of which depends on the interaction atom i has with other atoms $\{k\}$ in the same vicinity. If any of these atoms are close enough to form bonds with atom i, the function $\gamma_{ij}(\vec{r}^N)$ moderates the strength of the i - j bond and imposes an angular dependence through (5.101). By this token, $\gamma_{ij}(\vec{r}^N)$ is density dependent, since it varies with the number of atoms in the vicinity of atom i. It should be clear from this that unless the atoms i and j are in identical environments (i.e. have the same number of bonded neighbours), the bond made by atom i with j will not be the same as the bond made by atom j with i. So the summations appearing in (5.95) must range over *both* occurrences of the i, j pair.

It is also worth noting that the truncation function f_c generally makes the interaction short ranged. So the sum over j in (5.95) (and likewise the sum over k in (5.100) does not really extend over the full list of N atoms, which represents

a major cost reduction. However, the sum over i in (5.95) is inevitably of order N.

The Tersoff potential has twelve parameters in all. This is a relatively large parameter set, implying that fitting the potential to experimental data is a tricky task. However, not all the parameters are utilised for fitting; some take assumed values. In the case of mixed system (e.g. semiconductor alloys) it is assumed that the parameters can be resolved into contributions from specific atoms in the following manner:

$$\begin{aligned} A_{ij} &= \sqrt{A_i A_j}, & B_{ij} &= \sqrt{B_i B_j}, \\ S_{ij} &= \sqrt{S_i S_j}, & R_{ij} &= \sqrt{R_i R_j}, \\ a_{ij} &= (a_i + a_j)/2, & b_{ij} &= (b_i + b_j)/2, \end{aligned} \tag{5.103}$$

then if atoms i and j differ in species type, these equations define appropriate mixing rules. For the remaining parameters we have

$$\chi_{ij} = 1, \qquad \omega_{ij} = 1, \tag{5.104}$$

if i and j are the same species and

$$\chi_{ij} = \chi_{ji}, \qquad \omega_{ij} = \omega_{ji}, \tag{5.105}$$

if they are different species. Note that (5.105) may not hold in general if a more flexible fitting scheme is adopted.

The force on an arbitrary atom n, is formally given by

$$\vec{f}_n = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j\neq i}^{N}\frac{\partial \phi_{ij}^{T}(r_{ij})}{\partial \vec{r}_n}, \tag{5.106}$$

where

$$\begin{aligned} \frac{\partial \phi_{ij}^{T}(r_{ij})}{\partial \vec{r}_n} = {} & [f_A(r_{ij}) - \gamma_{ij}(\vec{r}^{\,N}) f_B(r_{ij})] f'_C(r_{ij}) \frac{\vec{r}_{ij}}{r_{ij}} (\delta_{jn} - \delta_{in}) + \\ & f_C(r_{ij}) \left[f'_A(r_{ij}) - \gamma_{ij}(\vec{r}^{\,N}) f'_B(r_{ij}) \right] \frac{\vec{r}_{ij}}{r_{ij}} (\delta_{jn} - \delta_{in}) - \\ & f_C(r_{ij}) f_B(r_{ij}) \frac{\partial \gamma_{ij}(\vec{r}^{\,N})}{\partial \vec{r}_n}, \end{aligned} \tag{5.107}$$

where a dash $(')$ indicates a first derivative with respect to the argument i.e.

$$f'_A(r_{ij}) = -a_{ij} A_{ij} \exp(-a_{ij} r_{ij}), \qquad f'_B(r_{ij}) = -b_{ij} B_{ij} \exp(-b_{ij} r_{ij}), \tag{5.108}$$

$$f'_C(r_{ij})=\begin{cases} 0 & : r_{ij}<R_{ij} \\ \dfrac{-\pi}{2(S_{ij}-R_{ij})}\sin[\pi(r_{ij}-R_{ij})/(S_{ij}-R_{ij})] & : R_{ij}<r_{ij}<S_{ij} \\ 0 & : r_{ij}>S_{ij}. \end{cases} \tag{5.109}$$

We also have

$$\frac{\partial \gamma_{ij}(\vec{r}^N)}{\partial \vec{r}_n}=-\frac{\chi_{ij}\beta_i^{\eta_i}}{2}L_{ij}^{\eta_i-1}(\vec{r}^N)(1+\beta_i^{\eta_i}L_{ij}^{\eta_i}(\vec{r}^N))^{-1/(2\eta_i)-1}\frac{\partial L_{ij}(\vec{r}^N)}{\partial \vec{r}_n}, \tag{5.110}$$

with

$$\begin{aligned}\frac{\partial L_{ij}(\vec{r}^N)}{\partial \vec{r}_n}=\sum_{k\neq i,j}^{N}\omega_{ik}\Bigg[& g(\theta_{jik})f'_C(r_{ik})\frac{\vec{r}_{ik}}{r_{ik}}(\delta_{kn}-\delta_{in})-f_C(r_{ik})\frac{g'(\theta_{jik})}{\sin\theta_{jik}} \\ & \left\{\frac{1}{r_{ij}r_{ik}}[\vec{r}_{ij}(\delta_{kn}-\delta_{in})+\vec{r}_{ik}(\delta_{jn}-\delta_{in})]-\right. \\ & \left.\cos\theta_{jik}\left[\frac{\vec{r}_{ij}}{r_{ij}^2}(\delta_{jn}-\delta_{in})+\frac{\vec{r}_{ik}}{r_{ik}^2}(\delta_{kn}-\delta_{in})\right]\right\}\Bigg],\end{aligned} \tag{5.111}$$

and

$$g'(\theta_{jik})=\frac{2c_i^2\sin\theta_{jik}(h_i-\cos\theta_{jik})}{[d_i^2+(h_i-\cos\theta_{jik})^2]^2}. \tag{5.112}$$

Notice that the $\sin\theta_{jik}$ appearing in this formula cancels that appearing in (5.111) (5.111) so it is prudent to remove it from both when expressing them in computer code.

Combining the equations (5.106) to (5.112) gives the net force on the arbitrary atom n as

$$\begin{aligned}\vec{f}_n=-\frac{1}{2}\sum_{i=1}^{N}\sum_{j\neq i}^{N}\Bigg[& [f_A(r_{ij})-\gamma_{ij}(\vec{r}^N)f_B(r_{ij})]f'_C(r_{ij})\frac{\vec{r}_{ij}}{r_{ij}}(\delta_{jn}-\delta_{in})+ \\ & f_C(r_{ij})\left[f'_A(r_{ij})-\gamma_{ij}(\vec{r}^N)f'_B(r_{ij})\right]\frac{\vec{r}_{ij}}{r_{ij}}(\delta_{jn}-\delta_{in})+ \\ & f_C(r_{ij})f_B(r_{ij})\frac{\chi_{ij}\beta_i^{\eta_i}}{2}L_{ij}^{\eta_i-1}(\vec{r}^N)(1+\beta_i^{\eta_i}L_{ij}^{\eta_i}(\vec{r}^N))^{-1/(2\eta_i)-1} \\ & \sum_{k\neq i,j}^{N}\omega_{ik}\Bigg[g(\theta_{jik})f'_C(r_{ik})\frac{\vec{r}_{ik}}{r_{ik}}(\delta_{kn}-\delta_{in})-f_C(r_{ik})\frac{g'(\theta_{jik})}{\sin\theta_{jik}} \\ & \left\{\frac{1}{r_{ij}r_{ik}}[\vec{r}_{ij}(\delta_{kn}-\delta_{in})+\vec{r}_{ik}(\delta_{jn}-\delta_{in})]-\right. \\ & \left.\cos\theta_{jik}\left[\frac{\vec{r}_{ij}}{r_{ij}^2}(\delta_{jn}-\delta_{in})+\frac{\vec{r}_{ik}}{r_{ik}^2}(\delta_{kn}-\delta_{in})\right]\right\}\Bigg],\end{aligned} \tag{5.113}$$

When coding an algorithm to calculate the atomic forces, it is helpful to recall that the sums over atoms i in equations (5.95) and (5.106) represent the search for atoms that are central in each angular term associated with the angle θ_{jik} in the above formulae. Having identified atom i, the search then branches to locate the atoms j and k, which are assumed to be spatially close to atom i. Note however, that these summations generate *both* θ_{jik} and θ_{kij}, which represent the same angle, but allow for the possibility that $\omega_{ik} \neq \omega_{jk}$, so these two cases are not necessarily identical with regard to potential energy or force contributions. They are certainly different for alloys.

Equation (5.113) is the most general form for the net force on an atom n. This atom can occur as any one of the three indexed atoms i, j, k which identify three different rôles for the atom in the full force scheme. Atoms indexed by i represent those atoms at the centre of the incipient bond angle formed by the triplet i, j, k. An atom with index j is the one with which atom i forms the bond represented by the potential function (5.96). Lastly atom k is one which, through its proximity to i, moderates the $i - j$ bond. These physically distinct rôles provide different contributions to the net force on each atom. Therefore the net force acting on an atom because of its rôle as the centre of an angular bond is given by

$$\begin{aligned}
\vec{f}_n = \frac{1}{2}\sum_{j\neq n}^{N} & [f_A(r_{nj}) - \gamma_{nj}(\vec{r}^{\,N}) f_B(r_{nj})] f'_C(r_{nj}) \frac{\vec{r}_{nj}}{r_{nj}} + \\
& f_C(r_{nj}) \left[f'_A(r_{nj}) - \gamma_{nj}(\vec{r}^{\,N}) f'_B(r_{nj}) \right] \frac{\vec{r}_{nj}}{r_{nj}} + \\
& f_C(r_{nj}) f_B(r_{nj}) \frac{\chi_{nj}\beta_n^{\eta_n}}{2} L_{nj}^{\eta_n - 1}(\vec{r}^{\,N}) \left(1 + \beta_n^{\eta_n} L_{nj}^{\eta_n}(\vec{r}^{\,N})\right)^{-1/(2\eta_n)-1} \\
\sum_{k\neq n,\, j}^{N} & \omega_{nk} \left[g(\theta_{jnk}) f'_C(r_{nk}) \frac{\vec{r}_{nk}}{r_{nk}} - f_C(r_{nk}) \frac{g'(\theta_{jnk})}{\sin\theta_{jnk}} \right. \\
& \left. \left\{ \frac{1}{r_{nj} r_{nk}} [\vec{r}_{nj} + \vec{r}_{nk}] - \cos\theta_{jnk} \left[\frac{\vec{r}_{nj}}{r_{nj}^2} + \frac{\vec{r}_{nk}}{r_{nk}^2} \right] \right\} \right],
\end{aligned} \tag{5.114}$$

while the net force arising from an atom's rôle as a pair bonding (non-central) atom is

$$\begin{aligned}
\vec{f}_n = -\frac{1}{2}\sum_{i\neq n}^{N} & [f_A(r_{in}) - \gamma_{in}(\vec{r}^{\,N}) f_B(r_{in})] f'_C(r_{in}) \frac{\vec{r}_{in}}{r_{in}} + \\
& f_C(r_{in}) \left[f'_A(r_{in}) - \gamma_{in}(\vec{r}^{\,N}) f'_B(r_{in}) \right] \frac{\vec{r}_{in}}{r_{in}} - \\
& f_C(r_{in}) f_B(r_{in}) \frac{\chi_{in}\beta_i^{\eta_i}}{2} L_{in}^{\eta_i - 1}(\vec{r}^{\,N}) \left(1 + \beta_i^{\eta_i} L_{in}^{\eta_i}(\vec{r}^{\,N})\right)^{-1/(2\eta_i)-1} \\
\sum_{k\neq i,\, n}^{N} & \omega_{ik} \left[f_C(r_{ik}) \frac{g'(\theta_{nik})}{\sin\theta_{nik}} \left\{ \frac{\vec{r}_{ik}}{r_{in} r_{ik}} - \cos\theta_{nik} \left[\frac{\vec{r}_{in}}{r_{in}^2} \right] \right\} \right],
\end{aligned} \tag{5.115}$$

and the net force from an atom's rôle as a moderating atom is

$$\vec{f}_n = -\frac{1}{2}\sum_{i=1}^{N}{}'\sum_{j\neq i}^{N}{}' f_C(r_{ij}) f_B(r_{ij}) \frac{\chi_{ij}\beta_i^{\eta_i}}{2} L_{ij}^{\eta_i - 1}(\vec{r}^N)\left(1+\beta_i^{\eta_i} L_{ij}^{\eta_i}(\vec{r}^N)\right)^{-1/(2\eta_i)-1} \omega_{in}$$
$$\left[g(\theta_{jin}) f'_C(r_{in}) \frac{\vec{r}_{in}}{r_{in}} - f_C(r_{in}) \frac{g'(\theta_{jin})}{\sin\theta_{jin}} \left\{ \frac{\vec{r}_{ij}}{r_{ij} r_{in}} - \cos\theta_{jin} \left[\frac{\vec{r}_{in}}{r_{in}^2} \right] \right\} \right], \qquad (5.116)$$

in which the dash $(')$ on the summations indicates that $i, j \neq n$.

Considering the forces from the viewpoint of a triplet of atoms i, j, k where atom i takes the rôle of central atom, j the second bond atom and k the moderating atom, the equations (5.114) - (5.116) provide the contribution to the forces on each of these atoms arising from the Tersoff potential, which are shown in equations (5.117). Once again, the sum of all the contributions in (5.117) is zero, correctly indicating that the Tersoff potential does not exert a net force on the system.

$$\vec{f}_i \leftarrow \frac{1}{2}\Bigg([f_A(r_{ij}) - \gamma_{ij}(\vec{r}^N) f_B(r_{ij})] f'_C(r_{ij}) \frac{\vec{r}_{ij}}{r_{ij}} +$$
$$f_C(r_{ij}) \left[f'_A(r_{ij}) - \gamma_{ij}(\vec{r}^N) f'_B(r_{ij}) \right] \frac{\vec{r}_{ij}}{r_{ij}} +$$
$$f_C(r_{ij}) f_B(r_{ij}) \frac{\chi_{ij}\beta_i^{\eta_i}}{2} L_{ij}^{\eta_i - 1}(\vec{r}^N)\left(1+\beta_i^{\eta_i} L_{ij}^{\eta_i}(\vec{r}^N)\right)^{-1/(2\eta_i)-1}$$
$$\omega_{ik} \Bigg[g(\theta_{jik}) f'_C(r_{ik}) \frac{\vec{r}_{ik}}{r_{ik}} - f_C(r_{ik}) \frac{g'(\theta_{jik})}{\sin\theta_{jik}}$$
$$\left\{ \frac{1}{r_{ij} r_{ik}} [\vec{r}_{ij} + \vec{r}_{ik}] - \cos\theta_{jik} \left[\frac{\vec{r}_{ij}}{r_{ij}^2} + \frac{\vec{r}_{ik}}{r_{ik}^2} \right] \right\} \Bigg] \Bigg),$$

$$\vec{f}_j \leftarrow -\frac{1}{2}\Bigg([f_A(r_{ij}) - \gamma_{ij}(\vec{r}^N) f_B(r_{ij})] f'_C(r_{ij}) \frac{\vec{r}_{ij}}{r_{ij}} + \qquad (5.117)$$
$$f_C(r_{ij}) \left[f'_A(r_{ij}) - \gamma_{ij}(\vec{r}^N) f'_B(r_{ij}) \right] \frac{\vec{r}_{ij}}{r_{ij}} -$$
$$f_C(r_{ij}) f_B(r_{ij}) \frac{\chi_{ij}\beta_i^{\eta_i}}{2} L_{ij}^{\eta_i - 1}(\vec{r}^N)\left(1+\beta_i^{\eta_i} L_{ij}^{\eta_i}(\vec{r}^N)\right)^{-1/(2\eta_i)-1}$$
$$\omega_{ik} \Bigg[f_C(r_{ik}) \frac{g'(\theta_{jik})}{\sin\theta_{jik}} \left\{ \frac{\vec{r}_{ik}}{r_{ij} r_{ik}} - \cos\theta_{jik} \left[\frac{\vec{r}_{ij}}{r_{ij}^2} \right] \right\} \Bigg] \Bigg),$$

$$\vec{f}_k \leftarrow -\frac{1}{2} f_C(r_{ij}) f_B(r_{ij}) \frac{\chi_{ij}\beta_i^{\eta_i}}{2} L_{ij}^{\eta_i - 1}(\vec{r}^N)\left(1+\beta_i^{\eta_i} L_{ij}^{\eta_i}(\vec{r}^N)\right)^{-1/(2\eta_i)-1} \omega_{ik}$$
$$\left[g(\theta_{jik}) f'_C(r_{ik}) \frac{\vec{r}_{ik}}{r_{ik}} - f_C(r_{ik}) \frac{g'(\theta_{jik})}{\sin\theta_{jik}} \left\{ \frac{\vec{r}_{ij}}{r_{ij} r_{ik}} - \cos\theta_{jik} \left[\frac{\vec{r}_{ik}}{r_{ik}^2} \right] \right\} \right].$$

At first glance the equations (5.117) suggest that a practical scheme to calculate the forces would require a simple run over atoms i (a loop of size *O(N)*), and for each i a local search over near neighbour atoms to locate atoms j and k, then the forces on the three atoms could be computed via (5.117). Such a scheme is thwarted however by the dependence of the forces on the terms $\gamma_{ij}(\vec{r}^N)$ and $L_{ij}(\vec{r}^N)$, which indicates that a complete loop over k is required for each pair i, j before the forces $\vec{f}_i$ and $\vec{f}_j$ (though not $\vec{f}_k$) can be obtained from (5.117).

5.3.5 Metal Forces

Potentials for metals differ markedly from most non-bonded potentials in that they cannot be described by simplistic n-body interactions in the manner of preceding sections. This is because metals have fully de-localised electrons that properly require a quantum mechanical treatment that is not compatible with the classical procedures of molecular dynamics. Nevertheless, it is possible to account for some aspects of metallic bonding in an empirical way by introducing density dependence as a factor. The resulting potentials and forces are then many-body in kind. At first glance this implies that metal potentials have a high-order dependence on the number of atoms in the system simulated and are computationally expensive to incorporate. However, by careful design of the empirical density function, such potentials need not scale in cost beyond the $O(N^2)$ of pair potentials, as this section will show.

There are two approaches to constructing an empirical metal potential, though in fact they look very similar. The first is the Embedded Atom Model (EAM) [43],[44] and the second is the Finnis-Sinclair Model (FSM) [45]. These potentials are suitable for calculating structural, or thermodynamic, properties of metals, but (obviously) not electronic properties. The configuration energy for both of these can be written in the form:

$$\Phi(\vec{r}^N)=\frac{1}{2}\sum_{i=1}^{N}\sum_{j\neq i}^{N}\phi_{ij}^{metal}(r_{ij})+\sum_{i=1}^{N}F_i^{metal}(\rho_i), \qquad (5.118)$$

in which the first (double) sum on the right represents a repulsive pair potential arising from electrostatic interactions and the inter-penetration of bound electronic orbitals on each atom and the second sum represents the embedding energy of the atoms in the bulk metal. The latter is dependent on the density, ρ_i, surrounding each atom i, through the function F_i^{metal}. In both the EAM and FSM the density can be written as

$$\rho_i=\sum_{j\neq i}^{N}\rho_{ij}(r_{ij}), \qquad (5.119)$$

which indicates the the density is calculated from the atomic pair density, or the coordination number of atoms j surrounding atom i. (The range of j in the summation in (5.119) is 1 to N, except that condition $i\neq j$ applies.)

The pair potential $\phi_{ij}(r_{ij})$ is often very simple in FSM (see examples below), but it is

rarely analytical in form for EAM and is supplied in the literature in a tabulated form, for which interpolation and numerical differentiation are necessary to obtain the required potential energy and atomic forces. This one of the main differences between the EAM and FSM metal potentials.

The other main difference is in the treatment of alloys. In the EAM [46] the contribution an atom j makes to the density (ρ_i) of atom i is different from that atom i makes to the density (ρ_j) of atom j. But in the FSM treatment of alloys [47], these two contributions are equal. In fact for two metal species A and B we have

$$\begin{aligned} \rho_{ij}^{AB} &= \rho_{ij}^{BB} \qquad \rho_{ij}^{BA} = \rho_{ij}^{AA}, && (EAM) \\ \rho_{ij}^{AB} &= \rho_{ij}^{BA} = \sqrt{\rho_{ij}^{AA}\,\rho_{ij}^{BB}}. && (FSM) \end{aligned} \tag{5.120}$$

We should also note that for both models

$$\phi_{ij}^{AB}(r_{ij}) = \phi_{ij}^{BA}(r_{ij}), \tag{5.121}$$

though the *mixing rules* are different for EAM [46] and FSM [47].

The force on an arbitrary metal atom n is given by

$$\vec{f}_n = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j\neq i}^{N} \phi'^{metal}_{ij}(r_{ij})\frac{\partial r_{ij}}{\partial \vec{r}_n} - \sum_{i=1}^{N} F'^{metal}_{i}(\rho_i)\sum_{j\neq i}^{N} \rho'_{ij}(r_{ij})\frac{\partial r_{ij}}{\partial \vec{r}_n}, \tag{5.122}$$

in which the dash $(')$ against a function indicates a derivative with respect to the argument. Equation (5.122) can be written as

$$\vec{f}_n = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j\neq i}^{N} \phi'^{metal}_{ij}(r_{ij})\frac{\vec{r}_{ij}}{r_{ij}}(\delta_{jn}-\delta_{in}) - \sum_{i=1}^{N} F'^{metal}_{i}(\rho_i)\sum_{j\neq i}^{N} \rho'_{ij}(r_{ij})\frac{\vec{r}_{ij}}{r_{ij}}(\delta_{jn}-\delta_{in}). \tag{5.123}$$

The fact that n can occur in both sums over i and j allows us to write

$$\begin{aligned} \vec{f}_n = \frac{1}{2}\sum_{j\neq n}^{N} \phi'^{metal}_{nj}(r_{nj})\frac{\vec{r}_{nj}}{r_{nj}} - \frac{1}{2}\sum_{i\neq n}^{N} \phi'^{metal}_{in}(r_{in})\frac{\vec{r}_{in}}{r_{in}} + \\ F'^{metal}_{n}(\rho_n)\sum_{j\neq n}^{N} \rho'_{nj}(r_{nj})\frac{\vec{r}_{nj}}{r_{nj}} - \sum_{i\neq n}^{N} F'^{metal}_{i}(\rho_i)\rho'_{in}(r_{in})\frac{\vec{r}_{in}}{r_{in}}. \end{aligned} \tag{5.124}$$

By re-defining the indices in the sums and noting that $\vec{r}_{ij} = -\vec{r}_{ji}$, this becomes

$$\vec{f}_n = \sum_{j\neq n}^{N} \left(\phi'^{metal}_{nj}(r_{nj}) + F'^{metal}_{n}(\rho_n)\rho'_{nj}(r_{nj}) + F'^{metal}_{j}(\rho_j)\rho'_{jn}(r_{nj})\right)\frac{\vec{r}_{nj}}{r_{nj}}, \tag{5.125}$$

which, in the case of the FSM *only*, simplifies further to

$$\vec{f}_n = \sum_{j \neq n}^{N} \left(\phi'^{metal}_{nj}(r_{nj}) + [F'^{metal}_{n}(\rho_n) + F'^{metal}_{j}(\rho_j)]\rho'_{nj}(r_{nj}) \right) \frac{\vec{r}_{nj}}{r_{nj}}. \tag{5.126}$$

The distinction between (5.125) and (5.126) only matters when dealing with alloys.

Examples of FSM potentials include the following.

i. The Finnis-Sinclair potential [45] is defined by the formulae

$$\begin{aligned}
&\phi^{metal}_{ij}(r_{ij}) = (r_{ij} - c)^2 (c_0 + c_1 r_{ij} + c_2 r_{ij}^2), \\
&\phi'^{metal}_{ij}(r_{ij}) = 2(r_{ij} - c)(c_0 + c_1 r_{ij} + c_2 r_{ij}^2) + (r_{ij} - c)^2 (c_1 + 2c_2 r_{ij}), \\
&\rho_{ij}(r_{ij}) = (r_{ij} - d)^2 + \beta (r_{ij} - d)^3 / d, \\
&\rho'_{ij}(r_{ij}) = 2(r_{ij} - d) + 3\beta (r_{ij} - d)^2 / d, \\
&F^{metal}_{i}(\rho_i) = -A\sqrt{\rho_i}, \qquad F'^{metal}_{i}(\rho_i) = -\frac{A}{2\sqrt{\rho_i}}.
\end{aligned} \tag{5.127}$$

where c_0, c_1, c_2, β and A are parameters and c, d are cut-offs. Notice the square root dependence of F^{metal}_i on ρ_i, which is common in FSM potentials.

ii. The Sutton-Chen potential [48] is defined by

$$\begin{aligned}
&\phi^{metal}_{ij}(r_{ij}) = \varepsilon \left(\frac{a}{r_{ij}} \right)^n, \qquad \phi'^{metal}_{ij}(r_{ij}) = -\frac{n\varepsilon}{r_{ij}} \left(\frac{a}{r_{ij}} \right)^n, \\
&\rho_{ij}(r_{ij}) = \left(\frac{a}{r_{ij}} \right)^m, \qquad \rho'_{ij}(r_{ij}) = -\frac{m}{r_{ij}} \left(\frac{a}{r_{ij}} \right)^m, \\
&F^{metal}_{i}(\rho_i) = -c\varepsilon\sqrt{\rho_i}, \qquad F'^{metal}_{i}(\rho_i) = -\frac{c\varepsilon}{2\sqrt{\rho_i}}.
\end{aligned} \tag{5.128}$$

where a, ε, c, n, m are parameters.

iii. The Gupta potential has the following definitions

$$\begin{aligned}
&\phi^{metal}_{ij}(r_{ij}) = A \exp\left(-\frac{p}{r_0}(r_{ij} - r_0) \right), \qquad \phi'^{metal}_{ij}(r_{ij}) = -\frac{pA}{r_0} \exp\left(-\frac{p}{r_0}(r_{ij} - r_0) \right), \\
&\rho_{ij}(r_{ij}) = \exp\left(-\frac{2q_{ij}}{r_0}(r_{ij} - r_0) \right), \qquad \rho'_{ij}(r_{ij}) = -\frac{2q_{ij}}{r_0} \exp\left(-\frac{2q_{ij}}{r_0}(r_{ij} - r_0) \right), \\
&F^{metal}_{i}(\rho_i) = -B\sqrt{\rho_i}, \qquad F'^{metal}_{i}(\rho_i) = -\frac{B}{2\sqrt{\rho_i}}.
\end{aligned} \tag{5.129}$$

in which A, B, p, r_0, q_{ij} are parameters.

As was mentioned previously, the EAM potentials are not generally described in analytical form. In the literature they appear in tabulated form. Three tables are usually supplied for each potential: $\phi^{metal}_{ij}(r_{ij})$ and $\rho_{ij}(r_{ij})$, both tabulated *versus*

r_{ij}, and $F^{metal}(\rho_i)$ tabulated against ρ_i.

A word should be said about long range corrections for the metal potentials. Such corrections are not applied in EAM potential, though perhaps they should be. However, the lack of an analytical form makes extrapolation to infinity impossible. With FSM potentials, long range corrections can be applied. It is arguable that there is no need to apply a correction to the $\phi_{ij}^{metal}(r_{ij})$ contributions. These generally represent the the short range repulsion between atoms and should decay rapidly to a negligible magnitude well before any practical cut-off is applied. However the functions $\rho_{ij}(r_{ij})$ do not decay so quickly. An analytical long range correction can be obtained as follows.

Generalising (5.119) to include contributions from infinity gives

$$\begin{aligned}\rho_i &= \sum_{j\neq i}^{\infty} \rho_{ij}(r_{ij}) \\ &= \sum_{j\neq i}^{r_{ij}\leq r_{cut}} \rho_{ij}(r_{ij}) + \sum_{j}^{r_{ij}>r_{cut}} \rho_{ij}(r_{ij}) \\ &= \rho_i^0 + \delta\rho_i\end{aligned} \tag{5.130}$$

where

$$\rho_i^0 = \sum_{j\neq i}^{r_{ij}\leq r_{cut}} \rho_{ij}(r_{ij}), \tag{5.131}$$

is the density calculated including all the $\rho_{ij}(r_{ij})$ contributions within the cut-off range r_{cut} and

$$\delta\rho_i = \sum_{j}^{r_{ij}>r_{cut}} \rho_{ij}(r_{ij}) \equiv 4\pi\bar{\rho}\int_{r_{cut}}^{\infty} \rho_{ij}(r) r^2 dr \tag{5.132}$$

is the long range correction, in which $\bar{\rho}$ is the mean atomic density. If this correction is applied to the density ρ_i appearing in the above formulae for the potential energy (5.118) and force (5.124), no further corrections are necessary.

5.4 Molecular Torques

To reprise what we saw in section 2.6 of chapter 2, a rigid molecule undergoes two kinds of motion: a translation of the molecule's centre of mass in response to the *molecular force* acting on the molecule and a rotation about its centre of mass in response to the *molecular torque*. The molecular force, $\vec{F}$, is simply the vector sum of the atomic forces, $\vec{f}_i$, acting on all its constituent atoms:

$$\vec{F} = \sum_{i=1}^{N_a} \vec{f}_i, \tag{5.133}$$

where N_a is the number of atoms in the rigid molecule. The molecular torque, $\vec{\tau}$, is also constructed from the atomic forces, but combined with the displacement $\vec{d}_i$, of the atoms from the molecular centre of mass:

$$\vec{\tau} = \sum_{i=1}^{N_a} \vec{d}_i \times \vec{f}_i, \tag{5.134}$$

in which

$$\vec{d}_i = \vec{r}_i - \vec{R}, \tag{5.135}$$

where $\vec{r}_i$ is the position of the $i'th$ atom in space and $\vec{R}$ is the molecular centre of mass given by

$$\vec{R} = \frac{1}{M} \sum_{i=1}^{N_a} m_i \vec{r}_i, \tag{5.136}$$

and M is the *molecular mass* equivalent to the sum of the atomic masses m_i, as in

$$M = \sum_{i=1}^{N_a} m_i. \tag{5.137}$$

Equations (5.134) to (5.137) provide a complete specification of the molecular torque when all the forces involved are of the simple atomic kind. However, not all molecular torques arise in this way and there are two notable exceptions. The first is where the simple atoms are replaced by electrostatic *point multipoles,* such as dipoles and quadrupoles. The second is where the rigid molecule is not represented as a framework of point atoms, but as a single non-spherical entity, examples of which include the Gay-Berne model [26] and the Gaussian density model [27] in which a single non-spherical site represents a linear or planar molecule. These two exceptions require further consideration.

5.4.1 Torques in Multipole Systems

When multipoles are present at various sites in a rigid molecule, then in an electric field (not necessarily uniform) a molecular torque arises from both atomic forces and what might be called *atomic torques*. For example an electric point dipole $\vec{\mu}$ in an electric field $\vec{E}$ experiences a torque given as

$$\vec{\tau} = \vec{\mu} \times \vec{E}. \tag{5.138}$$

In addition, the atom (or site) the dipole is associated with may also experience van der Waals or other forces, which will generate a contribution to the molecular torque in the rigid molecule of the kind shown in (5.134). The question is how to combine the effects of both these torques in the rotational dynamics. We shall demonstrate this by considering a simple model.

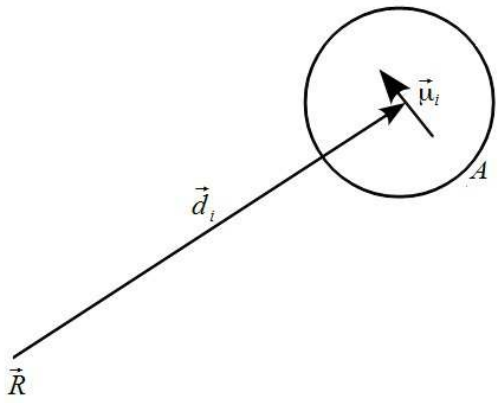

Figure 5.7: A Point Dipole within a van der Waals Site

We define a rigid molecule composed of N_a atoms, each of which functions as a van der Waals site and is located at a position $\vec{d}_i$ defined with respect to the molecular centre of mass at $\vec{R}$ (see figure 5.7). At each site a finite dipole $\vec{\mu}_i$ is also located, which is defined as

$$\vec{\mu}_i = q_i \vec{u}_i, \tag{5.139}$$

where $\vec{u}_i$ is a (small) displacement vector between two charges q_i^+ and q_i^- with $q_i = q_i^+ = -q_i^-$. It should be understood that the direction of vector $\vec{u}_i$ is fixed relative to the rigid molecule and rotates with it. It is in no sense "free" to rotate on its own.

The centre of the dipole $\vec{\mu}_i$ is at $\vec{d}_i$, so the positions of charges q_i^+ and q_i^- are given by

$$\begin{aligned} \vec{d}_i^{\,+} &= \vec{d}_i + \frac{1}{2}\vec{u}_i, \\ \vec{d}_i^{\,-} &= \vec{d}_i - \frac{1}{2}\vec{u}_i. \end{aligned} \tag{5.140}$$

In an electric field $\vec{E}$, the force on each charge is

$$\begin{aligned} \vec{f}_i^{\,+} &= q_i^+ \vec{E}, \\ \vec{f}_i^{\,-} &= q_i^- \vec{E}, \end{aligned} \tag{5.141}$$

and, we assume, there is also the van der Waals force $\vec{f}_i^{\,vdw}$ that acts on the site centre. From the positions of the charges and the site centres and the forces acting, we can write the molecular torque as

$$\begin{aligned}\vec{\tau} &= \sum_{i=1}^{N_a}\left[\vec{d}_i\times\vec{f}_i^{vdw}+(\vec{d}_i+\frac{1}{2}\vec{u}_i)\times\vec{f}_i^{+}+(\vec{d}_i-\frac{1}{2}\vec{u}_i)\times\vec{f}_i^{-}\right],\\ &= \sum_{i=1}^{N_a}\left[\vec{d}_i\times\vec{f}_i^{vdw}+q_i(\vec{d}_i+\frac{1}{2}\vec{u}_i)\times\vec{E}-q_i(\vec{d}_i-\frac{1}{2}\vec{u}_i)\times\vec{E}\right],\\ &= \sum_{i=1}^{N_a}\left[\vec{d}_i\times\vec{f}_i^{vdw}+q_i\vec{u}_i\times\vec{E}\right]=\sum_{i=1}^{N_a}\left[\vec{d}_i\times\vec{f}_i^{vdw}+\vec{\mu}_i\times\vec{E}\right]\end{aligned} \tag{5.142}$$

From this we see that the molecular torque is a simple vector sum:

$$\vec{\tau}=\vec{\tau}^{vdw}+\vec{\tau}^{\mu}, \tag{5.143}$$

where

$$\vec{\tau}^{vdw}=\sum_{i=1}^{N_a}\vec{d}_i\times\vec{f}^{vdw}, \tag{5.144}$$

is the molecular torque contribution arising from the van der Waals forces and

$$\vec{\tau}^{\mu}=\sum_{i=1}^{N_a}\vec{\mu}_i\times\vec{E}, \tag{5.145}$$

is the molecular torque contribution arising from the interaction of the dipoles with the electric field. From this we draw the conclusion that torques arising from *any* multipolar interactions can simply be added vectorially to the overall molecular torque. Clearly the same result would apply if they were point, rather than finite, multipoles.

5.4.2 Torques from Non-Spherical Interactions

In this section we describe how to calculate the torque in cases where the intermolecular interaction is described by non-spherical interactions of the kind typified by the Gay-Berne [26] and Gaussian density [27] models. This approach, due to Allen and Germano [49], can also be used to obtain torque expressions for electrostatic multipoles, since it only requires knowledge of the analytic expression for the system configuration energy, but we shall not consider this here.

In section 2.6 of chapter 2, we mentioned that the rotation matrix, $\boldsymbol{R}$, can be used to define the *orientation* of a molecule. We make that choice here and assume that the interaction potential energy between two non-spherical molecules labelled i and j can be written as $\phi(\vec{r}_{ij},\boldsymbol{R}_i,\boldsymbol{R}_j)$, signifying that it depends only on the centre-to-centre displacement between the molecules i and j, and the orientations of the two molecules, represented by their respective rotation matrices $\boldsymbol{R}_i$ and $\boldsymbol{R}_j$. (We will not need to consider an explicit expression for $\phi(\vec{r}_{ij},\boldsymbol{R}_i,\boldsymbol{R}_j)$ in this section.)

We begin by considering a pair of molecules in some arbitrary, fixed configuration (see figure 5.8) and create an arbitrary *unit vector* $\vec{n}$ associated with molecule j. Next we write a plausible expression for the torque experienced by molecule j that has

the direction $\vec{n}$ and arises from the action of molecule i on j. This we put in the form:

$$\vec{\tau}_j^n = -\vec{n}\frac{\partial\phi(\vec{r}_{ij}, \boldsymbol{R}_i, \boldsymbol{R}_j)}{\partial\theta_j^n} \tag{5.146}$$

According to this expression the torque vector direction is along the unit vector $\vec{n}$ (by construction) and its magnitude is equal to the (negative) rate of change of the interaction energy with the rotation angle θ_j^n around $\vec{n}$. It therefore resembles force, which is the (negative) rate of change of interaction energy with respect to a displacement. So, intuitively this definition of the torque is sensible.

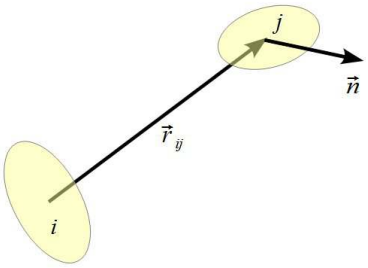

Figure 5.8: Non-Spherical Molecular Interactions

To proceed further we re-define the rotation matrix as follows

$$\boldsymbol{R}_j = \left[\vec{a}_j^1, \vec{a}_j^2, \vec{a}_j^3\right] \tag{5.147}$$

In which the columns of the matrix are represented by *column vectors* $\vec{a}_j^k$, with $(k=1,\ldots,3)$. It should be noted that, due to the property of all rotation matrices that $\boldsymbol{R}\tilde{\boldsymbol{R}}=\boldsymbol{1}$, it follows that all the vectors $\vec{a}_j^1, \vec{a}_j^2, \vec{a}_j^3$, are *unit* vectors[19]. We can now write

$$\frac{\partial\phi(\vec{r}_{ij}, \boldsymbol{R}_i, \boldsymbol{R}_j)}{\partial\theta_j^n} = \sum_{\nu=1}^{3}\frac{\partial\phi(\vec{r}_{ij}, \boldsymbol{R}_i, \boldsymbol{R}_j)}{\partial\vec{a}_j^\nu}\cdot\frac{\partial\vec{a}_j^\nu}{\partial\theta_j^n}, \tag{5.148}$$

which follows from the chain rule (note the scalar product on the right). It can also be shown that

$$\frac{\partial\vec{a}_j^\nu}{\partial\theta_j^n} = \vec{n}\times\vec{a}_j^\nu \tag{5.149}$$

which may be understood from the observation that rotating $\vec{a}_j^k$ through a small angle $\delta\theta^n$ about $\vec{n}$, requires that the change in $\vec{a}_j^\nu$ be perpendicular to both $\vec{n}$ and $\vec{a}_j^\nu$ (see figure 5.9).

19 Recall it was shown in section 4.6.8 that these vectors are the unit vectors defining the principal frame of reference, expressed in the laboratory frame.

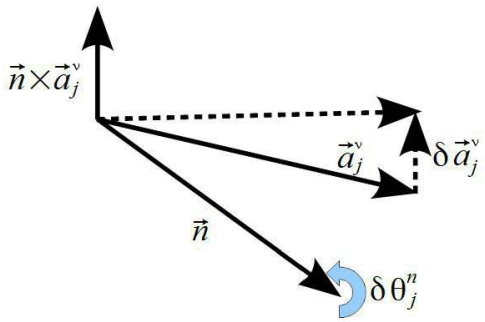

Figure 5.9: Rotating vector $\vec{a}_j^{\nu}$ about vector $\vec{n}$.

Hence (5.148) becomes

$$\frac{\partial\phi(\vec{r}_{ij},\boldsymbol{R}_i,\boldsymbol{R}_j)}{\partial\theta_j^n}=\sum_{\nu=1}^{3}\frac{\partial\phi(\vec{r}_{ij},\boldsymbol{R}_i,\boldsymbol{R}_j)}{\partial\vec{a}_j^{\nu}}\cdot\vec{n}\times\vec{a}_j^{\nu}=\vec{n}\cdot\sum_{\nu=1}^{3}\vec{a}_j^{\nu}\times\frac{\partial\phi(\vec{r}_{ij},\boldsymbol{R}_i,\boldsymbol{R}_j)}{\partial\vec{a}_j^{\nu}}, \tag{5.150}$$

and so (5.146) can be written as

$$\vec{\tau}_j^n=-\vec{n}\left(\vec{n}\cdot\sum_{\nu=1}^{3}\vec{a}_j^{\nu}\times\frac{\partial\phi(\vec{r}_{ij},\boldsymbol{R}_i,\boldsymbol{R}_j)}{\partial\vec{a}_j^{\nu}}\right). \tag{5.151}$$

Equation (5.151) has the form $\vec{\tau}_j^n=-\vec{n}(\vec{n}\cdot\vec{\tau}_j)$, which is a common vector form describing the component, or *projection*, of a vector, $\vec{\tau}_j$, in the direction of a vector $\vec{n}$. So

$$\vec{\tau}_j=-\sum_{\nu=1}^{3}\vec{a}_j^{\nu}\times\frac{\partial\phi(\vec{r}_{ij},\boldsymbol{R}_i,\boldsymbol{R}_j)}{\partial\vec{a}_j^{\nu}}. \tag{5.152}$$

This must be the molecular torque vector we seek.

As an example application we consider a simple case: an electric dipole $\vec{\mu}$ in a uniform electric field $\vec{E}$. The potential energy, $\phi(\vec{\mu},\vec{E})$, is given by

$$\phi(\vec{\mu},\vec{E})=-\vec{\mu}\cdot\vec{E}=-(\mu_x E_x+\mu_y E_y+\mu_z E_z). \tag{5.153}$$

To introduce the rotation matrix we write $\vec{\mu}$ as

$$\vec{\mu}=\boldsymbol{R}\vec{\mu}_0, \tag{5.154}$$

with

$$\vec{\mu}_0=\begin{bmatrix}0\\0\\\mu\end{bmatrix},\qquad \boldsymbol{R}=\left[\vec{a}^1,\vec{a}^2,\vec{a}^3\right], \tag{5.155}$$

where μ is the *dipole magnitude* and the form of $\boldsymbol{R}$ comes from (5.147). It follows that

$$\mu_x=\mu(a^3)_x, \qquad \mu_y=\mu(a^3)_y, \qquad \mu_z=\mu(a^3)_z, \tag{5.156}$$

Where $(a^3)_x$, $(a^3)_y$ and $(a^3)_z$ are the components of $\vec{a}^3$ *vis.*

$$\vec{a}^3=\begin{bmatrix}(a^3)_x\\(a^3)_y\\(a^3)_z\end{bmatrix} \tag{5.157}$$

We therefore calculate the required derivatives as

$$\frac{\partial\phi(\vec{\mu},\vec{E})}{\partial\vec{a}^1}=\frac{\partial\phi(\vec{\mu},\vec{E})}{\partial\vec{a}^2}=\vec{0} \quad \text{and} \quad \frac{\partial\phi(\vec{\mu},\vec{E})}{\partial\vec{a}^3}=-\mu\begin{bmatrix}E_x\\E_y\\E_x\end{bmatrix}. \tag{5.158}$$

Inserting (5.157) and (5.158) into (5.152) gives

$$\vec{\tau}=\mu\begin{bmatrix}(a^3)_x\\(a^3)_y\\(a^3)_x\end{bmatrix}\times\begin{bmatrix}E_x\\E_y\\E_x\end{bmatrix}, \quad \text{or} \quad \vec{\tau}=\vec{\mu}\times\vec{E}. \tag{5.159}$$

The result is correct if somewhat trivial. However we can now tackle more difficult cases.

5.4.3 Forces and Torques for the Gay-Berne Potential

The Gay-Berne potential [26] is an adaptation of the Lennard-Jones pair potential to an ellipsoidal form. In a system of uniform ellipsoids, it is written as

$$\phi_{ij}^{GB}(\vec{r}_{ij},\vec{e}_i,\vec{e}_j)=4\varepsilon(\vec{s}_{ij},\vec{e}_i,\vec{e}_j)\left(\left\{\frac{\sigma_0}{r_{ij}-\sigma(\vec{s}_{ij},\vec{e}_i,\vec{e}_j)+\sigma_0}\right\}^{12}-\left\{\frac{\sigma_0}{r_{ij}-\sigma(\vec{s}_{ij},\vec{e}_i,\vec{e}_j)+\sigma_0}\right\}^{6}\right), \tag{5.160}$$

in which $\vec{r}_{ij}$ is the centre-to-centre vector between ellipsoids labelled i and j and $\vec{s}_{ij}=\vec{r}_{ij}/|\vec{r}_{ij}|$ is a unit vector directed along $\vec{r}_{ij}$. The vectors $\vec{e}_i$ and $\vec{e}_j$ are unit vectors aligned along the longest axis of each ellipsoid and which define their orientations. The parameter σ_0 represents the the *width* of the ellipsoid taken at its narrowest cross section. It is also the the point of zero potential when two ellipsoids approach side-on, i.e. at $r_{ij}=\sigma_0$, analogous to the approach of two Lennard-Jones spheres at $r_{ij}=\sigma$. We may also define a parameter σ_1, corresponding to the *length* of the ellipsoids, at which zero potential occurs for an end-to-end approach for $r_{ij}=\sigma_1$, (see figure 5.10.) It is always assumed that $\sigma_1>\sigma_0$.

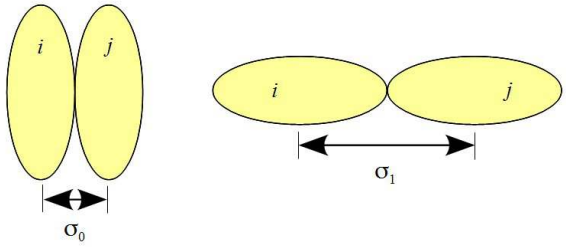

Figure 5.10: Side-on and End-to-End Gay-Berne Ellipsoids

As might be anticipated from the form (5.160) the function $\sigma(\vec{s}_{ij},\vec{e}_i,\vec{e}_j)$ defines the s*hape* of the ellipsoid, which is expressed as

$$\sigma(\vec{s}_{ij},\vec{e}_i,\vec{e}_j)=\sigma_0\left[1-\frac{\chi}{2}\left\{\frac{(\vec{s}_{ij}\cdot(\vec{e}_j+\vec{e}_i))^2}{1+\chi\vec{e}_i\cdot\vec{e}_j}+\frac{(\vec{s}_{ij}\cdot(\vec{e}_j-\vec{e}_i))^2}{1-\chi\vec{e}_i\cdot\vec{e}_j}\right\}\right]^{-1/2}, \tag{5.161}$$

where parameter χ is given as

$$\chi=\frac{\sigma_1^2-\sigma_0^2}{\sigma_1^2+\sigma_0^2}. \tag{5.162}$$

On the other hand, the function $\varepsilon(\vec{s}_{ij},\vec{e}_i,\vec{e}_j)$ defines the energy of interaction between the ellipsoids. It has the form

$$\varepsilon(\vec{s}_{ij},\vec{e}_i,\vec{e}_j)=\varepsilon_0\varepsilon_1^{\mu}(\vec{e}_i,\vec{e}_j)\varepsilon_2^{\nu}(\vec{s}_{ij},\vec{e}_i,\vec{e}_j) \tag{5.163}$$

where μ and ν are exponents and typically $\mu=1$ and $\nu=2.$ ε_0 is an energy parameter and

$$\varepsilon_1(\vec{e}_i,\vec{e}_j)=\left[1-\chi^2(\vec{e}_i\cdot\vec{e}_j)^2\right]^{-1/2} \tag{5.164}$$

and

$$\varepsilon_2(\vec{s}_{ij},\vec{e}_i,\vec{e}_j)=1-\frac{\chi'}{2}\left[\frac{(\vec{s}_{ij}\cdot(\vec{e}_j+\vec{e}_i))^2}{1+\chi'\vec{e}_i\cdot\vec{e}_j}+\frac{(\vec{s}_{ij}\cdot(\vec{e}_j-\vec{e}_i))^2}{1-\chi'\vec{e}_i\cdot\vec{e}_j}\right] \tag{5.165}$$

with

$$\chi'=\frac{1-(\varepsilon_2/\varepsilon_1)^{1/\nu}}{1+(\varepsilon_2/\varepsilon_1)^{1/\nu}}. \tag{5.166}$$

The parameters ε_1 and ε_2 in (5.166) allow some "tuning" in the preferred geometry of association between ellipsoid pairs. The origin of the form of this potential is given in [26], our purpose here is to show how forces and torques may be derived from it.

The configuration energy for a system of uniform ellipsoids is

$$\phi(\vec{r}^{\,N},\vec{e}^{\,N})=\sum_{i=1}^{N-1}\sum_{j>i}^{N}\phi_{ij}^{GB}(\vec{r}_{ij},\vec{e}_i,\vec{e}_j), \tag{5.167}$$

which is consistent with the pair interaction form. The force on arbitrary ellipsoid n is therefore

$$\vec{f}_n^{\,GB}=-\sum_{i=1}^{N-1}\sum_{j>i}^{N}\frac{\partial\phi_{ij}^{GB}(\vec{r}_{ij},\vec{e}_i,\vec{e}_j)}{\partial\vec{r}_n}. \tag{5.168}$$

From (5.160) we obtain

$$\begin{aligned}\frac{\partial\phi_{ij}^{GB}(\vec{r}_{ij},\vec{e}_i,\vec{e}_j)}{\partial\vec{r}_n}=&\frac{\phi_{ij}^{GB}(\vec{r}_{ij},\vec{e}_i,\vec{e}_j)}{\varepsilon(\vec{s}_{ij},\vec{e}_i,\vec{e}_j)}\frac{\partial\varepsilon(\vec{s}_{ij},\vec{e}_i,\vec{e}_j)}{\partial\vec{r}_n}-\frac{24\,\varepsilon(\vec{s}_{ij},\vec{e}_i,\vec{e}_j)}{(r_{ij}-\sigma(\vec{s}_{ij},\vec{e}_i,\vec{e}_j)+\sigma_0)}\\&\left(2W_{ij}^{12}-W_{ij}^{6}\right)\left(\frac{\partial r_{ij}}{\partial\vec{r}_n}-\frac{\partial\sigma(\vec{s}_{ij},\vec{e}_i,\vec{e}_j)}{\partial\vec{r}_n}\right),\end{aligned} \tag{5.169}$$

where, for brevity, we have defined W_{ij} as

$$W_{ij}=\frac{\sigma_0}{r_{ij}-\sigma(\vec{s}_{ij},\vec{e}_i,\vec{e}_j)+\sigma_0}. \tag{5.170}$$

We now define a vector function

$$\vec{G}_{ij}(\chi)=\chi\left\{\frac{\vec{s}_{ij}\cdot\vec{e}_{ij}^{\,+}(\vec{e}_{ij}^{\,+}-(\vec{s}_{ij}\cdot\vec{e}_{ij}^{\,+})\vec{s}_{ij})}{1+\chi\vec{e}_i\cdot\vec{e}_j}+\frac{\vec{s}_{ij}\cdot\vec{e}_{ij}^{\,-}(\vec{e}_{ij}^{\,-}-(\vec{s}_{ij}\cdot\vec{e}_{ij}^{\,-})\vec{s}_{ij})}{1-\chi\vec{e}_i\cdot\vec{e}_j}\right\}, \tag{5.171}$$

where vectors $\vec{e}_{ij}^{\,+}$ and $\vec{e}_{ij}^{\,-}$ are

$$\vec{e}_{ij}^{\,+}=\vec{e}_j+\vec{e}_i \qquad \text{and} \qquad \vec{e}_{ij}^{\,-}=\vec{e}_j-\vec{e}_i, \tag{5.172}$$

so that the derivatives in (5.169) can be written as follows

$$\frac{\partial r_{ij}}{\partial\vec{r}_n}=\vec{s}_{ij}(\delta_{jn}-\delta_{in}), \tag{5.173}$$

$$\frac{\partial\varepsilon(\vec{s}_{ij},\vec{e}_i,\vec{e}_j)}{\partial\vec{r}_n}=-\frac{\nu\varepsilon(\vec{s}_{ij},\vec{e}_i,\vec{e}_j)}{\varepsilon_2(\vec{s}_{ij},\vec{e}_i,\vec{e}_j)r_{ij}}\vec{G}_{ij}(\chi')(\delta_{jn}-\delta_{in}), \tag{5.174}$$

$$\frac{\partial\sigma(\vec{s}_{ij},\vec{e}_i,\vec{e}_j)}{\partial\vec{r}_n}=\frac{\sigma^3(\vec{s}_{ij},\vec{e}_i,\vec{e}_j)}{2\sigma_0^2 r_{ij}}\vec{G}_{ij}(\chi)(\delta_{jn}-\delta_{in}). \tag{5.175}$$

With these we can now write the contribution to the forces on ellipsoids i and j arising from a given pair potential, $\phi_{ij}^{GB}(\vec{r}_{ij},\vec{e}_i,\vec{e}_j)$, as

$$\vec{f}_i^{GB} \leftarrow -\left(\frac{\nu \phi_{ij}^{GB}(\vec{r}_{ij}, \vec{e}_i, \vec{e}_j)}{\varepsilon_2(\vec{s}_{ij}, \vec{e}_i, \vec{e}_j) r_{ij}} \vec{G}_{ij}(\chi') + \frac{24\,\varepsilon(\vec{s}_{ij}, \vec{e}_i, \vec{e}_j)}{(r_{ij} - \sigma(\vec{s}_{ij}, \vec{e}_i, \vec{e}_j) + \sigma_0)} \left(2W_{ij}^{12} - W_{ij}^{6}\right) \left[\vec{s}_{ij} - \frac{\sigma^3(\vec{s}_{ij}, \vec{e}_i, \vec{e}_j)}{2\sigma_0^2 r_{ij}} \vec{G}_{ij}(\chi) \right] \right),$$

$$\vec{f}_j^{GB} \leftarrow \left(\frac{\nu \phi_{ij}^{GB}(\vec{r}_{ij}, \vec{e}_i, \vec{e}_j)}{\varepsilon_2(\vec{s}_{ij}, \vec{e}_i, \vec{e}_j) r_{ij}} \vec{G}_{ij}(\chi') + \frac{24\,\varepsilon(\vec{s}_{ij}, \vec{e}_i, \vec{e}_j)}{(r_{ij} - \sigma(\vec{s}_{ij}, \vec{e}_i, \vec{e}_j) + \sigma_0)} \left(2W_{ij}^{12} - W_{ij}^{6}\right) \left[\vec{s}_{ij} - \frac{\sigma^3(\vec{s}_{ij}, \vec{e}_i, \vec{e}_j)}{2\sigma_0^2 r_{ij}} \vec{G}_{ij}(\chi) \right] \right). \tag{5.176}$$

It is apparent that these contributions are equal in magnitude and opposite in sign, as would be the case for simple Lennard-Jones sites. The *total* force acting on an ellipsoid n can be written as

$$\vec{f}_n^{GB} = -\sum_{j \neq n}^{N} \left(\frac{\nu \phi_{nj}^{GB}(\vec{r}_{nj}, \vec{e}_n, \vec{e}_j)}{\varepsilon_2(\vec{s}_{nj}, \vec{e}_n, \vec{e}_j) r_{nj}} \vec{G}_{nj}(\chi') + \frac{24\,\varepsilon(\vec{s}_{nj}, \vec{e}_n, \vec{e}_j)}{(r_{nj} - \sigma(\vec{s}_{nj}, \vec{e}_n, \vec{e}_j) + \sigma_0)} \left(2W_{nj}^{12} - W_{nj}^{6}\right) \left[\vec{s}_{nj} - \frac{\sigma^3(\vec{s}_{nj}, \vec{e}_n, \vec{e}_j)}{2\sigma_0^2 r_{nj}} \vec{G}_{nj}(\chi) \right] \right). \tag{5.177}$$

To calculate the torque we apply the procedure described in section 5.4.2 . Staring with the torque on ellipsoid j, we first define the vector $\vec{e}_j$ using the rotation matrix $\boldsymbol{R}_j$

$$\vec{e}_j = \boldsymbol{R}_j \vec{e}_j^{\,0}, \tag{5.178}$$

in which

$$\boldsymbol{R}_j = \left[\vec{a}_j^{\,1}, \vec{a}_j^{\,2}, \vec{a}_j^{\,3} \right] \quad \text{and} \quad \vec{e}_j^{\,0} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}. \tag{5.179}$$

Vector $\vec{e}_j^{\,0}$ is a unit vector directed along the longer axis of the ellipsoid in its local frame. It can be seen immediately from (5.178) and (5.179) that

$$\vec{e}_j = \vec{a}_j^{\,3}. \tag{5.180}$$

The torque contribution to j according to equation (5.152) is therefore

$$\vec{\tau}_j \leftarrow -\vec{e}_j \times \frac{\partial \phi_{ij}^{GB}(\vec{r}_{ij}, \boldsymbol{R}_i, \boldsymbol{R}_j)}{\partial \vec{e}_j}, \tag{5.181}$$

and the partial derivative appearing here is

$$\frac{\partial \phi_{ij}^{GB}(\vec{r}_{ij},\vec{e}_i,\vec{e}_j)}{\partial \vec{e}_j}=\frac{\phi_{ij}^{GB}(\vec{r}_{ij},\vec{e}_i,\vec{e}_j)}{\varepsilon(\vec{s}_{ij},\vec{e}_i,\vec{e}_j)}\frac{\partial \varepsilon(\vec{s}_{ij},\vec{e}_i,\vec{e}_j)}{\partial \vec{e}_j}+\frac{24\,\varepsilon(\vec{s}_{ij},\vec{e}_i,\vec{e}_j)}{(r_{ij}-\sigma(\vec{s}_{ij},\vec{e}_i,\vec{e}_j)+\sigma_0)}\left(2W_{ij}^{12}-W_{ij}^{6}\right)\left(\frac{\partial \sigma(\vec{s}_{ij},\vec{e}_i,\vec{e}_j)}{\partial \vec{e}_j}\right), \tag{5.182}$$

To proceed further we define the vector function

$$\vec{J}_{ij}(\chi)=\left(\chi\left[\frac{\vec{s}_{ij}\cdot\vec{e}_{ij}^{+}}{1+\chi\vec{e}_i\cdot\vec{e}_j}+\frac{\vec{s}_{ij}\cdot\vec{e}_{ij}^{-}}{1-\chi\vec{e}_i\cdot\vec{e}_j}\right]\vec{s}_{ij}-\frac{\chi^2}{2}\left[\left\{\frac{\vec{s}_{ij}\cdot\vec{e}_{ij}^{+}}{1+\chi\vec{e}_i\cdot\vec{e}_j}\right\}^2-\left\{\frac{\vec{s}_{ij}\cdot\vec{e}_{ij}^{-}}{1-\chi\vec{e}_i\cdot\vec{e}_j}\right\}^2\right]\vec{e}_i\right), \tag{5.183}$$

and write

$$\frac{\partial \varepsilon(\vec{s}_{ij},\vec{e}_i,\vec{e}_j)}{\partial \vec{e}_j}=\frac{\mu\,\varepsilon(\vec{s}_{ij},\vec{e}_i,\vec{e}_j)}{\varepsilon_1(\vec{e}_i,\vec{e}_j)}\frac{\partial \varepsilon_1(\vec{e}_i,\vec{e}_j)}{\partial \vec{e}_j}+\frac{\nu\,\varepsilon(\vec{s}_{ij},\vec{e}_i,\vec{e}_j)}{\varepsilon_2(\vec{s}_{ij},\vec{e}_i,\vec{e}_j)}\frac{\partial \varepsilon_2(\vec{s}_{ij},\vec{e}_i,\vec{e}_j)}{\partial \vec{e}_j}, \tag{5.184}$$

with

$$\begin{aligned}\frac{\partial \varepsilon_1(\vec{e}_i,\vec{e}_j)}{\partial \vec{e}_j}&=\chi^2\varepsilon_1^3(\vec{e}_i,\vec{e}_j)(\vec{e}_i\cdot\vec{e}_j)\vec{e}_i,\\ \frac{\partial \varepsilon_2(\vec{s}_{ij},\vec{e}_i,\vec{e}_j)}{\partial \vec{e}_j}&=-\vec{J}_{ij}(\chi'),\end{aligned} \tag{5.185}$$

and

$$\frac{\partial \sigma(\vec{s}_{ij},\vec{e}_i,\vec{e}_j)}{\partial \vec{e}_j}=\frac{\sigma^3(\vec{s}_{ij},\vec{e}_i,\vec{e}_j)}{2\sigma_0^2}\vec{J}_{ij}(\chi). \tag{5.186}$$

Combining results (5.182) to (5.186) gives the final torque as

$$\begin{aligned}\vec{\tau}_j\leftarrow-\frac{\phi_{ij}^{GB}(\vec{r}_{ij},\vec{e}_i,\vec{e}_j)}{\varepsilon(\vec{s}_{ij},\vec{e}_i,\vec{e}_j)}\Big[&\mu\chi^2\varepsilon_0\varepsilon_1^{\mu+2}(\vec{e}_i,\vec{e}_j)\varepsilon_2^{\nu}(\vec{s}_{ij},\vec{e}_i,\vec{e}_j)(\vec{e}_i\cdot\vec{e}_j)[\vec{e}_j\times\vec{e}_i]-\\ &\nu\varepsilon_0\varepsilon_1^{\mu}(\vec{e}_i,\vec{e}_j)\varepsilon_2^{\nu-1}(\vec{s}_{ij},\vec{e}_i,\vec{e}_j)[\vec{e}_j\times\vec{J}_{ij}(\chi')]\Big]-\\ &\frac{12\,\varepsilon(\vec{s}_{ij},\vec{e}_i,\vec{e}_j)\sigma^3(\vec{s}_{ij},\vec{e}_i,\vec{e}_j)}{\sigma_0^2(r_{ij}-\sigma(\vec{s}_{ij},\vec{e}_i,\vec{e}_j)+\sigma_0)}\left(2W_{ij}^{12}-W_{ij}^{6}\right)[\vec{e}_j\times\vec{J}_{ij}(\chi)].\end{aligned} \tag{5.187}$$

The torque acting on ellipsoid i may be obtained in a similar manner. The result is

$$\begin{aligned}\vec{\tau}_i\leftarrow-\frac{\phi_{ij}^{GB}(\vec{r}_{ij},\vec{e}_i,\vec{e}_j)}{\varepsilon(\vec{s}_{ij},\vec{e}_i,\vec{e}_j)}\Big[&\mu\chi^2\varepsilon_0\varepsilon_1^{\mu+2}(\vec{e}_i,\vec{e}_j)\varepsilon_2^{\nu}(\vec{s}_{ij},\vec{e}_i,\vec{e}_j)(\vec{e}_i\cdot\vec{e}_j)[\vec{e}_i\times\vec{e}_j]-\\ &\nu\varepsilon_0\varepsilon_1^{\mu}(\vec{e}_i,\vec{e}_j)\varepsilon_2^{\nu-1}(\vec{s}_{ij},\vec{e}_i,\vec{e}_j)[\vec{e}_i\times\vec{I}_{ij}(\chi')]\Big]-\\ &\frac{12\,\varepsilon(\vec{s}_{ij},\vec{e}_i,\vec{e}_j)\sigma^3(\vec{s}_{ij},\vec{e}_i,\vec{e}_j)}{\sigma_0^2(r_{ij}-\sigma(\vec{s}_{ij},\vec{e}_i,\vec{e}_j)+\sigma_0)}\left(2W_{ij}^{12}-W_{ij}^{6}\right)[\vec{e}_i\times\vec{I}_{ij}(\chi)],\end{aligned} \tag{5.188}$$

where

$$\vec{I}_{ij}(\chi)=\left(\chi\left[\frac{\vec{s}_{ij}\cdot\vec{e}_{ij}^{+}}{1+\chi\vec{e}_i\cdot\vec{e}_j}-\frac{\vec{s}_{ij}\cdot\vec{e}_{ij}^{-}}{1-\chi\vec{e}_i\cdot\vec{e}_j}\right]\vec{s}_{ij}-\frac{\chi^2}{2}\left[\left\{\frac{\vec{s}_{ij}\cdot\vec{e}_{ij}^{+}}{1+\chi\vec{e}_i\cdot\vec{e}_j}\right\}^2-\left\{\frac{\vec{s}_{ij}\cdot\vec{e}_{ij}^{-}}{1-\chi\vec{e}_i\cdot\vec{e}_j}\right\}^2\right]\vec{e}_j\right). \tag{5.189}$$

Note that the torque $\vec{\tau}_i$ is generally not equal and opposite to the torque $\vec{\tau}_j$. Also it is most efficient to calculate $\vec{G}_{ij}(\chi),\ \vec{J}_{ij}(\chi)$ and $\vec{I}_{ij}(\chi)$ *together*, as they all use the same scalar products and share various ratios. To ensure these formulae are used correctly, remember that the vectors $\vec{s}_{ij}$ and $\vec{r}_{ij}$ have direction $i\to j$ i.e. $\vec{r}_{ij}=\vec{r}_j-\vec{r}_i$ *etc*.

5.4.4 Forces and Torques for the Gaussian Density Potential

The Gaussian density model [27] resembles the Gay-Berne model in that it permits representation of ellipsoidal molecules. However, we shall consider a slightly more general case in which the ellipsoids have an elliptical rather than circular cross section perpendicular to the longest axis. According to this model a molecule is visualised as a Gaussian distribution of the form

$$g_i(\vec{r}_i)\,dv_i=\sqrt{\frac{|\boldsymbol{A}_i|}{(2\pi)^3}}\exp\left[-\vec{\xi}\cdot\boldsymbol{A}_i\cdot\vec{\xi}\right]dv_i \tag{5.190}$$

where dv_i is a volume element of the Gaussian molecule indexed i and

$$\vec{\xi}=\vec{r}_i-\vec{R}_i, \tag{5.191}$$

where $\vec{R}_i$ is the location of the centre of the Gaussian in the laboratory frame. The pre-exponential term is the *normalisation constant* that ensures the Gaussian volume integral is unity i.e.

$$\int_{-\infty}^{\infty} g_i(\vec{r}_i)\,dv_i=1. \tag{5.192}$$

$\boldsymbol{A}_i$ is a real, symmetric, 3×3 matrix which defines the shape and orientation of the Gaussian. $|\boldsymbol{A}_i|$ is its determinant. The matrix $\boldsymbol{A}_i$ encapsulates the orientational dependence of the Gaussian in the following manner. We may write

$$\boldsymbol{A}_i=\boldsymbol{R}_i\boldsymbol{A}_i^0\tilde{\boldsymbol{R}}_i, \tag{5.193}$$

where $\boldsymbol{R}_i$ is the *rotation matrix* for Gaussian i and $\tilde{\boldsymbol{R}}_i\,(=\boldsymbol{R}_i^{-1})$ is its transpose. The matrix $\boldsymbol{A}_i^0$ is a unique, *diagonal* matrix, the diagonal elements of which define the mean-square widths of the Gaussian distribution in the local frame. The local frame vectors correspond with the *principal axes* of the Gaussian ellipsoid and also the symmetry axes. So the orientation of the Gaussian is defined by the frame in which the matrix $\boldsymbol{A}_i$ is diagonal, which is also the frame in which the moment of inertia

tensor is diagonal. The diagonal elements of $\boldsymbol{A}_i^0$ define the shape of the Gaussian in the following way: if the elements are all equal, the Gaussian will be spherical; if two are the same it will be ellipsoidal with a circular cross section; and if all three are different it will be an ellipsoid with an elliptical cross section. In a system of uniform Gaussians the index i in $\boldsymbol{A}_i^0,$ can be dropped since they all will have the same diagonal elements.

The pair interaction energy, $\phi_{ij}^G(\vec{r}_{ij}),$ between two Gaussians labelled i and j is obtained as an integral:

$$\phi_{ij}^G(\vec{r}_{ij})=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g_i(\vec{r}_i)g_j(\vec{r}_j)u_{ij}(r_{ij})\,dv_i dv_j, \tag{5.194}$$

in which $u_{ij}(r_{ij})$ describes the interaction energy between the volume elements dv_i and $dv_j.$ For convenience this is assumed to be a sum of Gaussian terms:

$$u_{ij}(r_{ij})=\sum_{k=1}^{k_g} c_k \exp(-d_k r_{ij}^2), \tag{5.195}$$

where $k_g \leq 3.$ The coefficients, $c_k,$ and exponents, $d_k,$ are obtained by fitting to some less convenient analytical form such as the Lennard-Jones potential. Substituting this form into (5.194) and integrating leads to the result

$$\phi_{ij}^G(\vec{R}_{ij}, \boldsymbol{A}_i, \boldsymbol{A}_j)=\sum_{k=1}^{k_g} \phi_{ij}^k(\vec{R}_{ij}, \boldsymbol{A}_i, \boldsymbol{A}_j), \tag{5.196}$$

where

$$\phi_{ij}^k(\vec{R}_{ij}, \boldsymbol{A}_i, \boldsymbol{A}_j)=c_k\sqrt{|\boldsymbol{E}_{ij}^k|}\exp\left(-d_k \vec{R}_{ij}\cdot\boldsymbol{E}_{ij}^k\cdot\vec{R}_{ij}\right), \tag{5.197}$$

with

$$\boldsymbol{E}_{ij}^k=\left(\mathbf{1}+d_k\left[\boldsymbol{A}_i^{-1}+\boldsymbol{A}_j^{-1}\right]\right)^{-1}, \tag{5.198}$$

in which $\boldsymbol{E}_{ij}^k$ is a symmetric matrix and $\mathbf{1}$ is the unit matrix. In equations (5.196) to (5.198) the vector $\vec{R}_{ij}$ represents the centre-to-centre displacement between the Gaussians i and $j,$ i.e. $\vec{R}_{ij}=\vec{R}_j-\vec{R}_i.$ Since this is a pair interaction, we can write the configuration energy for a system of Gaussians as

$$\phi(\vec{r}^N, \boldsymbol{A}^N)=\sum_{i=1}^{N-1}\sum_{j>i}^{N}\sum_{k=1}^{k_g} \phi_{ij}^k(\vec{R}_{ij}, \boldsymbol{A}_i, \boldsymbol{A}_j). \tag{5.199}$$

The force on a Gaussian n is given by

$$\vec{f}_n = -\sum_{i=1}^{N-1}\sum_{j>i}^{N}\sum_{k=1}^{k_g}\frac{\partial}{\partial \vec{R}_n}\phi_{ij}^k(\vec{R}_{ij}, \boldsymbol{A}_i, \boldsymbol{A}_j) \tag{5.200}$$

where

$$\frac{\partial}{\partial \vec{R}_n}\phi_{ij}^k(\vec{R}_{ij}, \boldsymbol{A}_i, \boldsymbol{A}_j) = -2\,d_k\,\phi_{ij}^k(\vec{R}_{ij}, \boldsymbol{A}_i, \boldsymbol{A}_j)\,\boldsymbol{E}_{ij}^k\,\vec{R}_{ij}(\delta_{jn} - \delta_{in}). \tag{5.201}$$

The force contributions to Gaussians i and j due to the pair potential (5.197) are

$$\begin{aligned}\vec{f}_i &\leftarrow -\sum_{k=1}^{k_g} 2\,d_k\,\phi_{ij}^k(\vec{R}_{ij}, \boldsymbol{A}_i, \boldsymbol{A}_j)\,\boldsymbol{E}_{ij}^k\,\vec{R}_{ij},\\ \vec{f}_j &\leftarrow \sum_{k=1}^{k_g} 2\,d_k\,\phi_{ij}^k(\vec{R}_{ij}, \boldsymbol{A}_i, \boldsymbol{A}_j)\,\boldsymbol{E}_{ij}^k\,\vec{R}_{ij},\end{aligned} \tag{5.202}$$

which are equal and opposite. The total force on Gaussian n is therefore

$$\vec{f}_n = \sum_{i\neq n}^{N}\sum_{k=1}^{k_g} 2\,d_k\,\phi_{in}^k(\vec{R}_{in}, \boldsymbol{A}_i, \boldsymbol{A}_n)\,\boldsymbol{E}_{in}^k\,\vec{R}_{in}. \tag{5.203}$$

According to (5.152) the torque on Gaussian j is given by

$$\vec{\tau}_j = -\sum_{k=1}^{k_g}\sum_{\nu=1}^{3}\vec{a}_j^{\,\nu}\times\frac{\partial\phi_{ij}^k(\vec{R}_{ij}, \boldsymbol{A}_i, \boldsymbol{A}_j)}{\partial \vec{a}_j^{\,\nu}}, \tag{5.204}$$

where the partial derivative in (5.204) expands to

$$\frac{\partial}{\partial \vec{a}_j^{\,\nu}}\phi_{ij}^k(\vec{R}_{ij}, \boldsymbol{A}_i, \boldsymbol{A}_j) = \phi_{ij}^k(\vec{R}_{ij}, \boldsymbol{A}_i, \boldsymbol{A}_j)\left(\frac{1}{2|\boldsymbol{E}_{ij}^k|}\frac{\partial|\boldsymbol{E}_{ij}^k|}{\partial \vec{a}_j^{\,\nu}} - d_k\,\vec{R}_{ij}\cdot\frac{\partial \boldsymbol{E}_{ij}^k}{\vec{a}_j^{\,\nu}}\cdot\vec{R}_{ij}\right). \tag{5.205}$$

To obtain the derivative of the matrix $\boldsymbol{E}_{ij}^k$ we first note that for a non-singular matrix $\boldsymbol{M}$, with inverse $\boldsymbol{M}^{-1}$, that is dependent on a variable x we have

$$\frac{\partial \boldsymbol{M}^{-1}}{\partial x} = -\boldsymbol{M}^{-1}\frac{\partial \boldsymbol{M}}{\partial x}\boldsymbol{M}^{-1}, \tag{5.206}$$

which is easily proved by differentiating the relation $\boldsymbol{M}\,\boldsymbol{M}^{-1} = \boldsymbol{1}$. Applying this to the definition of $\boldsymbol{E}_{ij}^k$ in (5.198) gives

$$\frac{\partial \boldsymbol{E}_{ij}^k}{\partial \vec{a}_j^{\,\nu}} = -d_k\,\boldsymbol{E}_{ij}^k\frac{\partial \boldsymbol{A}_j^{-1}}{\partial \vec{a}_j^{\,\nu}}\boldsymbol{E}_{ij}^k = d_k\,\boldsymbol{E}_{ij}^k\,\boldsymbol{A}_j^{-1}\frac{\partial \boldsymbol{A}_j}{\partial \vec{a}_j^{\,\nu}}\boldsymbol{A}_j^{-1}\,\boldsymbol{E}_{ij}^k, \tag{5.207}$$

which, when combined with (5.193), becomes

$$\begin{aligned}\frac{\partial \boldsymbol{E}_{ij}^k}{\partial \vec{a}_j^{\nu}} &= d_k \boldsymbol{E}_{ij}^k \boldsymbol{A}_j^{-1}\left(\frac{\partial \boldsymbol{R}_j}{\partial \vec{a}_j^{\nu}} \boldsymbol{A}_j^0 \tilde{\boldsymbol{R}}_j + \boldsymbol{R}_j \boldsymbol{A}_j^0 \frac{\partial \tilde{\boldsymbol{R}}_j}{\partial \vec{a}_j^{\nu}}\right) \boldsymbol{A}_j^{-1} \boldsymbol{E}_{ij}^k, \\ &= d_k \boldsymbol{E}_{ij}^k \boldsymbol{A}_j^{-1}\left(\frac{\partial \boldsymbol{R}_j}{\partial \vec{a}_j^{\nu}} \tilde{\boldsymbol{R}}_j \boldsymbol{A}_j + \boldsymbol{A}_j \boldsymbol{R}_j \frac{\partial \tilde{\boldsymbol{R}}_j}{\partial \vec{a}_j^{\nu}}\right) \boldsymbol{A}_j^{-1} \boldsymbol{E}_{ij}^k, \\ &= d_k \boldsymbol{E}_{ij}^k \boldsymbol{A}_j^{-1} \frac{\partial \boldsymbol{R}_j}{\partial \vec{a}_j^{\nu}} \tilde{\boldsymbol{R}}_j \boldsymbol{E}_{ij}^k + d_k \boldsymbol{E}_{ij}^k \boldsymbol{R}_j \frac{\partial \tilde{\boldsymbol{R}}_j}{\partial \vec{a}_j^{\nu}} \boldsymbol{A}_j^{-1} \boldsymbol{E}_{ij}^k.\end{aligned} \tag{5.208}$$

So we can write, from the right of (5.205):

$$d_k \vec{R}_{ij} \cdot \frac{\partial \boldsymbol{E}_{ij}^k}{\vec{a}_j^{\nu}} \cdot \vec{R}_{ij} = d_k^2 \left(\vec{U}_{ij}^k \cdot \frac{\partial \boldsymbol{R}_j}{\vec{a}_j^{\nu}} \tilde{\boldsymbol{R}}_j \cdot \vec{V}_{ij}^k + \vec{V}_{ij}^k \cdot \boldsymbol{R}_j \frac{\partial \tilde{\boldsymbol{R}}_j}{\vec{a}_j^{\nu}} \cdot \vec{U}_{ij}^k\right) \tag{5.209}$$

where we have defined the vectors

$$\begin{aligned}\vec{U}_{ij}^k &= \boldsymbol{A}_j^{-1} \boldsymbol{E}_{ij}^k \vec{R}_{ij}, \\ \vec{V}_{ij}^k &= \boldsymbol{E}_{ij}^k \vec{R}_{ij}.\end{aligned} \tag{5.210}$$

The differentiation of $\boldsymbol{R}_j$ with respect to the μ - component of $\vec{a}_j^{\nu}$ yields a matrix that is zero everywhere except at element $\mu\nu$ where it is unity. The terms in brackets right of (5.209) both collapse easily giving the same result, so (5.209) becomes

$$d_k \vec{R}_{ij} \cdot \frac{\partial \boldsymbol{E}_{ij}^k}{\vec{a}_j^{\nu}} \cdot \vec{R}_{ij} = 2 d_k^2 (\tilde{\boldsymbol{R}}_j \cdot V_{ij}^k)^{\nu} \vec{U}_{ij}^k, \tag{5.211}$$

in which $(\tilde{\boldsymbol{R}}_j \cdot V_{ij}^k)^{\nu}$ is the ν - component of vector $\tilde{\boldsymbol{R}}_j \cdot V_{ij}^k$

To obtain the derivative of $|\boldsymbol{E}_{ij}^k|$ in (5.205) we first expand the determinant as

$$|\boldsymbol{E}_{ij}^k| = \sum_{\alpha=1}^{3} \sum_{\beta=1}^{3} \sum_{\gamma=1}^{3} \epsilon_{\alpha\beta\gamma} (E_{ij}^k)^{1\alpha} (E_{ij}^k)^{2\beta} (E_{ij}^k)^{3\gamma}, \tag{5.212}$$

where $\epsilon_{\alpha\beta\gamma}$ is known as the Levi-Civita tensor which is defined as follows:

$$\begin{aligned}&\epsilon_{\alpha\beta\gamma} \neq 0, \quad \textit{only if } \alpha \neq \beta \neq \gamma, \\ &\epsilon_{\alpha\beta\gamma} = 1, \quad \textit{if } \alpha\beta\gamma \textit{ is even permutation}, \\ &\epsilon_{\alpha\beta\gamma} = -1, \quad \textit{if } \alpha\beta\gamma \textit{ is odd permutation.}\end{aligned} \tag{5.213}$$

(For a 3×3 matrix, such as $\boldsymbol{E}_{ij}^k$, the *even* permutations are $123, 231, 312$ and the *odd* permutations are $132, 213, 321.$) Using this expansion it can be shown that

$$\frac{\partial |\boldsymbol{E}_{ij}^k|}{\partial \vec{a}_j^{\nu}} = \sum_{\alpha=1}^{3} \sum_{\beta=1}^{3} \left(\frac{\partial (E_{ij}^k)_{\alpha\beta}}{\partial \vec{a}_j^{\nu}} (E_{ij}^k)_{\alpha\beta}^{-1}\right). \tag{5.214}$$

Inserting (5.208) into (5.214) leads to

$$\frac{1}{|\boldsymbol{E}_{ij}^k|}\frac{\partial|\boldsymbol{E}_{ij}^k|}{\partial(\vec{a}_j^{\,\nu})^\mu}=\sum_{\alpha=1}^{3}\sum_{\beta=1}^{3}\left(d_k\left\{\boldsymbol{E}_{ij}^k\boldsymbol{A}_j^{-1}\frac{\partial\boldsymbol{R}_j}{\partial\boldsymbol{R}_j^{\mu\nu}}\tilde{\boldsymbol{R}}_j\boldsymbol{E}_{ij}^k+\boldsymbol{E}_{ij}^k\boldsymbol{R}_j\frac{\partial\tilde{\boldsymbol{R}}_j}{\partial\boldsymbol{R}_j^{\mu\nu}}\boldsymbol{A}_j^{-1}\boldsymbol{E}_{ij}^k\right\}^{\alpha\beta}((E_{ij}^k)^{-1})^{\alpha\beta}\right), \tag{5.215}$$

where $(\vec{a}_j^{\,\nu})^\mu$ is the μ - component of $\vec{a}_j^{\,\nu}$, which corresponds to $(\boldsymbol{R}_j)^{\mu\nu}=R_j^{\mu\nu}$, the $\mu\nu$ - component of matrix $\boldsymbol{R}_j$. We also note that the derivative of $\boldsymbol{R}_j$ with respect to $R_j^{\mu\nu}$ is a matrix that is zero everywhere, except at location $\mu\nu$, where it is unity. Thus (5.215) can be written as

$$\frac{1}{|\boldsymbol{E}_{ij}^k|}\frac{\partial|\boldsymbol{E}_{ij}^k|}{\partial(\vec{a}_j^{\,\nu})^\mu}=\sum_{\alpha=1}^{3}\sum_{\beta=1}^{3}\left(d_k\left\{(\boldsymbol{E}_{ij}^k\boldsymbol{A}_j^{-1})^{\alpha\mu}(\tilde{\boldsymbol{R}}_j\boldsymbol{E}_{ij}^k)^{\nu\beta}+(\boldsymbol{E}_{ij}^k\boldsymbol{A}_j^{-1})^{\beta\mu}(\tilde{\boldsymbol{R}}_j\boldsymbol{E}_{ij}^k)^{\nu\alpha}\right\}((E_{ij}^k)^{-1})^{\alpha\beta}\right). \tag{5.216}$$

Expanding the matrix products in (5.216) and summing over α and β gives the result

$$\frac{1}{|\boldsymbol{E}_{ij}^k|}\frac{\partial|\boldsymbol{E}_{ij}^k|}{\partial(\vec{a}_j^{\,\nu})^\mu}=2d_k(\boldsymbol{A}_j^{-1}\boldsymbol{E}_{ij}^k\boldsymbol{R}_j)^{\mu\nu}\quad\text{or}\quad\frac{1}{|\boldsymbol{E}_{ij}^k|}\frac{\partial|\boldsymbol{E}_{ij}^k|}{\partial\vec{a}_j^{\,\nu}}=2d_k(\boldsymbol{A}_j^{-1}\boldsymbol{E}_{ij}^k)\cdot\vec{a}_j^{\,\nu} \tag{5.217}$$

Inserting (5.211) and (5.217) into (5.205) gives the derivative of the potential energy, $\phi_{ij}^k(\vec{R}_{ij},\boldsymbol{A}_i,\boldsymbol{A}_j)$, with respect to $\vec{a}_j^{\,\nu}$ as

$$\frac{\partial}{\partial\vec{a}_j^{\,\nu}}\phi_{ij}^k(\vec{R}_{ij},\boldsymbol{A}_i,\boldsymbol{A}_j)=d_k\phi_{ij}^k(\vec{R}_{ij},\boldsymbol{A}_i,\boldsymbol{A}_j)\left[(\boldsymbol{A}_j^{-1}\boldsymbol{E}_{ij}^k)\vec{a}_j^{\,\nu}-2d_k(\tilde{\boldsymbol{R}}_j\cdot\vec{V}_{ij}^k)^\nu\vec{U}_{ij}^k\right]. \tag{5.218}$$

Both terms inside the square brackets of (5.218) are vectors. Combining this result with equation (5.204) and rearranging gives the following expression for the torque:

$$\vec{\tau}_j=-\sum_{k=1}^{k_g}d_k\phi_{ij}^k(\vec{R}_{ij},\boldsymbol{A}_i,\boldsymbol{A}_j)\left(\sum_{\nu=1}^{3}\vec{a}_j^{\,\nu}\times\left[(\boldsymbol{A}_j^{-1}\boldsymbol{E}_{ij}^k)\vec{a}_j^{\,\nu}-2d_k(\tilde{\boldsymbol{R}}_j\vec{V}_{ij}^k)^\nu\vec{U}_{ij}^k\right]\right). \tag{5.219}$$

The torque on molecule i is equivalently:

$$\vec{\tau}_i=-\sum_{k=1}^{k_g}d_k\phi_{ij}^k(\vec{R}_{ij},\boldsymbol{A}_i,\boldsymbol{A}_j)\left(\sum_{\nu=1}^{3}\vec{a}_i^{\,\nu}\times\left[(\boldsymbol{A}_i^{-1}\boldsymbol{E}_{ij}^k)\vec{a}_i^{\,\nu}-2d_k(\tilde{\boldsymbol{R}}_i\vec{V}_{ij}^k)^\nu\vec{W}_{ij}^k\right]\right), \tag{5.220}$$

where

$$\vec{W}_{ij}^k=\boldsymbol{A}_i^{-1}\boldsymbol{E}_{ij}^k\vec{R}_{ij}. \tag{5.221}$$

The vector products appearing in (5.219) can be simplified. Firstly define vector $\vec{P}_{ij}^k$ as

$$\vec{P}_{ij}^k=\sum_{\nu=1}^{3}\vec{a}_j^{\,\nu}\times(\boldsymbol{A}_j^{-1}\boldsymbol{E}_{ij}^k)\vec{a}_j^{\,\nu}, \tag{5.222}$$

then

$$
\begin{aligned}
(\vec{P}_{ij}^{\,k})^{\alpha} &= \sum_{\nu=1}^{3}\sum_{\beta=1}^{3}\sum_{\gamma=1}^{3} \epsilon_{\alpha\beta\gamma}(\vec{a}_j^{\,\nu})^{\beta}\big((\boldsymbol{A}_j^{-1}\boldsymbol{E}_{ij}^{k})\vec{a}_j^{\,\nu}\big)^{\gamma}, \\
&= \sum_{\nu=1}^{3}\sum_{\beta=1}^{3}\sum_{\gamma=1}^{3}\sum_{\lambda=1}^{3} \epsilon_{\alpha\beta\gamma}(\vec{a}_j^{\,\nu})^{\beta}\big(\boldsymbol{A}_j^{-1}\boldsymbol{E}_{ij}^{k}\big)^{\gamma\lambda}(\vec{a}_j^{\,\nu})^{\lambda}, \\
&= \sum_{\nu=1}^{3}\sum_{\beta=1}^{3}\sum_{\gamma=1}^{3}\sum_{\lambda=1}^{3} \epsilon_{\alpha\beta\gamma}\big(\boldsymbol{A}_j^{-1}\boldsymbol{E}_{ij}^{k}\big)^{\gamma\lambda}\boldsymbol{R}_j^{\beta\nu}\boldsymbol{R}_j^{\lambda\nu}, \\
&= \sum_{\beta=1}^{3}\sum_{\gamma=1}^{3}\sum_{\lambda=1}^{3} \epsilon_{\alpha\beta\gamma}\big(\boldsymbol{A}_j^{-1}\boldsymbol{E}_{ij}^{k}\big)^{\gamma\lambda}\delta_{\beta\lambda}, \\
&= \sum_{\beta=1}^{3}\sum_{\gamma=1}^{3} \epsilon_{\alpha\beta\gamma}\big(\boldsymbol{A}_j^{-1}\boldsymbol{E}_{ij}^{k}\big)^{\gamma\beta}, \\
&= \sum_{\beta=1}^{3}\sum_{\gamma=1}^{3} \epsilon_{\alpha\beta\gamma}\big(\boldsymbol{E}_{ij}^{k}\boldsymbol{A}_j^{-1}\big)^{\beta\gamma}.
\end{aligned}
\tag{5.223}
$$

Hence

$$
\vec{P}_{ij}^{\,k} = \boldsymbol{\epsilon}:\big(\boldsymbol{E}_{ij}^{k}\boldsymbol{A}_j^{-1}\big). \tag{5.224}
$$

Next, we define vector $\vec{Q}_{ij}^{\,k}$ as

$$
\vec{Q}_{ij}^{\,k} = \sum_{\nu=1}^{3}\vec{a}_j^{\,\nu}\times\big(\tilde{\boldsymbol{R}}_j\vec{V}_{ij}^{\,k}\big)^{\nu}\vec{U}_{ij}^{\,k}, \tag{5.225}
$$

then

$$
\begin{aligned}
(\vec{Q}_{ij}^{\,k})^{\alpha} &= \sum_{\nu=1}^{3}\sum_{\beta=1}^{3}\sum_{\gamma=1}^{3} \epsilon_{\alpha\beta\gamma}(\vec{a}_j^{\,\nu})^{\beta}\big(\tilde{\boldsymbol{R}}_j\vec{V}_{ij}^{\,k}\big)^{\nu}\big(\vec{U}_{ij}^{\,k}\big)^{\gamma}, \\
&= \sum_{\nu=1}^{3}\sum_{\beta=1}^{3}\sum_{\gamma=1}^{3}\sum_{\lambda=1}^{3} \epsilon_{\alpha\beta\gamma}\boldsymbol{R}_j^{\beta\nu}\tilde{\boldsymbol{R}}_j^{\nu\lambda}\big(\vec{V}_{ij}^{\,k}\big)^{\lambda}\big(\vec{U}_{ij}^{\,k}\big)^{\gamma}, \\
&= \sum_{\beta=1}^{3}\sum_{\gamma=1}^{3}\sum_{\lambda=1}^{3} \epsilon_{\alpha\beta\gamma}\delta_{\beta\lambda}\big(\vec{V}_{ij}^{\,k}\big)^{\lambda}\big(\vec{U}_{ij}^{\,k}\big)^{\gamma}, \\
&= \sum_{\beta=1}^{3}\sum_{\gamma=1}^{3} \epsilon_{\alpha\beta\gamma}\big(\vec{V}_{ij}^{\,k}\big)^{\beta}\big(\vec{U}_{ij}^{\,k}\big)^{\gamma}.
\end{aligned}
\tag{5.226}
$$

Hence

$$
\vec{Q}_{ij}^{\,k} = \vec{V}_{ij}^{\,k}\times\vec{U}_{ij}^{\,k}. \tag{5.227}
$$

Putting (5.223) and (5.227) back into (5.219) gives the final result

$$
\vec{\tau}_j = -\sum_{k=1}^{k_g} d_k\,\phi_{ij}^{k}\big(\vec{R}_{ij},\boldsymbol{A}_i,\boldsymbol{A}_j\big)\Big(\boldsymbol{\epsilon}:\big(\boldsymbol{E}_{ij}^{k}\boldsymbol{A}_j^{-1}\big) - 2\,d_k\,\vec{V}_{ij}^{\,k}\times\vec{U}_{ij}^{\,k}\Big). \tag{5.228}
$$

Similarly, the torque on molecule i is

$$\vec{\tau}_i = -\sum_{k=1}^{k_g} d_k \phi_{ij}^k \left(\vec{R}_{ij}, \boldsymbol{A}_i, \boldsymbol{A}_j \right) \left(\boldsymbol{\epsilon} : \left(\boldsymbol{E}_{ij}^k \boldsymbol{A}_i^{-1} \right) - 2 d_k \vec{V}_{ij}^k \times \vec{W}_{ij}^k \right). \tag{5.229}$$

The equations (5.228) and (5.229) can be written in other ways, see [49] and [27].

Chapter 6

Coulombic Forces

6.1 Introduction

Coulombic forces, which are *electrostatic* in origin[20], represent a particular difficulty in molecular simulation because they are *long ranged*. Given that most Coulombic systems are nominally infinite (or at least pseudo-infinite), this is a worry. In practical terms this means the influence of Coulombic interactions cannot be expected to decay to negligibility over the range of a few atomic diameters. Indeed Coulombic forces are particularly pathological in this regard, as can be seen from a simple example.

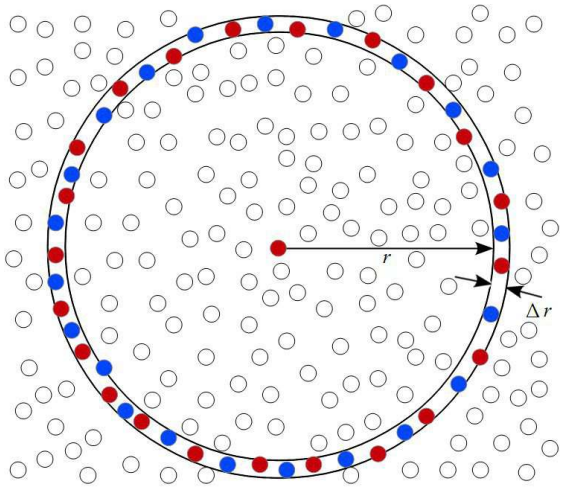

Figure 6.1. Long-ranged Effects in Condensed Phase Ionic Systems

Figure 6.1 represents a condensed phase ionic system. The nominated *central* ion is surrounded by other ions, with like and unlike charges. (Overall the system is electrically neutral and the ions are considered to be *point entities*.) The ions in the (3D) *spherical shell* indicated, which has width Δr, are all at the same approximate distance from the central ion. The overall charge in the whole shell will vary with the radius r. The question is what effect does the net charge in a given shell have on the central ion?

Denoting the average positive and negative charge densities as a function of r by $\rho^+(r)$ and $\rho^-(r)$ the charge on a shell, q^{shell}, is given approximately by

$$q^{shell} = 4\pi\left(\rho^+(r)+\rho^-(r)\right)r^2\Delta r, \tag{6.1}$$

20 In this chapter, unless specified otherwise, we assume that the Coulombic system is overall charge neutral.

This shell of charge contributes to the electrostatic potential of the central ion, with charge $q^{ion},$ by an amount

$$\Delta\phi^{ion}=\frac{1}{4\pi\epsilon_0}\frac{q^{ion}q^{shell}}{r}=\frac{1}{\epsilon_0}q^{ion}\left(\rho^{+}(r)+\rho^{-}(r)\right)r\,\Delta r. \tag{6.2}$$

The remarkable thing about this result is that this potential no longer has the usual $1/r$ dependence, but in fact is most strongly governed by its dependence on $\rho^{+}(r)$ and $\rho^{-}(r),$ neither of which go to zero at long range. Thus the collective influence of distant ions is at least as significant as that of closer ions in determining the physics of the system. However, at extreme long range, $\rho^{+}(r)$ and $\rho^{-}(r)$ approach equal and opposite constant values (the uniform charge densities of the bulk system) and, as a result, are expected to cancel each other. So there must exist a natural cut-off at which the influence of distant charges is expected to vanish. This reduces the problem to one of determining the range at which the radial distribution functions of both the ionic species can be considered to reach the bulk values. Unfortunately this is not known *a priori*, but it *is* known that the oscillations in the radial distribution functions can persist for many molecular diameters [50]. In the absence of any fore-knowledge of this range, we are reduced to guessing what the critical range is in any given system.

Many different approaches have been devised for handling Coulombic systems in molecular simulation, with differing degrees of reliability and trustworthiness. Arguably the method best suited to simulations of condensed phases is the venerable Ewald method [51]. This method is both accurate and computationally flexible, though it has an undeserved reputation for being complicated and, by inference, slow. An adaptation of the method using the Fast Fourier Transform (FFT) and known as the Smoothed Particle Mesh Ewald (SPME) [52] is both highly efficient and adaptable to large scale parallel computation [53]. It is currently the method of choice for large scale simulations. Its main rival is the Fast Multipole Method [54], which is itself rather complicated and seems not to have been adopted to the same extent by the molecular simulation community, but it scales linearly with problem size, something the Ewald based methods cannot claim. Other methods commonly used in the past (but hopefully not today) are based on physical intuition at best or desperation at worst. These will be covered here, if only to inform unthinking users of their dangers.

We should also mention other issues that can arise, such as the presence of an overall charge in the system, or the occurrence of a net system dipole. Other complications include polarisable ions or point multipoles in the system. All of these require additional consideration.

6.2 Direct Summation Methods

In the case of crystalline solids, there is a long history of ingenious direct-summation methods being used to obtain the Coulombic energy. Unfortunately these are too specialised to be of much use in molecular simulation, where the ions spend most of their time displaced from their ideal sites. Direct summation of the Coulomb energy is however perfectly acceptable and accurate when applied to a finite system, such as a

cluster of ions, provided every possible pair interaction is considered. The Coulombic configuration energy, $\Phi^C(\{\vec{r}_i\})$, is then given as

$$\Phi^C(\{\vec{r}_i\})=\frac{1}{4\pi\epsilon_0}\sum_{i=1}^{N-1}\sum_{j=i+1}^{N}\frac{q_i q_j}{r_{ij}}, \tag{6.3}$$

in which $r_{ij}=|\vec{r}_j-\vec{r}_i|$ is the inter-ionic distance. Note that none of the N ions in the system is excluded from the double sum. The force on an arbitrary ion, n, is then given as

$$\vec{f}_n=-\frac{\partial\Phi^C(\{\vec{r}_i\})}{\partial\vec{r}_n}=\frac{1}{4\pi\epsilon_0}\sum_{i=1}^{N-1}\sum_{j=i+1}^{N}\frac{q_i q_j}{r_{ij}^3}(\delta_{jn}-\delta_{in})\vec{r}_{ij}=\frac{q_n}{4\pi\epsilon_0}\sum_{j\neq n}^{N}\frac{q_j}{r_{jn}^3}\vec{r}_{jn}. \tag{6.4}$$

Alternatively, we can write that the contributions to the forces on the ions i and j from a single Coulombic pair interaction as

$$\vec{f}_i\leftarrow-\frac{1}{4\pi\epsilon_0}\frac{q_i q_j}{r_{ij}^3}\vec{r}_{ij}, \qquad \vec{f}_j\leftarrow\frac{1}{4\pi\epsilon_0}\frac{q_i q_j}{r_{ij}^3}\vec{r}_{ij}. \tag{6.5}$$

These force contributions being equal and opposite, as required.

A commonly attempted, but erroneous, approximation to the Coulomb sum is to truncate the sum (6.3) by the application of a cut-off condition:

$$\Phi^C(\{\vec{r}_i\})=\frac{1}{4\pi\epsilon_0}\sum_{i=1}^{N-1}\sum_{j=i+1}^{N}\frac{q_i q_j}{r_{ij}}H(r_{cut}-r_{ij}), \tag{6.6}$$

where $H(x)$ is the Heaviside function, defined as

$$\begin{aligned} H(x)&=0 \qquad when\ x<0,\\ H(x)&=1 \qquad when\ x\geq 0. \end{aligned} \tag{6.7}$$

In this case the force on the ion n is

$$\vec{f}_n=\frac{q_n}{4\pi\epsilon_0}\sum_{j\neq n}^{N}\left\{\frac{q_j}{r_{jn}^3}H(r_{cut}-r_{jn})+\frac{q_j}{r_{jn}^2}\delta(r_{cut}-r_{jn})\right\}\vec{r}_{jn}, \tag{6.8}$$

The function $\delta(x)$ is the *Dirac delta function*, which is the derivative of the Heaviside function and is zero everywhere except at $x=0$. The first term right of (6.8) is clearly the normal Coulombic force truncated at r_{cut}. However the second term represents an *impulse force*, which results from the cut off condition. This is normally discarded in molecular dynamics, but is nevertheless symptomatic of unexpected behaviour at the cut off. It should also be noted that the use of truncation in (6.6) implies that artificial fluctuations in the potential energy will occur, the magnitude of which is determined by r_{cut}. In many Coulombic systems these fluctuations are likely to be unacceptably large.

Attempts have been made to mitigate these defects by truncating and shifting the Coulombic potential so that it is zero at the cut-off and at longer range i.e.

$$\Phi^C(\{\vec{r}_i\}) = \frac{1}{4\pi\epsilon_0}\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} q_i q_j \left(\frac{1}{r_{ij}} - \frac{1}{r_{cut}}\right), \quad \text{with} \quad r_{ij} < r_{cut}. \tag{6.9}$$

This removes artificial fluctuations and impulse forces caused by the cut-off. However, such modifications cannot restore the lost long range effects of the Coulombic interaction. The system dynamics will inevitably be affected to an uncertain degree and the loss of terms that contribute to the configuration energy and virial mean that the system cohesive energy and pressure are likely to be substantially wrong. It follows that isothermal-isobaric simulations are not possible in such systems, which is a significant disadvantage.

Some practitioners have used a modified version of the Coulomb sum in which a variable dielectric function is introduced to dampen the range of interaction, as in

$$\Phi^C(\{\vec{r}_i\}) = \frac{1}{4\pi\epsilon_0}\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} \frac{q_i q_j}{\epsilon(r_{ij}) r_{ij}}, \quad \text{with} \quad r_{ij} < r_{cut}. \tag{6.10}$$

A common practice is to assume that $\epsilon(r)$ is linear in r, which is tantamount to modifying the distance dependence of Coulomb's law to $1/r^2$, which undoubtedly reduces the long-ranged aspect of the Coulombic interaction. This is arguable on the grounds that "screening" of more distant charges by the (dielectric) medium is a valid macroscopic description of what occurs. (This is a similar argument to that appearing in section 6.1 above). However, the mathematical dependence of ϵ on r is merely a convenient assumption and to screen close inter-atomic interactions in this way is questionable and likely to affect local structure significantly.

More sophisticated schemes exist, combining both trunctation and screening, such as that devised by Fennell and Gezelter [55] (based on the earlier scheme by Wolf et al. [56]) in which a screening function (in the form of a *complementary error-function*) is applied to all Coulombic interactions and steps are taken to ensure that both Coulombic potential and force are continuous at the cut-off. The Coulombic potential in this case is

$$\Phi^C(\{\vec{r}_i\}) = \frac{1}{4\pi\epsilon_0}\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} q_i q_j \left(\frac{erfc(\alpha r_{ij})}{r_{ij}} - \frac{erfc(\alpha r_{cut})}{r_{cut}} + \left(\frac{erfc(\alpha r_{cut})}{r_{cut}^2} + \frac{2\alpha}{\sqrt{\pi}}\frac{\exp(-\alpha^2 r_{cut}^2)}{r_{cut}}\right)(r_{ij} - r_{cut})\right) \quad r_{ij} < r_{cut}, \tag{6.11}$$

in which α represents the *screening constant*, which determines the range of the screening. This is barely recognisable as a description of the Coulombic potential (which is encapsulated in the $erfc(\alpha r_{ij})/r_{ij}$ term) as most of the terms are designed to handle the cut-off requirements. The force on an arbitrary ion, n, in this case is given by

$$\vec{f}_n = \frac{q_n}{4\pi\epsilon_0}\sum_{j\neq 1}^{N} q_j \left(\frac{erfc(\alpha r_{jn})}{r_{jn}^2} + \frac{2\alpha}{\sqrt{\pi}}\frac{\exp(-\alpha^2 r_{jn}^2)}{r_{jn}} - \frac{erfc(\alpha r_{cut})}{r_{cut}^2} - \frac{2\alpha}{\sqrt{\pi}}\frac{\exp(-\alpha^2 r_{cut}^2)}{r_{cut}} \right) \frac{\vec{r}_{jn}}{r_{jn}} \qquad r_{jn} < r_{cut}. \tag{6.12}$$

It should be apparent from these formulae that, at the cut-off, the pair terms make no contribution to either the potential or the force. The authors report that this method provides an accurate description of Coulombic systems, provided r_{cut} is taken large enough. This implies it works best for large systems.

It is noteworthy that the choice of the complementary error function as the screening function in this method undoubtedly derives from it use in the Ewald sum (see section 6.4) where it performs a similar role.

This section shows that deficiencies in direct summation methods are quite common and not always obvious. In the past they were frequently used as methods of last resort, on account of the computational expense of obtaining an accurate Coulomb sum. However, given the power of modern computing resources, their continued use is now unjustifiable. Hopefully nobody today would think seriously about using them, other than for finite systems and with the possible exception of the Fennell and Gezelter method applied to large systems.

6.3 The Reaction Field Method

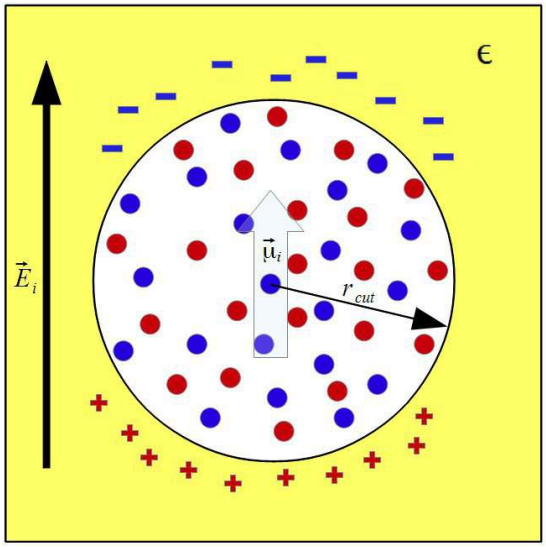

Figure 6.2: The Reaction Field Method

The Reaction Field Method [57] is a compromise between direct summation and macroscopic approximation. For a given ion, i, Coulomb interactions are calculated explicitly out to the specified cut-off distance, r_{cut}, but a long range correction is then applied, based on the assumption that, beyond the cut-off, the system can be approximated as a polarisable dielectric medium. For the given ion, i, the cut-off also defines a spherical cavity in the dielectric medium centred on the ion (see figure 6.2) and the other ions within the cavity constitute an electric dipole μ_i that polarises the dielectric medium. The polarisation, in turn, engenders a *reaction field*, $\vec{E}_i$, that interacts with the ion i. This interaction constitutes the long-ranged correction. According to the method described by Neumann [57], the system Coulombic configuration energy is given by

$$\Phi^C(\{\vec{r}_i\}) = \frac{1}{4\pi\epsilon_0}\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} q_i q_j \left(\frac{1}{r_{ij}} + \frac{B_0 r_{ij}^2}{2 r_{cut}^3}\right), \tag{6.13}$$

in which the second term in brackets on the right represents the reaction field correction. The constant B_0 is given as

$$B_0 = \frac{2(\epsilon-1)}{(2\epsilon+1)}, \tag{6.14}$$

in which ϵ is the dielectric constant of the polarisable medium. This implies that, in order to use this method, ϵ must be known beforehand, which may be problematical in an unknown system. However, in many ionic systems ϵ is expected to be large and in such cases setting $B_0 = 1$ is reasonable as a first approximation.

The force on an arbitrary ion, n, is given by

$$\vec{f}_n = \frac{q_n}{4\pi\epsilon_0}\sum_{j\neq n}^{N} q_j\left(\frac{1}{r_{jn}^3}-\frac{B_0}{r_{cut}^3}\right)\vec{r}_{jn} \qquad r_{jn}<r_{cut} \tag{6.15}$$

and the force contributions to the ions i and j arising from the pair interaction are

$$\vec{f}_i \leftarrow -\frac{q_i q_j}{4\pi\epsilon_0}\left(\frac{1}{r_{jn}^3}-\frac{B_0}{r_{cut}^3}\right)\vec{r}_{ij}, \qquad \vec{f}_j \leftarrow \frac{q_i q_j}{4\pi\epsilon_0}\left(\frac{1}{r_{jn}^3}-\frac{B_0}{r_{cut}^3}\right)\vec{r}_{ij}. \qquad r_{ij}<r_{cut} \tag{6.16}$$

The interesting thing about the reaction field method is that it makes a real attempt to deal correctly with the long ranged aspect of Coulombic forces and does not make *ad hoc* changes to the basic physics. To that extent it is better founded than most methods. It does have problems however; if the constant B_0 is far from unity, large fluctuations in the configuration energy can arise and the pair forces are discontinuous at the cut-off, giving rise to artificial impulse forces. This takes us back to the various *fixes* described for the direct summation methods. The best fix to apply in this case is probably the Fennel and Geselter method [55] The larger cut-off required by this method is enough to improve things on its own.

6.4 The Ewald Method

In molecular simulation work the Ewald method [51] (or the Ewald sum) is arguably the best choice for handling Coulombic interactions. It is robust and highly accurate and guaranteed to converge in any periodic system with relatively little *fine tuning*. A few precautions are necessary for proper use however and these will be outlined here, but on the whole there is no real difficulty in using the Ewald method to obtain reliable results. It is not the easiest method to program, but it *is* the easiest of the *accurate* methods, and while it may not be the fastest in all circumstances, its variant - the Smoothed Particle-Mesh Ewald (SPME) - is among the fastest for condensed phase systems. No one engaging in condensed phase simulation today can afford not to know about the Ewald method in some detail.

6.4.1 Description of the Ewald Method

P.P. Ewald developed his method for summing the Coulombic configuration energy in ideal, periodic lattices, which is normally given by:

$$\Phi^C\left(\{\vec{r}_i\}\right) = \frac{1}{4\pi\epsilon_0}\sum_{\vec{L}=\vec{0}}^{\vec{\infty}}\sum_{i=1}^{N-1}\sum_{j=i+1}^{N}\frac{q_i q_j}{|\vec{L}+\vec{r}_j-\vec{r}_i|}, \tag{6.17}$$

in which $\vec{L}$ is a lattice vector, $\vec{r}_i$ a position vector for the $i'th$ ion with charge q_i and N is the number of ions in the unit cell. This is obviously a generalisation of equation (6.3) for an infinite, periodic lattice. Ewald's was the first general method for lattice sums, guaranteed to converge whatever the symmetry properties of the unit cell. Since most molecular simulations assume periodic boundaries, the method can be used directly in that field also. The unit cell of the original method becomes the

simulation cell in molecular dynamics or Monte Carlo.

The method is based on an ingenious transformation in which the Coulomb sum (6.17) is separated into two sums, one of which is in normal coordinates, or *real* space, and the other is in the space of the crystallographic reciprocal lattice, or *reciprocal* space. In both spaces the sums are formally infinite but, by a suitable choice of a *convergence parameter*, can be guaranteed to converge, so overall convergence of the Ewald sum is also guaranteed. This is in marked contrast to the unmodified Coulomb sum (6.17).

We shall first write the Ewald sum here to introduce the important variables and then later explain how it is derived. The sum, for a system that is overall charge neutral, is written as

$$\Phi^E(\{\vec{r}_i\}) = \frac{1}{2\epsilon_0 V}\sum_{\vec{k}\neq\vec{0}}^{\vec{\infty}} \frac{\exp(-k^2/4\alpha^2)}{k^2}\left|\sum_{j=1}^{N} q_j \exp(i\vec{k}\cdot\vec{r}_j)\right|^2 + \frac{1}{2}\frac{1}{4\pi\epsilon_0}\sum_{\vec{L}=\vec{0}}^{\vec{\infty}}\sum_{j=1}^{N}\sum_{n=1}^{N}{}' \frac{q_j q_n erfc(\alpha r_{jn}^L)}{r_{jn}^L} - \frac{\alpha}{4\pi^{3/2}\epsilon_0}\sum_{j=1}^{n} q_j^2. \tag{6.18}$$

Three terms are evident on the right of (6.18). The first is the so-called *reciprocal space sum* and the second is the *real space* sum. The final term is called the *self interaction correction*. We note the presence of the parameter α in all three terms. This is the *Ewald convergence parameter* that must be "tuned" to produce optimal convergence. Elsewhere, N is the number of ions in the unit or simulation cell, r_{jn}^L represents the distance between ions in the infinite lattice:

$$r_{jn}^L = \left|\vec{L} + \vec{r}_j - \vec{r}_n\right| \tag{6.19}$$

and $\vec{k}$ is a reciprocal space vector:

$$\vec{k} = \mu\vec{u} + \nu\vec{v} + \lambda\vec{w}, \tag{6.20}$$

in which μ, ν, λ are integers and $\vec{u}, \vec{v}, \vec{w}$ are the *basis vectors* of the reciprocal lattice defined by

$$\vec{u} = \frac{2\pi}{V}\vec{b}\times\vec{c}, \qquad \vec{v} = \frac{2\pi}{V}\vec{c}\times\vec{a}, \qquad \vec{w} = \frac{2\pi}{V}\vec{a}\times\vec{b}, \tag{6.21}$$

where $\vec{a}, \vec{b}, \vec{c}$ are the basis vectors for the periodic lattice (either of the crystallographic unit cell or the simulation cell) and V is the volume of the periodic cell, which is given by

$$V = |\vec{a}\cdot(\vec{b}\times\vec{c})| = |\vec{b}\cdot(\vec{c}\times\vec{a})| = |\vec{c}\cdot(\vec{a}\times\vec{b})|. \tag{6.22}$$

The reciprocal space term in (6.18) requires a sum over the components of each $\vec{k}$ vector and is therefore a triple sum. This is potentially an infinite sum over the vectors $\vec{k}$. However the range of this sum is moderated by the presence of the Gaussian

term $\exp(-k^2/4\alpha^2)/k^2$. As the magnitude of k^2 increases, this factor is driven to smaller values and the terms become negligible at some finite range. Control of the rate of convergence is afforded by the parameter α, such that decreasing its value shortens the range of convergence. Also present in the reciprocal space sum is the term we may write as

$$S(\vec{k})=\left|\sum_{j=1}^{N} q_j \exp(i\vec{k}\cdot\vec{r}_j)\right|^2=\left(\sum_{j=1}^{N} q_j \exp(i\vec{k}\cdot\vec{r}_j)\right)\left(\sum_{j=1}^{N} q_j \exp(-i\vec{k}\cdot\vec{r}_j)\right), \tag{6.23}$$

which is known as the *structure factor* or $S(\vec{k})$ which is a commonly encountered function in crystallographic theory and encapsulates the periodicity of the system. Despite the presence of the imaginary number, $i=\sqrt{-1}$, the structure factor is *real*, as the final expansion in (6.23) reveals it to be the product of a complex number with its complex conjugate. Lastly, it should be noted that the sum over $\vec{k}$ excludes the zero vector $\vec{k}=\vec{0}$. This mathematical necessity has subtle consequences in the physics of the Ewald sum, as will be evident later.

The real space term contains a sum over components of the $\vec{L}$ vectors, which represent periodic images of the simulation cell. Restrictions apply to the indices appearing here. Firstly the index j always refers to an ion in the *central cell* of the infinite periodic system, which corresponds to the cell at $\vec{L}=\vec{0}$. This is necessary to restrict the Coulombic energy to a total for N ions. Index n meanwhile can refer to ions inside or outside the central cell with $\vec{L}\geq\vec{0}$. If both j and n are in the central cell, then the condition $j\neq n$ applies. This is indicated by the dash $(')$ adjacent to the third summation sign. The real space sum is also a potentially infinite sum. However, here again the range is moderated, this time by the *complementary error function* $erfc(\alpha r_{nj}^L)$. This function drives down towards zero as the argument increases and so effectively truncates the infinite sum at a finite range. Control over the convergence range is again governed by the parameter α. As its name implies, the complementary error function is related to the error function that arises in statistics:

$$erf(\alpha r)=1-erfc(\alpha r), \tag{6.24}$$

where

$$erf(\alpha r)=\frac{2}{\sqrt{\pi}}\int_0^{\alpha r}\exp(-x^2)\,dx, \quad \text{and} \quad erfc(\alpha r)=\frac{2}{\sqrt{\pi}}\int_{\alpha r}^{\infty}\exp(-x^2)\,dx. \tag{6.25}$$

The self interaction correction is a simple sum of the squares of the charges in the periodic cell. This is an artefact of the construction of the Ewald method. It is a large term and therefore important, but it also happens to be a constant for any given system, so its cost implications are minimal in the context of simulation.

An important thing to note about the Ewald sum is that (6.18) returns the same value for $\Phi^C(\{\vec{r}_i\})$ no matter what value is assigned to α, provided the real and reciprocal space sums are computed far enough. Variation in α changes the relative

weight of the calculation in each domain. Setting $\alpha=0$ changes the Ewald sum to the standard Coulomb sum (6.17) and the reciprocal space sum and self interaction correction become zero. In this circumstance convergence cannot be guaranteed. Increasing α casts more of the calculation into reciprocal space and guarantees convergence. A balance needs to be struck between the volume of numerical calculation in real and reciprocal space to obtain the computationally optimal form.

6.4.2 Theory of the Ewald Sum

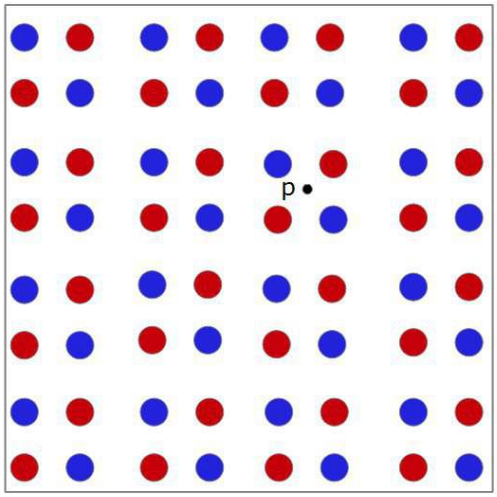

Figure 6.3: A "Probe" Unit Positive Charge in a Periodic Lattice of Ions.

In this section we present an outline of the derivation of the Ewald sum. In an infinite lattice of charged point ions shown in figure 6.3 we first determine the Coulombic potential of a unit positive point charge p (called the *probe* charge) at an arbitrary position $\vec{r}$.

Formally this is given by the Coulomb sum:

$$\phi(\vec{r})=\frac{1}{4\pi\epsilon_0}\sum_{\vec{L}=\vec{0}}^{\vec{\infty}}\sum_{j=1}^{N}\frac{q_j}{|\vec{r}-\vec{r}_j+\vec{L}|}. \tag{6.26}$$

As seen previously, the sum (6.26) is in reality infinite. To make it manageable we imagine each ion j is surrounded by a normalised Gaussian density distribution of opposite charge which has the form

$$\rho_j(\vec{r}-\vec{r}_j)dv=-q_j\left(\frac{\alpha^2}{\pi}\right)^{3/2}\exp\left(-\alpha^2|\vec{r}-\vec{r}_j|^2\right)dv. \tag{6.27}$$

centred on the ion at $\vec{r}_j$. The constant α determines the mean-square width of the Gaussian, which is $1/(2\alpha^2)$. To obtain the Coulombic potential of the probe charge, we must include this charge in the sum, and since this is a distribution, it must be integrated from centre of each ion, j, to the limit of the Gaussian distribution at infinity. In which case the sum (6.26) becomes

$$\phi_1(\vec{r}) = \frac{1}{4\pi\epsilon_0}\sum_{\vec{L}=\vec{0}}^{\vec{\infty}}\sum_{j=1}^{N}\frac{q_j}{|\vec{r}-\vec{r}_j+\vec{L}|}\left(1-\frac{2\alpha}{\sqrt{\pi}}\int_0^{|\vec{r}-\vec{r}_j+\vec{L}|}\exp(-\alpha^2 r^2)\,dr\right), \tag{6.28}$$

where we have used the the notation $\phi_1(\vec{r})$ to distinguish this from the true Coulomb sum (6.26). Note that in obtaining (6.28) we have made use of a well known electrostatic theorem which says that the potential at a point p, due to a spherical charge distribution centred at a point q, is equivalent to that of the spherically integrated charge over the range $|\vec{r}_p-\vec{r}_q|$ located entirely at the point q. According to (6.24) and (6.25), equation (6.28)can be written as

$$\begin{aligned}\phi_1(\vec{r}) &= \frac{1}{4\pi\epsilon_0}\sum_{\vec{L}=\vec{0}}^{\vec{\infty}}\sum_{j=1}^{\infty}\frac{q_j}{|\vec{r}-\vec{r}_j+\vec{L}|}\left(1-erf\left(\alpha|\vec{r}-\vec{r}_j+\vec{L}|\right)\right),\\ &= \frac{1}{4\pi\epsilon_0}\sum_{\vec{L}=\vec{0}}^{\vec{\infty}}\sum_{j=1}^{\infty}\frac{q_j}{|\vec{r}-\vec{r}_j+\vec{L}|}erfc\left(\alpha|\vec{r}-\vec{r}_j+\vec{L}|\right).\end{aligned} \tag{6.29}$$

Though formally (6.29) is an infinite sum, the $erfc$ function guarantees that it will converge in a finite spatial range. This is entirely keeping with the idea that surrounding each ion with a "cloud" of opposite charge will mitigate its effects at long range. This is the basis for the real space part of the Ewald sum.

Obviously, (6.29) does not represent the real Coulomb sum. It is necessary to correct for the artifice of surrounding the ions with counteracting Gaussian charges. For this purpose we now consider the related problem of the unit point charge p interacting solely with the lattice of Gaussian charges. To solve this problem, we make use of *Poisson's equation*, a classic electrostatic equation relating the potential, $\phi_2(\vec{r})$, at a point $\vec{r}$ in a charge distribution to the charge density density, $\rho(\vec{r})$, at that point:

$$\nabla^2\phi_2(\vec{r}) = \left(\frac{\partial^2}{\partial x^2}+\frac{\partial^2}{\partial y^2}+\frac{\partial^2}{\partial z^2}\right)\phi_2(\vec{r}) = -\frac{\rho(\vec{r})}{\epsilon_0}. \tag{6.30}$$

The charge density at any point in the Gaussian lattice is

$$\rho(\vec{r}) = \sum_{\vec{L}=\vec{0}}^{\vec{\infty}}\sum_{j=1}^{N}\rho_j(\vec{r}-\vec{r}_j+\vec{L}). \tag{6.31}$$

A common way of solving a partial differential equation like (6.30) in which the system is periodic in three dimensions, is to resort to a 3D *Fourier Expansion*. Because of the underlying periodicity, both the potential and charge density can be written in periodic form as a Fourier series:

$$\begin{aligned}\phi_2(\vec{r}) &= \sum_{\vec{k}=\vec{0}}^{\vec{\infty}}\phi_{\vec{k}}\exp(i\vec{k}\cdot\vec{r}), \qquad \text{(a)}\\ \rho(\vec{r}) &= \sum_{\vec{k}=\vec{0}}^{\vec{\infty}}\rho_{\vec{k}}\exp(i\vec{k}\cdot\vec{r}), \qquad \text{(b)}\end{aligned} \tag{6.32}$$

in which $\rho_{\vec{k}}$ and $\phi_{\vec{k}}$ are *Fourier coefficients* and $\exp(i\vec{k}\cdot\vec{r})$ is a *complex exponential function*. The $\vec{k}$ vectors appearing here are those described in equations (6.20) to (6.22). The coefficients $\rho_{\vec{k}}$ can be obtained analytically, starting from (6.31), but $\phi_{\vec{k}}$ must be obtained by solving Poisson's equation as we will now show. Inserting the identities (6.32) into Poisson's equation (6.30) leads to the Fourier form

$$\sum_{\vec{k}=\vec{0}}^{\vec{\infty}} k^2 \phi_{\vec{k}} \exp(i\vec{k}\cdot\vec{r}) = \frac{1}{\epsilon_0} \sum_{\vec{k}=\vec{0}}^{\vec{\infty}} \rho_{\vec{k}} \exp(i\vec{k}\cdot\vec{r}). \tag{6.33}$$

By comparing Fourier components it follows that

$$k^2 \phi_{\vec{k}} = \frac{1}{\epsilon_0} \rho_{\vec{k}}, \tag{6.34}$$

or

$$\phi_{\vec{k}} = \frac{\rho_{\vec{k}}}{\epsilon_0 k^2}, \quad \textit{provided} \quad \vec{k} \neq \vec{0} \tag{6.35}$$

Clearly in (6.35) the condition $\vec{k} \neq \vec{0}$ is mathematically essential, but it does suggest we are missing something. It is traditional to neglect the $\vec{k} = \vec{0}$ term and assume it is harmless, since $\vec{k} = \vec{0}$ in reciprocal space is equivalent to $\vec{r} = \vec{\infty}$ in real space and we expect to be able to identify any such contribution that may arise on physical grounds. For now we simply make a note to watch out for unexpected effects.

From (6.35) we can easily obtain $\phi_{\vec{k}}$ from $\rho_{\vec{k}}$ and through (6.32)(a) obtain the required potential $\phi_2(\vec{r})$ we seek. It remains to specify the coefficients $\rho_{\vec{k}}$. Multiplying both sides of equation (6.32)(b) by $\exp(-i\vec{k}'\cdot\vec{r})$ gives

$$\rho(\vec{r}) \exp(-i\vec{k}'\cdot\vec{r}) = \sum_{\vec{k}=\vec{0}}^{\vec{\infty}} \rho_{\vec{k}} \exp(i\vec{k}\cdot\vec{r}) \exp(-i\vec{k}'\cdot\vec{r}). \tag{6.36}$$

Then integrating over $\vec{r}$ in real space leads to the result that

$$\rho_{\vec{k}} = \frac{1}{V} \int^{V} \rho(\vec{r}) \exp(-i\vec{k}\cdot\vec{r})\, dv, \tag{6.37}$$

in which the integral takes place over the volume, V, of the periodic simulation cell in accordance with the standard Fourier treatment. This result arises directly from the orthogonality of the Fourier terms i.e.

$$\int^{V} \exp(i\vec{k}\cdot\vec{r}) \exp(-i\vec{k}'\cdot\vec{r})\, dv = \begin{Bmatrix} 0, & \textit{if } \vec{k} \neq \vec{k}' \\ V, & \textit{if } \vec{k} = \vec{k}' \end{Bmatrix}. \tag{6.38}$$

Substituting (6.31) into (6.37) leads to

$$\rho_{\vec{k}}=\frac{1}{V}\sum_{\vec{L}=\vec{0}}^{\vec{\infty}}\sum_{j=1}^{N}\int^{V}\rho_j(\vec{r}-\vec{r}_j+\vec{L})\exp(-i\vec{k}\cdot\vec{r})\,dv. \tag{6.39}$$

This expression can be simplified if it is realised that integrating every Gaussian from every periodic cell over the central volume V, is the same as integrating every Gaussian in the central cell to infinity, so (6.39) becomes:

$$\rho_{\vec{k}}=\frac{1}{V}\sum_{j=1}^{N}\int_{-\infty}^{\infty}\rho_j(\vec{r}-\vec{r}_j)\exp(-i\vec{k}\cdot\vec{r})\,dv. \tag{6.40}$$

Substituting the form of the Gaussian (6.27) into (6.40) yields a standard *Fourier transform* of a Gaussian. The result is

$$\rho_{\vec{k}}=\frac{1}{V}\sum_{j=1}^{N}q_j\exp(-k^2/4\alpha^2)\exp(-i\vec{k}\cdot\vec{r}_j). \tag{6.41}$$

Combining this result with (6.35) allows us to write (6.32)(a) as

$$\phi_2(\vec{r})=-\frac{1}{V\epsilon_0}\sum_{\vec{k}\neq\vec{0}}^{\vec{\infty}}\frac{\exp(-k^2/4\alpha^2)}{k^2}\sum_{j=1}^{N}q_j\exp(-i\vec{k}\cdot\{\vec{r}-\vec{r}_j\}). \tag{6.42}$$

Note that condition $\vec{k}\neq\vec{0}$ applies here, since we are neglecting the $\vec{k}=\vec{0}$ term. This result is the potential energy at the point p due to the lattice of Gaussians that was superimposed on the point charges of the original lattice. It is obviously the basis for the reciprocal space part of the Ewald sum.

We may now subtract $\phi_2(\vec{r})$ in (6.42) from $\phi_1(\vec{r})$ in (6.29) to obtain the true potential, $\phi(\vec{r})$, at point p:

$$\begin{aligned}\phi(\vec{r})=&\frac{1}{4\pi\epsilon_0}\sum_{\vec{L}=\vec{0}}^{\vec{\infty}}\sum_{j=1}^{N}\frac{q_j}{|\vec{r}-\vec{r}_j+\vec{L}|}erfc(\alpha|\vec{r}-\vec{r}_j+\vec{L}|)+\\&\frac{1}{V\epsilon_0}\sum_{\vec{k}\neq\vec{0}}^{\vec{\infty}}\frac{\exp(-k^2/4\alpha^2)}{k^2}\sum_{j=1}^{N}q_j\exp(-i\vec{k}\cdot\{\vec{r}-\vec{r}_j\}).\end{aligned} \tag{6.43}$$

Having obtained the potential at an arbitrary point in the lattice, we now adapt this to the circumstance in which the point of interest is one of the ions in the lattice, say ion with index n. We can attempt this by setting $\vec{r}=\vec{r}_n$ in (6.43), which amounts to saying the probe charge now has the location $\vec{r}_n$ and a charge of q_n. However, this leads to a superposition of the probe charge on top of the ion already there, so additional modifications are necessary. Recall that in real space the site at $\vec{r}_n$ is occupied by a point charge q_n and a Gaussian $\rho(\vec{r}-\vec{r}_n)$ with charge $-q_n$. If we simply remove both and impose the condition $j\neq n$ when $\vec{L}=\vec{0}$ in the sum over ions, that is sufficient to correct the real space sum. In reciprocal space the site is occupied by a Gaussian with charge q_n and it is tempting to remove this also. Unfortunately doing this has a dramatic impact on the efficiency of the reciprocal space sum calculation. (In short, it is difficult to extract the required terms from the structure

factor, as any attempt will prove.) Fortunately, there is a simple alternative: leave the Gaussian in place and subtract the interaction between the Gaussian and the point charge from the final result. This is the origin of the self interaction correction mentioned above.

The interaction between the point charge q_n and a Gaussian of the same charge is given by

$$\begin{aligned}\phi^{corr} &= \frac{q_n^2}{4\pi\epsilon_0}\left(\frac{\alpha^2}{\pi}\right)^{3/2}\int_{-\infty}^{\infty}\frac{\exp(-\alpha^2 r^2)}{r}dv \\ &= \frac{q_n^2}{\epsilon_0}\left(\frac{\alpha^2}{\pi}\right)^{3/2}\int_0^{\infty}\exp(-\alpha^2 r^2)\,r\,dr = \frac{\alpha q_n^2}{2\pi^{3/2}\epsilon_0}\end{aligned} \tag{6.44}$$

equation (6.43) therefore becomes

$$\begin{aligned}\phi_n(\vec{r}_n) = &\frac{q_n}{4\pi\epsilon_0}\sum_{\vec{L}=\vec{0}}^{\vec{\infty}}\sum_{j=1}^{N}{}' \frac{q_j}{|\vec{r}_n-\vec{r}_j+\vec{L}|}erfc(\alpha|\vec{r}_n-\vec{r}_j+\vec{L}|)+ \\ &\frac{q_n}{V\epsilon_0}\sum_{\vec{k}=\vec{0}}^{\vec{\infty}}\frac{\exp(-k^2/4\alpha^2)}{k^2}\sum_{j=1}^{N}q_j\exp(-i\vec{k}\cdot\{\vec{r}_n-\vec{r}_j\})-\frac{\alpha q_n^2}{2\pi^{3/2}\epsilon_0},\end{aligned} \tag{6.45}$$

where $\phi_n(\vec{r}_n)$ represents the potential energy of the $n'th$ ion. The Ewald Coulombic configuration energy for all N ions is therefore given by

$$\Phi^E\left(\{\vec{r}_i\}\right) = \frac{1}{2}\sum_{n=1}^{n}\phi_n(\vec{r}_n), \tag{6.46}$$

which, on insertion of (6.45) leaves the final form for the real space term as

$$\frac{1}{2}\frac{1}{4\pi\epsilon_0}\sum_{\vec{L}=\vec{0}}^{\vec{\infty}}\sum_{j=1}^{N}\sum_{n=1}^{N}{}' \frac{q_j q_n erfc(\alpha r_{jn}^L)}{r_{jn}^L}, \tag{6.47}$$

the reciprocal space term as

$$\frac{1}{2\epsilon_0 V}\sum_{\vec{k}\neq\vec{0}}^{\vec{\infty}}\frac{\exp(-k^2/4\alpha^2)}{k^2}\left(\sum_{n=1}^{N}q_n\exp(-i\vec{k}\cdot\vec{r}_n)\right)\left(\sum_{j=1}^{N}q_j\exp(i\vec{k}\cdot\vec{r}_j)\right) \tag{6.48}$$

and the self interaction correction term as

$$-\frac{\alpha}{4\pi^{3/2}\epsilon_0}\sum_{j=1}^{n}q_j^2. \tag{6.49}$$

These are identical to the terms appearing in (6.18).

The force acting on the $m'th$ ion is obtained by the usual differentiation:

$$\vec{f}_m = -\frac{\partial}{\partial \vec{r}_m}\Phi^E(\{\vec{r}_j\}),$$
$$= \frac{q_m}{\epsilon_0 V}\sum_{\vec{k}\neq\vec{0}}^{\vec{\infty}} \frac{\exp(-k^2/4\alpha^2)}{k^2}\Im\left\{\exp(i\vec{k}\cdot\vec{r}_m)\left(\sum_{j=1}^{N} q_j \exp(-i\vec{k}\cdot\vec{r}_j)\right)\right\}\vec{k} + \qquad (6.50)$$
$$\frac{q_m}{4\pi\epsilon_0}\sum_{\vec{L}=\vec{0}}^{\vec{\infty}}\sum_{j=1}^{N}{}' \; q_j\left\{\frac{erfc(\alpha r_{jm}^L)}{r_{jm}^L} + \frac{2\alpha}{\sqrt{\pi}}\exp(-\alpha^2(r_{jm}^L)^2)\right\}\frac{\vec{r}_{jm}^L}{(r_{jm}^L)^2},$$

in which $\Im\{z\}$ represents the *imaginary* part of the complex argument z. We have also used the fact that

$$\frac{d}{dr}erfc(\alpha r) = -\frac{2\alpha}{\sqrt{\pi}}\exp(-\alpha^2 r^2). \qquad (6.51)$$

Note that the force obtained from (6.50) is *real* as required.

6.4.3 Using the Ewald Sum

Ideally it is desirable to find a value of α that minimises the overall cost for the combined real and reciprocal space calculations. In molecular simulation the simulation cell is usually constructed from many replicas of the crystallographic unit cell, so it is expected that convergence of the real space part will occur within the dimensions of the simulation cell. Therefore a sum over cell images (represented by the $\vec{L}$ vectors) is not called for. A common practice is to choose α so that the real space sum converges within the same cut-off range as the van der Waals potentials in the system and all required images of ions are generated using the minimum image convention. The reciprocal space sum, which uses the same value of α, is then set to a cut-off range large enough to complete the calculation of the full Ewald sum to the required accuracy. The advantage of this approach is that it easily accommodates the Ewald sum within the periodic boundary and minimum image schemes of most simulation programs. This alone does not guarantee optimal performance but it is easy to proceed from there to a more optimal scheme by trial and error.

If we assume that a cut-off at range r_{cut} is applied in real space and that q_{max} is the largest absolute charge possessed by any ion in the system, then the largest potential term at the cut-off will have magnitude

$$\delta_1 = \frac{q_{max}^2}{4\pi\epsilon_0}\frac{erfc(\alpha r_{cut})}{r_{cut}} \qquad (6.52)$$

In principle one can decide what value of δ_1 is accurate enough for our purposes and solve (6.52) as a transcendental equation to obtain the working value of α. However, since the function $erfc(x)$ is bounded by the function $\exp(-x^2)$ over the range $x = 0 \rightarrow \infty$, it is acceptable to work with the approximation

$$\delta_1 \approx \frac{q_{max}^2}{4\pi\epsilon_0}\frac{\exp(-\alpha^2 r_{cut}^2)}{r_{cut}} \tag{6.53}$$

In which case, taking the natural logarithm of (6.53),we can estimate α from

$$\alpha \approx \frac{1}{r_{cut}}\sqrt{-\log\left(\frac{4\pi\epsilon_0 r_{cut}\delta_1}{q_{max}^2}\right)}. \tag{6.54}$$

Since the logarithm is a slowly varying function, we can see that α scales approximately with the reciprocal of the cut-off distance r_{cut}.

To determine the required range in reciprocal space, based on the value of α obtained from real space considerations, we use the formula

$$\delta_2 = \frac{S(\vec{k}_{max})}{2\epsilon_0 V}\frac{\exp(-k_{max}^2/4\alpha^2)}{k_{max}^2}, \tag{6.55}$$

which is based on the reciprocal space vector, $\vec{k}_{max}$, that supplies the first term of the reciprocal space sum (6.48) contributing less than an amount δ_2 to the overall Coulombic energy. The structure factor $S(\vec{k}_{max})$ has a maximum theoretical value of

$$S(\vec{k}_{max}) = \left(\sum_{j=1}^{N}|q_j|\right)^2, \tag{6.56}$$

where $|q_j|$ is the absolute charge on ion j. This can be substituted into (6.55) and the resulting transcendental equation solved by iteration. However, if all that is required is some guarantee of convergence, we can simplify this by setting the k_{max}^2 in the denominator of (6.55) to unity and rearranging to give the equation

$$k_{max}^2 = -4\alpha^2\log\left(\frac{2\epsilon_0 V\delta_2}{S(\vec{k}_{max})}\right) \tag{6.57}$$

in which $S(\vec{k}_{max})$ is given by (6.56). Provided the resulting k_{max}^2 is greater than unity, convergence to δ_2 will be satisfied. If we regard the logarithm here as a constant, we can see that k_{max} scales linearly with α, or from (6.54) with the reciprocal of r_{cut}.

Having obtained k_{max}^2 this can be used to set limits on the sum over $\vec{k}$ in the reciprocal space term in (6.18). This means specifying three integers: μ_{max}, ν_{max} and λ_{max}, representing the largest integers that can be used in equation (6.20) defining the vectors $\vec{k}$. So from (6.20) we can write

$$k_{max}^2 \le \mu_{max}^2\vec{u}\cdot\vec{u}, \qquad k_{max}^2 \le \nu_{max}^2\vec{v}\cdot\vec{v}, \qquad k_{max}^2 \le \lambda_{max}^2\vec{w}\cdot\vec{w}, \tag{6.58}$$

for which vectors $\vec{u}, \vec{v}$ and $\vec{w}$ are defined in (6.21). It should be clear that equations (6.58) require *integer* solutions for μ_{max}, ν_{max} and λ_{max}.

One useful point worth noting is that for every vector, $\vec{k},$ appearing in the reciprocal space sum, there is a corresponding vector, $-\vec{k},$ which makes the *same* energy and force contribution to the overall Ewald sum. The obvious thing to do when computing the Ewald sum is to restrict the reciprocal space sum to one half of reciprocal space and double the contribution of each $\vec{k}$ vector used. At a stroke the cost of the reciprocal space sum is thereby halved. Rules to achieve this can be written as follows.

- In all cases, λ runs from 0 to λ_{max} only.
- When $\lambda=0,$ ν runs from 0 to ν_{max} only.
- When $\lambda=\nu=0,$ μ runs from 1 to μ_{max} only.
- When $\lambda=0$ and $\nu>0,$ μ runs from $-\mu_{max}$ to μ_{max}.
- When $\lambda>0,$ μ runs from $-\mu_{max}$ to μ_{max} and ν from $-\nu_{max}$ to ν_{max}.

6.4.4 Performance Scaling

The Ewald sum is rather more complicated than the simple pair interactions we have encountered previously. So it is worth considering how its performance scales with system size. This can be done in a rather *ad hoc* manner which is nevertheless instructive. First we note that the self interaction correction does not contribute to this exercise, since it is calculated once only in any simulation and therefore makes a negligible contribution to the overall computational cost.

The real space sum in (6.18) obviously scales as order $O(N^2),$ where N is the number of ions in the simulation cell, but it also scales according to the cut-off condition we apply. We will write this as

$$t_1 = A\,N^2 r_{cut}^3, \tag{6.59}$$

where t_1 is the time required to complete the real space sum and A is a constant (to be determined). We have included the factor r_{cut}^3 to reflect the dependence of t_1 on the cut-off sphere surrounding each ion.

The reciprocal space term is linearly dependent on N since, in calculating $S(\vec{k})$ it is necessary to sum over all ions once only and take the complex conjugate to get the second sum. It also scales as $k_{max}^3,$ which effectively represents the volume of reciprocal space summed over. However as we noted in the previous section, k_{max} itself scales approximately as the reciprocal of $r_{cut},$ so we can write

$$t_2 = B\,N / r_{cut}^3, \tag{6.60}$$

where t_2 is the time required to compute the reciprocal space sum and B is another constant (to be determined). The total time to calculate the Ewald sum is therefore

$$t = t_1 + t_2 = A N^2 r_{cut}^3 + B N / r_{cut}^3 . \quad (6.61)$$

This formula shows that, for a given system size, N, the cost of the Ewald sum scales as r_{cut}^3 in real space and as $1/r_{cut}^3$ in reciprocal space. The relative cost of each sum being determined by the constants A and B and the system size N. The constants can be obtained easily, by timing two short simulations of the same ionic system with different set values for r_{cut}. Constants A and B can be extracted from the compared times using (6.61).

Once A and B are known the computing time t can be minimised by optimising (6.61) with respect to r_{cut} as follows.

$$\frac{d\,t}{d\,r_{cut}} = 3\mathrm{A}\, N^2 r_{cut}^2 - 3\, B N / r_{cut}^4 = 0, \quad (6.62)$$

from which we obtain

$$r_{cut} = \{ B / (NA) \}^{1/6} . \quad (6.63)$$

This represents the value required to obtain an optimal Ewald sum. However, if the value obtained this way implies a cut-off sphere with a diameter greater than the width of the simulation cell, we must default to the largest cut-off compatible with the minimum image convention. Substituting (6.63) back into (6.61) leads to the result

$$t = 2\sqrt{AB}\, N^{3/2} . \quad (6.64)$$

So the Ewald sum scales as $O\left(N^{3/2}\right)$ rather than the $O\left(N^2\right)$ of the parent Coulomb sum.

6.4.5 Rigid Molecules and Bonds

The Ewald approach described so far can be applied directly to molecules composed of atoms bearing charges if required. However, there are occasions when this is not ideal. This exception applies mostly to rigid molecules, but it can sometimes apply to flexible molecules and to models involving polarisable atoms, if they are described by the shell model (see section 6.9). The problem is that in these cases the Coulombic interaction between the atoms in the same molecule are not usually part of the model and we need some way to leave them out. This is trivial to do in the real space sum, but in the reciprocal space sum attempts to do so requires de-constructing the reciprocal space sum, which we should not attempt if we want an efficient algorithm.

The simplest thing to do is to calculate the coulombic interactions with the full Ewald sum and the subtract the unwanted Coumbic terms afterwards. In effect we define a new potential – the ionic molecule correction potential, $\phi_c(\{\vec{r}_i\})$, which compensates for the unwanted terms:

$$\phi_c(\{\vec{r}_i\}) = -\frac{1}{4\pi\epsilon_0}\sum_{n=1}^{N_m}\sum_{j=1}^{n_s-1}\sum_{m>j}^{n_s}\frac{q_n^j q_n^m}{r_n^{jm}}, \tag{6.65}$$

In which the sum over n is a sum over the N_m molecules in the system and the sums over j and m are over the n_s atomic sites in each molecule. Charges $q_{n,j}$ and $q_{n,m}$ are the charges on sites j and m on the $n'th$ molecule. The distance $r_{n,jm}$ is the separation between the charges i.e.

$$r_n^{jm} = |\vec{r}_n^m - \vec{r}_n^j|. \tag{6.66}$$

While this is formally correct, there is the danger that r_n^{jm} is a very small quantity in the molecular context and subtraction of the resulting large terms from the overall Coulomb energy may lead to numerical inaccuracy. (Similar considerations apply to the forces.)

We can do better if we first extract the real space terms for the intra-molecular interactions and write them as a separate term:

$$\phi_M(\{\vec{r}_i\}) = \frac{1}{4\pi\epsilon_0}\sum_{n=1}^{N_m}\sum_{j=1}^{n_s-1}\sum_{m>j}^{n_s}\frac{q_n^j q_n^m}{r_n^{jm}}\,erfc(\alpha r_n^{jm}). \tag{6.67}$$

Now, adding the correction $\phi_c(\{\vec{r}_i\})$, to this gives, with the help of (6.24):

$$\begin{aligned}\phi_M'(\{\vec{r}_i\}) &= \frac{1}{4\pi\epsilon_0}\sum_{n=1}^{N_m}\sum_{j=1}^{n_s-1}\sum_{m>j}^{n_s}\frac{q_n^j q_n^m}{r_n^{jm}}\left(erfc(\alpha r_n^{jm})-1\right),\\ &= -\frac{1}{4\pi\epsilon_0}\sum_{n=1}^{N_m}\sum_{j=1}^{n_s-1}\sum_{m>j}^{n_s}\frac{q_n^j q_n^m}{r_n^{jm}}\,erf(\alpha r_n^{jm}).\end{aligned} \tag{6.68}$$

The terms appearing here are now weighted by $erf(\alpha r_n^{jm})$ which is small when the argument αr_n^{jm} is small, so the issue of subtracting large numbers is diminished.

In practice, what all this amounts to is a simple rule: when calculating the real space sum, on encountering terms referring to Coulombic interactions we need to remove, we simply replace $erfc(\alpha r_n^{jm})$ by $-erf(\alpha r_n^{jm})$ and calculate that instead. The final resulting system Coulombic energy will then be correctly compensated for unwanted interactions. The force contribution arising from these pair terms terms take the form

$$\begin{aligned}\vec{f}_n^m &\leftarrow \frac{q_n^m q_n^j}{4\pi\epsilon_0}\left\{\frac{erf(\alpha r_n^{jm})}{r_n^{jm}} - \frac{2\alpha}{\sqrt{\pi}}\exp\left(-\alpha^2(r_n^{jm})^2\right)\right\}\frac{\vec{r}_n^{jm}}{(r_n^{jm})^2},\\ \vec{f}_n^j &\leftarrow -\frac{q_n^m q_n^j}{4\pi\epsilon_0}\left\{\frac{erf(\alpha r_n^{jm})}{r_n^{jm}} - \frac{2\alpha}{\sqrt{\pi}}\exp\left(-\alpha^2(r_n^{jm})^2\right)\right\}\frac{\vec{r}_n^{jm}}{(r_n^{jm})^2}.\end{aligned} \tag{6.69}$$

6.4.6 Systems with a Net Charge

The description of the Ewald sum so far has been concerned with systems that are electrically neutral. Here we address the possibility that the simulation cell has an overall net charge, q^{tot}, where

$$q^{tot} = \sum_{i=1}^{N} q_i . \tag{6.70}$$

We cannot, in principle, apply the Ewald sum to such a system because, from a theoretical viewpoint, the Coulombic energy is infinite (though in fact the sum does return a finite result, with an uncertain meaning). However, we note that the Coulombic energy of a neutral system is finite. So instead we apply the Ewald sum to a system in which the net charge q^{tot} in the cell is screened by an opposite uniform charge density, ρ^{scr}, defined as

$$\rho^{scr} = -q^{tot}/V , \tag{6.71}$$

where V is the volume of the simulation cell. This screening charge ensures that the system is overall charge neutral. We now seek to adapt the Ewald sum to include the effect of this background charge.

It is evident that we need to add to the Ewald sum (6.18) the following additional terms:

1. The interaction between the Gaussian-screened point ions used in the *real space* sum and the background charge density: term (Φ_1^{scr}).
2. The interaction between the lattice of Gaussian charges used in the *reciprocal space* sum and the background charge density: term (Φ_2^{scr}).
3. The self interaction of the background charge density: term (Φ_3^{scr}).

To obtain term 1 we write the interaction between a Gaussian-screened ion, i, and the charge $\rho^{scr} dv$ directly as

$$\begin{aligned} \phi_i^{scr_1} &= \frac{q_i \rho^{scr}}{4\pi\epsilon_0} \int_{-\infty}^{\infty} \frac{erfc(\alpha r)}{r} dv \\ &= \frac{q_i \rho^{scr}}{\epsilon_0} \int_0^{\infty} erfc(\alpha r) r \, dr \end{aligned} \tag{6.72}$$

The second form of the integral is obtained by integrating over the angular coordinates. The integral over r is obtained by parts, using the derivative of $erfc(\alpha r)$ given in equation (6.51). The result is

$$\phi_i^{scr_1} = \frac{q_i \rho^{scr}}{4 \epsilon_0 \alpha^2}. \tag{6.73}$$

The total contribution to the Ewald sum from all ions in the cell is therefore

$$\Phi_1^{scr} = \frac{1}{2} \sum_{i=1}^{N} \phi_i^{scr_1} = -\frac{(q^{tot})^2}{8 \epsilon_0 V \alpha^2}, \tag{6.74}$$

where we have made use of both (6.71) and (6.73). Note that the derivative of this with respect to $\vec{r}_i$ is zero, so the term exerts no forces on the ions.

To calculate the second term we start with the potential at a point $\vec{r}$ due to the screening charge density ρ^{scr}, which we write as $\phi^{scr}(\vec{r})$. Then, in principle this can be used to obtain the Coulombic energy of a Gaussian charge distribution embedded in the screening charge as:

$$\phi_i^{scr_2} = q_i \left(\frac{\alpha^2}{\pi}\right)^{3/2} \int_{-\infty}^{\infty} \phi^{scr}(\vec{r}) \exp(-\alpha^2 r^2) dv. \tag{6.75}$$

The function $\phi^{scr}(\vec{r})$ can be obtained as follows. Consider a unit point charge at a position $\vec{r}$ in the uniform charge density ρ^{scr}. We surround this by a spherical shell of thickness $\delta r'$ and radius r' (with origin at at $\vec{r}$), which contains a charge δq given as

$$\delta q = 4 \pi \rho^{scr} r'^2 \delta r'. \tag{6.76}$$

The contribution of this charge to the potential of the unit charge at $\vec{r}$ is

$$\delta \phi^{scr}(\vec{r}) = \frac{\rho^{scr}}{\epsilon_0} r' \delta r. \tag{6.77}$$

The total potential due to a sphere of uniform density ρ^{scr} and of radius R is therefore

$$\phi^{scr}(\vec{r}) = \frac{\rho^{scr}}{\epsilon_0} \int_0^R r' d r' = \frac{\rho^{scr} R^2}{2 \epsilon_0}. \tag{6.78}$$

We will make the temporary assumption that $R \to \infty$, without R becoming infinite at this stage, so that $\phi^{scr}(\vec{r})$ is a finite constant. Putting the result back into (6.75) gives

$$\phi_i^{scr_2} = \frac{q_i \rho^{scr} R^2}{2 \epsilon_0}. \tag{6.79}$$

The term Φ_2^{scr} can then be given as

$$\Phi_2^{scr} = \frac{1}{2}\sum_{i=1}^{N} \phi_i^{scr_2} = \frac{(q^{tot})^2 R^2}{4\epsilon_0 V}. \tag{6.80}$$

We now calculate Φ_3^{scr} by following the same path as Φ_2^{scr} above, we again proceed via (6.78) and eventually obtain

$$\Phi_3^{scr} = -\frac{1}{2}\frac{(\rho^{scr})^2 R^2}{2\epsilon_0}\int^{V} dv = -\frac{(q^{tot})^2 R^2}{4\epsilon_0 V}. \tag{6.81}$$

Thus we find that terms Φ_2^{scr} and Φ_3^{scr} are equal and opposite and thus cancel one another. This is irrespective of the value of R, so we presume it holds even as R proceeds to infinity, as we require. So we are left with Φ_1^{scr} as the only surviving term. We conclude from this that the Coulombic energy of an ionic system with a uniform neutralising background charge density is given by the Ewald sum (6.18) plus the term Φ_1^{scr} given in (6.74). Obviously, this correction is zero when the system has no net charge.

6.4.7 Systems with a Net Dipole

In our discussion of the Ewald method we have made frequent mention of the neglect of the $\vec{k} = \vec{0}$ term of the reciprocal space sum, a necessary assumption to make a practical scheme. We have also noted that this neglect may have some repercussions with respect to the treatment of the infinitely long range. This is essentially a *boundary condition* issue that we have simply swept under the carpet.
We now address this by considering what happens when the simulation cell possesses a net dipole, which may be a permanent property of the system or one which simply occurs instantaneously as the result of fluctuations in the ionic positions.

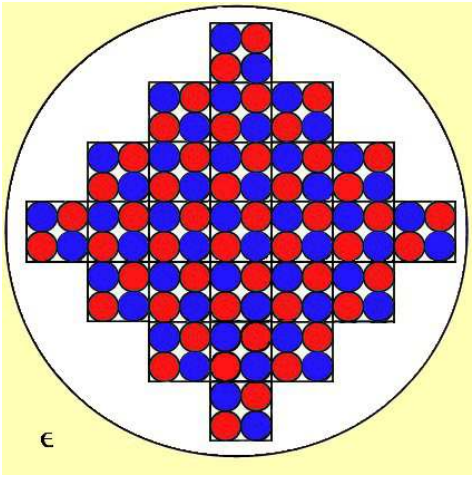

Figure 6.4. A Finite Coulombic System Embedded in a Dielectric Medium (ϵ)

Consider the system presented in figure 6.4, which represents a 3D, near-spherical, ionic cluster composed of many simulation cells. Each cell is packed with the others in a manner resembling the infinite periodic system. The boundary of this system is the interface (defined by a sphere) between the cluster and the dielectric medium in

which the cluster is embedded. It is legitimate to ask what affect the dielectric medium may have on the Coulombic energy of the cluster. For a system in which the periodic cells have no net electric dipole, we expect no effects. But if the periodic cells possess a dipole, then the cluster as a whole is dipolar and, as was seen in the case of the reaction field model of section 6.3 , the dipole is expected to polarise the dielectric medium which, in turn, will generate a reaction field that interacts with the cluster dipole. So in this case an additional energy term is required, beyond the inter-ionic Coulombic energy. It is also clear that it does not matter how big the ionic cluster is or how many periodic cells are required to construct it, the dipole - reaction field interaction exists on all size scales as long as the cluster is embedded in the dielectric medium. It follows that this applies even as the cluster approaches infinite size. Since the Ewald sum specifically does not allow for any boundary effects, this is clearly a contribution that it does not incorporate. This deficiency was first identified by DeLeeuw, Perram and Smith [58].

The correction for this effect unfortunately depends on the shape of the supposed boundary, but if it is assumed that the material sample is spherical then it is easily calculated. Assuming the sample sits in a spherical cavity in the dielectric medium, the material within the cavity has a polarisation $\vec{P}$ defined as

$$\vec{P}=\frac{\vec{\mu}}{V}=\frac{1}{V}\sum_{i=1}^{N} q_i \vec{r}_i, \tag{6.82}$$

where V is the volume of the simulation cell and $\vec{\mu}$ its net dipole moment. The reaction field generated by the polarisation of the external medium, with dielectric constant ϵ, is given by standard electrostatic theory as

$$\vec{E}=-\frac{\vec{P}}{\epsilon_0(2\epsilon+1)}. \tag{6.83}$$

Finally, the interaction of this field with the net dipole of the simulation cell gives the dipole-polarisation correction to the Ewald sum as

$$\Phi^{DP}=-\frac{1}{2}\vec{\mu}\cdot\vec{E}=\frac{1}{2V\,\epsilon_0(2\epsilon+1)}\left(\sum_{i=1}^{N} q_i \vec{r}_i\right)^2. \tag{6.84}$$

For the majority of cases we would assume the ionic system is embedded in a vacuum, in which case $\epsilon=1$ in (6.84).

With this correction we can write the Coulombic energy of an ionic system as

$$\Phi^{C}=\Phi^{E}+\Phi^{DP}. \tag{6.85}$$

Note that, when $\epsilon=\infty$, which corresponds to the system being embedded in metal, then the Ewald and the Coulomb energies are identical. For this reason the standard Ewald sum is said to give the Coulombic energy in a “tinfoil” boundary condition.

The polarisation correction exerts a force of the ions, which is given by

$$\vec{f}_i^{\,DP} = -\frac{q_i}{V\epsilon_0(2\epsilon+1)}\left(\sum_{j=1}^{N} q_j \vec{r}_j\right). \tag{6.86}$$

This force is obviously the response of the ions to the reaction field and it evidently works in opposition to the forces that give rise to the dipole $\vec{\mu}$.

The one last thing to say about the dipole-polarisation correction is that it is not often used! The main reason for not including it is the unknown nature of the long range boundary of real systems.

6.5 The Smoothed Particle-Mesh Ewald

The Smoothed Particle Mesh Ewald sum (SPME) is an alternative form of the Ewald sum devised by Essmann *et al.* [52]. which is more computationally efficient than the form described in the preceding sections. The difference is entirely in the treatment of the reciprocal space sum, which is handled using the Fast Fourier Transform (FFT). Surprisingly the SPME method owes nothing to the earlier Particle-Particle Particle-Mesh (P3M) method of Hockney and Eastwood [16] which also exploits the FFT and reputedly has a superior control of truncation errors. Nevertheless, it is SPME that dominates the world of applications, presumably because it is the simpler method. In the following brief description we will focus on the reciprocal space component, since the real space treatment does not differ from the conventional Ewald sum.

We begin by extracting the reciprocal space term from the Ewald formula (6.18) thus

$$\Phi^F(\{\vec{r}_i\}) = \frac{1}{2\epsilon_0 V}\sum_{\vec{k}\neq\vec{0}}^{\vec{\infty}} \frac{\exp(-k^2/4\alpha^2)}{k^2}\left|\sum_{j=1}^{N} q_j \exp(i\vec{k}\cdot\vec{r}_j)\right|^2, \tag{6.87}$$

where Φ^F symbolises the Fourier (i.e. reciprocal space) part of the sum. The objective of the SPME method is to replace this with an alternative FFT-based form:

$$\Phi^F(\{\vec{r}_i\}) = \frac{1}{2\epsilon_0 V}\sum_{k_1,k_2,k_3} \check{\boldsymbol{G}}(k_1,k_2,k_3)\boldsymbol{Q}(k_1,k_2,k_3), \tag{6.88}$$

in which both $\boldsymbol{G}$ and $\boldsymbol{Q}$ are $3\mathrm{D}$ arrays resulting from a mapping on to a $3\mathrm{D}$ regular grid and $\check{\boldsymbol{G}}$ is the 3D discrete Fourier transform of $\boldsymbol{G}$. We will describe the construction of these arrays presently, but first we will describe the process by which (6.87) is discretised onto a 3D grid.

The central modification required is to the structure factor term appearing in (6.87) and defined in (6.23). This, as was shown in the derivation of the Ewald sum (see section 6.4.2), is based on a Fourier series expansion of the Coulombic potential. It is in principle amenable to reformulation as a 3D discrete Fourier transform, otherwise known as the FFT. However, the location of each ion, represented by the vector $\vec{r}_j$, is not, in general, on the regular points of the 3D grid required by the FFT.

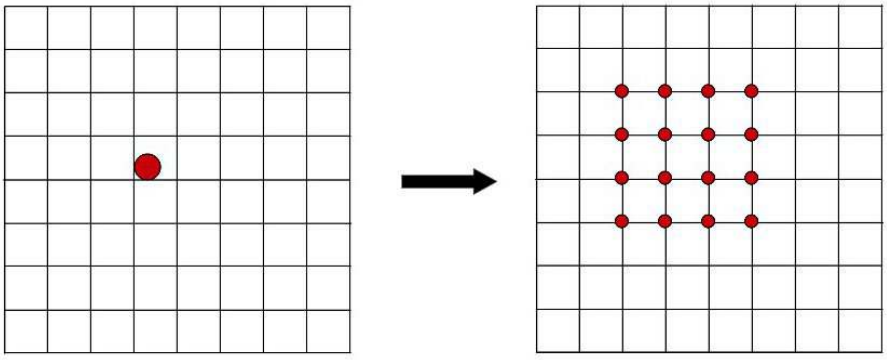

Figure 6.5. Charge Allocation in the SPME Scheme

To handle this, the SPME method employs a charge allocation scheme to disperse the charge of each ion onto neighbouring sites on the regular 3D grid (as shown in figure 6.5), such that the original charge can be obtained by interpolation of the distributed charges. The scheme for doing this employs the mathematics of (complex) *Cardinal B-Splines*, $M_n(u)$, which (in 1D) are defined for argument u as:

$$M_n(u)=\frac{1}{(n-1)!}\sum_{k=0}^{n}(-1)^k\frac{n!}{k!(n-k)!}Max(u-k,0)^{(n-1)}, \tag{6.89}$$

where $n\approx 8-12$ is the *order* of the B-Spline. In practice only 1D B-Splines are needed. Note that whenever $u>k$ the sum is effectively truncated. Low orders of n, such as $n=0$ and $n=1$ can be calculated directly, but higher orders can be generated using the recursion formula

$$M_n(u)=\frac{u}{n-1}M_{n-1}(u)+\frac{n-u}{n-1}M_{n-1}(u-1). \tag{6.90}$$

The Fourier terms from the structure factor can be expressed using B-Splines as follows

$$\exp(2\pi i u_j k/K)=b(k)\sum_{p=-\infty}^{\infty}M_n(u_j-p)\exp(2\pi i k\, p/K), \tag{6.91}$$

in which u_j is the fractional coordinate, s_j, of ion j scaled by an integer factor K, which is the number of grid points in the same direction as coordinate u_j. Variables p and k are integers. The left hand side of (6.91) thus represents an expansion of the Fourier term on a regular grid. Note that the formally infinite sum over p, is in fact finite as a result of the dependence of the expansion on K. The factor $b(k)$ has the form

$$b(k)=\exp(2\pi i(n-1)k/K)\left[\sum_{p=0}^{n-2}M_n(p+1)\exp(2\pi i k\, p/K)\right]^{-1}. \tag{6.92}$$

These formulae allow the calculation of the structure factor in a form appropriate for a grid:

$$S(\vec{k}) = \left| b_1(k_1) b_2(k_2) b_3(k_3) \check{\boldsymbol{Q}}(k_1, k_2, k_3) \right|^2 \tag{6.93}$$

The array $\check{\boldsymbol{Q}}$ appearing in (6.93) is discrete 3D Fourier transform of the so-called *charge array*, $\boldsymbol{Q}$, which is defined as

$$\boldsymbol{Q}(k_1, k_2, k_3) = \sum_{j=1}^{N} q_j \sum_{n_1, n_2, n_3} M_n(u_{1j} - k_1 - n_1 K_1) M_n(u_{2j} - k_2 - n_2 K_2) M_n(u_{3j} - k_3 - n_3 K_3), \tag{6.94}$$

where k_1, k_2 and k_3 are integers and the sums over n_1, n_2 and n_3 gather contributions over periodic cell images. As a result of this procedure we can now write (6.87) as

$$\Phi^F(\{\vec{r}_i\}) = \frac{1}{2\epsilon_0 V} \sum_{\vec{k} \neq \vec{0}}^{\vec{k}_{max}} \frac{\exp(-k^2/4\alpha^2)}{k^2} \left| b_1(k_1) b_2(k_2) b_3(k_3) \check{\boldsymbol{Q}}(k_1, k_2, k_3) \right|^2, \tag{6.95}$$

where $\vec{k}_{max}$ is the largest k-vector compatible with the grid size. We now construct a 3D array, $\boldsymbol{G}$, which is defined as

$$\boldsymbol{G}(k_1, k_2, k_3) = \frac{\exp(-k^2/4\alpha^2)}{k^2} \boldsymbol{B}(k_1, k_2, k_3) \check{\boldsymbol{Q}}^*(k_1, k_2, k_3) \tag{6.96}$$

in which k_1, k_2 and k_3 are again integers, with $k^2 = k_1^2 + k_2^2 + k_3^2$. Array $\check{\boldsymbol{Q}}^*$ is the *complex conjugate* of the discrete Fourier transform of $\boldsymbol{Q}$ and $\boldsymbol{B}$ is a 3D array defined as

$$\boldsymbol{B}(k_1, k_2, k_3) = |b_1(k_1)|^2 |b_2(k_2)|^2 |b_3(k_3)|^2. \tag{6.97}$$

So (6.95) now becomes

$$\Phi^F(\{\vec{r}_i\}) = \frac{1}{2\epsilon_0 V} \sum_{\vec{k} \neq \vec{0}}^{\vec{\infty}} \boldsymbol{G}(k_1, k_2, k_3) \check{\boldsymbol{Q}}(k_1, k_2, k_3), \tag{6.98}$$

which, by a property of Fourier transforms (as shown in the appendix to reference [52]) is equivalent to (6.88).

The force on an ion j arising from Φ^E is formally obtained from

$$\vec{f}_j = -\frac{\partial}{\partial \vec{r}_j} \Phi(\{\vec{r}_j\}) = -\frac{1}{\epsilon_0 V} \sum_{k_1, k_2, k_3} \check{\boldsymbol{G}}(k_1, k_2, k_3) \frac{\partial}{\partial \vec{r}_j} \boldsymbol{Q}(k_1, k_2, k_3), \tag{6.99}$$

for which the following recursion formula is useful

$$\frac{d}{du} M_n(u) = M_{n-1}(u) - M_{n-1}(u-1). \tag{6.100}$$

Despite any impression that may be gained from appearances, these formulae are easy to manage and quick to compute. Furthermore highly optimised 3D FFT

routines are widely available, as an internet search will demonstrate. This makes for a very efficient algorithm. There is also a very efficient distributed $3D$ FFT available [59], from which an SPME scheme for parallel computers has been produced [53] (see chapter 9).

Lastly, it is interesting to note that the original scheme described by Essmann is also applicable to van der Waals potentials as well as to the Coulombic potential.

6.6 The Ewald Sum and Ionic Surfaces

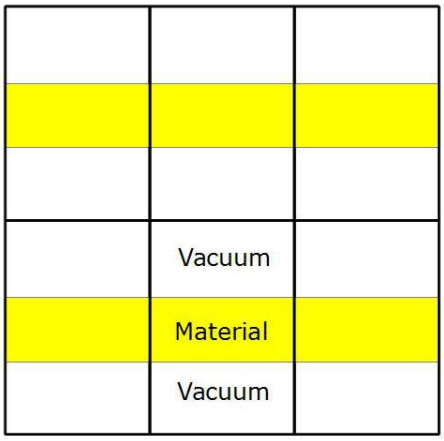

Figure 6.6: The Slab Model for Ionic Surfaces

Ionic surface are often modelled in studies of surface diffusion and molecular adsorption and the like, which are often part of a more general study of heterogeneous catalysis. Here, as much as anywhere, there is a demand for accurate treatment of electrostatic interactions. Unfortunately, the Ewald sum, as we have described it so far, is not suitable for such systems for the obvious reason that it is designed for systems with 3D periodicity, while surfaces are periodic in 2D only. This can be compensated for, to some extent, by using the "slab" model (figure 6.6), which entails constructing a periodic cell containing a vacuum gap which, on periodic repetition generates parallel layers, or slabs, of material featuring internal surfaces. It is hoped that both the slab of material and the vacuum gap are large enough for the internal surfaces to represent the free material surface accurately. However, this needs to be confirmed in each particular case.

Alternatively, one can devise a 2D periodic version of the Ewald sum and several authors have done so (e.g. Parry [60] and Heyes [61]) but we will focus on an elegant form due to Hautman and Klein [62]. The method starts with the simple triangle shown in figure 6.7, where s represents a distance between two ions in the XY plane of the surface, z is a distance perpendicular to the surface plane and r is the distance between the ions in 3D space. We will always assume that $z<s$, or, in terms of the physical dimensions of the system in 3D, symbolized by the ranges: L_x, L_y, L_z, we require that $L_z<L_x, L_y$.

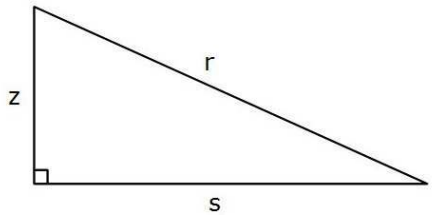

Figure 6.7: The Basic Variables for the Hautman-Klein-Ewald Method.

For a right angled triangle we may write

$$r = s\left(1 + z^2/s^2\right)^{1/2} \quad \text{and hence} \quad \frac{1}{r} = \frac{1}{s}\frac{1}{\left(1 + z^2/s^2\right)^{1/2}} \tag{6.101}$$

Expanding the denominator in (6.101) in a Maclaurin series gives

$$\frac{1}{r} = \sum_{n=0}^{\infty} a_n \frac{z^{2n}}{s^{2n+1}} \quad \text{with} \quad a_n = (-1)^n \frac{(2n)!}{2^{2n}(n!)^2}. \tag{6.102}$$

Formula (6.102) is an expansion of $1/r$ which converges quickly in a finite range provided z is small in relation to s. In the spirit of the Ewald sum, Hautman and Klein then defined $1/r$ in the form

$$\frac{1}{r} = \left(\frac{1}{r} - \sum_{n=0}^{\infty} a_n z^{2n} \frac{h_n(s;\alpha)}{s^{2n+1}}\right) + \left(\sum_{n=0}^{\infty} a_n z^{2n} \frac{h_n(s;\alpha)}{s^{2n+1}}\right), \tag{6.103}$$

in which $h_n(s;\alpha)$ is a *convergence function* of order n (see below) and the parameter α plays the same role as the convergence parameter in the Ewald sum. The first bracketed term on the right of (6.103) is a damped $1/r$ function which becomes the *real space* part of the 2D periodic Ewald sum and the second bracketed term, the compensating *reciprocal space* part. Following the procedures described by Hautman and Klein [62], we may write the 2D periodic configuration energy (for a neutral system) as

$$\begin{aligned}\Phi^{HKE}\left(\{\vec{r}_i\}\right) = {} & \frac{1}{4\epsilon_0 A}\sum_{n=0}^{n_{max}} a_n \sum_{\vec{g}\neq\vec{0}} f_n(g;\alpha)\, g^{2n-1} \sum_{m=0}^{2n} (-1)^m C_m^{2n} Z_m(\vec{g})\, Z_{2n-m}^*(\vec{g}) + \\ & \frac{1}{8\pi\epsilon_0}{\sum_{j,k}}' q_j q_k \sum_{\vec{L}} \left(\frac{1}{r_{\vec{L}jk}} - \sum_{n=0}^{n_{max}} a_n z_{jk}^{2n} \frac{h_n(s_{\vec{L}jk};\alpha)}{s_{\vec{L}jk}^{2n+1}}\right) + \\ & \frac{1}{8\pi\epsilon_0}\sum_{j=1}^{N} q_j^2 \sum_{L>0} \frac{\left(1 - h_0(L;\alpha)\right)}{L} - \frac{\alpha}{\pi^{3/2}\epsilon_0}\sum_{j=1}^{N} q_j^2,\end{aligned} \tag{6.104}$$

where C_m^{2n} are binomial coefficients and

$$Z_m(\vec{g}) = \sum_{j=1}^{N} q_j z_j^m \exp\left(i\,\vec{g}\cdot\vec{s}_j\right). \tag{6.105}$$

Variable A is the area of the system in the XY plane. Distance s_{jk} is the inter-ionic distance between ions j and k *in the XY plane*. $\vec{L}$ is a vector of the 2D periodic lattice, which can be written as

$$\vec{L}=n_1\vec{a}+n_2\vec{b}, \tag{6.106}$$

where n_1 and n_2 are integers and $\vec{a}$ and $\vec{b}$ are the lattice basis vectors in the XY plane. Likewise $\vec{g}$ is a vector of the 2D reciprocal lattice, such that

$$\vec{g}=g_1\vec{u}+g_2\vec{v}, \tag{6.107}$$

where g_1 and g_2 are again integers and $\vec{u}$ and $\vec{v}$ are the reciprocal lattice basis vectors, which are related to vectors $\vec{a}$ and $\vec{b}$ of (6.106) by

$$\vec{u}=\frac{2\pi}{(a_x b_y-a_y b_x)}\begin{bmatrix} b_y \\ -b_x \end{bmatrix} \quad \text{and} \quad \vec{v}=\frac{2\pi}{(a_x b_y-a_y b_x)}\begin{bmatrix} -a_y \\ a_x \end{bmatrix} \tag{6.108}$$

The form given in (6.104) resembles the standard Ewald sum (6.18) and the reciprocal space, real space and self interaction terms can be readily identified. There is apparently an additional term, the third on the right of (6.104), which concerns the interaction of ions with their own images in neighbouring periodic cells. (No such interactions arise in the $\vec{L}=\vec{0}$ cell, since these are forbidden self interactions.) The term is needed because $j=k$ is forbidden in the real space term (second on right). It is therefore a necessary part of the real space sum. Incidentally, note that since the system is *not periodic* in the z-direction, the simulation cell is not replicated in that direction and image ions therefore always have the same z-coordinate i.e. $z_{jk}=0.$ This helps to explain the form of the third term.

The functions $h_n(s,\alpha)$ and $f_n(g;\alpha)$ are the convergence functions corresponding to the Gaussian functions of the standard 3D Ewald method and in fact $f_n(g;\alpha)$ is a Fourier transform of $h_n(s,\alpha).$ The sums over n in (6.104) stop at some order n_{max} which also limits the order of $h_n(s,\alpha).$ Usually $n\leq 2.$ Explicitly we can write

$$h_0(s;\alpha)=erf(\alpha s) \quad \text{and} \quad \frac{h_n(s;\alpha)}{s^{2n+1}}=\frac{1}{a_n(2n)!}\nabla^2\left(\frac{h_0(s,\alpha)}{s}\right), \tag{6.109}$$

with

$$f_0(g;\alpha)=erfc(g/2\alpha) \quad \text{and} \quad f_n(g;\alpha)=\frac{1}{a_n(2n)!}f_0(g;\alpha). \tag{6.110}$$

In practice the functions f_n are easy to obtain, but the functions h_n are relatively tedious by comparison, though a great number of them is not required.

The force on an ion is obtained by the usual differentiation:

$$
\begin{aligned}
f_k^{\beta} = -\frac{\partial}{\partial r_k^{\beta}} \Phi^{HKE}\left(\{\vec{r}_i\}\right) = -\frac{1}{4\epsilon_0 A} \sum_{n=0}^{n_{max}} a_n \sum_{\vec{g}\neq\vec{0}} f_n(g;\alpha)\, g^{2n-1} \\
\sum_{m=0}^{2n} (-1)^m C_m^{2n} \left(Z_m(\vec{g}) \frac{\partial}{\partial r_k^{\beta}} Z_{2n-m}^*(\vec{g}) + Z_{2n-m}^*(\vec{g}) \frac{\partial}{\partial r_k^{\beta}} Z_m(\vec{g}) \right) + \\
\frac{q_k}{4\pi\epsilon_0} \sum_{j\neq k}^{N} q_j \sum_{\vec{L}} \left(\frac{r_{\vec{L}\,jk}^{\beta}}{r_{\vec{L}\,jk}^3} + \sum_{n=0}^{n_{max}} a_n \frac{\partial}{\partial r_k^{\beta}} \left\{ z_{jk}^{2n} \frac{h_n(s_{\vec{L}\,jk};\alpha)}{s_{\vec{L}\,jk}^{2n+1}} \right\} \right),
\end{aligned}
\tag{6.111}
$$

where r_k^{β} is one of $x_k,\ y_k,\ z_k$ and (noting that x_k and y_k behave similarly)

$$
\begin{aligned}
\frac{\partial Z_m(\vec{g})}{\partial x_k} &= i g_x q_k z_k^m \exp\left(i\vec{g}\cdot\vec{s}_k\right), \\
\frac{\partial Z_m(\vec{g})}{\partial z_k} &= m q_k z_k^{m-1} \exp\left(i\vec{g}\cdot\vec{s}_k\right),
\end{aligned}
\tag{6.112}
$$

and

$$
\begin{aligned}
\frac{\partial}{\partial x_k}\left(z_{jk}^{2n} \frac{h_n(s_{\vec{L}\,jk};\alpha)}{s_{\vec{L}\,jk}^{2n+1}} \right) &= x_k \frac{z_{jk}^{2n}}{s_{\vec{L}\,jk}} \frac{\partial}{\partial s_{\vec{L}\,jk}} \left(\frac{h_n(s_{\vec{L}\,jk};\alpha)}{s_{\vec{L}\,jk}^{2n+1}} \right), \\
\frac{\partial}{\partial z_k}\left(z_{jk}^{2n} \frac{h_n(s_{\vec{L}\,jk};\alpha)}{s_{\vec{L}\,jk}^{2n+1}} \right) &= 2n\, z_{jk}^{2n-1} \frac{h_n(s_{\vec{L}\,jk};\alpha)}{s_{\vec{L}\,jk}^{2n+1}}.
\end{aligned}
\tag{6.113}
$$

The Hautman-Klein-Ewald method is simple to use and, on the face of it, a neatly constructed algorithm. However, the condition $L_z < L_x,\ L_y$ turns out to be quite severe, perhaps more like $L_z \ll L_x,\ L_y,$ to ensure accuracy in the direction perpendicular to the surface. In turn this implies a need for large surfaces which means bigger systems than perhaps is first contemplated. Running the sums over $\vec{L}$ to at least the nearest neighbour cells is recommended, but note this makes the calculation of the real space energy appear considerably more expensive than the 3D Ewald sum, where normally only a single minimum image is included.

6.7 The Ewald Sum and Point Multipoles

In some systems the electrostatic interaction between molecules is best described by point multipoles rather than point charges. These can be regarded as a superposition of point charges, dipoles, quadrupoles and so on. Such a representation of the charge distribution in a molecule is hard to obtain experimentally, at least beyond the quadrupole, but are relatively simple to compute by quantum chemistry methods. Multipolar interactions have differing effective ranges. For example dipole-dipole interactions diminish as $1/r^3$ and dipole-quadrupole interactions as $1/r^4$ (see table 6.1). Some of these interactions are considered to be long ranged and some are not. A reasonable criterion for deciding the issue is to regard interactions decaying faster than $1/r^6$ to be short ranged, since this is the distance dependence of the van der Waals dispersion force, which is usually considered to be short ranged.

	C	D	Q	O	H
C	r^{-1}	r^{-2}	r^{-3}	r^{-4}	r^{-5}
D	r^{-2}	r^{-3}	r^{-4}	r^{-5}	r^{-6}
Q	r^{-3}	r^{-4}	r^{-5}	r^{-6}	r^{-7}
O	r^{-4}	r^{-5}	r^{-6}	r^{-7}	r^{-8}
H	r^{-5}	r^{-6}	r^{-7}	r^{-8}	r^{-9}

Table 6.1. Distance Dependence of Multipolar Terms (C=charge, D=dipole, Q=quadrupole, O=octupole, H=hexadecapole.)

Wherever the boundary between long and short ranged interactions is drawn it is clear that some method is required to handle the long ranged interactions properly. Fortunately the Ewald sum can be adapted to this purpose. We will do this for multipoles defined up to the quadrupole, which is enough to be useful. Extension beyond this is possible, but the number of terms rapidly becomes unmanageable. To get beyond the quadrupole it is best to use a spherical harmonic representation of the multipoles [63], [64], though this is not an easy option.

We start by defining some mathematical operators that will facilitate the adaptation.

The Taylor expansion for a scalar function of three variables, $F(\vec{r})=F(x,y,z),$ is

$$F(\vec{r}+\delta\vec{r})=F(\vec{r})+\sum_{\alpha}\delta r^{\alpha}\frac{\partial}{\partial r^{\alpha}}F(\vec{r})+\frac{1}{2}\sum_{\alpha}\sum_{\beta}\delta r^{\alpha}\delta r^{\beta}\frac{\partial^{2}}{\partial r^{\alpha}\partial r^{\beta}}F(\vec{r})+etc. \tag{6.114}$$

Now consider a cluster of n_c point charges, $q_k,$ with $(k=1,\ldots,n_c),$ located around the origin of coordinates, $(0,0,0),$ at positions $\vec{r}_k$ and a unit point "probe" charge at a remote position $\vec{r}$ far away from the origin i.e. $r\gg r_k.$ To the probe charge the cluster represents a multipole, $M_A,$ for which we will develop an operator, $\hat{M}_A.$

The potential energy of the probe charge is given by

$$\phi(\vec{r})=\frac{1}{4\pi\epsilon_0}\sum_{k=1}^{n_c}\frac{q_k}{\left|\vec{r}-\vec{r}_k\right|}. \tag{6.115}$$

Since $r\gg r_k,$ the Taylor expansion allows us to rewrite this in the form

$$\phi(\vec{r})=\frac{1}{4\pi\epsilon_0}\left(C_A-\sum_{\alpha}D_A^{\alpha}\partial^{\alpha}+\sum_{\alpha}\sum_{\beta}Q_A^{\alpha\beta}\partial^{\alpha\beta}\right)\left(\frac{1}{r}\right), \tag{6.116}$$

which has been truncated at the quadrupole term and where

$$C_A = \sum_{k=1}^{n_c} q_k \quad \text{is the net charge (scalar),} \tag{6.117}$$

$$D_A^\alpha = \sum_{k=1}^{n_c} q_k r_k^\alpha \quad \text{is the net dipole (vector),} \tag{6.118}$$

$$Q_A^{\alpha\beta} = \frac{1}{2}\sum_{k=1}^{n_c} q_k r_k^\alpha r_k^\beta \quad \text{is the net quadrupole (matrix).} \tag{6.119}$$

These are all *tensors* of different *rank* as indicated by the number of indices. It is worth noting that although the quadrupole formally has 9 components, only 6 are unique as it is a symmetric matrix. In (6.116) we have also defined the differential operators

$$\partial^\alpha = \frac{\partial}{\partial r^\alpha}, \quad \text{and} \quad \partial^{\alpha\beta} = \frac{\partial^2}{\partial r^\alpha \partial r^\beta}. \tag{6.120}$$

We can now define the *first* multipole operator (which acts upon the function $(1/r)$ in (6.116)) as

$$\hat{M}_A = C_A - \sum_\alpha D_A^\alpha \partial^\alpha + \sum_\alpha \sum_\beta Q_A^{\alpha\beta} \partial^{\alpha\beta}. \tag{6.121}$$

This formula can be expressed more economically if we adopt the *Einstein convention* for tensor products, in which a repeated index implies a sum over that index. Thus (6.121) becomes

$$\hat{M}_A = C_A - D_A^\alpha \partial^\alpha + Q_A^{\alpha\beta} \partial^{\alpha\beta}. \tag{6.122}$$

With the first multipole operator, equation (6.116) can be written as

$$\phi(\vec{r}) = \frac{1}{4\pi\epsilon_0} \hat{M}_A \left(\frac{1}{r}\right). \tag{6.123}$$

This key result shows how the potential at a point $\vec{r}$ due to a multipole M_A at the origin can be obtained by applying the operator $\hat{M}_A$ to the function $1/4\pi\epsilon_0 r,$ which happens to represent the potential due to a single unit positive charge placed at the origin.

In the same way, the potential of a multipole M_B at the position $\vec{r},$ due to the multipole M_A at the origin (i.e. the electrostatic energy of two interacting multipoles), can be obtained as

$$\phi_{AB}(\vec{r}) = \hat{M}_B \phi(\vec{r}) = \frac{1}{4\pi\epsilon_0} \hat{M}_B \hat{M}_A \left(\frac{1}{r}\right), \tag{6.124}$$

where we have the *second* multipole operator

$$\hat{M}_B = C_B + D_B^\alpha \partial^\alpha + Q_B^{\alpha\beta} \partial^{\alpha\beta}. \tag{6.125}$$

Once again we are assuming that the displacements of charges that make up the second multipole M_B around the point $\vec{r}$ are small in relation to r.

It is useful to expand the product $\hat{M}_B \hat{M}_A$, of the first and second operators to give

$$\hat{M}_B \hat{M}_A = P_{AB}^0 + (P_{AB}^1)^\alpha \partial^\alpha + (P_{AB}^2)^{\alpha\beta} \partial^{\alpha\beta} + (P_{AB}^3)^{\alpha\beta\gamma} \partial^{\alpha\beta\gamma} + (P_{AB}^4)^{\alpha\beta\gamma\delta} \partial^{\alpha\beta\gamma\delta}, \tag{6.126}$$

where each P_{AB}^n is a tensor of rank n defined as follows

$$\begin{aligned}
&P_{AB}^0 = C_A C_B, \\
&(P_{AB}^1)^\alpha = C_A D_B^\alpha - C_B D_A^\alpha, \\
&(P_{AB}^2)^{\alpha\beta} = C_A Q_B^{\alpha\beta} + C_B Q_A^{\alpha\beta} - D_A^\alpha D_B^\beta, \\
&(P_{AB}^3)^{\alpha\beta\gamma} = -D_A^\alpha Q_B^{\beta\gamma} + D_B^\alpha Q_A^{\beta\gamma}, \\
&(P_{AB}^4)^{\alpha\beta\gamma\delta} = Q_A^{\alpha\beta} Q_B^{\gamma\delta}.
\end{aligned} \tag{6.127}$$

The force acting on multipole M_B is obtained by differentiating (6.124) and is therefore

$$\vec{F}_B = -\frac{1}{4\pi\epsilon_0} \vec{\nabla} \hat{M}_B \hat{M}_a \left(\frac{1}{r}\right). \tag{6.128}$$

From (6.126) and (6.127) we can write

$$\left(\vec{\nabla} \hat{M}_B \hat{M}_A\right)^\eta = P_{AB}^0 \partial^\eta + (P_{AB}^1)^\alpha \partial^{\eta\alpha} + (P_{AB}^2)^{\alpha\beta} \partial^{\eta\alpha\beta} + (P_{AB}^3)^{\alpha\beta\gamma} \partial^{\eta\alpha\beta\gamma} + (P_{AB}^4)^{\alpha\beta\gamma\delta} \partial^{\eta\alpha\beta\gamma\delta}, \tag{6.129}$$

which defines the F_B^η component of the force $\vec{F}_B$.

The torque acting on multipole M_B is obtained from the vector product of each charge displacement vector $q_k \vec{r}_k$ with the force at $\vec{r}_k$ summed over all charges i.e.

$$\vec{\tau}_B = -\sum_{k=1}^{n_c} q_k \vec{r}_k \times \vec{\nabla} \phi(\vec{r} + \vec{r}_k). \tag{6.130}$$

This equation can be expanded as

$$\tau_B^\eta = -\sum_{k=1}^{n_c} q_k \sum_\mu \sum_\nu \epsilon_{\eta\mu\nu} r_k^\mu \frac{\partial}{\partial r^\nu} \phi(\vec{r} + \vec{r}_k), \tag{6.131}$$

which provides the τ_B^η component of the torque $\vec{\tau}_B$. The symbol $\epsilon_{\eta\mu\nu}$ is the Levi-Civita tensor which is defined by the properties:

$$\begin{aligned}
&\epsilon_{\eta\mu\nu}=0, \quad \text{if } \eta=\mu \text{ or } \eta=\nu \text{ or } \mu=\nu,\\
&\epsilon_{\eta\mu\nu}=1, \quad \text{if } \eta\,\mu\,\nu \text{ is an } \textit{even} \text{ permutation},\\
&\epsilon_{\eta\mu\nu}=-1, \quad \text{if } \eta\,\mu\,\nu \text{ is an } \textit{odd} \text{ permutation}.
\end{aligned} \tag{6.132}$$

Re-ordering the summations in (6.131) gives

$$\tau_B^\eta=-\sum_\mu\sum_\nu \epsilon_{\eta\mu\nu}\sum_{k=1}^{n_c} q_k r_k^\mu \frac{\partial}{\partial r^\nu}\phi(\vec{r}+\vec{r}_k). \tag{6.133}$$

Expanding $\phi(\vec{r}+\vec{r}_k)$ using the Taylor expansion (6.114) allows the derivative in (6.133) to be worked through, then summing over k gives

$$\tau_B^\eta=-\epsilon_{\eta\mu\nu}\left(D_B^\mu\partial^\nu+2Q_B^{\mu\lambda}\partial^{\nu\lambda}\right)\phi(\vec{r}). \tag{6.134}$$

This equation can also be written as

$$\tau_B^\eta=\frac{1}{4\pi\epsilon_0}\hat{\tau}_B^\eta\hat{M}_A\left(\frac{1}{r}\right), \tag{6.135}$$

where

$$\hat{\tau}_B^\eta=-\epsilon_{\eta\mu\nu}\left(D_B^\mu\partial^\nu+2Q_B^{\mu\lambda}\partial^{\nu\lambda}\right) \tag{6.136}$$

is a *torque operator*. We can expand the product $\hat{\tau}_B^\eta\hat{M}_A$ from (6.135) as

$$\hat{\tau}_B^\eta\hat{M}_A=-\epsilon_{\eta\mu\nu}\left((T_{AB}^1)^\mu\partial^\nu+(T_{AB}^2)^{\mu\lambda}\partial^{\nu\lambda}+(T_{AB}^3)^{\mu\lambda\sigma}\partial^{\nu\lambda\sigma}+(T_{AB}^4)^{\mu\lambda\sigma\chi}\partial^{\nu\lambda\sigma\chi}\right), \tag{6.137}$$

where each T_{AB}^n is a tensor of rank n defined as follows

$$\begin{aligned}
&(T_{AB}^1)^\mu=C_A D_B^\mu,\\
&(T_{AB}^2)^{\mu\lambda}=2C_A Q_B^{\mu\lambda}-D_B^\mu D_A^\lambda,\\
&(T_{AB}^3)^{\mu\lambda\sigma}=D_B^\mu Q_A^{\lambda\sigma}-2D_A^\sigma Q_B^{\mu\lambda},\\
&(T_{AB}^4)^{\mu\lambda\sigma\chi}=2Q_A^{\sigma\chi}Q_B^{\mu\lambda}.
\end{aligned} \tag{6.138}$$

Our aim from this point is to apply the operators we have developed to constructing an Ewald sum for multipoles. We start with a lattice of *unit point charges* into which we insert a unit positive "probe" charge at some arbitrary point. The operators can be used to convert the potential energy of the probe charge into one corresponding to a probe point multipole in a lattice of point multipoles.

The Ewald formula for the potential of a unit probe charge in a lattice of unit point charges is obtained from equation (6.43), which, for convenience, we rewrite as

$$\phi_0(\vec{r}) = \frac{1}{4\pi\epsilon_0}\sum_{j=1}^{N} B_0(|\vec{r}-\vec{r}_j|) + \frac{1}{V\epsilon_0}\sum_{\vec{k}\neq\vec{0}}^{\vec{\infty}} A(\vec{k})\sum_{j=1}^{N}\exp(-i\vec{k}\cdot\{\vec{r}-\vec{r}_j\}), \tag{6.139}$$

where we have dropped the real space sum over $\vec{L}$ and assume a short-ranged cut-off suitable for the application of the minimum image convention. (This is not essential, but it simplifies our formulae at no cost to any ultimate application.) We have also introduced two functions:

$$A(\vec{k}) = \frac{\exp(-k^2/4\eta^2)}{k^2}, \tag{6.140}$$

(which is merely a convenient abbreviation) and

$$B_0(|\vec{r}-\vec{r}_j|) = \frac{erfc(\eta|\vec{r}-\vec{r}_j|)}{|\vec{r}-\vec{r}_j|}, \tag{6.141}$$

which is an important function in what follows. (Note that in this section we represent the Ewald constant as η rather than the usual α to avoid confusion with the components of the multipole tensors.)

We now use the combined operator shown in (6.124) to convert both the unit charges of the lattice and the probe charge at $\vec{r}$ into point multipoles:

$$\phi_p(\vec{r}) = \frac{1}{4\pi\epsilon_0}\sum_{j=1}^{N} \hat{M}_p\hat{M}_j B_0(|\vec{r}-\vec{r}_j|) + \frac{1}{V\epsilon_0}\sum_{\vec{k}\neq\vec{0}}^{\vec{\infty}} A(\vec{k})\sum_{j=1}^{N}\hat{M}_p\hat{M}_j\exp(-i\vec{k}\cdot\{\vec{r}-\vec{r}_j\}), \tag{6.142}$$

where $\hat{M}_p$ represents the probe multipole and $\hat{M}_j$ a lattice multipole. As we have seen from the expansion (6.126) and the definitions (6.127) the operator $\hat{M}_p\hat{M}_j$ is a series of operators of the kind $(P_{pj}^n)^{\alpha\beta\cdots}\partial^{\alpha\beta\cdots}$ where P_{pj}^n is a tensor constructed from multipole components and $\partial^{\alpha\beta\cdots}$ is a differential operator. We therefore now examine the effect of the differential operators on equation (6.142).

Multiple differentiation of the function $B_0(u)$ with respect to argument $u=|\vec{r}-\vec{r}_j|$ is facilitated by first defining a series of functions $B_n(u),$ with $n>0,$ which are generated using the following recursion relation:

$$B_n(u) = \frac{1}{u^2}\left((2n-1)B_{n-1}(u) + \frac{(2\eta^2)^n}{\eta\sqrt{\pi}}\exp(-\eta^2u^2)\right). \tag{6.143}$$

It can then be shown by mathematical induction that

$$\frac{\partial}{\partial u^\alpha}B_n(u) = -u^\alpha B_{n+1}(u). \tag{6.144}$$

From this we can efficiently obtain the higher derivatives of $B_0(u),$ which we write as the tensor functions $R^n(u),$ with rank n:

$$
\begin{aligned}
&R^0(u)=B_0(u),\\
&\left(R^1(u)\right)^{\alpha}=\partial^{\alpha}B_0(u)=-B_1(u)u^{\alpha},\\
&\left(R^2(u)\right)^{\alpha\alpha}=\partial^{\alpha\alpha}B_0(u)=B_2(u)u^{\alpha}u^{\alpha}-B_1(u),\\
&\left(R^2(u)\right)^{\alpha\beta}=\partial^{\alpha\beta}B_0(u)=B_2(u)u^{\alpha}u^{\beta},\\
&\left(R^3(u)\right)^{\alpha\alpha\alpha}=\partial^{\alpha\alpha\alpha}B_0(u)=-B_3(u)u^{\alpha}u^{\alpha}u^{\alpha}+3\,B_2(u)u^{\alpha},\\
&\left(R^3(u)\right)^{\alpha\alpha\beta}=\partial^{\alpha\alpha\beta}B_0(u)=-B_3(u)u^{\alpha}u^{\alpha}u^{\beta}+B_2(u)u^{\beta},\\
&\left(R^3(u)\right)^{\alpha\beta\gamma}=\partial^{\alpha\beta\gamma}B_0(u)=-B_3(u)u^{\alpha}u^{\beta}u^{\gamma},\\
&\left(R^4(u)\right)^{\alpha\alpha\alpha\alpha}=\partial^{\alpha\alpha\alpha\alpha}B_0(u)=B_4(u)u^{\alpha}u^{\alpha}u^{\alpha}u^{\alpha}-6\,B_3(u)u^{\alpha}u^{\alpha}+3\,B_2(u),\\
&\left(R^4(u)\right)^{\alpha\alpha\alpha\beta}=\partial^{\alpha\alpha\alpha\beta}B_0(u)=B_4(u)u^{\alpha}u^{\alpha}u^{\alpha}u^{\beta}-3\,B_3(u)u^{\alpha}u^{\beta},\\
&\left(R^4(u)\right)^{\alpha\alpha\beta\beta}=\partial^{\alpha\alpha\beta\beta}B_0(u)=B_4(u)u^{\alpha}u^{\alpha}u^{\beta}u^{\beta}-B_3(u)\left(u^{\alpha}u^{\alpha}+u^{\beta}u^{\beta}\right)+B_2(u),\\
&\left(R^4(u)\right)^{\alpha\alpha\beta\gamma}=\partial^{\alpha\alpha\beta\gamma}B_0(u)=B_4(u)u^{\alpha}u^{\alpha}u^{\beta}u^{\gamma}-B_3(u)u^{\beta}u^{\gamma},\\
&\left(R^5(u)\right)^{\alpha\alpha\alpha\alpha\alpha}=\partial^{\alpha\alpha\alpha\alpha\alpha}B_0(u)=-B_5(u)u^{\alpha}u^{\alpha}u^{\alpha}u^{\alpha}u^{\alpha}+10\,B_4(u)u^{\alpha}u^{\alpha}u^{\alpha}-15\,B_3(u)u^{\alpha},\\
&\left(R^5(u)\right)^{\alpha\alpha\alpha\alpha\beta}=\partial^{\alpha\alpha\alpha\alpha\beta}B_0(u)=-B_5(u)u^{\alpha}u^{\alpha}u^{\alpha}u^{\alpha}u^{\beta}+6\,B_4(u)u^{\alpha}u^{\alpha}u^{\beta}-3\,B_3(u)u^{\beta},\\
&\left(R^5(u)\right)^{\alpha\alpha\alpha\beta\beta}=\partial^{\alpha\alpha\alpha\beta\beta}B_0(u)=-B_5(u)u^{\alpha}u^{\alpha}u^{\alpha}u^{\beta}u^{\beta}+B_4(u)\left(3\,u^{\alpha}u^{\beta}u^{\beta}+u^{\alpha}u^{\alpha}u^{\alpha}\right)-3\,B_3(u)u^{\alpha},\\
&\left(R^5(u)\right)^{\alpha\alpha\alpha\beta\gamma}=\partial^{\alpha\alpha\alpha\beta\gamma}B_0(u)=-B_5(u)u^{\alpha}u^{\alpha}u^{\alpha}u^{\beta}u^{\gamma}+3\,B_4(u)u^{\alpha}u^{\beta}u^{\gamma},\\
&\left(R^5(u)\right)^{\alpha\alpha\beta\beta\gamma}=\partial^{\alpha\alpha\beta\beta\gamma}B_0(u)=-B_5(u)u^{\alpha}u^{\alpha}u^{\beta}u^{\beta}u^{\gamma}+B_4(u)\left(u^{\alpha}u^{\alpha}u^{\gamma}+u^{\beta}u^{\beta}u^{\gamma}\right)-B_3(u)u^{\gamma},
\end{aligned}
\tag{6.145}
$$

In these formulae we have assumed that $\alpha\neq\beta\neq\gamma$ and exploited the fact that the tensors $R^n(u)$ are unchanged by any permutation of the shown indices to reduce the number of formulae we have to specify. Apart from the tensors of rank 5 (which are needed later to calculate the force), these tensors together with the tensors P^n_{pj} defined in equations (6.127) are all we need to process the real space part of (6.142).

Now we consider the action of the differential operators $\partial^{\alpha\beta\ldots}$ on functions $\exp(-i\vec{k}\cdot\vec{u})$, where $\vec{u}=\vec{r}-\vec{r}_j$. These are easily obtained as

$$
\begin{aligned}
&\partial^{\alpha}\exp(-i\vec{k}\cdot\vec{u})=-i\,k^{\alpha}\exp(-i\vec{k}\cdot\vec{u}),\\
&\partial^{\alpha\beta}\exp(-i\vec{k}\cdot\vec{u})=-k^{\alpha}k^{\beta}\exp(-i\vec{k}\cdot\vec{u}),\\
&\partial^{\alpha\beta\gamma}\exp(-i\vec{k}\cdot\vec{u})=i\,k^{\alpha}k^{\beta}k^{\gamma}\exp(-i\vec{k}\cdot\vec{u}),\\
&\partial^{\alpha\beta\gamma\delta}\exp(-i\vec{k}\cdot\vec{u})=k^{\alpha}k^{\beta}k^{\gamma}k^{\delta}\exp(-i\vec{k}\cdot\vec{u}),\\
&\partial^{\alpha\beta\gamma\delta\mu}\exp(-i\vec{k}\cdot\vec{u})=-i\,k^{\alpha}k^{\beta}k^{\gamma}k^{\delta}k^{\mu}\exp(-i\vec{k}\cdot\vec{u}).
\end{aligned}
\tag{6.146}
$$

Once again these may be combined with the tensors P^n_{pj} to complete the processing of the reciprocal space part of (6.142).

With the expansions (6.145) and (6.146) equation (6.142) becomes

$$
\phi_p(\vec{r})=\frac{1}{4\pi\epsilon_0}\sum_{n=0}^{4}\sum_{j=1}^{N}P^n_{jp}:R^n_{jp}(|\vec{r}-\vec{r}_j|)+\frac{1}{V\epsilon_0}\sum_{\vec{k}\neq\vec{0}}^{\vec{\infty}}A(\vec{k})\sum_{n=0}^{4}\sum_{j=1}^{N}(-i)^n P^n_{jp}:K^n\exp\left(-i\vec{k}\cdot\{\vec{r}-\vec{r}_j\}\right),
$$

(6.147)

in which we have the rank n tensor functions

$$R^n_{jp}(|\vec{r}-\vec{r}_j|)\equiv R^n(u) \quad \text{with} \quad u=|\vec{r}-\vec{r}_j| \tag{6.148}$$

which follows directly from definitions (6.145), and

$$(K^n)^{\alpha\beta\cdots}=k^\alpha k^\beta \ldots etc. \tag{6.149}$$

The operation indicated as $A:B$ in (6.147) implies a scalar product of the tensors A and B i.e. each element of A is multiplied by the corresponding element of B and summed to obtain the scalar result. It follows that A and B must have the same rank if the product $A:B$ is to be meaningful.

So far equation (6.147) represents the potential of a probe multipole in a periodic lattice of multipoles, but we require the potential of a lattice multipole. To obtain this we first assign to the probe multipole the characteristics of the point multipole at the lattice site of interest. Next we consider the interaction between the probe and the multipole on the site as the probe is moved into superposition on the same site. This represents a *self interaction* which must be extracted from equation (6.147). In determining the self interaction correction, we must bear in mind that in the Ewald prescription the multipole on the lattice site is not a simple object - it is a Gaussian multipole in reciprocal space and a point multipole screened by a Gaussian multipole in real space.

We consider the real space terms first. This is a sum of scalar products of the form $P^n_{jp}:R^n_{jp}(u)$ in which the P^n_{jp} tensor has no distance dependence. Thus we need only consider how the tensor functions $R^n_{jp}(u)$ behave as $u\to 0$ to work out the effect of the superposition. From equations (6.145) we see that this is partly decided by terms like u^α, $u^{\alpha\beta}$ etc. (which readily go to zero as $u\to 0,$) and the functions $B_n(u),$ which have a more complicated behaviour. Since all functions $B_n(u)$ are derived from $B_0(u)$ through the recursion (6.143), we find the following Maclaurin expansions useful

$$B_0(u)=\frac{1}{u}-\frac{2\eta}{\sqrt{\pi}}\left(1-\frac{(\eta u)^2}{3}+\frac{(\eta u)^4}{10}-\frac{(\eta u)^6}{42}+\frac{(\eta u)^8}{216}-\ldots\right) \tag{6.150}$$

and

$$\frac{2\eta}{\sqrt{\pi}}\exp\left(-\eta^2u^2\right)=\frac{2\eta}{\sqrt{\pi}}\left(1-(\eta u)^2+\frac{(\eta u)^4}{2}-\frac{(\eta u)^6}{6}+\frac{(\eta u)^8}{24}-\ldots\right), \tag{6.151}$$

with the aid of which and recursion relation (6.143) we can show that for small u

$$B_n(u)=\frac{(2n)!}{2^n n!u^{2n+1}}-\frac{(2\eta^2)^{n+1}}{\sqrt{\pi}\eta(2n+1)}+O(u). \tag{6.152}$$

This form can be used in place of $B_n(u)$ in (6.145) to investigate what happens to functions $R^n_{jp}(u)$ when $u \to 0.$ Clearly the term of order $O(u)$ in (6.152) vanishes. Meanwhile the second term on the right remains constant. The first term however rises to infinity, but it can be shown that this term describes the interaction between the probe multipole and the *unscreened* point multipole on the lattice site. For this reason we may simply remove the term since it represents an interaction that does not exist when the probe replaces the site multipole. Feeding this back into equations (6.145) as $u \to 0$ leads to the following

$$\begin{aligned} R^0_{pp}(0) &= B_0(0) = -2\eta/\sqrt{\pi}, \\ (R^2_{pp})^{\alpha\alpha}(0) &= -B_1(0) = 4\eta^2/3\sqrt{\pi}, \\ (R^4_{pp})^{\alpha\alpha\alpha\alpha}(0) &= 3B_2(0) = -24\eta^5/5\sqrt{\pi}, \\ (R^4_{pp})^{\alpha\alpha\beta\beta}(0) &= B_2(0) = -8\eta^5/5\sqrt{\pi}, \end{aligned} \tag{6.153}$$

with all other tensor components being zero.

These constant terms survive from (6.145) as $u \to 0$ and, since the bare multipole terms have been extracted, these account for the interaction between the probe and the screening Gaussian multipole. In real space we would be justified in removing these terms, since they do not formally exist, however if they are retained they will compensate for the interaction between the probe and the Gaussian multipole which is included in reciprocal space. So in fact retaining these surviving terms provides the full self interaction correction we seek. Gathering all these terms gives the self interaction correction as

$$\phi^S_p = \frac{1}{4\pi\epsilon_0}\left(P^0_{pp} : R^0_{pp}(0) + P^r_{pp} : R^2_{pp}(0) + P^4_{pp} : R^4_{pp}(0)\right). \tag{6.154}$$

Then using tensor definitions (6.127) and the reduced forms (6.153) this becomes

$$\phi^S_p = -\frac{2\eta}{4\epsilon_0\pi^{3/2}}\left(C^2_p + 2\eta^2\left\{\frac{1}{3}\left[D^\alpha_p D^\alpha_p - 2C_p Q^{\alpha\beta}_p \delta_{\alpha\beta}\right] + \frac{2\eta^2}{5}\left[2Q^{\alpha\beta}_p Q^{\alpha\beta}_p + \left(Q^{\alpha\beta}_p \delta_{\alpha\beta}\right)^2\right]\right\}\right). \tag{6.155}$$

This can also be written as

$$\phi^S_p = -\frac{2\eta}{4\epsilon_0\pi^{3/2}}\left(C^2_p + 2\eta^2\left\{\frac{1}{3}\left[D^2_p - 2C_p Q_p : \mathbf{1}\right] + \frac{2\eta^2}{5}\left[2Q_p : Q_p + \left(Q_p : \mathbf{1}\right)^2\right]\right\}\right), \tag{6.156}$$

where $\mathbf{1}$ is the unit matrix and operation $(:)$ indicates a scalar product.

We can now write the potential energy of a multipole in a lattice of multipoles as

$$\phi_p(\vec{r}_p)=\frac{1}{4\pi\epsilon_0}\sum_{n=0}^{4}\sum_{j\neq p}^{N}P^n_{jp}:R^n_{jp}(|\vec{r}_p-\vec{r}_j|)+ \\ \frac{1}{V\epsilon_0}\sum_{\vec{k}\neq\vec{0}}^{\vec{\infty}}A(\vec{k})\sum_{n=0}^{4}\sum_{j=1}^{N}(-i)^n P^n_{jp}:K^n\exp(-i\vec{k}\cdot\{\vec{r}_p-\vec{r}_j\})+\phi^S_p, \tag{6.157}$$

where multipole p is now a member of the lattice of multipoles.

From (6.157) the lattice configuration energy for N multipoles can be written as

$$\Phi^M(\{\vec{r}_i\})=\frac{1}{2}\frac{1}{4\pi\epsilon_0}\sum_{n=0}^{4}\sum_{p}^{N}\sum_{j}^{N}{}'P^n_{jp}:R^n_{jp}(|\vec{r}_p-\vec{r}_j|)+ \\ \frac{1}{2V\epsilon_0}\sum_{\vec{k}\neq\vec{0}}^{\vec{\infty}}A(\vec{k})\left|\sum_{j=1}^{N}\sum_{n=0}^{2}p^n_j:K^n\exp(-i\vec{k}\cdot\vec{r}_j)\right|^2+\frac{1}{2}\sum_{j=1}^{N}\phi^S_j, \tag{6.158}$$

which follows from summing (6.157) over multipoles $p.$ The new tensors p^n_j are given by

$$p^0_j=C_j, \qquad p^1_j=-i\vec{D}_j \qquad \text{and} \qquad p^2_j=-\boldsymbol{Q}_j. \tag{6.159}$$

The form of the reciprocal space part in (6.158) is based on the decomposition

$$\sum_{n=0}^{4}\sum_{j=1}^{N}\sum_{p=1}^{N}(-i)^n P^n_{jp}:K^n\exp(-i\vec{k}\cdot\{\vec{r}_p-\vec{r}_j\})= \\ \sum_{p=1}^{N}\sum_{m=0}^{2}p^m_p:K^m\exp(-i\vec{k}\cdot\vec{r}_p)\sum_{j=1}^{N}\sum_{n=0}^{2}(p^n_j)^*:K^n\exp(i\vec{k}\cdot\vec{r}_j), \tag{6.160}$$

which is a separation of terms for multipoles p and j. Note the complex conjugate $(p^n_j)^*$ appearing in (6.160).

The force on a lattice multipole p is obtained by differentiation of (6.157) using $\vec{\nabla}_p$, which according to (6.129) and (6.145) leads to

$$f^\eta_p=-\frac{1}{4\pi\epsilon_0}\sum_{n=0}^{4}\sum_{j\neq p}^{N}P^n_{jp}\partial^\eta R^n_{jp}(r_{jp})+\frac{1}{V\epsilon_0}\sum_{\vec{k}\neq\vec{0}}^{\vec{\infty}}A(\vec{k})k^\eta\Re\left\{i\sum_{m=0}^{2}p^m_p:K^m\exp(-i\vec{k}\cdot\vec{r}_p)M^*(\vec{k})\right\}. \tag{6.161}$$

where symbol $\Re$ indicates the real part of the following complex term and

$$r_{jp}=|\vec{r}_p-\vec{r}_j|. \tag{6.162}$$

and

$$M(\vec{k})=\sum_{j=1}^{N}\sum_{n=0}^{n=2}(p^n_j):K^n\exp(-i\vec{k}\cdot\vec{r}_j). \tag{6.163}$$

Note that (6.161) uses the complex conjugate of $M(\vec{k})$. The terms $\partial^{\eta} R^{\eta}_{jp}(r_{jp})$ can be obtained directly from the formulae (6.145).

The torque acting on a multipole on a lattice site is obtained by applying the operator (6.137) to the Ewald potential (6.139) (which applies to a probe at an off-lattice site) and, as previously, then removing the terms corresponding to any self interaction as the probe is moved on to a lattice site. The only modification required in this case is to set the condition $j \neq p$ in the real space sum i.e. direct removal of the self interaction term. No modification is required for the reciprocal space sum, since a Gaussian multipole exerts no torque on an identical multipole superimposed on the same site (and there are no infinities in the formulae). The result is

$$\tau^{\eta}_{p} = -\frac{1}{4\pi\epsilon_0}\sum_{j\neq p}^{N} \epsilon_{\eta\alpha\beta}\Big((T^1_{jp})^{\alpha} (R^1_{jp}(r_{jp}))^{\beta} + (P^2_{jp})^{\alpha\gamma} (R^2_{jp}(r_{jp}))^{\beta\gamma} + (T^3_{jp})^{\alpha\gamma\delta} (R^3_{jp}(r_{jp}))^{\beta\gamma\delta} + (T^4_{jp})^{\alpha\gamma\delta\mu} (R^4_{jp}(r_{jp}))^{\beta\gamma\delta\mu} \Big) + \frac{1}{V\epsilon_0}\sum_{\vec{k}\neq\vec{0}}^{\vec{\infty}} A(\vec{k}) \Re\left\{ \epsilon_{\eta\alpha\beta}\Big(i D^{\alpha}_{p}(K^1)^{\beta} + 2 Q^{\alpha\gamma}_{p}(K^2)^{\beta\gamma} \Big) \exp(-i\vec{k}\cdot\vec{r}_{p}) M^{*}(\vec{k}) \right\}, \tag{6.164}$$

where the tensors T^{n}_{jp} are specified in (6.138).

Writing a program based on these equations is clearly a non-trivial task. Nevertheless the described method is probably the simplest approach to the problem of simulating multipole systems. The most difficult aspect is handling the real space terms in the formulae for the energy (6.158), force (6.161) and torque (6.164). However, it is encouraging to note that the construction of the tensors $R^{n}_{jp}(u)$, which depend on the functions $B_n(u)$, is greatly facilitated by use of the recursion relation (6.143). Furthermore the number of non-zero components of the higher $R^{n}_{jp}(u)$ tensors is strictly limited, as may be grasped by the list given in (6.145).

Some further comments are in order. Firstly, the definition of the quadrupole here is based on the form (6.119), which is *not* the form that is that commonly encountered in the literature. The matrix $\boldsymbol{Q}$ used here is, strictly speaking, the *second moment* of the charge distribution. The standard multipole, $\boldsymbol{\Xi}$, can be obtained from $\boldsymbol{Q}$ via the relation:

$$\boldsymbol{\Xi}^{\alpha\beta} = 3\boldsymbol{Q}^{\alpha\beta} - \delta_{\alpha\beta} Tr\,\boldsymbol{Q} \tag{6.165}$$

where $Tr\,\boldsymbol{Q}$ is the sum of the diagonal elements of $\boldsymbol{Q}$. Matrix $\boldsymbol{\Xi}$ is the so-called *traceless* multipole. It is not possible to convert from $\boldsymbol{\Xi}$ to $\boldsymbol{Q}$ unless the trace of $\boldsymbol{Q}$ is known beforehand. This is not a problem if the components are calculated by quantum mechanics.

Finally, it is important to note that this method is based on multipoles that are defined with respect to the *laboratory frame* and as a result the multipole components change as the molecules rotate in space. Recalculation of the rotated multipole is accomplished using the standard rotation matrix. i.e. if

$$\vec{r} = \boldsymbol{R}\,\vec{r}_o \tag{6.166}$$

where $\vec{r}_o$ is a vector in the body (or principal) frame of reference, $\vec{r}$ is the corresponding vector in the laboratory frame and $\boldsymbol{R}$ is the rotation matrix (see chapter 4), then conversion of the a multipole from the body frame to the laboratory frame is accomplished by the transformations for the charge (C), dipole $(\vec{D})$ and quadrupole $(\boldsymbol{Q})$:

$$C = C_o, \qquad D^\alpha = \sum_{\beta=1}^{3} \boldsymbol{R}^{\alpha\beta} D_o^\beta, \qquad \boldsymbol{Q}^{\alpha\beta} = \sum_{\mu=1}^{3}\sum_{\nu=1}^{3} \boldsymbol{Q}_o^{\mu\nu}\,\boldsymbol{R}^{\alpha\mu}\,\boldsymbol{R}^{\beta\nu} \tag{6.167}$$

where the subscript $(_o)$ indicates the multipole in the body frame.

6.8 The Fast Multipole Method

The Ewald sum is ideal for periodic systems, but it is not well suited to systems that have no periodicity. For such systems the Fast Multipole Method (FMM) is arguably the best choice. We shall describe the method due to Greengard and Rokhlin [54]. It has excellent scaling characteristics, as well as being supremely elegant in construction, and is particularly efficient for large scale systems. The description will be as brief as possible. The details may be found in Greengard and Rokhlin's article [54].

The FMM is a hierarchical cell method based on electrostatic multipoles. In this context a multipole, M, is defined as a *spherical harmonic* expansion of the charge distribution of a collection of ions within a sphere of radius R_m as in figure 6.8:

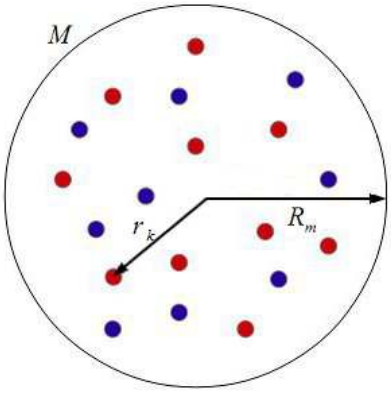

Figure 6.8:Distribution of Charges in a Sphere

A spherical harmonic multipole component, M_{lm}, is defined by the formula

$$M_{lm} = \sum_{k=1}^{n_c} q_k r_k^l Y_{lm}^*(\theta_k, \phi_k) \quad \text{where} \quad r_k < R_m, \tag{6.168}$$

in which $r_k = |\vec{r}_k|$ is the distance of the charge q_k from the centre of the sphere and

θ_k, ϕ_k are the associated angular coordinates. Y_{lm} is a standard (complex) spherical harmonic function. Using (6.168) multipoles of any order l, m can be obtained for the same set of charges, though they are expected to become increasingly insignificant as $l \to \infty$. Greengard and Rokhlin provide formulae to estimate the errors for any chosen truncation of this and similar expansions [54].

At a point P, located at $\vec{r}$, from the centre of charge cluster, M, with $r > R_m$, the electrostatic potential may be obtained from the expansion (see figure 6.9):

$$\phi(r,\theta,\phi) = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} \frac{M_{lm}}{r^{l+1}} Y_{lm}(\theta,\phi), \quad \text{where} \quad r > R_m. \tag{6.169}$$

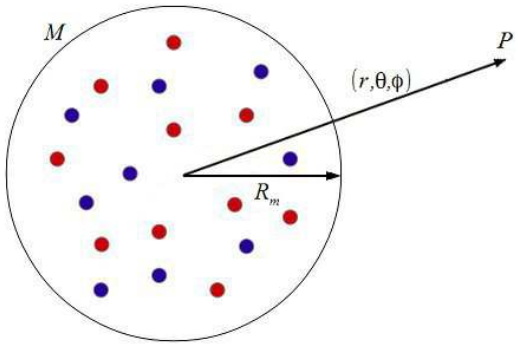

Figure 6.9:The Potential Outside a Sphere of Charges

At the point P, we may define the centre of a second sphere of radius R_p (see figure 6.10). It can then be shown that provided the two spheres do not overlap $(i.e.\ r > R_m + R_p)$, at any point $\vec{u}$ within that sphere, the potential can be calculated from the so called *local* expansion:

$$\phi(u,\alpha,\beta) = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} L_{lm} u^l Y_{lm}(\alpha,\beta), \quad \text{where} \quad \begin{matrix} u < R_p, \\ r > R_m + R_p. \end{matrix} \tag{6.170}$$

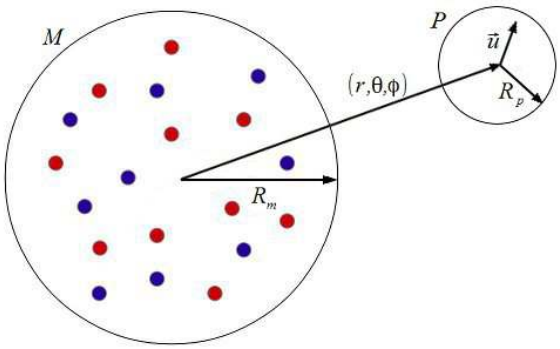

Figure 6.10: The Local Expansion of the Multipole Potential

Variables u, α, β are the spherical polar coordinates of $\vec{u}$ and L_{lm} is a coefficient of the local expansion defined as

$$L_{lm}=\sum_{j=0}^{\infty}\sum_{k=-j}^{j}\frac{M_{jk}J_{mk}A_{jk}A_{lm}Y_{(l+j)(k-m)}(\alpha,\beta)}{A_{(l+j)(k-m)}}r^{l+j+1}, \tag{6.171}$$

where

$$A_{lm}=\frac{(-1)^l}{\sqrt{(l+m)!(l-m)!}} \quad \text{and} \quad J_{mk}=\begin{cases}(-1)^l & if \quad km\le 0\\ (-1)^l(-1)^{min(|k|,|m|)} & if \quad km>0.\end{cases} \tag{6.172}$$

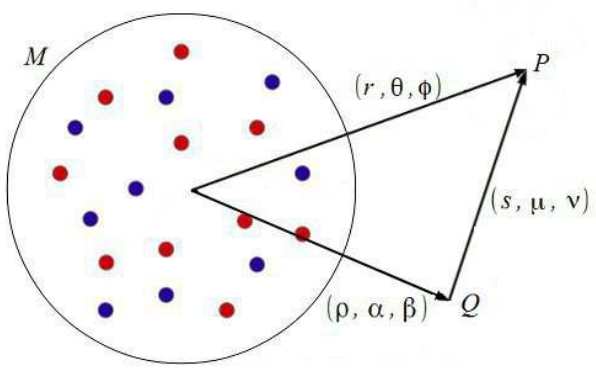

Figure 6.11: The Shift Theorem (from M to Q).

Further valuable tools in the FMM approach are the shift theorems. For example, the influence at a point P in space due to any given multipole, M, can be completely described by another, fictitious, multipole, Q, located somewhere else in space (see figure 6.11). The multipole Q is fictitious in the sense that it is not physically present, it merely represents an abstract alternative to the multipole M that has the same effect at P. In other words the potential given by equation (6.169) can just as well be described by the following expansion

$$\phi(s,\mu,\nu)=\sum_{l=0}^{\infty}\sum_{m=-l}^{l}\frac{Q_{lm}}{s^{l+1}}Y_{lm}(\mu,\nu), \tag{6.173}$$

where (s,μ,ν) are the polar coordinates of the point P with respect to the centre of multipole Q and

$$Q_{lm}=\sum_{j=0}^{l}\sum_{k=-j}^{j}\frac{M_{(l-j)(m-k)}J_{k(m-k)}A_{jk}A_{(l-j)(m-k)}\rho^j Y_{jk}^*(\alpha\beta)}{A_{lm}}, \tag{6.174}$$

in which (ρ,α,β) are the polar coordinates of the position of Q with respect to the centre of multipole M. There are several such theorems needed for the FMM scheme.

The FMM is implemented using a *tree structure*, which is described as follows. The entire system is first enclosed by a cubic simulation cell. This initial cube is denoted level 0. The cube is then divided into 8 smaller cubes, equal in size, and filling the original cube. This is level 1. Level 2 is a further 8-fold division on the level 1 cubes. Sub-division continues to a maximum level, L_{max}, where the smallest cells contain

only a small number of ions (some may have zero population). Ultimately, the total number of cubes in the tree structure is $8^{L_{max}}$. Notationally the cubes at a level L, are said to be the *child* cells of the *parent* enclosing cell at level $L-1$. Figure 6.12 shows the division of the system as far as level 2. Implementation of the FMM scheme takes place within this tree structure. Two passes through the structure occur: an *upward* pass followed by a *downward* pass, at the end of which all the required interactions have been calculated. We will now describe each pass.

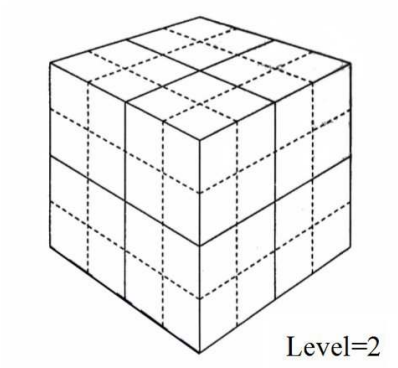

Figure 6.12: The FMM tree Structure Down to Level 2.

Upward Pass:

- To start the multipoles, $M_{lm}^{L_{max}}$, for each cell at level L_{max} are calculated.
- Using the shift formulae, the multipoles from the 8 child cells are used to create the multipole of the parent cell, at the parent cell centre (see figure 6.13).
- From level L_{max} the construction of parent multipoles continues sequentially up to level 2. (This is as far as it is necessary to go for a non-periodic system.)

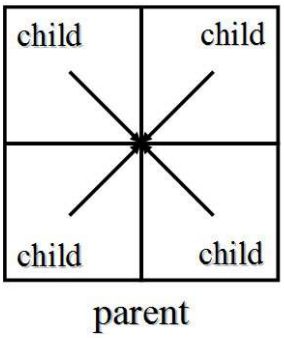

Figure 6.13: Construction of Parent Multipole from the (8) Child Multipoles

Downward Pass:

- Starting at level 2, work down sequentially to level L_{max}.
- At level L, for all cells, add to the local expansion contributions from multipoles of all cells of their interaction list. These are the cells at the same level which are *not* nearest neighbours nor next nearest neighbours, but *are* the child cells of the nearest neighbour or next nearest neighbour cell of the parent cell. (See figure 6.14 for further clarification.)

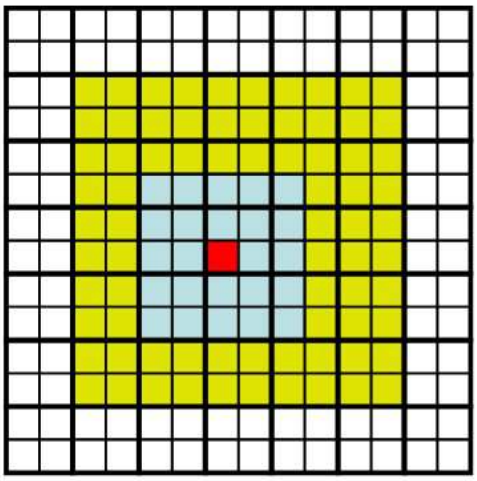

Figure 6.14: The Interaction List of Cells (yellow) of the Red Cell.
(No other cells at the same level are considered.)

- The local expansions for each cell at level L are shifted to the 8 child cells at the next level $(L+1)$.
- At level L_{max} the potential (and force) is obtained from the local expansion plus the direct ion-ion interactions from the nearest neighbour and next nearest neighbour cells.

A key feature of this scheme is that the approximations implicit in the use of truncated multipole expansions are kept within acceptable accuracy. So, for example, the *interaction list* described above does not incorporate contributions from neighbouring cells closer that two cell widths away. This is consistent with the restriction $r > R_m + R_p$ in equation (6.170).Greengard and Rohklin showed that the FMM scales as an $O(N)$ algorithm. If the accepted error is ϵ, the order of the local expansions is $\approx \log_2 \epsilon$ and the time to completion is of order $N(e[\log_2 \epsilon]^2 + f)$, where e and f are machine constants. Data storage is of order $O(N)$.

It would be naive to think this is an easy algorithm to code up(!) It is complicated to implement and, as it turns out, rather expensive in computer time, despite the $O(N)$ dependence. Nevertheless, it is an important example of a class of algorithms that dominate large scale computation, where the $O(N)$ scaling eventually wins out over the cost of managing the many expansion coefficients. For systems beyond 20,000 atoms, it is a serious alternative to the SPME method.

6.9 Ionic Polarisation

Thus far in this chapter we have dealt with systems in which the ions are represented by simple point charges or multipoles. Successful as such models are they do not account for one very important property of many ionic materials: polarisation. In such systems it is found experimentally that placing a sample in an electric field *induces* the atoms to develop a dipole moment. This is expressed by the equation

$$\vec{\mu}_i = \alpha_i \vec{E}\,, \tag{6.175}$$

where $\vec{\mu}$ represents the induced dipole on an atom i, $\vec{E}$ is the applied electric field and α_i is the *electric polarisability* of the atom. Since polarisation can have a profound effect on the electric properties of materials, it is important to be able to incorporate it into molecular dynamics studies.

Arguably the most successful model including the polarisability, and certainly the most widely used, is the so called shell model devised by Dick and Overhauser [65]. The model is a simple one, it assumes that an ion can be represented by a charged core surrounded by a massless shell of opposing charge to which it is coupled by a harmonic spring (figure 6.15). The charges on the core and shell are not necessarily equal and opposite and their sum is the net charge on the ion. Formally there is no electrostatic interaction between the core and shell in the same atom, but they can interact with other charged or polarised atoms or an imposed electric field. The core and shell respond individually to the local electric field, but the separation of the charges is governed by the respective strengths of the harmonic spring and the field. In the absence of any local electric field, the core and shell relax to the same position. The potential energy of the stretched spring is equivalent to the polarisation energy of the atom, from which the system polarisation energy can be calculated. The parameters of the model, namely the charges on the core and shell and the force constant of the harmonic spring, are obtained by fitting the physical properties of the model to the experimental structural and electric properties of the real material.

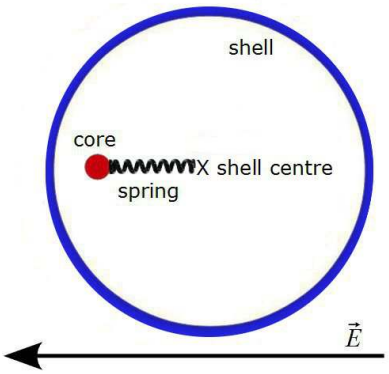

Figure 6.15: The Shell Polarisation Model

This simple model works well in static lattice studies, where it first found application, but its incorporation into molecular dynamics is problematic on account of the zero mass of the shell. Clearly any object of zero mass cannot be sensibly driven by normal dynamical procedures. Two methods have been devised to deal with this: the Adiabatic Shell Model [66] and the Relaxed Shell Model [67] , which we describe below. Both of these methods are widely used.

6.9.1 The Adiabatic Shell Model

The Adiabatic Shell Model is due to Fincham and Mitchell [66]. In this model the shell is assigned a small fraction of the total mass of the ion, with the aim of obtaining an

object that can be driven by conventional molecular dynamics. The shell mass thus becomes another parameter in the overall model. However, simply assigning an arbitrary mass to the shell effectively changes the original ion into a *pseudo*-diatomic molecule with internal vibrational and rotational kinetic energy that are not part of the intended scheme. To deal with this, care is taken to ensure that the dynamics of the core-shell unit representing each ion is *adiabatic*, meaning that it does not exchange kinetic energy with any other degree of freedom in the system. As a result core-shell units that start out with little or no internal kinetic energy maintain this state as far as possible and only the stretching of the core-shell spring makes any change to the overall internal energy.

Fortunately, systems containing harmonic potentials, such as the core-shell spring, are often notoriously non-ergodic. The exchange of kinetic energy between degrees of freedom is naturally inefficient. The Adiabatic Shell Model therefore aims to exacerbate this tendency. This is done by ensuring the natural vibrational frequency of the core-shell unit is significantly higher than that of any other degree of freedom in the system. This weakens the coupling to other degrees of freedom and helps to maintain adiabaticity in the dynamics. The natural frequency of vibration, $\nu,$ of the core-shell vibration is given classically by the formula

$$\nu=\frac{1}{2\pi}\left(\frac{k}{mx(1-x)}\right)^{1/2}, \tag{6.176}$$

where m is the total mass of the ion and x is the fraction of the ionic mass assigned to the shell. Evidently, by adjusting $x,$ the frequency of the vibration can be "tuned" to any required frequency. Comparison of this frequency with other degrees of freedom can be obtained by simulating the system initially with a rigid (non-polarisable) ionic model and generating the velocity auto-correlation function of the system as a whole. The Fourier transform of the correlation function provides a spectrum of all frequencies in the system. The aim is to adjust x in (6.176) to guarantee ν is a factor of 3 or more above the highest frequency occurring. Once this has been settled, the polarisable system may be modelled using conventional molecular dynamics. The polarisable ions in this model are able to align with the local electric field and polarise in response to the local field strength.

There are some other things to note about this method. Firstly the small mass of the shell means that shell motion is rapid. In turn this implies the use of a smaller time step, $(\Delta t),$ than is used in a rigid ion simulation. Inevitably this means that simulation times will be longer if they are to range as far through phase space. Secondly, the adiabaticity of the dynamics needs to be monitored at all times. The temperature or kinetic energy of the core-shell unit should not be allowed to rise to more than (say) $10\ K,$ for a simulation set at room temperature. If this occurs, it implies that the shell mass is incorrectly set. Inevitably some kinetic energy will leak into the core-shell units, so it is a matter of judgement and experience to decide if the simulation is compromised. Finally, temperature scaling or thermostat methods used with the Adiabatic Shell Model, should not be allowed to affect the internal kinetic energy of the core-shell unit. It is not advisable to attempt to control of the internal kinetic energy in any way. It would act as a "heat sink" in the system and invalidate the assumption of adiabaticity.

6.9.2 The Relaxed Shell Model

The Relaxed Shell Model was devised by Lindan and Gillan [67]. In this method the zero mass of the shell is retained. As in any molecular dynamics simulation the forces acting on the cores and shells are calculated as normal (including the forces from the core-shell springs), but the integration algorithm that drives the dynamics is only used to update the positions of the cores. Updating the shell positions it is handled separately, by an optimisation procedure such as *Conjugate Gradients*, that relaxes the shells to zero force. The zero force on the shells at all times means they never acquire kinetic energy and thus avoid the issues associated with internal degrees of freedom. Meanwhile the core-shell separation encapsulates the ionic polarisation as required. The system must be initialised and thermostatted in a manner preventing the shells acquiring velocity.

Each time step in this method requires an optimisation which itself may need several force evaluations as the shells move to minimum force. Thus in principle it is a computationally expensive option. However the fact that a normal time step (Δt) can be used means that its disadvantage per step in comparison with the Adiabatic Shell method is offset by its faster progression through phase space. In terms of time to obtain an equivalent statistical accuracy, it is at much less of a disadvantage.

Chapter 7

Calculating the Pressure

7.1 Introduction

Pressure is an important thermodynamic property of any system and methods to calculate it are an essential part of molecular dynamics. One method, the Clausius Virial Theorem, was shown in section 3.5.3 and instructive though this is, it is not necessarily the best method for calculating pressure in the general system. The method we shall describe in this chapter is based on the thermodynamic approach described in section 3.19.3 , where the all-important equation (3.123)was derived. We reproduce it here:

$$P=-\left\langle\left(\frac{\partial H^N(\vec{r}^N,\vec{p}^N)}{\partial V}\right)_T\right\rangle. \tag{7.1}$$

This relationship shows that the pressure in any system is an ensemble average of the derivative of the system Hamiltonian with respect to volume at constant temperature. The derivative evaluated at any time during a simulation may be regarded as the instantaneous pressure of the system, which includes natural system fluctuations. In this chapter we shall apply equation (7.1) to a variety of molecular systems and see how pressure depends upon the molecular model. To begin with we will consider molecules in which the atoms are not subject to rigid constraints, such as fixed bond lengths or rigid bodies and later move on to molecules where such constraints apply.

In molecular systems free of constraints, the system Hamiltonian can be split into kinetic and potential energy parts in the following way

$$H^N(\vec{r}^N,\vec{p}^N)=K^N(\vec{p}^N)+\Phi^N(\vec{r}^N), \tag{7.2}$$

where the position vectors $\vec{r}^N$ and momentum vectors $\vec{p}^N$ have fully independent components. It is apparent from (7.2) that, in order to calculate the derivative of the Hamiltonian with respect to the volume, as required by (7.1), the volume dependence of each of these terms must be made explicit. To do this we introduce a scaling relation:

$$\vec{r}^N=V^{1/3}\vec{s}^N, \tag{7.3}$$

where V is the system volume and $\vec{s}^N$ is the so-called *scaled position* which is invariant to isotropic changes in the system volume. The time derivative of this relation is

$$\dot{\vec{r}}^N = V^{1/3}\left(\dot{\vec{s}}^N + \frac{\dot{V}}{3V}\vec{s}^N\right), \tag{7.4}$$

so we may write the system kinetic energy as

$$K^N(\dot{\vec{r}}^N) \equiv \frac{1}{2}\sum_{i=1}^{N} m_i V^{2/3}\left(\dot{\vec{s}}_i + \frac{\dot{V}}{3V}\vec{s}_i\right)^2. \tag{7.5}$$

According to Hamiltonian mechanics (see section 2.5) the momentum $\vec{\pi}^N$ *conjugate* to $\vec{s}^N$ is given by differentiation of this with respect to $\dot{\vec{s}}^N$. So for each atom we have

$$\vec{\pi}_i = \frac{\partial K^N(\dot{\vec{r}}^N)}{\partial \dot{\vec{s}}_i} = m_i V^{2/3}\left(\dot{\vec{s}}_i + \frac{\dot{V}}{3V}\vec{s}_i\right). \tag{7.6}$$

From this we see that

$$\vec{\pi}_i = V^{1/3} m_i \dot{\vec{r}}_i = V^{1/3}\vec{p}_i. \tag{7.7}$$

Using relations (7.3) and (7.7) we can now re-write the Hamiltonian (7.2) as

$$H^N(\vec{r}^N, \vec{p}^N) \equiv H^N(V^{1/3}\vec{s}^N, V^{-1/3}\vec{\pi}^N) = K^N(V^{-1/3}\vec{\pi}^N) + \Phi^N(V^{1/3}\vec{s}^N), \tag{7.8}$$

which is a form that permits the differentiation with respect to the volume. This can be worked through as follows

$$\begin{aligned}
\frac{\partial H^N(V^{1/3}\vec{s}^N, V^{-1/3}\vec{\pi}^N)}{\partial V} &= \frac{\partial K^N(V^{-1/3}\vec{\pi}^N)}{\partial V} + \frac{\partial \Phi^N(V^{1/3}\vec{s}^N)}{\partial V}, \\
&= \frac{\partial K^N(\vec{p}^N)}{\partial \vec{p}^N}\cdot\frac{\partial(V^{-1/3}\vec{\pi}^N)}{\partial V} + \frac{\partial \Phi^N(\vec{r}^N)}{\partial \vec{r}^N}\cdot\frac{\partial(V^{1/3}\vec{s}^N)}{\partial V} \\
&= -\frac{1}{3V}\frac{\partial K^N(\vec{p}^N)}{\partial \vec{p}^N}\cdot\vec{p}^N + \frac{1}{3V}\frac{\partial \Phi^N(\vec{r}^N)}{\partial \vec{r}^N}\cdot\vec{r}^N \\
&= -\frac{1}{3V}\left(2\mathrm{K}^N(\vec{p}^N) - \Psi^N(\vec{r}^N)\right),
\end{aligned} \tag{7.9}$$

where the kinetic term is easily obtained and we have simply defined the virial $\Psi^N(\vec{r}^N)$ as

$$\Psi^N(\vec{r}^N) = \frac{\partial \Phi^N(\vec{r}^N)}{\partial \vec{r}^N}\cdot\vec{r}^N. \tag{7.10}$$

Evidently, from (7.9) this can also be written as

$$\Psi^N(\vec{r}^N)=3V\frac{\partial\Phi^N(\vec{r}^N)}{\partial V}, \tag{7.11}$$

which is itself a useful relation.

Thus we have from equations (7.1) and (7.9) a result that is applicable to *all* systems of fully flexible molecules:

$$P=\frac{1}{3V}\left(2\left\langle K^N(\vec{p}^N)\right\rangle-\left\langle\Psi^N(\vec{r}^N)\right\rangle\right). \tag{7.12}$$

The resemblance of this result to that given by the Clausius virial theorem (section 3.5.3) should be noted. Interestingly we have not yet made any comments about boundary conditions, which were important in the Clausius theorem. This is because the derivation is based on the bulk system without reference to such conditions. Nevertheless, most molecular dynamics simulations assume periodic boundaries, so we must ensure when calculating the kinetic energy and virial that the calculations are done in a way that properly incorporates them. In practice, the kinetic energy is not a problem as the momentum is unaffected by periodic boundaries, but for the virial this means we must use the minimum image convention (see section 1.2.5) when calculating inter-atomic distances.

We will now proceed to apply this approach to a variety of molecular systems.

7.2 Simple Atomic System with Pair Forces

The Hamiltonian for such a system is

$$H^N(\vec{r}^N,\vec{p}^N)=\sum_{i=1}^{N}\frac{p_i^2}{2m_i}+\sum_{i=2}^{N}\sum_{j=1}^{i-1}\phi_{ij}(r_{ij}), \tag{7.13}$$

where r_{ij} is the scalar separation between atoms i and j i.e. $\vec{r}_{ij}=\vec{r}_j-\vec{r}_i$. This may be transformed using the scaled variables into

$$H^N(V^{1/3}\vec{s}^N,V^{-1/3}\vec{\pi}^N)=V^{-2/3}\sum_{i=1}^{N}\frac{\pi_i^2}{2m_i}+\sum_{i=2}^{N}\sum_{j=1}^{i-1}\phi_{ij}(V^{1/3}s_{ij}), \tag{7.14}$$

It is easy to show that

$$\frac{\partial H^N}{\partial V}=-\frac{2V^{-5/3}}{3}\sum_{i=1}^{N}\frac{\pi_i^2}{2m_i}+\frac{V^{-2/3}}{3}\sum_{i=2}^{N}\sum_{j=1}^{i-1}\phi'_{ij}(V^{1/3}s_{ij})s_{ij}, \tag{7.15}$$

where $\phi'_{ij}(x)$ is the first derivative of $\phi_{ij}(x)$ with respect to the argument x. Equation (7.15) can be transformed back into the original variables to obtain

$$\frac{\partial H^N}{\partial V} = -\frac{2}{3V}\sum_{i=1}^{N}\frac{p_i^2}{2m_i} + \frac{1}{3V}\sum_{i=2}^{N}\sum_{j=1}^{i-1}\phi'_{ij}(r_{ij})r_{ij}, \tag{7.16}$$

The pressure in this system is therefore:

$$P = \frac{1}{3V}\left\langle 2\sum_{i=1}^{N}\frac{p_i^2}{2\mathrm{m}_i} - \sum_{i=2}^{N}\sum_{j=1}^{i-1}\phi'(r_{ij})r_{ij}\right\rangle. \tag{7.17}$$

This can be written in a form involving the pair forces through the identity

$$\vec{f}_{ij}\cdot\vec{r}_{ij} = -\frac{1}{r_{ij}}\phi'_{ij}(r_{ij})\vec{r}_{ij}\cdot\vec{r}_{ij}, \tag{7.18}$$

then (7.17) becomes

$$P = \frac{1}{3V}\left\langle 2\sum_{i=1}^{N}\frac{p_i^2}{2\mathrm{m}_i} + \sum_{i=2}^{N}\sum_{j=1}^{i-1}\vec{f}_{ij}\cdot\vec{r}_{ij}\right\rangle, \tag{7.19}$$

Which is the well-known expression for the pressure in a system with pair forces only. The term involving the momenta is of course twice the average kinetic energy and the term involving the forces is the average pair force virial $\Psi^N(\vec{r}^N)$ i.e.

$$\Psi^N(\vec{r}^N) = -\sum_{i=2}^{N}\sum_{j=1}^{i-1}\vec{f}_{ij}\cdot\vec{r}_{ij}. \tag{7.20}$$

As remarked in section 3.5.3 , the minimum image convention must be used when calculating this property.

7.3 Rigid Ions and the Ewald Sum

Since the Coulombic force is a pair force, the kinetic contribution to the pressure is the same as in equation (7.19). So we shall focus on calculating the Coulombic virial $\Psi_C^N(\vec{r}^N)$ using relation (7.11). Now, it is well known that in a Coulombic system the virial is simply related to the potential energy via

$$\Psi_C^N(\vec{r}^N) = -\Phi_C^N(\vec{r}^N), \tag{7.21}$$

where

$$\Phi_C^N(\vec{r}^N) = \frac{1}{4\pi\epsilon_0}\sum_{i=2}^{N}\sum_{j=1}^{i-1}\frac{q_i q_j}{r_{ij}}. \tag{7.22}$$

This may be easily demonstrated by inserting the expression for the Coulombic pair force

$$\vec{f}_{ij}=\frac{1}{4\pi\epsilon_0}\frac{q_i q_j}{r_{ij}^3}\vec{r}_{ij}, \tag{7.23}$$

into the formula for the pair force virial (7.20). However, in the case of the Ewald sum method [51], [68], [69], this property of the Coulombic virial is not obvious and some useful points emerge if we attempt a full derivation.

The Coulombic potential energy for a system of point ions is given by the Ewald formula (6.18) which we described in chapter 6 and reproduce here:

$$\begin{aligned}\Phi_C^N=&\frac{1}{2V\epsilon_0}\sum_{\vec{k}\neq\vec{0}}^{\vec{\infty}}\frac{\exp(-k^2/4\alpha^2)}{k^2}\left|\sum_{j=1}^{N}q_j\exp(-i\vec{k}\cdot\vec{r}_j)\right|^2+\\ &\frac{1}{2}\frac{1}{4\pi\epsilon_0}\sum_{\vec{L}=\vec{0}}^{\vec{\infty}}\sum_{j=1}^{N}\sum_{n=1}^{N}{}'\frac{q_j q_n}{|\vec{L}+\vec{r}_j-\vec{r}_n|}\operatorname{erfc}(\alpha|\vec{L}+\vec{r}_j-\vec{r}_n|)-\\ &\frac{\alpha}{4\pi^{3/2}\epsilon_0}\sum_{j=1}^{N}q_j^2.\end{aligned} \tag{7.24}$$

We now need to apply the procedure (7.11). It is apparent at once that the third term right of (7.24), (the so-called *self interaction correction*,) has no inherent volume dependence and so has a zero volume derivative. The second term (middle line) is the r*eal space* term of the Ewald sum, which we shall designate as $\Phi_R^N,$ is clearly a pair potential term of the kind described in section 7.2 and may be dealt with in the same way. We start by defining the inter-atomic separation

$$r_{Ljn}=|\vec{L}+\vec{r}_n-\vec{r}_j|, \tag{7.25}$$

for which the scaling transformation is

$$r_{Ljn}=V^{1/3}s_{Ljn}. \tag{7.26}$$

The derivative of the real space term with respect to the volume is then

$$\frac{\partial\Phi_R^N}{\partial V}=-\frac{1}{3V}\frac{1}{2}\frac{1}{4\pi\epsilon_0}\sum_{\vec{L}=\vec{0}}^{\vec{\infty}}\sum_{j=1}^{N}\sum_{n=1}^{N}{}'q_j q_n\left(\frac{\operatorname{erfc}(\alpha r_{Ljn})}{r_{Ljn}}+\frac{2\alpha}{\sqrt{\pi}}\exp(-\alpha^2 r_{Ljn}^2)\right), \tag{7.27}$$

where we have used the standard derivative

$$\frac{d}{dr}\operatorname{erfc}(\alpha r)=\frac{-2\alpha}{\sqrt{\pi}}\exp(-\alpha^2 r^2). \tag{7.28}$$

We turn now to the first term of (7.24), which is the r*eciprocal space* term, $(\Phi_F^N).$ We need the scaling transformation for the vectors $\vec{r}_j$ which we write as

$$\vec{r}_j = V^{1/3}\vec{s}_j = V^{1/3}\begin{bmatrix} s_j^x \\ s_j^y \\ s_j^z \end{bmatrix}, \tag{7.29}$$

but we also need to specify the volume dependence of the $\vec{k}$ vectors. The definition of the $\vec{k}$ vectors for the general case is more complicated than we need for this demonstration so we shall assume that the simulation cell is cubic. In which case we can write

$$\vec{k} = \frac{2\pi}{V^{1/3}}\begin{bmatrix} l \\ m \\ n \end{bmatrix}, \tag{7.30}$$

where l, m, n are integers. In this form the dependence of $\vec{k}$ on the system volume V is explicit. From this it should be noted that the scalar product $\vec{k}\cdot\vec{r}_j$, which appears in the reciprocal space term is *independent of the volume.* The scalar product is in fact

$$\vec{k}\cdot\vec{r}_j = 2\pi\left(l s_j^x + m s_j^y + n s_j^z\right), \tag{7.31}$$

where it is evident that the volume dependence vanishes. Thus there is no need to calculate the derivative of the term $\left|\sum_{j=1}^{N} q_j \exp\left(-i\vec{k}\cdot\vec{r}_j\right)\right|^2$. (This is true for *all* periodic lattices, regardless of their symmetry.)

We also note from (7.30) that

$$k^2 = \vec{k}\cdot\vec{k} = \frac{4\pi^2}{V^{2/3}}\left(l^2, m^2, n^2\right), \tag{7.32}$$

and so

$$\frac{\partial k^2}{\partial V} = -\frac{2k^2}{3V}. \tag{7.33}$$

Differentiating Φ_F^N is thus quite straightforward and we are led to the formula

$$\frac{\partial \Phi_F^N}{\partial V} = -\frac{1}{3V}\frac{1}{2V\epsilon_0}\sum_{\vec{k}\neq\vec{0}}^{\vec{\infty}}\left(1 - \frac{k^2}{2\alpha^2}\right)\frac{\exp\left(-k^2/4\alpha^2\right)}{k^2}\left|\sum_{j=1}^{N} q_j \exp\left(-i\vec{k}\cdot\vec{r}_j\right)\right|^2, \tag{7.34}$$

for the volume derivative of the reciprocal space term.

We may now combine (7.27) and (7.34) to obtain the virial. However, before doing this we note that a simplification is possible. Extracting the second term from (7.27) we have

$$-\frac{1}{3V}\frac{1}{2}\frac{1}{4\pi\epsilon_0}\left\{\sum_{\vec{L}=\vec{0}}^{\vec{\infty}}\sum_{j=1}^{N}\sum_{n=1}^{N} q_j q_n\left(\frac{2\alpha}{\sqrt{\pi}}\exp\left(-\alpha^2 r_{Ljn}^2\right)\right)-\frac{2\alpha}{\sqrt{\pi}}\sum_{j=1}^{N} q_j^2\right\}, \tag{7.35}$$

where we have cancelled the restriction $n\neq j$ when $\vec{L}=\vec{0}$ in the sum over n and compensated for this by inserting the last term involving the squares of the charges. We now isolate the following term from (7.34)

$$\frac{1}{3V}\frac{1}{2V\epsilon_0}\sum_{\vec{k}\neq\vec{0}}^{\vec{\infty}}\left(\frac{k^2}{2\alpha^2}\right)\frac{\exp\left(-k^2/4\alpha^2\right)}{k^2}\left|\sum_{j=1}^{N} q_j\exp\left(-i\vec{k}\cdot\vec{r}_j\right)\right|^2, \tag{7.36}$$

and note that this is the Fourier series expansion of the first term appearing in (7.35) (multiplied by $-1.$) Thus adding (7.35) and (7.36) leaves only the term

$$+\frac{1}{3V}\frac{\alpha}{4\pi^{3/2}\epsilon_0}\sum_{j=1}^{N} q_j^2, \tag{7.37}$$

in which we recognise the self interaction correction of equation (7.24).

Adding (7.37) to the remaining terms of (7.27) and (7.34) and multiplying the result by $3V$ as required by (7.11) confirms the result (7.21).

This is not quite the complete story however. It is often the case when dealing with molecules rather than free atoms that the interactions between pairs of atoms that are considered to be chemically bonded to each other are entirely described by an explicit bond potential. In these cases it is normal to exclude any non-bonded interaction that might exist between them. This exclusion covers both van der Waals and electrostatic interactions. The simplest strategy for dealing with this is to identify bonded atom pairs during the non-bonded force processing and bypass the calculations (i.e. leave them out). However, when processing the reciprocal space terms of the Ewald sum it is difficult to do this.

There are two ways to deal with this. The first is simply to calculate the contribution of the excluded atom pairs to the overall system Coulombic configuration energy and virial for a second time and subtract these contributions from the original results. (In this case the direct form of Coulombic interaction shown in (7.22) must be used.) This of course means extra computation and, because bond lengths are significantly shorter than non-bonded atomic separations, it sometimes requires subtracting one large number from another, which does not help with accuracy.

The second approach is to modify the calculation of the corresponding term in the real-space part of the Ewald sum. Instead of calculating the term

$$\frac{q_j q_n}{r_{jn}}\operatorname{erfc}\left(\alpha r_{jn}\right), \tag{7.38}$$

we calculate

$$\frac{q_j q_n}{r_{jn}}(\mathrm{erfc}(\alpha r_{jn})-1), \tag{7.39}$$

where the subtraction of the offending term is obvious. The expression (7.39) can of course be written more simply as

$$-\frac{q_j q_n}{r_{jn}}\mathrm{erf}(\alpha r_{jn}), \tag{7.40}$$

where $\mathrm{erf}(x)$ is known as the *error function*, which is a standard function, as is $\mathrm{erfc}(x).$ Using (7.40) in place of (7.38) corrects for the unwanted inclusion of the Coulombic interaction between a bonded pair of atoms. The correction to the virial is of course the negative of this.

7.4 Flexible Polyatomic Molecules (Intra-molecular Potentials)

The Hamiltonian for a system of flexible polyatomic molecules (i.e. without constrained bonds and angles *etc.*) is commonly of the form:

$$H^N(\vec{r}^N,\vec{p}^N)=\sum_{i=1}^{N}\frac{p_i^2}{2\mathrm{m}_i}+\sum_{i=2}^{N}\sum_{j<i}\phi^{nb}(r_{ij})+\sum_{k=1}^{N_b}\phi_k^b(r_{ab})+\sum_{k=1}^{N_t}\phi_k^t(\vec{r}_a,\vec{r}_b,\vec{r}_c)+\sum_{k=1}^{N_f}\phi_k^f(\vec{r}_a,\vec{r}_b,\vec{r}_c,\vec{r}_d) \tag{7.41}$$

Clearly, the kinetic energy, nonbonded terms (ϕ^{nb}) and bonded terms $(\phi_k^b),$ which are pair potentials, may be treated in exactly the same way as the previous case. What is needed in addition is the virial term arising from the three body (ϕ_k^t) and from the four body terms $(\phi_k^f).$ These we consider on a case-by-case basis.

In dealing with three and four body terms, the following cases are common:

- ***Three body potentials of the form***: $\phi_k^t(\hat{r}_{ab},\hat{r}_{bc}),$ (where $\hat{r}_{ab}=\vec{r}_{ab}/r_{ab}$ *etc.*). The potential depends on the relative positions of three atoms, but not on the magnitudes of their separations. It is the angles between the vectors that determine the potential (e.g. the bond angle potential, commonly written as $\phi_k^t(\theta).$) The scaling transformation has no effect on such potentials because the explicit volume dependence vanishes. For this reason these potentials make *no* contribution to the pressure. For example

  $$\phi_k^t(\hat{r}_{ab},\hat{r}_{bc})=A_{abc}(\theta-\theta_0)^2 \tag{7.42}$$

  is a typical harmonic form, where A_{abc} and θ_0 are constants. The angle θ is obtained from the scalar product of vectors $\hat{r}_{ab}$ and $\hat{r}_{bc}$ as

$$\theta = \cos^{-1}(\hat{r}_{ab} \cdot \hat{r}_{bc}). \tag{7.43}$$

This can be expanded and then transformed by the scaling relation to give

$$\theta = \cos^{-1}\left(\frac{\vec{r}_{ab} \cdot \vec{r}_{bc}}{r_{ab} r_{bc}}\right) = \cos^{-1}\left(\frac{\vec{s}_{ab} \cdot \vec{s}_{bc}}{s_{ab} s_{bc}}\right), \tag{7.44}$$

which establishes the fact that θ, and hence the potential $\phi_k^t(\hat{r}_{ab}\hat{r}_{bc})$, is independent of the system volume. The virial must therefore be zero in this case.

- ***Three body potentials of the form***: $\phi_k^t(r_{ab}, r_{bc}, r_{ac})$. The potential depends on the spatial separation between three atoms (e.g. the Axilrod-Teller potential [41]). After the scaling transformation this becomes $\phi_k^t(V^{1/3} s_{ab}, V^{1/3} s_{bc}, V^{1/3} s_{ac})$, which is a form that clearly has a derivative with respect to the volume V and therefore makes a contribution to the pressure. The precise nature of this depends on the analytical form of the derivative. For example, the Axilrod-Teller potential, which is

$$\phi_k^t(r_{ab}, r_{bc}, r_{ac}) = E_0\left(\frac{1 + 3\cos\theta_a \cos\theta_b \cos\theta_c}{(r_{ab} r_{bc} r_{ac})^3}\right) \tag{7.45}$$

becomes

$$\phi_k^t(V^{1/3} s_{ab}, V^{1/3} s_{bc}, V^{1/3} s_{ac}) = E_0\left(\frac{1 + 3\cos\theta_a \cos\theta_b \cos\theta_c}{V^3 (s_{ab} s_{bc} s_{ac})^3}\right) \tag{7.46}$$

the virial for which is then easily shown to be

$$\psi_k^t(r_{ab}, r_{bc}, r_{ac}) = -9\,\phi_k^t(r_{ab}, r_{bc}, r_{ac}). \tag{7.47}$$

- ***Three body potentials of the form***: $\phi_k^t(\vec{r}_{ab}, \vec{r}_{bc}) = \eta_{abc}(\theta) S(r_{ab}, r_{bc})$. The potential has both an angular part $\eta_{abc}(\theta)$ and a "switching" function $S(r_{ab}, r_{bc})$ which zeroes the potential at long range (e.g. the Leslie three body potential [70]). Provided the switching function is analytical, the scaling transformation gives $\eta_{abc}(\theta) S(V^{1/3} s_{ab}, V^{1/3} s_{bc})$, which clearly will contribute to the pressure. For example, we typically have

$$S(r_{ab}, r_{bc}) = C \exp(-(r_{ab} + r_{bc})/\rho), \tag{7.48}$$

where C and ρ are constants. So that under the scaling transformation

$$\phi_k^t(\vec{r}_{ab}, \vec{r}_{bc}) \equiv C\,\eta_{abc}(\theta) \exp\left(-V^{1/3}(s_{ab} + s_{bc})/\rho\right). \tag{7.49}$$

Then the virial is easily shown to be

$$\psi_k^t(\vec{r}_{ab},\vec{r}_{bc}) = -\left(\frac{r_{ab}+r_{bc}}{\rho}\right)\phi_k^t(\vec{r}_{ab},\vec{r}_{bc}). \tag{7.50}$$

- ***Four body potentials of the form***: $\phi_k^f(\hat{r_{ab}},\hat{r_{bc}},\hat{r_{cd}})$. The potential depends on the relative positions of four atoms, but not on the magnitudes of their separations. In other words they are functions of the angles between the displacement vectors. Potentials of this type include dihedral angle potentials [71],[72] and inversion potentials [71], which can be written as $\phi_k^f(\theta)$, where θ is the crucial angle. Here again, the scaling transformation has no effect on these potentials, so they have no virial and *cannot* contribute to the pressure.

  Sometimes however, dihedral angle potentials are defined with an included 1-4 interaction which depends on the separation r_{ad} in our notation. The overall potential is thus $\phi_k^f(\theta)+\phi_k^{1-4}(r_{ad})$ and since $\phi_k^{1-4}(r_{ad})$ becomes $\phi_k^{1-4}(V^{1/3}s_{ad})$ under the scaling transformation, the corresponding virial will make a contribution to the pressure. The analytical form of the derivative determines what the contribution will be. For example, if $\phi_k^{1-4}(r_{ad})$ has the Lennard-Jones form

$$\phi_k^{1-4}(r_{ad}) = 4\epsilon_{ad}\left(\left\{\frac{\sigma_{ab}}{r_{ab}}\right\}^{12} - \left\{\frac{\sigma_{ab}}{r_{ab}}\right\}^{6}\right), \tag{7.51}$$

  then following the scaling transformation, the virial for the *full* dihedral can be shown to be

$$\psi_k^f(\hat{r}_{ab},\hat{r}_{bc},\hat{r}_{cd},r_{ad}) = \psi_k^{1-4}(r_{ad}) = -24\epsilon_{ad}\left(2\left\{\frac{\sigma_{ab}}{r_{ab}}\right\}^{12} - \left\{\frac{\sigma_{ab}}{r_{ab}}\right\}^{6}\right). \tag{7.52}$$

- ***Four body potentials of the "calcite" form***: $\phi_k^f(\vec{r_{ab}},\vec{r_{ac}},\vec{r_{ad}})$. The potential is defined by the arrangement of three atoms around a central atom (*c.f.* the oxygen atoms in the calcite ion CO_3^{--}) and takes the form [39]

$$\phi_k^f(\vec{r_{ab}},\vec{r_{ac}},\vec{r_{ad}}) = Au^2 + Bu^4, \tag{7.53}$$

  where A and B are constants and u is the displacement distance of atom a above a plane defined by the atoms b, c and d, which is given by

$$u = \frac{\vec{r}_{ab}\cdot\vec{r}_{bc}\times\vec{r}_{bd}}{|\vec{r}_{bc}\times\vec{r}_{bd}|}. \tag{7.54}$$

  It is apparent from the definition of u, that it has the dimension of distance. So rather than apply the scaling transformation to the atomic displacement

vectors $\vec{r}_{ab}$ etc. we may apply it directly to the u itself using $u = V^{1/3} s$. Thus

$$\phi_k^f(V^{1/3}\vec{s}_{ab}, V^{1/3}\vec{s}_{ac}, V^{1/3}\vec{s}_{ad}) \equiv AV^{2/3}s^2 + BV^{4/3}s^4, \tag{7.55}$$

from which, via equation (7.11) we obtain the virial as

$$\psi_k^f(\vec{r}_{ab}, \vec{r}_{ac}, \vec{r}_{ad}) = 2Au^2 + 4Bu^4. \tag{7.56}$$

This result shows that sometimes, the viral for an apparently complicated potential can be obtained almost by inspection!

It is worth noting that in all these cases, calculating the required scalar derivative (where it exists), is invariably simpler than attempting to construct the pressure from the vector atomic forces.

7.5 Many-Body Potentials

The potentials in this category do not have a simple two- or three-body dependence of the kind described above. The number of atoms involved in the interaction is in fact defined by the local density, so they might be described as having a *variable n-body* dependence. Examples in this class include the Finnis-Sinclair [45], [48], [73] and Embedded Atom [43], [44] potentials for metals and the Tersoff potential [42] for covalent materials.

- Many-body metal potentials have the general form

$$\Phi_m^N(\vec{r}^N) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j\neq i}^{N}\phi_{ij}^m(r_{ij}) + \sum_{i=1}^{N}F(\rho_i), \tag{7.57}$$

where $\phi_{ij}^m(r_{ij})$ is a pair potential describing the short-range repulsion between atoms, $F(\rho_i)$ is a functional describing the energy of embedding an atom in the bulk density, and ρ_i is the local density, which is defined as

$$\rho_i = \sum_{j\neq i}^{N}\rho_{ij}(r_{ij}). \tag{7.58}$$

A spherical cut-off is usually applied in calculating the density and the pair potentials. The properties of the potential depend on the precise forms of the functions ϕ_{ij}, F and ρ_{ij}. Using the scaling transformation (7.3) with relation (7.11) allows us to write the corresponding virial as

$$\Psi_m^N(\vec{r}^N) = \frac{3V}{2}\sum_{i=1}^{N}\sum_{j\neq i}^{N}\frac{\partial\phi_{ij}^m(V^{1/3}s_{ij})}{\partial V} + 3V\sum_{i=1}^{N}\frac{F(\rho_i)}{\partial\rho_i}\sum_{j\neq i}^{N}\frac{\partial\rho_{ij}(V^{1/3}s_{ij})}{\partial V}, \tag{7.59}$$

which leads to

$$\Psi_m^N(\vec{r}^N)=\frac{1}{2}\sum_{i=1}^{N}\sum_{j\neq i}^{N}\phi_{ij}^{m}{}'(r_{ij})r_{ij}+\sum_{i=1}^{N}F'(\rho_i)\sum_{j\neq i}^{n}\rho'_{ij}(r_{ij})r_{ij}. \tag{7.60}$$

where the dash (') on each function indicates a first derivative with respect to the function argument. These derivatives are usually easy to obtain though a numerical derivative is required for Embedded Atom potentials (which are usually expressed in tabulated form).

- The Tersoff potential has the form

$$\Phi_T^N(\vec{r}^N)=\frac{1}{2}\sum_{i=1}^{N}\sum_{j\neq i}^{N}f_{ij}^{C}(r_{ij})\left(f_{ij}^{R}(r_{ij})-f_{ij}^{A}(r_{ij})\gamma_{ij}(\mathrm{X}_{ij})\right), \tag{7.61}$$

with

$$\gamma_{ij}(\mathrm{X}_{ij})=\chi_{ij}\left(1+\beta_i^{\eta_i}\mathrm{X}_{ij}^{\eta_i}\right)^{-1/2\eta_i} \tag{7.62}$$

and

$$\mathrm{X}_{ij}=\sum_{k\neq i,j}^{N}f_{ik}^{C}(r_{ik})\omega_{ik}\,g(\theta_{ijk}). \tag{7.63}$$

For brevity we simply note here that functions f_{ij}^{R} and f_{ij}^{A} are exponential decay functions and f_{ij}^{C} is a "switching" function that changes continuously from 1 to 0 over a narrow range as r_{ij} increases. All these functions are differentiable. The function $g(\theta_{ijk})$ depends on the angle between vectors $\hat{r}_{ij}$ and $\hat{r}_{ik}$ but not on the inter-atomic separation so it is not necessary to consider it in detail here. In addition χ_{ij}, β_i, η_i, ω_{ik} are constants. Using the scaling transformation (7.3) and the relation (7.11) leads to the virial

$$\begin{aligned}\Psi_T^N(\vec{r}^N)=\frac{1}{2}\sum_{i=1}^{N}\sum_{j\neq i}^{N}\Bigg(\;& f_{ij}^{c}{}'(r_{ij})[f_{ij}^{R}(r_{ij})-f_{ij}^{A}(r_{ij})\gamma_{ij}(\mathrm{X}_{ij})]r_{ij}+\\ & f_{ij}^{c}(r_{ij})[f_{ij}^{R}{}'(r_{ij})-f_{ij}^{A}{}'(r_{ij})\gamma_{ij}(\mathrm{X}_{ij})]r_{ij}-\\ & f_{ij}^{C}(r_{ij})f_{ij}^{A}(r_{ij})\gamma_{ij}{}'(\mathrm{X}_{ij})\sum_{k\neq i,j}^{N}\omega_{ik}g(\theta_{ijk})f_{ik}^{C}{}'(r_{ik})r_{ik}\;\Bigg).\end{aligned} \tag{7.64}$$

Once again, the dash symbol $(')$ indicates the first derivative of the function with respect to the argument. The derivatives of f_{ij}^{R}, f_{ij}^{A} and f_{ij}^{C} are easy to obtain. The derivative $\gamma_{ij}{}'$ is

$$\gamma_{ij}{}'(\mathrm{X}_{ij})=\frac{1}{2}\chi_{ij}\beta_i^{\eta_i}\mathrm{X}_{ij}^{\eta_i-1}\left(1+\beta_i^{\eta_i}\mathrm{X}_{ij}^{\eta_i}\right)^{-(1+1/2\eta_i)}. \tag{7.65}$$

Thus the scaling transformation approach for obtaining the virial for such potentials is not intrinsically difficult. However it does not follow that writing efficient programs to

perform the calculations is trivial. Both the virial and the potential energy imply a triple sum over the atoms, which can be done concurrently.

7.6 Extensible Diatomic Molecules

This is clearly a special case of the fully flexible molecule with pair forces as discussed in section 7.4 . However, we will work through this case in some detail because it reveals some interesting differences from molecules with rigid components.

We start be defining the positions of the atoms a and b of the $i'th$ molecule in the simulation cell:

$$\begin{aligned} \vec{r}_i^{\,a} &= \vec{R}_i + \vec{b}_i^{\,a}, \\ \vec{r}_i^{\,b} &= \vec{R}_i + \vec{b}_i^{\,b}. \end{aligned} \tag{7.66}$$

Where $\vec{R}_i$ locates the molecule's centre of mass and vectors $\vec{b}_i^{\,a}$ and $\vec{b}_i^{\,b}$ respectively define the positions of atoms a and $b,$ with respect to the centre of mass. These are defined by the relations

$$\begin{aligned} \vec{b}_i^{\,a} &= -\frac{\mu_i}{m_i^a}\vec{b}_i, \\ \vec{b}_i^{\,b} &= \frac{\mu_i}{m_i^b}\vec{b}_i, \end{aligned} \tag{7.67}$$

where $\vec{b}_i$ is the bond vector (directed from atom *a* to *b*). Masses m_i^a and m_i^b are the atomic masses of a and b respectively and μ_i is the molecule's reduced mass, which is defined as

$$\mu_i = \frac{m_i^a m_i^b}{m_i^a + m_i^b}. \tag{7.68}$$

The mass of the molecule as a whole will be designated as $M_i.$

Substituting (7.67) into (7.66) and differentiating with respect to time gives the velocities $\dot{\vec{r}}_i^{\,a}$ and $\dot{\vec{r}}_i^{\,b}$ as

$$\begin{aligned} \dot{\vec{r}}_i^{\,a} &= \dot{\vec{R}}_i - \frac{\mu_i}{m_i^a}\dot{\vec{b}}_i \\ \dot{\vec{r}}_i^{\,b} &= \dot{\vec{R}}_i + \frac{\mu_i}{m_i^b}\dot{\vec{b}}_i \end{aligned} \tag{7.69}$$

From which the system kinetic energy can be written as

$$K^N(\dot{\vec{R}}^{N_m},\dot{\vec{b}}^{N_m})=\frac{1}{2}\sum_{i=1}^{N_m}\left(M_i\dot{R}_i^2+\mu_i\dot{b}_i^2\right), \tag{7.70}$$

where N_m is the number of molecules in the system. Following Hamilton's definition (section 2.5 , equation (2.17)), we may obtain the conjugate momenta

$$\begin{aligned}\vec{P}_i&=M_i\dot{\vec{R}}_i,\\ \vec{q}_i&=\mu_i\dot{\vec{b}}_i.\end{aligned} \tag{7.71}$$

Vectors $\vec{R}_i$, $\vec{P}_i$, $\vec{b}_i$ and $\vec{q}_i$, are thus the *generalised coordinates* for the system and the Hamiltonian is a function of these i.e.

$$H^N(\vec{\Gamma}^N)\equiv H^N\left(\vec{R}^{N_m},\vec{b}^{N_m},\vec{P}^{N_m},\vec{q}^{N_m}\right), \tag{7.72}$$

The appropriate scaled coordinates for such a system are

$$\begin{aligned}\vec{R}_i=V^{1/3}\vec{S}_i,\quad \vec{\Pi}_i=V^{2/3}M_i(\dot{\vec{S}}_i+\vec{S}_i\dot{V}/3V),\quad \vec{P}_i=V^{-1/3}\vec{\Pi}_i\\ \vec{b}_i=V^{1/3}\vec{d}_i,\quad \vec{\lambda}_i=V^{2/3}\mu_i(\dot{\vec{d}}_i+\vec{d}_i\dot{V}/3V),\quad \vec{q}_i=V^{-1/3}\vec{\lambda}_i\end{aligned} \tag{7.73}$$

According to our procedures the pressure is therefore given by

$$P=-\left\langle\left(\frac{\partial}{\partial V}H^N(V^{1/3}\vec{S}^{N_m},V^{1/3}\vec{d}^{N_m},V^{-1/3}\vec{\Pi}^{N_m},V^{-1/3}\vec{\lambda}^{N_m})\right)_T\right\rangle, \tag{7.74}$$

where, assuming atom-atom pair forces, the Hamiltonian is

$$\begin{aligned}H^N\left(V^{1/3}\vec{S}^{N_m},V^{1/3}\vec{d}^{N_m},V^{-1/3}\vec{\Pi}^{N_m},V^{-1/3}\vec{\lambda}^{N_m}\right)=V^{-2/3}\sum_{i=1}^{N_m}\frac{\Pi_i^2}{2M_i}+V^{-2/3}\sum_{i=1}^{N_m}\frac{\lambda_i^2}{2\mu_i}\\ +\sum_{i=2}^{N_m}\sum_{j=1}^{i-1}\sum_{a=1}^{2}\sum_{b=1}^{2}\phi_{ij}^{ab}(V^{1/3}s_{ij}^{ab})\\ +\sum_{i=1}^{N_m}\phi_i^{ab}(V^{1/3}d_i)\end{aligned} \tag{7.75}$$

with

$$\vec{s}_{ij}^{ab}=V^{-1/3}\vec{r}_{ij}^{ab}=V^{-1/3}\left(\vec{R}_j-\vec{R}_i+\vec{b}_j^b-\vec{b}_i^a\right), \tag{7.76}$$

i.e. $\vec{r}_{ij}^{ab}$ is the separation vector between atom a on molecule i and atom b on molecule j and $\vec{s}_{ij}^{ab}$ is its scaled equivalent. ϕ_{ij}^{ab} is the non-bonded interaction between atoms on different molecules, while ϕ_i^{ab} is the potential of the diatomic bond. (Incidentally, it is worth noting that the structure of the quadruple sum involving the ϕ_{ij}^{ab} potentials is such that the two atoms on the same molecule cannot interact via non-bonding potentials.) Differentiating (7.75) with respect to the volume leads to

$$P=\frac{2}{3\mathrm{V}}\left\langle\left(V^{-2/3}\sum_{i=1}^{N_m}\frac{\Pi_i^2}{2M_i}+V^{-2/3}\sum_{i=1}^{N_m}\frac{\lambda_i^2}{2}\mu_i\right)\right\rangle$$
$$-\frac{1}{3\mathrm{V}}\left\langle\left(V^{1/3}\sum_{i=2}^{N_m}\sum_{j=1}^{i-1}\sum_{a=1}^{2}\sum_{b=1}^{2}\phi'^{ab}_{ij}(r^{ab}_{ij})s^{ab}_{ij}+V^{1/3}\sum_{i=1}^{N_m}\phi'^{b}_{i}(b_i)d_i\right)\right\rangle, \tag{7.77}$$

where the dash $(')$ indicates a derivative with respect to the argument. This reduces further to

$$P=\frac{1}{3\mathrm{V}}\left\langle 2\sum_{i=1}^{N_m}\left(\frac{P_i^2}{2M_i}+\frac{q_i^2}{2\mu_i}\right)+\sum_{i=2}^{N_m}\sum_{j=1}^{i-1}\sum_{a=1}^{2}\sum_{b=1}^{2}\vec{f}^{ab}_{ij}\cdot\vec{r}^{ab}_{ij}+\sum_{i=1}^{N_m}\vec{f}^{ab}_{i}\cdot\vec{b}_i\right\rangle, \tag{7.78}$$

where $\vec{f}^{ab}_{ij}$ and $\vec{f}^{ab}_{i}$ are respectively the non-bonded forces between atoms on different molecules and the force due to the bond in the diatomic molecule.

This result is consistent with the general case described in section 7.4 . Note the presence of the last term right of (7.78) and also the contribution of the momentum $\vec{q}_i$ to the kinetic energy. Both arise from the existence of the bond in the diatomic molecules. It is also noticeable that the kinetic term must also account for both the molecular vibrational and rotational energy, since $\vec{P}_i$ accounts entirely for the translational momentum of the whole molecule. This suggests that the rotational energy contributes to the pressure, but this is not the case, as we will show later.

7.7 Rigid Diatomic Molecules

This system is similar to that in the preceding section, except that the bond length in each molecule is held fixed and there is no vibrational motion within the molecules. So once again the vector $\vec{R}_i$ defines the location of the $i'th$ molecule's centre of mass and the vector $\vec{b}_i$ defines the bond direction and magnitude. Therefore equations (7.66) through to (7.70) still apply in this case. However we now choose to define the bond vector $\vec{b}_i$ in terms of the constant bond length b_i and two orientation angles[21] (ϕ_i,θ_i), which function as generalised coordinates. The bond vector and its time derivative (which is required to calculate the kinetic energy) are thus given by

$$\vec{b}_i=b_i\begin{bmatrix}\cos\theta_i\cos\phi_i\\ \cos\theta_i\sin\phi_i\\ \sin\theta_i\end{bmatrix},\qquad \dot{\vec{b}}_i=b_i\begin{bmatrix}-\sin\theta_i\cos\phi_i\dot{\theta}_i-\cos\theta_i\sin\phi_i\dot{\phi}_i\\ -\sin\theta_i\sin\phi_i\dot{\theta}_i+\cos\theta_i\cos\phi_i\dot{\phi}_i\\ \cos\theta_i\dot{\theta}_i\end{bmatrix} \tag{7.79}$$

The momenta conjugate to $\vec{R}_i$, ϕ_i and θ_i (denoted $\vec{P}_i$, κ_i and λ_i respectively) are obtained from the kinetic energy (7.70) by the usual differentiation. The result is

21 Two angles are sufficient because the molecule is linear.

$$\begin{aligned}\vec{P}_i &= M_i \dot{\vec{R}}_i ,\\ \kappa_i &= \mu_i b_i^2 \cos^2(\theta_i)\dot{\phi} ,\\ \lambda_i &= \mu_i b_i^2 \dot{\theta} \quad .\end{aligned} \tag{7.80}$$

The Hamiltonian for a system composed of N_m molecules may be written as

$$H^N\left(\vec{R}^{N_m}, \phi^{N_m}, \theta^{N_m}, \vec{P}^{N_m}, \kappa^{N_m}, \lambda^{N_m}\right), \tag{7.81}$$

and to obtain the pressure in this system we need to differentiate the Hamiltonian with respect to the system volume. To do that we must first apply the familiar scaling transformation to the generalised coordinates and conjugate momenta:

$$\begin{aligned}\vec{R}_i &= V^{1/3}\vec{S}_i ,\\ \vec{\Pi}_i &= V^{2/3} M_i \left(\dot{\vec{S}}_i + \vec{S}_i \dot{V}/3V\right),\\ \vec{P}_i &= V^{-1/3}\vec{\Pi}_i .\end{aligned} \tag{7.82}$$

The pressure is thus given by

$$P = -\left\langle \left(\frac{\partial}{\partial V} H^N\left(V^{1/3}\vec{S}^{N_m}, \phi^{N_m}, \theta^{N_m}, V^{-1/3}\vec{\Pi}^{N_m}, \kappa^{N_m}, \lambda^{N_m}\right)\right)_T \right\rangle . \tag{7.83}$$

It should be clear that there can be no scaling of the angular coordinates nor their conjugate momenta. This follows from the fact that the bond vector $\vec{b}_i$ has a fixed length which, like its orientation, is unaffected by rescaling the volume.

Assuming all the interactions are atom-atom pair interactions we obtain

$$P = \frac{1}{3V}\left\langle 2\sum_{i=1}^{N_m}\frac{P_i^2}{2M_i} - \sum_{i=2}^{N_m}\sum_{j=1}^{i-1}\sum_{a=1}^{2}\sum_{b=1}^{2}\frac{1}{r_{ij}^{ab}}\phi'^{ab}_{ij}\left(r_{ij}^{ab}\right)\vec{r}_{ij}^{ab}\cdot\vec{R}_{ij}\right\rangle, \tag{7.84}$$

where

$$\vec{R}_{ij} = \vec{R}_j - \vec{R}_i \quad \text{and} \quad \vec{r}_{ij}^{ab} = \vec{R}_j - \vec{R}_i + \vec{b}_j^b - \vec{b}_i^a \tag{7.85}$$

are the vector displacements between molecular centres of mass and the vector displacement between individual atoms on *different* molecules. In (7.84) we recognise the pair forces acting between the atoms as

$$\vec{f}_{ij}^{ab} = -\frac{1}{r_{ij}^{ab}}\phi'^{ab}_{ij}\left(r_{ij}^{ab}\right)\vec{r}_{ij}^{ab} . \tag{7.86}$$

We also note that relations (7.85) can be rearranged into the form:

$$\vec{R}_{ij} = \vec{r}_{ij}^{ab} - \left(\vec{b}_j^b - \vec{b}_i^a\right). \tag{7.87}$$

So equation (7.84) may be finally rewritten as

$$P=\frac{1}{3V}\left\langle 2\sum_{i=1}^{N_m}\frac{P_i^2}{2M_i}+\sum_{i=2}^{N_m}\sum_{j=1}^{i-1}\sum_{a=1}^{2}\sum_{b=1}^{2}\vec{f}_{ij}^{ab}\cdot\vec{r}_{ij}^{ab}-\sum_{i=1}^{N_m}\sum_{a=1}^{2}\vec{f}_i^a\cdot\vec{b}_i^a\right\rangle, \tag{7.88}$$

where $\vec{f}_i^a$ is the net force acting on atom a of molecule i. The differences between equation (7.88) which applies to rigid diatomic molecules and equation (7.78) which applies to extensible diatomic molecules merit some comments.

Firstly, looking at the kinetic energy contributions to the pressure, we see that in (7.88) there is no contribution to the pressure from the *internal* motion of the molecules. The absence of a vibrational contribution is expected, but there is no contribution from the rotational energy either. However the extensible molecule equation (7.78) undoubtedly contains both vibrational *and* rotational energy. To explain this mystery we must look at the last term appearing on the right of (7.88).

We must first make clear that this term does not have the same meaning as the last term on the right of (7.78). In (7.78) the last term is clearly a virial contribution from the extensible bond potential and in that sense is a normal *intra-molecular* part of the virial. The last term of (7.88) however concerns the interactions of each molecule with every other molecule and is not intra-molecular in origin. In fact in this case the term is a manifestation of the constraint force which appears as an additional term to satisfy the requirement that each molecule maintains the specified bond length. Indeed the construction of this term shows clearly that it opposes the forces that act on the atoms along the bond direction.

Examination of the constraint force however shows that it is entirely dependent on inter-atomic forces. It is not therefore the complete constraint force, which should also include contributions to resist the centrifugal forces due to molecular rotation. The absence of rotational constraint forces is therefore consistent with the absence of a rotational kinetic energy term in equation (7.88), as is demonstrated by the following argument.

If a diatomic molecule i spins with an *instantaneous* angular velocity ω_i, the rotational kinetic energy of the molecule is given by

$$K_i^R=\frac{1}{2}m_i^a(\omega_i b_i^a)^2+\frac{1}{2}m_i^b(\omega_i b_i^b)^2. \tag{7.89}$$

The centripetal (constraint) force acting on each atom is

$$\vec{g}_i^a=-m_i^a\omega_i^2\vec{b}_i^a \tag{7.90}$$

and the contribution to the virial due to these forces in one molecule is

$$\psi_i^R=-\vec{g}_i^a\cdot\vec{b}_i^a-\vec{g}_i^b\cdot\vec{b}_i^b=m_i^a(\omega_i b_i^a)^2+m_i^b(\omega_i b_i^b)^2 \tag{7.91}$$

From which it is apparent that

$$2K_i^R - \psi_i^R = 0. \tag{7.92}$$

So we see that the centripetal virial, ψ_i^R, cancels the rotational kinetic energy $2K_i^R$ for each molecule. This is why neither term appears in the pressure equation (7.88). Furthermore, we can say that, since this cancellation is true at each instant, rotational motion does not contribute in the extensible case either, however in equation (7.78) we are unaware of this because both the rotational kinetic energy and the corresponding virial correction are implicitly included.

These observations show that when constraints are used in the molecular model, the constraint forces make a significant contribution to the pressure. The forces may or may not include centripetal contributions, depending on the method used to solve the dynamics, and this makes a difference to how the pressure is calculated. For example, in rigid body dynamics, the centripetal contributions are *not* included and so the rotational kinetic energy of the molecules is not included either. In the dynamics of fully flexible molecules, the total kinetic energy includes the rotational energy, though this may not be explicitly represented and the centripetal forces never make any explicit appearance, but nevertheless they exist and the two automatically compensate for each other in the pressure calculation. In the SHAKE algorithm, as applied to bond constraints, a similar compensation occurs, provided we remember to include the constraint forces (which automatically include centripetal contributions) when calculating the pressure.

7.8 Rigid Polyatomic Molecules

Rigid polyatomic molecules resemble rigid diatomic molecules in many respects. The main difference is that the orientation now requires three Euler angles to specify (see section 2.8). We shall represent these these collectively as $\Omega_i \equiv (\phi_i, \theta_i, \psi_i)$. The location of an atom a belonging to molecule i in space is $\vec{r}_i^a$ which can be expanded as

$$\vec{r}_i^a = \vec{R}_i + \vec{d}_i^a, \tag{7.93}$$

where $\vec{d}_i^a$ is the displacement of the atom from the centre of mass $\vec{R}_i$. The centre of mass is given by

$$\vec{R}_i = \frac{1}{M_i} \sum_{a=1}^{\eta_i} m_i^a \vec{r}_i^a, \qquad \text{with} \qquad M_i = \sum_{a=1}^{\eta_i} m_i^a \tag{7.94}$$

and η_i is the number of atoms in the $i'th$ molecule. We write the Hamiltonian for a system of N_m such molecules as

$$H^N\left(\vec{R}^{N_m}, \Omega^{N_m}, \vec{P}^{N_m}, \Lambda^{N_m}\right), \tag{7.95}$$

where $\vec{P}_i$ and $\Lambda_i \equiv \left(\lambda_i^\phi, \lambda_i^\theta, \lambda_i^\psi\right)$ are the momenta conjugate to $\vec{R}_i$ and Ω_i. $\vec{P}_i$ is the familiar centre of mass momentum:

$$\vec{P}_i = M_i \dot{\vec{R}}_i \tag{7.96}$$

and Λ_i is of course the molecular angular momentum expressed in terms of $\dot{\Omega}_i \equiv \left(\dot{\phi}_i, \dot{\theta}_i, \dot{\psi}_i\right)$ but we do not need to specify it further here, as it is not affected by the scaling transformation and has no further role to play in the derivation of the pressure. The scaling relation applies only to the centre of mass position vector and translational momentum:

$$\begin{aligned} \vec{R}_i &= V^{1/3} \vec{S}_i, \\ \vec{\Pi}_i &= V^{2/3} M_i \left(\dot{\vec{S}}_i + \vec{S}_i \dot{V} / 3V \right), \\ \vec{P}_i &= V^{-1/3} \vec{\Pi}_i. \end{aligned} \tag{7.97}$$

The system pressure is given by

$$P = -\left\langle \left(\frac{\partial}{\partial V} H^N \left(V^{1/3} \vec{S}^{N_m}, \Omega^{N_m}, V^{-1/3} \vec{\Pi}^{N_m}, \Lambda^{N_m} \right) \right)_T \right\rangle \tag{7.98}$$

Assuming all the forces arise from atom-atom pair interactions, we define

$$\vec{r}_{ij}^{ab} = \vec{R}_{ij} + \vec{d}_{ij}^{ab} \quad \text{with} \quad \vec{R}_{ij} = \vec{R}_j - \vec{R}_i \quad \text{and} \quad \vec{d}_{ij}^{ab} = \vec{d}_j^b - \vec{d}_i^a \tag{7.99}$$

and we note that

$$r_{ij}^{ab} = \left| V^{1/3} \vec{S}_{ij} + \vec{d}_{ij}^{ab} \right| \tag{7.100}$$

where $\vec{S}_{ij}$ is the scaled centre of mass displacement vector from molecule i to molecule j.

The Hamiltonian is then

$$\begin{aligned} H^N \left(\vec{V^{1/3}} S^{N_m}, \Omega^{N_m}, V^{-1/3} \vec{\Pi}^{N_m}, \Lambda^{N_m} \right) = V^{-2/3} \sum_{i=1}^{N_m} \frac{\Pi_i^2}{2M_i} + \frac{1}{2} \sum_{i=1^{N_m}} \vec{\omega}_i \cdot \boldsymbol{I}_i \cdot \vec{\omega}_i \\ + \sum_{i=1}^{N_m - 1} \sum_{j>i}^{N_m} \sum_{a=1}^{\eta_i} \sum_{b=1}^{\eta_j} \phi_{ij}^{ab} \left(\left| V^{1/3} \vec{R}_{ij} + \vec{d}_{ij}^{ab} \right| \right), \end{aligned} \tag{7.101}$$

where $\vec{\omega}_i$ is the angular velocity of the $i'th$ molecule. (Note that, according to this Hamiltonian, atoms on the same molecule do not interact.)

The pressure is therefore

$$P = \frac{1}{3V} \left\langle 2 \sum_{i=1}^{N_m} \frac{P_i^2}{2M_i} - \sum_{i=1}^{N_m - 1} \sum_{j>i}^{N_m} \sum_{a=1}^{\eta_i} \sum_{b=1}^{\eta_j} \frac{1}{r_{ij}^{ab}} \phi'^{ab}_{ij} \left(r_{ij}^{ab} \right) \vec{r}_{ij}^{ab} \cdot \vec{R}_{ij} \right\rangle \tag{7.102}$$

which, using the relation

$$\vec{R}_{ij}^{ab} = \vec{r}_{ij}^{ab} - (\vec{d}_j^b - \vec{d}_i^a), \tag{7.103}$$

may be written as

$$P = \frac{1}{3V}\left\langle 2\sum_{i=1}^{N_m}\frac{P_i^2}{2M_i} + \sum_{i=1}^{N_m-1}\sum_{j>i}^{N_m}\sum_{a=1}^{n_i}\sum_{b=1}^{n_j}\vec{f}_{ij}^{ab}\cdot\vec{r}_{ij}^{ab} - \sum_{i=1}^{N_m}\sum_{a=1}^{n_m}\vec{f}_i^a\cdot\vec{d}_i^a\right\rangle \tag{7.104}$$

where $\vec{f}_i^a$ is the net force acting on atom a of molecule i.

Incidentally, the last term on the right of (7.104) (and also (7.88) above) has a zero ensemble average in systems where there is unrestricted rotation of the molecules, since then there is no correlation between the vectors $\vec{d}_i^a$ and $\vec{f}_i^a$. However, in systems where rotation is restricted, this may not be the case, so it is safer in general to include the term in any simulations. The last term is often called the *atomic-to-molecular viral correction* or the *rigid molecule virial correction*.

7.9 Rigid Molecules and the Ewald Sum

This represents a hybrid of the systems described in sections 7.3 and 7.8 above. In principle this should not be more difficult than for any rigid molecule system, but unfortunately the requirement that atoms on the same molecule do not interact is problematic. This problem is similar to that arising from charged atoms connected via chemical bonds, which was described in section 7.3 , but here it is expanded to include all atoms in the molecule. We will present a more general solution here.

The configuration energy for a system of N_m rigid molecules with η_j (usually fractionally charged) atoms on a molecule j is

$$\begin{aligned}\Phi_C^N = {} & \frac{1}{2V\epsilon_0}\sum_{\vec{k}\neq\vec{0}}^{\vec{\infty}}\frac{\exp(-k^2/4\alpha^2)}{k^2}\left|\sum_{j=1}^{N_m}\sum_{a=1}^{\eta_j}q_j^a\exp(-i\vec{k}\cdot\vec{r}_j^a)\right|^2 + \\ & \frac{1}{2}\frac{1}{4\pi\epsilon_0}\sum_{\vec{L}=\vec{0}}^{\vec{\infty}}\sum_{j=1}^{N_m}\sum_{n=1}^{N_m}{}'\sum_{a=1}^{\eta_j}\sum_{b=1}^{\eta_n}\frac{q_j^a q_n^b}{\vec{r}_{Ljn}^{ab}}\operatorname{erfc}(\alpha\vec{r}_{Ljn}^{ab}) - \\ & \frac{\alpha}{4\pi^{3/2}\epsilon_0}\sum_{j=1}^{N_m}\sum_{a=1}^{\eta_j}q_j^{a\,2} - \frac{1}{2}\frac{1}{4\pi\epsilon_0}\sum_{j=1}^{N_m}\sum_{a=1}^{\eta_j}\sum_{b\neq j}^{\eta_j}\frac{q_j^a q_j^b}{r_{jj}^{ab}}\operatorname{erf}(\alpha r_{jj}^{ab}).\end{aligned} \tag{7.105}$$

where

$$r_{Ljn}^{ab} = \left|\vec{L} + \vec{r}_n^b - \vec{r}_j^a\right|, \qquad \vec{r}_j^a = \vec{R}_j + \vec{d}_j^a \text{ and } r_{jj}^{ab} = \left|\vec{r}_j^b - \vec{r}_j^a\right| = \left|\vec{d}_j^b - \vec{d}_j^a\right|. \tag{7.106}$$

But for the presence of the last term on the right of (7.105), which is the correction for *intra-molecular* interactions (see equation (7.40)). The derivation of the virial from this starting point parallels what was done in sections 7.3 and 7.8 and will not be repeated here. Instead we shall point out the features unique to this case.

When the scaling relation (7.97) appropriate to rigid molecules is inserted into (7.105)

and the differentiation with respect to volume performed, it is found that third and fourth terms on the right of (7.105) have zero derivatives. This is because the third term is a constant and the fourth because there can be no volume scaling of a rigid body (i.e. the positions $\vec{d}_i^a$ *etc*. are volume independent.)

Differentiation of the reciprocal space term resembles that given in section 7.3 but with one important difference. The exponential terms $\exp\left(-i\vec{k}\cdot\vec{r}_j^a\right)$ are now volume dependent on account of the term term $\vec{k}\cdot\vec{d}_j^a$ which arises from the definition of $\vec{r}_j^a$ given in (7.93). These provide extra terms in the derivative (compared with equation (7.34) previously):

$$\begin{aligned}\frac{\partial\Phi_F^N}{\partial V}=-\frac{1}{3\mathrm{V}}\frac{1}{2V\epsilon_0}\sum_{\vec{k}\neq\vec{0}}^{\vec{\infty}}\left(1-\frac{k^2}{2\alpha^2}\right)\frac{\exp(-k^2/4\alpha^2)}{k^2}\left|\sum_{j=1}^{N_m}\sum_{a=1}^{\eta_j}q_j^a\exp(-i\vec{k}\cdot\vec{r}_j^a)\right|^2 \\ +\frac{1}{3V}\sum_{j=1}^{N_m}\sum_{a=1}^{\eta_j}\vec{g}_j^a\cdot\vec{d}_j^a,\end{aligned} \tag{7.107}$$

Where $\vec{g}_j^a$ represents the Fourier component of the force on atom a of molecule j due to all the other molecules in the system. Its explicit form is

$$\vec{g}_j^a=-\frac{q_j^a}{V\epsilon_0}\sum_{\vec{k}\neq\vec{0}}^{\infty}\frac{\exp(-k^2/4\alpha^2)}{k^2}\Im\left(\left\{\sum_{n=1}^{N_m}\sum_{b=1}^{\eta_n}q_n^b\exp\left(i\vec{k}\cdot\vec{r}_n^b\right)\right\}\exp\left(-i\vec{k}\cdot\vec{r}_j^a\right)\right)\vec{k}\,, \tag{7.108}$$

where $\Im(C)$ is the imaginary part of a complex number C. Notice that this force includes contributions from within the same molecule, since the sums over n and b do not exclude this possibility.

Differentiation of the Real Space term of (7.105) follows straightforwardly from sections 7.3 and 7.8 above, with no additional complications.

$$\begin{aligned}\frac{\partial\Phi_R^N}{\partial V}=-\frac{1}{3\mathrm{V}}\frac{1}{2}\frac{1}{4\pi\epsilon_0}\sum_{\vec{L}=\vec{0}}^{\vec{\infty}}\sum_{j=1}^{N_m}\sum_{n=1}^{N_m}{}'\sum_{a=1}^{\eta_j}\sum_{b=1}^{\eta_n}q_j^aq_n^b\left(\frac{\mathrm{erfc}(\alpha r_{Ljn}^{ab})}{r_{Ljn}^{ab}}+\frac{2\alpha}{\sqrt{\pi}}\exp\left(-\alpha^2r_{Ljn}^{ab\,2}\right)\right) \\ +\frac{1}{3V}\sum_{j=1}^{N_m}\sum_{a=1}^{\eta_j}\vec{h}_j^a\cdot\vec{d}_j^a,\end{aligned} \tag{7.109}$$

where

$$\vec{h}_j^a=-\frac{q_j^a}{4\pi\epsilon_0}\sum_{\vec{L}=\vec{0}}^{\vec{\infty}}\sum_{n=1}^{N_m}{}'\sum_{b=1}^{\eta_n}\frac{q_n^b}{r_{Ljn}^{ab\,2}}\left(\frac{\mathrm{erfc}(\alpha r_{Ljn}^{ab})}{r_{Ljn}^{ab}}+\frac{2\alpha}{\sqrt{\pi}}\exp\left(-\alpha^2r_{Ljn}^{ab\,2}\right)\right)\vec{r}_{Ljn}^{ab} \tag{7.110}$$

is the Real Space contribution to the force on atom a of molecule j due to all other molecules n in the system.

We could now add the reciprocal space (7.107) and real space (7.109) components to obtain the virial. However, before doing this we first separate out the Gaussian terms

from (7.109) and add to it the following expression (on the understanding that it will be removed later):

$$-\frac{1}{3V}\frac{1}{2}\frac{1}{4\pi\epsilon_0}\sum_{j=1}^{N_m}\sum_{a=1}^{\eta_j}\sum_{b=1}^{\eta_j} q_j^a q_j^b \frac{2\alpha}{\sqrt{\pi}}\exp\left(-\alpha^2 r_{jj}^{ab\ 2}\right). \tag{7.111}$$

This represents the *intra-molecular* Gaussian terms that are excluded from (7.109). The extended set of Gaussian terms is therefore

$$-\frac{1}{3V}\frac{1}{2}\frac{1}{4\pi\epsilon_0}\sum_{\vec{L}=\vec{0}}^{\vec{\infty}}\sum_{j=1}^{N_m}\sum_{n=1}^{N_m}{}'\sum_{a=1}^{\eta_j}\sum_{b=1}^{\eta_n} q_j^a q_n^b\left(\frac{2\alpha}{\sqrt{\pi}}\right)\exp\left(-\alpha^2 r_{Ljn}^{ab\ 2}\right). \tag{7.112}$$

Next we extract the following terms from (7.107)

$$\frac{1}{3V}\frac{1}{2V\epsilon_0}\frac{1}{2\alpha^2}\sum_{\vec{k}\neq\vec{0}}^{\vec{\infty}}\exp\left(-k^2/4\alpha^2\right)\left|\sum_{j=1}^{N_m}\sum_{a=1}^{\eta_j} q_j^a\exp\left(-i\vec{k}\cdot\vec{r}_j^{\,a}\right)\right|^2 \tag{7.113}$$

and note that these collectively cancel the terms in (7.112) since the two expressions are equivalent via Fourier synthesis. Thus, when (7.107) and (7.109) are combined, we may use this cancellation to give the simplified result

$$\begin{aligned}\Psi_C^N = &-\frac{1}{2V\epsilon_0}\sum_{\vec{k}\neq\vec{0}}^{\vec{\infty}}\frac{\exp\left(-k^2/4\alpha^2\right)}{k^2}\left|\sum_{j=1}^{N_m}\sum_{a=1}^{\eta_j} q_j^a\exp\left(-i\vec{k}\cdot\vec{r}_j^{\,a}\right)\right|^2 - \\ &\frac{1}{2}\frac{1}{4\pi\epsilon_0}\sum_{\vec{L}=\vec{0}}^{\vec{\infty}}\sum_{j=1}^{N_m}\sum_{n=1}^{N_m}{}'\sum_{a=1}^{\eta_j}\sum_{b=1}^{\eta_n}\frac{q_j^a q_n^b}{\vec{r}_{Ljn}^{\,ab}}\operatorname{erfc}\left(\alpha\vec{r}_{Ljn}^{\,ab}\right)+\frac{\alpha}{4\pi^{3/2}\epsilon_0}\sum_{j=1}^{N_m}\sum_{a=1}^{\eta_j} q_j^{a\,2}+ \\ &\frac{1}{2}\frac{1}{4\pi\epsilon_0}\sum_{j=1}^{N_m}\sum_{a=1}^{\eta_j}\sum_{b\neq a}^{\eta_j} q_j^a q_j^b\frac{2\alpha}{\sqrt{\pi}}\exp\left(\alpha^2 r_{jj}^{ab2}\right)+\sum_{j=1}^{N_m}\sum_{a=1}^{\eta_j}\left(\vec{g}_j^{\,a}+\vec{h}_j^{\,a}\right)\cdot\vec{d}_j^{\,a}.\end{aligned} \tag{7.114}$$

Note that in (7.114) we have corrected for adding expression (7.111) by subtraction of the same, though we have also split off the $a=b$ terms from (7.111) to give term 3 (the familiar self interaction) and term 4 (evidently intra-molecular corrections) on the right hand side. Both these terms are constants and need to be evaluated only once during a simulation.

An even simpler form can be obtained for (7.114) if the intra-molecular contributions to $\vec{g}_j^{\,a}\cdot\vec{d}_j^{\,a}$ are separated out. Converting these to a Real Space representation we may write them as

$$+\frac{1}{2}\frac{1}{4\pi\epsilon_0}\sum_{j=1}^{N_m}\sum_{a=1}^{\eta_j}\sum_{b\neq a}^{\eta_j} q_j^a q_j^b\left(\frac{\operatorname{erf}\left(\alpha r_{jj}^{ab}\right)}{r_{jj}^{ab}}-\frac{2\alpha}{\sqrt{\pi}}\exp\left(-\alpha^2 r_{jj}^{ab\ 2}\right)\right). \tag{7.115}$$

Substituting this for the intra-molecular parts of $\vec{g}_j^{\,a}\cdot\vec{d}_j^{\,a}$ in (7.114) allows us to write

$$\Psi_C^N = -\frac{1}{2V\epsilon_0}\sum_{\vec{k}\neq\vec{0}}^{\vec{\infty}}\frac{\exp\left(-k^2/4\alpha^2\right)}{k^2}\left|\sum_{j=1}^{N_m}\sum_{a=1}^{\eta_j} q_j^a \exp\left(-i\vec{k}\cdot\vec{r}_j^a\right)\right|^2 -$$
$$\frac{1}{2}\frac{1}{4\pi\epsilon_0}\sum_{\vec{L}=\vec{0}}^{\vec{\infty}}\sum_{j=1}^{N_m}\sum_{n=1}^{N_m}{}'\sum_{a=1}^{\eta_j}\sum_{b=1}^{\eta_n}\frac{q_j^a q_n^b}{\vec{r}_{Ljn}^{ab}}\operatorname{erfc}\left(\alpha\vec{r}_{Ljn}^{ab}\right)+\frac{\alpha}{4\pi^{3/2}\epsilon_0}\sum_{j=1}^{N_m}\sum_{a=1}^{\eta_j} q_j^{a\,2}+ \qquad (7.116)$$
$$\frac{1}{2}\frac{1}{4\pi\epsilon_0}\sum_{j=1}^{N_m}\sum_{a=1}^{\eta_j}\sum_{b\neq a}^{\eta_j}\frac{q_j^a q_j^b}{r_{jj}^{ab}}\operatorname{erf}\left(\alpha r_{jj}^{ab}\right)+\sum_{j=1}^{N_m}\sum_{a=1}^{\eta_j}\vec{f}_j^a\cdot\vec{d}_j^a.$$

where $\vec{f}_j^a$ is the inter-molecular Coulombic force acting on an atom a in molecule j, which *is* obtained from differentiating (7.105) with respect to the vector $\vec{r}_j^a$ in the usual way. This force necessarily excludes any intra-molecular contributions. The form (7.116) (multiplied by -1) closely resembles (7.105) except for the presence of the last term, which is the atomic-to-molecular virial correction.

7.10 Calculating the Stress Tensor

To complete this chapter we will say something about the calculation of the stress tensor $\boldsymbol{\sigma}$, which is related to the isotropic pressure but is a (3×3) matrix defined by nine numbers (though this is reduced to six unique numbers by the requirement that the matrix be symmetric about the diagonal i.e. $\sigma_{\alpha\beta}=\sigma_{\beta\alpha}$ with $\alpha,\beta\equiv x,y,z$ *etc.*) The stress tensor describes the anisotropic aspects of pressure, such as uniaxial tension and shearing forces. The units of stress are force per unit area, as they are for pressure. The anisotropic application of forces on a system can produce bulk motion (including rotation). So model and experimental systems are set up so that the stress tensor is symmetric, which ensures that the system responds to an applied stress in a way that is characteristic of the material properties of the system, rather than simply producing bulk or rotational motion.

Calculating the stress tensor follows a similar procedure to the pressure calculation and we shall describe it for systems of N atoms. This is equally applicable when the atoms are combined into N_m *fully flexible* molecules, though we shall also mention some important points for rigid and constrained molecules later. As with calculating pressure, a derivative of the Hamiltonian is required, though in this case it is with respect to the simulation cell vectors, rather than the cell volume. We will begin with some preliminary discussion on the properties of the cell vectors.

The vector $\vec{r}_i$ locating the $i'th$ atom in a simulation cell can be written as

$$\vec{r}_i = u_i\vec{a}+v_i\vec{b}+w_i\vec{c}, \qquad (7.117)$$

where u_i, v_i, w_i are real numbers and $\vec{a}$, $\vec{b}$, $\vec{c}$ are the simulation cell vectors. Equation (7.117) may be expressed in matrix form as

$$\begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} = \begin{bmatrix} a_x b_x c_x \\ a_y b_y c_y \\ a_z b_z c_z \end{bmatrix} \begin{bmatrix} u_i \\ v_i \\ w_i \end{bmatrix}, \tag{7.118}$$

or more briefly as

$$\vec{r}_i = \boldsymbol{h}\,\vec{s}_i, \tag{7.119}$$

where $\boldsymbol{h}$ is the *cell matrix* and $\vec{s}_i$ is the *scaled atomic position*. Equation (7.119) is the scaling relation that underpins the derivation of the stress tensor. For later convenience we note the following properties of the matrix $\boldsymbol{h}$:

Matrix definition:

$$\boldsymbol{h} = \begin{bmatrix} a_x, b_x, c_x \\ a_y, b_y, c_y \\ a_z, b_z, c_z \end{bmatrix} \equiv \begin{bmatrix} h_{xx}, h_{xy}, h_{xz} \\ h_{yx}, h_{yy}, h_{yz} \\ h_{zx}, h_{zy}, h_{zz} \end{bmatrix}. \tag{7.120}$$

Matrix transpose ("flip" the matrix about the diagonal):

$$\tilde{\boldsymbol{h}} = \begin{bmatrix} h_{xx}, h_{yx}, h_{zx} \\ h_{xy}, h_{yy}, h_{zy} \\ h_{xz}, h_{yz}, h_{zz} \end{bmatrix}. \tag{7.121}$$

Matrix inverse:

$$\boldsymbol{h}^{-1} = \frac{1}{|\boldsymbol{h}|} \begin{bmatrix} h_{yy}h_{zz} - h_{zy}h_{yz}, h_{zy}h_{xz} - h_{xy}h_{zz}, h_{xy}h_{yz} - h_{yy}h_{xz} \\ h_{zx}h_{yz} - h_{yx}h_{zz}, h_{xx}h_{zz} - h_{zx}h_{xz}, h_{yx}h_{xz} - h_{xx}h_{yz} \\ h_{yx}h_{zy} - h_{zx}h_{yy}, h_{zx}h_{xy} - h_{xx}h_{zy}, h_{xx}h_{yy} - h_{yx}h_{xy} \end{bmatrix}. \tag{7.122}$$

such that

$$\boldsymbol{h}\,\boldsymbol{h}^{-1} = \boldsymbol{h}^{-1}\,\boldsymbol{h} = \mathbf{1}. \tag{7.123}$$

Matrix determinant:

$$|\boldsymbol{h}| = h_{xx}(h_{yy}h_{zz} - h_{zy}h_{yz}) + h_{xy}(h_{zx}h_{yz} - h_{yx}h_{zz}) + h_{xz}(h_{yx}h_{zy} - h_{zx}h_{yy}) = V, \tag{7.124}$$

where V is the volume of the simulation cell.

Using the definition (7.120) equation (7.124) is found to be equivalent to

$$V = \vec{a} \cdot (\vec{b} \times \vec{c}), \tag{7.125}$$

i.e. the *scalar triple product* of the cell vectors.

Armed with this information, the stress tensor for a system composed of simple atoms

(i.e. without constraints or rigid bodies etc.) may be defined by the following equation:

$$\boldsymbol{\sigma} = -\left\langle \frac{1}{V}\left(\frac{\partial H^N(\vec{\Gamma}^N)}{\partial \boldsymbol{h}}\right)_T \tilde{\boldsymbol{h}} \right\rangle. \tag{7.126}$$

The meaning of this equation becomes clearer when we realise that the partial derivative in (7.126) is a matrix (say **M**) such that

$$M_{\alpha\beta} = \left(\frac{\partial H^N(\vec{\Gamma}^N)}{\partial h_{\alpha\beta}}\right)_T, \tag{7.127}$$

with $\alpha,\beta \equiv x, y, z$ *etc*, so that matrix $\boldsymbol{\sigma}$ is obtained from a matrix product:

$$\boldsymbol{\sigma} = -\frac{1}{V}\langle \boldsymbol{M}\,\tilde{\boldsymbol{h}} \rangle. \tag{7.128}$$

As was the case when calculating the pressure, the Hamiltonian is not usually an explicit function of the matrix $\boldsymbol{h}$, so we make it so by using the scaling relation (7.119). We will also need the derivative of (7.119) with respect to time:

$$\dot{\vec{r}}_i = \boldsymbol{h}\dot{\vec{s}}_i + \dot{\boldsymbol{h}}\vec{s}_i, \tag{7.129}$$

from which we obtain the system kinetic energy:

$$K^N(\dot{\vec{r}}^N) = \frac{1}{2}\sum_{i=1}^{N} m_i\left(\boldsymbol{h}\dot{\vec{s}}_i + \dot{\boldsymbol{h}}\vec{s}_i\right)^2. \tag{7.130}$$

From this it can be shown that the momentum $\vec{\pi}_i$ conjugate to $\vec{s}_i$ is related to the Newtonian momentum $\vec{p}_i$ via

$$\vec{\pi}_i = \tilde{\boldsymbol{h}}\,\vec{p}_i \qquad \text{or} \qquad \vec{p}_i = \tilde{\boldsymbol{h}}^{-1}\vec{\pi}_i. \tag{7.131}$$

(Note, trivially, that the transpose of an inverse matrix and the inverse of a transpose matrix are the same.) We can now write the Hamiltonian for a system of N atoms as

$$H^N(\vec{r}^N, \vec{p}^N) \equiv H^N(\boldsymbol{h}_N\vec{s}^N, \tilde{\boldsymbol{h}}_N^{-1}\vec{\pi}^N) = K^N(\tilde{\boldsymbol{h}}_N^{-1}\vec{\pi}^N) + \Phi^N(\boldsymbol{h}_N\vec{s}^N), \tag{7.132}$$

where the matrices $\boldsymbol{h}_N$ and $\tilde{\boldsymbol{h}}_N^{-1}$ are the *N-dimensional* equivalents of $\boldsymbol{h}$ or $\boldsymbol{h}^{-1}$ as required. In practice we will only ever use the 3-dimensional versions. We now insert (7.132) into (7.126) and perform the required differentiation to obtain the required formula for the stress tensor. We will do this here for the kinetic and potential energy components separately.

The kinetic component is

$$K^N(\tilde{\boldsymbol{h}}_N^{-1}\vec{\pi}^N)=\frac{1}{2}\sum_{i=1}^{N}\frac{1}{m_i}\tilde{\vec{\pi}}_i\boldsymbol{h}^{-1}\tilde{\boldsymbol{h}}^{-1}\vec{\pi}_i. \tag{7.133}$$

where the tilde (~) over $\tilde{\vec{\pi}}$ indicates a *row vector*.

In order to differentiate this with respect to an element $h_{\alpha\beta}$ we use the general relation (where λ is any variable):

$$\frac{\partial \boldsymbol{h}^{-1}}{\partial\lambda}=-\boldsymbol{h}^{-1}\frac{\partial \boldsymbol{h}}{\partial\lambda}\boldsymbol{h}^{-1}, \tag{7.134}$$

(which is easily proved by differentiating equation (7.123) with respect to λ.) Thus (7.133) becomes on differentiation

$$\begin{aligned}\frac{\partial K^N(\tilde{\boldsymbol{h}}_N^{-1}\vec{\pi}^N)}{\partial h_{\alpha\beta}}&=\frac{1}{2}\sum_{i=1}^{N}\frac{1}{m_i}\tilde{\vec{\pi}}_i\left(\frac{\partial \boldsymbol{h}^{-1}}{\partial h_{\alpha\beta}}\tilde{\boldsymbol{h}}^{-1}+\boldsymbol{h}^{-1}\frac{\partial \tilde{\boldsymbol{h}}^{-1}}{\partial h_{\alpha\beta}}\right)\vec{\pi}_i\\&=-\frac{1}{2}\sum_{i=1}^{N}\frac{1}{m_i}\tilde{\vec{\pi}}_i\left(\boldsymbol{h}^{-1}\frac{\partial \boldsymbol{h}}{\partial h_{\alpha\beta}}\boldsymbol{h}^{-1}\tilde{\boldsymbol{h}}^{-1}+\boldsymbol{h}^{-1}\tilde{\boldsymbol{h}}^{-1}\frac{\partial \tilde{\boldsymbol{h}}}{\partial h_{\alpha\beta}}\tilde{\boldsymbol{h}}^{-1}\right)\vec{\pi}_i,\end{aligned} \tag{7.135}$$

which leads to

$$\frac{\partial K^N(\tilde{\boldsymbol{h}}_N^{-1}\vec{\pi}^N)}{\partial h_{\alpha\beta}}=-\frac{1}{2}\sum_{i=1}^{N}\frac{1}{m_i}\tilde{\vec{p}}_i\left(\frac{\partial \boldsymbol{h}}{\partial h_{\alpha\beta}}\boldsymbol{h}^{-1}+\tilde{\boldsymbol{h}}^{-1}\frac{\partial \tilde{\boldsymbol{h}}}{\partial h_{\alpha\beta}}\right)\vec{p}_i. \tag{7.136}$$

Now, it is obvious that the derivative of $\boldsymbol{h}$ with respect to $h_{\alpha\beta}$ is mostly a zero matrix, except for the element at α,β which is unity. Likewise the derivative of $\tilde{\boldsymbol{h}}$ with respect to $h_{\alpha\beta}$ is zero everywhere except for unity at element β,α. This allows a significant degree of simplification and (7.136) can be rewritten as

$$\frac{\partial K^N(\tilde{\boldsymbol{h}}_N^{-1}\vec{\pi}^N)}{\partial h_{\alpha\beta}}=-\sum_{i=1}^{N}\frac{1}{m_i}\left(p_i^{\alpha}\sum_{\gamma}h_{\beta\gamma}^{-1}p_i^{\gamma}\right). \tag{7.137}$$

Taking account of all elements $h_{\alpha\beta}$ leads to

$$\frac{\partial K^N(\tilde{\boldsymbol{h}}_N^{-1}\vec{\pi}^N)}{\partial \boldsymbol{h}}=-2\left(\sum_{i=1}^{N}\boldsymbol{k}_i\right)\tilde{\boldsymbol{h}}^{-1}. \tag{7.138}$$

where the elements of the *kinetic energy tensor* $\boldsymbol{k}_i$ are defined as

$$(\boldsymbol{k}_i)_{\alpha\beta}=\frac{p_i^{\alpha}p_i^{\beta}}{2m_i}=\frac{1}{2}m_i\dot{r}_i^{\alpha}\dot{r}_i^{\beta}. \tag{7.139}$$

Differentiating the potential energy component gives

$$\frac{\partial \Phi^N(\vec{r}^N)}{\partial h_{\alpha\beta}} = \sum_{i=1}^{N} \frac{\partial \Phi^N(\vec{r}^N)}{\partial \vec{r}_i} \cdot \frac{\partial \boldsymbol{h}\vec{s}_i}{\partial h_{\alpha\beta}} = -\sum_{i=1}^{N} \tilde{\vec{f}}_i \left(\frac{\partial \boldsymbol{h}}{\partial h_{\alpha\beta}} \right) \vec{s}_i , \tag{7.140}$$

where $\vec{f}_i$ is the force acting on atom i. Once again the derivative of $\boldsymbol{h}$ with respect to $h_{\alpha\beta}$ is a matrix that is zero everywhere except for unity at element α,β. This allows some simplification, leading to

$$\frac{\partial \Phi^N(\vec{r}^N)}{\partial h_{\alpha\beta}} = -\sum_{i=1}^{N} f_i^{\alpha} s_i^{\beta}, \qquad \text{or} \qquad \frac{\partial \Phi^N(\vec{r}^N)}{\partial \boldsymbol{h}} = \left(\sum_{i=1}^{N} \boldsymbol{\psi}_i \right) \tilde{\boldsymbol{h}}^{-1}, \tag{7.141}$$

where the elements of the *virial tensor* $\boldsymbol{\psi}_i$ are defined as

$$(\boldsymbol{\psi}_i)_{\alpha\beta} = -f_i^{\alpha} r_i^{\beta}. \tag{7.142}$$

Combining results (7.138) and (7.141) into (7.126) gives the final formula for the stress tensor.

$$\boldsymbol{\sigma} = \left\langle \frac{1}{V} \sum_{i=1}^{N} \left(2\boldsymbol{k}_i - \boldsymbol{\psi}_i \right) \right\rangle. \tag{7.143}$$

Alert readers will notice that this definition of the virial tensor (7.142) is not suitable for a system using periodic boundaries. More suitable forms can be constructed once the nature of the inter-atomic potentials is known. For example for pair forces we may write

$$\sum_{i=1}^{N} (\boldsymbol{\psi}_i)_{\alpha\beta} = -\frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N}{}' f_{ij}^{\alpha} r_{ij}^{\beta}. \tag{7.144}$$

The rule is always to use the minimum image of the separation $\vec{r}_{ij}$ for every interaction calculation.

There are other important points to be made before concluding this section. These are given below.

- The average of the diagonal elements of the stress tensor is equal to the isotropic pressure. This can easily be established by comparing the contributions from the diagonal elements of the kinetic and virial tensors to the isotropic kinetic energy and virial contributions.

- In the calculation of pressure in preceding sections it was possible to determine that certain (angle dependent) potentials made no contribution to the virial because they were invariant under the scaling transformation. In the context of stress tensor calculations, this merely means that the average of the diagonal elements is zero. It does *not* mean the stress tensor elements are all zero.

- Calculation of the stress tensor for rigid molecule systems proceeds in much the same way as the calculation of pressure. The location of the atom a in

molecule i is given by

$$\vec{r}_i^a = \vec{R}_i + \vec{d}_i^a \tag{7.145}$$

where $\vec{R}_i$ is the location of the molecule's centre of mass and $\vec{d}_i^a$ the displacement of the atom from the molecule centre. The scaling relation used in the stress tensor calculation is

$$\vec{r}_i^a = \boldsymbol{h}\vec{S}_i + \vec{d}^a \tag{7.146}$$

in which $\vec{S}_i$ is the scaled centre of mass and the displacement vector $\vec{d}_i^a$ is not affected by the scaling. Proceeding with the differentiation with respect to $h_{\alpha\beta}$ leads, as with the calculation of pressure, to a molecular virial with an *atomic-to-molecular viral correction*, which in this case is a matrix. This is straightforward, if tedious, but the important thing to note is that the correction matrix obtained is generally *not symmetric* and it needs to be made so by a simple transformation:

$$\begin{aligned} \sigma'_{\alpha\beta} &= (\sigma_{\alpha\beta} + \sigma_{\beta\alpha})/2 \\ \sigma'_{\beta\alpha} &= (\sigma_{\alpha\beta} + \sigma_{\beta\alpha})/2 \end{aligned} \tag{7.147}$$

If this correction is not applied, simulations which are dynamically dependent on the stress tensor (e.g. the $N\sigma T$ ensemble) will develop *bulk rotation*.

7.11 Pressure and "Frozen" Atoms

In molecular dynamics it is often an advantage to fix the position of some atoms so that they are unaffected by the normal dynamics of the system. Such "frozen" atoms then form a fixed background to the events occurring in the simulation but, at the same time continue to exert an influence. This is of use, for example, in studies of catalysis, where atoms remote from the active site exert background forces on the process, but otherwise may not be dynamically involved. A question that arises in such applications is: how do these atoms contribute to the system pressure?

To answer this question we will adapt the Clausius virial theorem from section 3.5.3 to this case and begin by considering the first N_f atoms from a system of N atoms as frozen. What effect do they exert on the remaining system? Clearly forces arise from them which act on the other atoms, but they are not themselves meaningfully acted upon - inasmuch as they are not moved by these forces. The principal effect is to exclude the mobile atoms from the volume that surrounds them, and thus effectively reduce the system volume.

We begin by writing the virial equation for the full N atom system as

$$\left\langle \sum_{i=1}^{N} m_i \dot{r}_i^2 \right\rangle + \left\langle \sum_{i=1}^{N-1} \sum_{j>i}^{N} \vec{r}_{ij} \cdot \vec{f}_{ij} \right\rangle - 3\mathrm{PV} = 0, \tag{7.148}$$

in which we assume the first N_f atoms are frozen and the remainder are free to move dynamically. The first term on the left may now be written as

$$\left\langle \sum_{i=1}^{N} m_i \dot{r}_i^2 \right\rangle = \left\langle \sum_{i>N_f}^{N} m_i \dot{r}_i^2 \right\rangle = 2K' \tag{7.149}$$

where K' is the kinetic energy of the dynamically free atoms only.

The virial term may be expanded into the following contributions

$$\left\langle \sum_{i=1}^{N-1} \sum_{j>i}^{N} \vec{r}_{ij} \cdot \vec{f}_{ij} \right\rangle = \left\langle \sum_{i=1}^{N_f-1} \sum_{j>i}^{N_f} \vec{r}_{ij} \cdot \vec{f}_{ij} \right\rangle + \left\langle \sum_{i>N_f}^{N-1} \sum_{j>i}^{N} \vec{r}_{ij} \cdot \vec{f}_{ij} \right\rangle + \left\langle \sum_{i=1}^{N_f} \sum_{j>N_f}^{N} \vec{r}_{ij} \cdot \vec{f}_{ij} \right\rangle \tag{7.150}$$

The first term on the right, contains contributions from the frozen atoms alone. If we insist that these atoms are fixed in space this term cannot contribute to the pressure. The forces between them do not contribute to the dynamics of the system and are thus dynamically irrelevant. We might also say that their mutual interaction energy does not scale with the system volume and cannot contribute thermodynamically to the pressure. So we may neglect this term altogether.

The second term we may identify with the virial of the freely moving atoms:

$$\Psi' = \left\langle \sum_{i>N_f}^{N-1} \sum_{j>i}^{N} \vec{r}_{ij} \cdot \vec{f}_{ij} \right\rangle \tag{7.151}$$

The third term involves both frozen and mobile atoms. It turns out this can be identified with a reduction in system volume as suggested above. The third term is thus the frozen atom virial correction term.

Consider a single pair of atoms, one frozen and one free. The virial contribution is $\vec{r}_{12} \cdot \vec{f}_{12}$, which because the force and inter-atomic vectors are parallel, reduces to $r_{12} f_{12}$. If we assume for the moment that the system is composed of uniform hard spheres, this contribution is finite only when $r_{12} = \sigma$, where σ is the atomic diameter. Then $\vec{f}_{12}$ corresponds to an impulse force. From the viewpoint of the frozen atom, the sum of such impulse forces give rise to a pressure acting on the surface of a sphere of radius σ centred on the frozen atom. This pressure is, of course, the system pressure P. Thus we may write for the hard sphere case (by analogy with the wall effect of the cubic system boundary):

$$\left\langle \sum_{i=1}^{N_f} \sum_{j>N_f}^{N} \vec{r}_{ij} \cdot \vec{f}_{ij} \right\rangle = \sum_{i=1}^{N_f} \sigma 4\pi\sigma^2 P = 3 N_f V_h P \tag{7.152}$$

Where V_h is the excluded volume around a frozen atom. The pressure force in this case acts in the same direction as the vector $\vec{r}_{ij}$ and leads to a positive result. Thus we see that the additional term is effectively a correction to the system volume due to

the finite size of the frozen atoms. i.e. the pressure equation is now

$$3P\left(V - N_f V_h\right) = \left(2K' - \Psi'\right) \tag{7.153}$$

In systems with continuous potentials however, we cannot express the correction so neatly, but at least we see what it means. In practice, for continuous potentials, it is more straightforward to calculate the virial for the free atoms (7.151) and the frozen atom virial term (left side of (7.152)) at the same time.

Thus, in summary, the only operational differences between calculating pressure for a system of mobile atoms and one containing frozen atoms are:

1. When calculating the kinetic energy, the contributions from the frozen atoms are zero.
2. When calculating the virial, assume all forces between frozen atoms are zero, but include all other pair forces.

Chapter 8

Some Molecular Dynamics Methodology

8.1 Introduction

In this chapter we describe some of the methodology of molecular dynamics, covering the techniques that are to be found in most programs and about which it is useful to know something if they are to be used with some efficiency. Some of this is basic stuff, which is tempting to skip over, but it's not often discussed and it pays to take a look if only to clarify issues that are buried in the folk law of molecular simulation. We start by describing the simplest possible structure for a molecular dynamics program, on which all more advanced programs, including those designed for parallel applications, are ultimately based.

The basic structure of a molecular dynamics program is shown in figure 8.1. On the left is a program employing the leapfrog Verlet algorithm, while on the right is a program employing the velocity Verlet algorithm. (See the equations presented in section 4.2.)

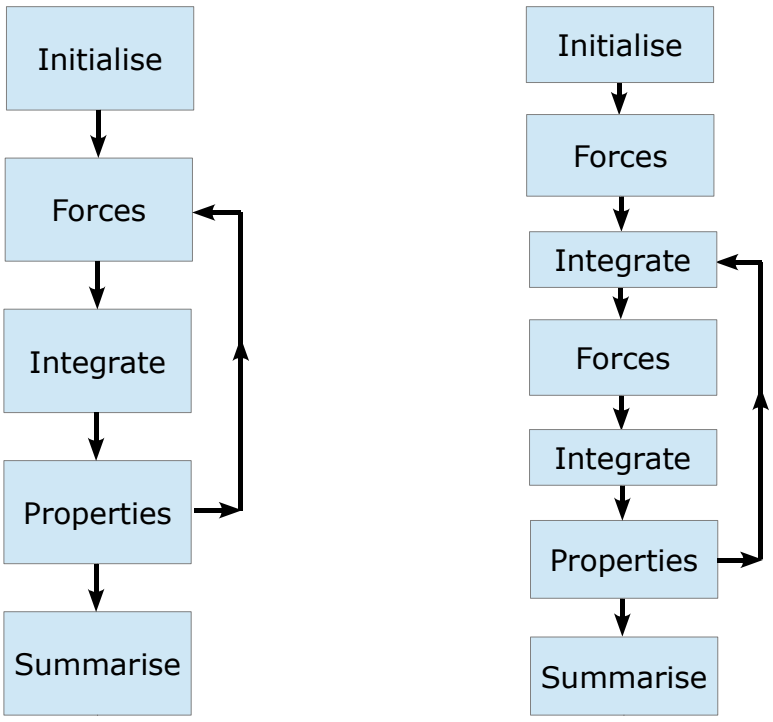

Figure 8.1 Basic Program Structures for Molecular Dynamics
(left leapfrog, right velocity Verlet)

The two programs are closely related and have comparable blocks of code in their

structure. Both have an *Initialise* block, in which the initial spatial configuration of the atoms is defined and the starting atomic velocities assigned (with subtraction of centre of mass motion and scaling to temperature). In the case of the velocity Verlet program, this is followed by an additional *Forces* block, since initialisation in this case also requires knowledge of the initial forces and torques acting on the molecules.

The central part of each program might be called the *molecular dynamics engine*, which generates the trajectories of the atoms and molecules as a function of time. This is executed a great number of times in a simulation and inevitably is where most of the computational cost of a simulation lies. There are three blocks in this region: the *Forces* block, which performs the calculation of the forces acting in the system; the *Integrate* blocks, which perform the integration of the equations of motion; and the *Properties* block which calculates the system properties. Note that the leapfrog algorithm requires one call to the integration block, while the velocity Verlet algorithm requires two, in accordance with the structures of these algorithms as described in Chapter 4. The Properties block is usually not active during the initial equilibration period of the simulation, since any data gathered from this period will not be representative of the equilibrium thermodynamic state.

The last block is the S*ummarise* block, which calculates the final average properties of the system, writes out important data files and closes down the program tidily, so that it can be restarted and continued if required.

The Forces block is usually considerably more time consuming than either the Integrate or Properties blocks and in consequence it is where most effort is concentrated in designing an efficient program. Of the many forms of force that occur in a molecular system, (see Chapter 5) the pair forces are usually the most dominant in terms of cost. This is because the pair forces are always longer ranged than the others and involve a larger number of interactions. For this reason much attention is paid to calculating these efficiently.

With these program outlines in mind, we shall now look at the issue of how the algorithms scale on a computer.

8.2 Algorithm Performance and Scaling

From the basic program structures shown in figure 8.1 it is possible to get an idea of how the leapfrog and velocity Verlet based programs scale when running on a computer. The objective of such an exercise is to learn how the cost of a simulation depends on common requirements such as the number of atoms, the size of the cut-off applied to the forces calculations and so on. To simplify matters we will assume a serial, or single processor computer and the simplest of applications: the noble gas system with Lennard-Jones interactions. Furthermore we will ignore the computational cost of the Initialise and Summarise blocks in the code, since these are executed once only whenever the program is run, whereas the Forces and Integrate blocks may be repeated many thousands of times. We will also leave out the Properties block, since it is not usually executed every time step and in any case its requirements differ in different applications. (However it should become apparent how an estimate for it can be obtained once we see how the Forces and Integrate blocks are dealt with.)

With regard to the Forces block we know that a system of N atoms has potentially $N(N-1)/2$ possible pair interactions, provided we are using the minimum image convention (section 1.2.5), which considers only one image of each possible pair of atoms. The cost of calculating an interaction is composed of two parts. The first is the calculation of the minimum image inter-atomic separation, which is required to determine if the interaction is within the cut-off range r_{cut}, and must be done for all possible atom pairs. The second is the calculation of the actual pair interaction (inter-atomic force and pair potential). We will assume that the cost of calculating one pair interaction is α, typically in seconds, and the cost of calculating an inter-atomic separation we assume to be α'. On this basis the cost of calculating *all* the inter-atomic separations is simply $\alpha' N(N-1)/2$ while the cost of calculating all the pair interactions is $f\alpha N(N-1)/2.$ The factor f is required to account for the effect of the cut-off, since only a fraction, f, of all the possible interactions is actually calculated. For any given simulation with a reasonably constant density, f can be assumed to be constant. For example, for a cubic simulation cell with the largest cut-off permitted by the minimum image convention $f=\pi/6$ or $f\approx 1/2,$ which is the ratio of the volume of the cut-off sphere, $(4\pi r_{cut}^3/3),$ to the volume of the simulation cell, $V_0,$ which in this case equals $8r_{cut}^3.$ Thus the total cost *per time step* for the pair interaction calculations in the Forces block is

$$t_{force}=(f\alpha+\alpha')N(N-1)/2. \tag{8.1}$$

It is evident that t_{force} is *quadratically dependent* upon $N,$ which is the expected outcome for pair interactions.

Before proceeding further, we should ask if the formula given in (8.1) is applicable under all circumstances. The answer is no. We have in fact implicitly assumed that we will systematically consider all possible pair interactions in the system, which is an *all-pairs* approach. There are however alternative approaches which explicitly avoid considering all possible pairs. A notable example is the linked cell approach of Hockney and Eastwood [16] which we will describe later and which is almost linearly dependent on $N.$ For now we will continue with an all-pairs approach since this is the most obvious and is widely used.

To be useful the formula (8.1) requires reasonable estimated values for α, α' and $f.$ The last of these can be accurately obtained in most cases from the ratio of the volume of the cut-off sphere to the volume of the simulation cell, as outlined above. As for α and $\alpha',$ it is best to obtain these experimentally rather than attempt to calculate them *a priori* from the fundamental characteristics of the computer's *central processing unit*. Attempting the same simulation twice with different specified cut-off distances, while timing the passage through the Forces block, will allow α and α' to be obtained from (8.1) by solving a pair of simultaneous equations. (But note that f is different in the two cases!) The values of α and α' obtained can be used to estimate t_{force} for different system sizes and different cut-off conditions.

Estimation of the cost of the Integrate block proceeds in a similar manner. Integration for the simple atomic case clearly depends linearly on the number of atoms $N,$ since it is necessary to integrate the equations of motion for each atom independently and without exception. So for the leapfrog algorithm, we can immediately write the time

per time step as

$$t^{lf}_{integrate} = N\beta, \tag{8.2}$$

Where β represents the time to integrate the equations of motion (both position and velocity) for just one atom. In the leapfrog case only one pass through the Integrate block is made, so this represents the full cost per time step. In the case of the velocity Verlet algorithm, figure 8.1 indicates two passes through the Integrate block are required, so the time is better accounted for using

$$t^{v}_{integrate} = \frac{3}{2} N\beta. \tag{8.3}$$

Here we have inserted a factor of $3/2$ rather than 2 because the second pass through the Integration block updates the atomic velocities but not the positions. In practice this is not going to have a great impact on the relative performances of these two algorithms, as the Forces block usually dominates the overall cost. As for the Forces block, the empirical parameter β can be estimated by timing the passage through the block for different values of N.

We therefore can get a simple idea of how a program will perform from estimating a few parameters and consulting the equation

$$t_{time\,step} = t_{force} + t_{integrate}, \tag{8.4}$$

Where the terms appearing are defined by equations (8.1),(8.2) and (8.3). This equation is suitable for estimating how the simulation scales in cost as a function of the number of atoms, the cut-off used and the system volume. It should be reasonably accurate for most cases.

It has to be admitted that this analysis is not particularly profound in this case, but we will consider more complicated situations later, which will be more insightful. The main purpose of deriving this expression is to show that performance scaling is amenable to a simple analysis.

8.3 Boundary Conditions

Every molecular dynamics simulation is enclosed by a boundary condition of some kind, whether or not one was knowingly specified, and whatever boundary condition that is, it has a marked effect on the properties of the system being simulated. It is therefore extremely important to understand the nature of the boundary condition being used, to ensure that it is appropriate for the study being undertaken. Most boundaries have the effect of removing any surfaces from the system, so that the structural and dynamical properties of the system are not affected by surface forces, which can otherwise dominate the system. But they also have less obvious effects, such as inhibiting the occurrence of phase transitions because the imposed boundary conditions are incommensurate with the structure of the new phase. Also, the dimensions of the system, as defined by the boundary conditions, place limits on the range of correlations in space and time that can be studied. If a system is periodic on a length scale, λ, spatial correlations beyond the range $\lambda/2$ are directly affected

by the imposed periodicity. Time correlations on a time scale of λ/c (where c is the velocity of sound) may potentially also show artefacts due to the periodicity. This is particularly the case when energetic events occur, as in radiation damage studies. So researchers should at all times be mindful of the boundary conditions being used and consider how they might be affecting the physical properties they are investigating. It is also useful to remember, if you are a programmer, the impact the boundary condition has throughout the computer programs used to perform the simulations. Every one of the coding blocks referred to in the previous section need to manage the boundary condition somewhere.

The kinds of boundary conditions commonly encountered in molecular dynamics as the following. (Only periodic boundaries will be dealt with in detail here however.)

1. *Periodic boundaries.* The great majority of molecular dynamics simulations employ periodic boundary conditions, which are manifested through the shape of the simulation cell. These range from simple cubic boundaries (as encountered in Chapter 1) through orthorhombic boundaries to triclinic boundaries, which are recognisable from their unit cell counterparts in crystallography, but other forms are possible (see below).

2. *Vacuum boundaries.* Simulations of a system in a vacuum are sometimes undertaken, for example to find the preferred shape of a nano-crystal or to study fusion (i.e. sintering) of nano-crystals. In such systems, surface effects are of key interest, so periodic boundaries may be inappropriate. (However modelling isolated molecules, such as proteins, without boundaries is inadvisable, since the tertiary structure will be much influenced by surface forces.)

3. *Stochastic boundaries.* This resembles the case without boundaries, but is more suitable for the study of molecules in solution. If the effects of a periodic boundary are considered unacceptable an alternative is to place the solute molecule within a liquid drop, which is confined in space by a stochastic boundary to control evaporation. (The same arrangement can be used to study surface tension in the drop.) The stochastic boundary consists of random forces, which operate at a chosen range from the system centre and serve to mimic the effect of solvent molecules at this range. This alone is not sufficient to counter the effects of surface tension at the boundary, which is counteracted to some degree by a *potential of mean force* based on the known structure of the pure solvent.

4. *Slab boundaries.* If the study of the infinite liquid or solid surface is required, then it is possible to set up a boundary condition that is periodic in a plane and non-periodic in a direction perpendicular to the plane. Such a boundary is often used for example in surfactant studies and heterogeneous catalysis. Note that the pressure in such a system is *undefined*, since the volume is arbitrary. The surface stress tensor (surface tension) can be defined however.

5. *Dynamic/mechanical boundaries.* Such boundaries are much used in *non-equilibrium dynamics* and are largely concerned with the engineering properties

of materials and liquids. Mechanical stresses and shears can easily be applied to solids to determine elastic and inelastic responses and parameters associated with them, such as elastic moduli and yield stresses. Dynamical boundaries, such as the Lees-Edwards boundary [74], which imposes a velocity gradient in the fluid, are used to study viscosity in liquids. Dynamical and mechanical boundary conditions generally do mechanical work on the systems and so generate heat. This means an appropriate thermostat (or several thermostats) may be needed to define the system temperature and thus the associated properties, meaningfully. This is a large subject, but the works of Hoover [75] and Evans [76] provide excellent coverage.

8.3.1 Common Periodic Boundaries

The periodic boundary defines the shape and size of the simulation cell. The commonest periodic boundaries are cubic, orthorhombic and triclinic, as shown in figure 8.2. Typically, cubic periodic boundaries are used for isotropic liquids, while orthorhombic boundaries are used for studies of surfaces. The triclinic boundaries are used in studies of crystalline solids, where the shape of the simulation cell matches that of the unit cell of the corresponding crystalline system. It will be appreciated that the cubic and orthorhombic boundary conditions are special cases of the triclinic case. These three examples can be constructed by a multiple replication of a crystal unit cell i.e. if the unit cell is represented by the cell vectors $\vec{a}, \vec{b}, \vec{c}$ the simulation cell is represented by vectors $\vec{A}, \vec{B}, \vec{C},$ where

$$\vec{A}=l\vec{a}, \qquad \vec{B}=m\vec{b}, \qquad \vec{C}=n\vec{c}, \tag{8.5}$$

in which l, m, n are integers. This often provides a convenient way to generate model systems starting from the unit cell basis vectors and the few atoms within.

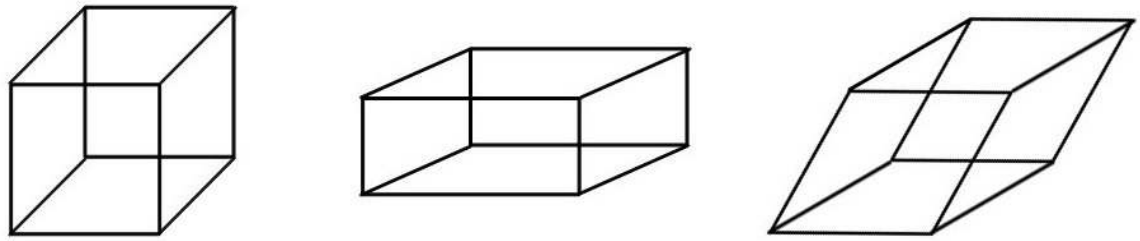

Figure 8.2: Cubic, Orthorhombic and Triclinic Simulation Cells

To employ these periodic boundaries in molecular simulation we need to adapt the minimum image convention introduced in chapter 1, section 1.2.5 to each particular case. To do this we need first to describe the position, $\vec{r}_i,$ of an atom, $i,$ within the simulation cell in terms of the cell vectors $\vec{A}, \vec{B}, \vec{C}.$ This we write as

$$\vec{r}_i=s_i^a\vec{A}+s_i^b\vec{B}+s_i^c\vec{C}, \tag{8.6}$$

which is a linear combination of the vectors $\vec{A}, \vec{B}, \vec{C}.$ Since $\vec{r}_i$ is a location *within* the cell, it follows that all the scalar numbers s_i^a, s_i^b, s_i^c are in the range $-1/2$ to $1/2,$ (assuming the origin of coordinates is in the centre of the cell). These are

called the *fractional coordinates* of atom i.

We can now write

$$\vec{r}_i = \boldsymbol{h}\vec{s}_i, \tag{8.7}$$

in which $\vec{s}_i$ is a vector composed of the fractional coordinates:

$$\vec{s}_i = \begin{bmatrix} s_i^a \\ s_i^b \\ s_i^c \end{bmatrix}, \tag{8.8}$$

and $\boldsymbol{h}$ is the *cell matrix* defined as

$$\boldsymbol{h} = \begin{bmatrix} A_x B_x C_x \\ A_y B_y C_y \\ A_z B_z C_z \end{bmatrix}, \tag{8.9}$$

where A_x, B_y *etc*. are components of the vectors $\vec{A}$, $\vec{B}$, $\vec{C}$.

Equation (8.7) can now be inverted:

$$\vec{s}_i = \boldsymbol{h}^{-1}\vec{r}_i, \tag{8.10}$$

in which matrix $\boldsymbol{h}^{-1}$ can always be obtained provided the vectors $\vec{A}$, $\vec{B}$, $\vec{C}$ do not all lie in the same plane (i.e. the volume of the cell is non-zero).

Now, if $\vec{r}_i$ is a location in a *neighbouring image cell* then we would write (8.6) as

$$\vec{r}_i = (s_i^a \pm 1)\vec{A} + (s_i^b \pm 1)\vec{B} + (s_i^c \pm 1)\vec{C}, \tag{8.11}$$

where the ± 1 indicates addition or subtraction of a full cell vector: $\vec{A}$, $\vec{B}$ or $\vec{C}$. In this case (8.7) becomes

$$\vec{r}_i = \boldsymbol{h}\vec{s}_i + \boldsymbol{h}\vec{I}, \tag{8.12}$$

Where the vector, $\vec{I}$, is

$$\vec{I} = \begin{bmatrix} \pm 1 \\ \pm 1 \\ \pm 1 \end{bmatrix}. \tag{8.13}$$

The equation corresponding to (8.10) is now

$$\vec{s}_i = \boldsymbol{h}^{-1}\vec{r}_i - \boldsymbol{h}^{-1}\boldsymbol{h}\vec{I}, \tag{8.14}$$

or

$$\vec{s}_i = \boldsymbol{h}^{-1}\vec{r}_i - \vec{I}\,. \tag{8.15}$$

Equation (8.15) tells us that atoms in a neighbouring (image) cell can be restored to the corresponding location in the simulation cell by first converting from $\vec{r}_i$ to $\vec{s}_i$ using (8.10), adding or subtracting 1 to the appropriate component of $\vec{s}_i$, and then reverting back to the coordinate $\vec{r}_i$ using (8.7). This is the basis for an algorithm for the minimum image construction, which we present below for the specific cases shown in figure 8.2.

8.3.1.1 The Triclinic case:

1: Construct $\boldsymbol{h}$ using (8.9) and calculate $\boldsymbol{h}^{-1}$,
2: Calculate $\vec{s}_i$ for all $i=1,\ldots,N$, using $\vec{s}_{ij}=\boldsymbol{h}^{-1}\vec{r}_{ij}$,
3: Calculate $\vec{s}_{ij}=\vec{s}_j-\vec{s}_i$ for all atom pairs,
4: Calculate $s_{ij}^{\alpha} \leftarrow s_{ij}^{\alpha} - Nint\left(s_{ij}^{\alpha}\right)$ for all atom pairs and all components α,
5: Calculate minimum image separation for all atom pairs using $\vec{r}_{ij}=\boldsymbol{h}\,\vec{s}_{ij}$.

The function $Nint(x)$ in step 4 is the nearest integer to argument x. (This is *not* merely the integer part of x.)

It is fair to ask if this really does provide the minimum image of the separation $\vec{r}_{ij}$, since we are using $\vec{s}_{ij}$ to obtain it and this is clearly not the same as using $\vec{r}_{ij}$ directly. We can recognise that it is, by assuming atom i is at the origin of the cell. (Since the system overall is periodic in 3D and we are free to choose the origin of the cell anywhere.) In which case $\vec{r}_{ij}$ is equivalent to $\vec{r}_j$ and using the transformation (8.15) on atom j would guarantee that we are working with an image of j that is in the same cell as i. Any atom that is in the same cell as one at the origin must be its nearest image. We can make this argument, whichever atom is taken as at the origin, so all distances $\vec{r}_{ij}$ transformed using (8.15) must be the minimum image distances.

8.3.1.2 The Cubic case:

In the special cases of the cubic and orthorhombic boundaries, the matrix $\boldsymbol{h}$ is diagonal, which makes the inverse $\boldsymbol{h}^{-1}$ trivial. The overall algorithm is also then much simpler.

$$\boldsymbol{h}_{cub} = a\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}, \qquad \boldsymbol{h}_{cub}^{-1} = \frac{1}{a}\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}. \tag{8.16}$$

The minimum image is given by

$$\begin{aligned} x_{ij} &\leftarrow x_{ij} - a\,Nint\left(x_{ij}/a\right) \\ y_{ij} &\leftarrow y_{ij} - a\,Nint\left(y_{ij}/a\right) \\ z_{ij} &\leftarrow z_{ij} - a\,Nint\left(z_{ij}/a\right) \end{aligned} \quad . \tag{8.17}$$

8.3.1.3 The Orthorhombic case:

The cell matrix is

$$\boldsymbol{h}_{ort} = \begin{bmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & c \end{bmatrix}, \qquad \boldsymbol{h}_{ort}^{-1} = \begin{bmatrix} a^{-1} & 0 & 0 \\ 0 & b^{-1} & 0 \\ 0 & 0 & c^{-1} \end{bmatrix}. \tag{8.18}$$

The minimum image is given by

$$\begin{aligned} x_{ij} &\leftarrow x_{ij} - a\,Nint\left(x_{ij}/a\right) \\ y_{ij} &\leftarrow y_{ij} - b\,Nint\left(y_{ij}/b\right) \\ z_{ij} &\leftarrow z_{ij} - c\,Nint\left(z_{ij}/c\right) \end{aligned} \quad . \tag{8.19}$$

The similarity of this to the cubic case is evident.

8.3.2 Unusual Periodic Boundaries

Simulation cells derived from the triclinic cell are not the only possibilities for periodic boundaries. The main criteria are that they are space filling and for practical purposes it helps if they have symmetry. In this category are the truncated octahedron, rhombic dodecahedron and the hexagonal prism shown in figure 8.3. These all happen to be Wigner-Seitz cells characterised by a high degree of reflection symmetry through the central point of the cell.

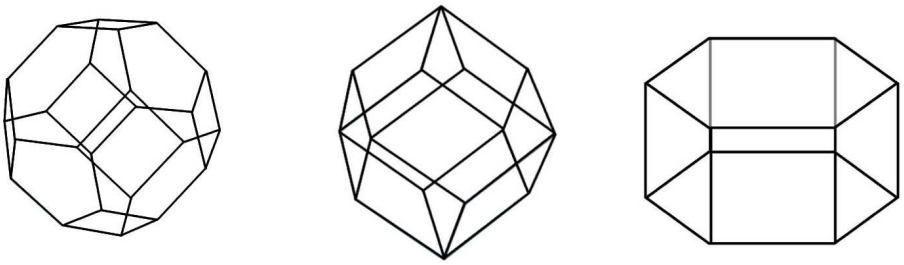

Figure 8.3: Truncated Octahedral, Rhombic Dodecahedral and Hexagonal Prism Simulation Cells

The main use of such boundaries is for solvation studies in the case of the truncated octahedron and rhombic dodecahedron, since they both offer a larger cut-off in relation to the cell volume than the simple cube. In the case of the hexagonal prism, this has proved convenient for studies of DNA molecules in crystalline arrangement or solution. (It is worth noting that the Ewald sum can be cast into a form compatible with these unusual boundary conditions. See references [77] and [78].)

The minimum image algorithms for these boundaries are derived from a parent orthorhombic cell that inscribes the simulation cell and shares a common centre. In all of these cases the inscribing orthorhombic cell has twice the volume of the simulation cell. (Note that while the inscribing cell is a useful construct when devising the minimum image scheme, it has no physical influence in the simulated system. In particular the N atoms of the system are not doubled to $2N$ to make the scheme work. However, for those uncomfortable with such esoteric simulation cells, it is always possible to reconstruct the fully populated parent orthorhombic cell and work with that. The price is the greater computational cost.)

For the simulation cells shown in figure 8.3, the minimum image algorithms are as follows.

8.3.2.1 The Truncated Octahedron:

The cell vectors for this case are represented by the matrix

$$\boldsymbol{h}_{toct} = a \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{8.20}$$

which is the same set of vectors as the parent cube. The minimum image scheme is shown in (8.21) in which the function $Sign(x)$ takes the value $+1$ if $x \geq 0$ and -1 if $x < 0$. Also $|x|$ represents the absolute value of x. In this algorithm the first three lines represent a possible shift of the difference $\vec{r}_{ij}$ to the minimum image inside of the cube of width a. The 4th line checks that $\vec{r}_{ij}$ is between opposite hexagonal faces of the truncated octahedron and, if this is not the case, ensures it is so in the following three lines. (The 4th line is based on the projection of the vector $\vec{r}_{ij}$ along the perpendicular vectors defining each hexagonal surface of the truncated octahedron.)

$$\begin{array}{l} x_{ij} \leftarrow x_{ij} - a\, Nint(x_{ij}/a) \\ y_{ij} \leftarrow y_{ij} - a\, Nint(y_{ij}/a) \\ z_{ij} \leftarrow z_{ij} - a\, Nint(z_{ij}/a) \\ if\,(|x_{ij}|+|y_{ij}|+|z_{ij}| > 0.75\,a)\{ \\ \quad x_{ij} \leftarrow x_{ij} - 0.5\,a\, Sign(x_{ij}) \\ \quad y_{ij} \leftarrow y_{ij} - 0.5\,a\, Sign(y_{ij}) \\ \quad z_{ij} \leftarrow z_{ij} - 0.5\,a\, Sign(z_{ij}) \\ \} \end{array}, \tag{8.21}$$

8.3.2.2 The Rhombic Dodecahedron:

The cell vectors for the rhombic dodecahedron are represented by the matrix

$$\boldsymbol{h}_{rhom}=a\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & \sqrt{2}\end{bmatrix}, \tag{8.22}$$

which define the parent orthorhombic cell, which in this case is a square prism with width and depth a and height $b=\sqrt{2}a$. The minimum image scheme is shown in (8.23). The resemblance to (8.21) is apparent. The first three lines again set the coordinate difference $\vec{r}_{ij}$ to its minimum image within parent orthorhombic cell. The 4th line checks that $\vec{r}_{ij}$ is between opposite rhombohedral faces of the rhombic dodecahedron and, in the following three lines, resets it within the cell if this is not the case.

$$\begin{array}{l} x_{ij}\leftarrow x_{ij}-a\,Nint\left(x_{ij}/a\right)\\ y_{ij}\leftarrow y_{ij}-a\,Nint\left(y_{ij}/a\right)\\ z_{ij}\leftarrow z_{ij}-b\,Nint\left(z_{ij}/b\right)\\ if\left(|x_{ij}|+|y_{ij}|+\sqrt{2}|z_{ij}|>a\right)\{\\ \quad x_{ij}\leftarrow x_{ij}-0.5\,a\,Sign\left(x_{ij}\right)\\ \quad y_{ij}\leftarrow y_{ij}-0.5\,a\,Sign\left(y_{ij}\right)\\ \quad z_{ij}\leftarrow z_{ij}-0.5\,b\,Sign\left(z_{ij}\right)\\ \}\end{array}, \tag{8.23}$$

8.3.2.3 The Hexagonal Prism:

The cell vectors here are represented by the matrix

$$\boldsymbol{h}_{hexa}=\begin{bmatrix}a & 0 & 0\\ 0 & a/\sqrt{3} & 0\\ 0 & 0 & b\end{bmatrix}, \tag{8.24}$$

which define an orthorhombic cell of width a, depth $a/\sqrt{3}$ and height b, where a is the distance between opposite rectangular faces of the hexagonal prism and b is arbitrary. The minimum image scheme is then

$$\begin{array}{l} x_{ij}\leftarrow x_{ij}-a\,Nint\left(x_{ij}/a\right)\\ y_{ij}\leftarrow y_{ij}-(a/\sqrt{3})\,Nint\left(\sqrt{3}\,y_{ij}/a\right)\\ z_{ij}\leftarrow z_{ij}-b\,Nint\left(z_{ij}/b\right)\\ if\left(\sqrt{3}|x_{ij}|+|y_{ij}|>a/\sqrt{3}\right)\{\\ \quad x_{ij}\leftarrow x_{ij}-0.5\,a\,Sign\left(x_{ij}\right)\\ \quad y_{ij}\leftarrow y_{ij}-a/(2\sqrt{3})\,Sign\left(y_{ij}\right)\\ \}\end{array}, \tag{8.25}$$

Again this works in much the same way as (8.21) and (8.23). The first three lines set $\vec{r}_{ij}$ to its minimum image within the parent orthorhombic cell. The remaining lines check if $\vec{r}_{ij}$ is between opposite rectangular faces of the hexagonal prism and adjusts it accordingly.

8.4 The Efficient Calculation of (Short Ranged) Pair Forces

8.4.1 The Direct Approach

As we have remarked already, the calculation of pair forces is the most expensive part of any simulation, so it pays to look closely at the components of the calculations to see where the procedure can be accelerated. The most obvious concern is to avoid unnecessary calculation, so for example when calculating the pair force $\vec{f}_{ij}$ (i.e. the force exerted by atom i on atom j), we should not repeat the whole calculation to obtain the force $\vec{f}_{ji}$ (the force exerted by atom j on atom i), since we know from Newton's third law that $\vec{f}_{ji} = -\vec{f}_{ij}$. This alone halves the cost of calculating the pair forces and it would be negligent not to incorporate this in a simulation program.

Where else can savings be made? There are clearly opportunities when using the cut-off condition, since its application allows us to discard a substantial proportion of possible pair interactions. A first attempt at suitable procedure might be as shown in (8.26).

```
1   for i=1→N−1{
2       for j=i+1→N{
3           x_ij ← MI(x_j − x_i)
4           y_ij ← MI(y_j − y_i)
5           z_ij ← MI(z_j − z_i)
6           r²_ij ← x²_ij + y²_ij + z²_ij
7           if (r²_ij < r²_cut){
8             r_ij ← √r²_ij
9             Φ ← Φ + Pot(r_ij)
10            f_ij ← Fce(r_ij)                    .        (8.26)
11            f^x_j ← f^x_j + f_ij x_ij/r_ij
12            f^y_j ← f^y_j + f_ij y_ij/r_ij
13            f^z_j ← f^z_j + f_ij z_ij/r_ij
14            f^x_i ← f^x_i − f_ij x_ij/r_ij
15            f^y_i ← f^y_i − f_ij y_ij/r_ij
16            f^z_i ← f^z_i − f_ij z_ij/r_ij
17          }
18      }
19  }
```

In this we have introduced the functions: $MI(x)$, is the minimum image of distance argument x ; $Pot(r)$, is the pair potential; and $Fce(r)$ is the *scalar* force for the distance r, defined as

$$f = -\frac{d}{dr} Pot(r). \tag{8.27}$$

The forms of these functions will depend on the circumstances. Lines 1-2, initiating the *double loop*, set the running values of i and j, with limits defined as 1 to $N-1$ and $i+1$ to N respectively, which confine each ij pair to one occurrence and also to prevent the redundant occurrence of pairs of the kind ii. The minimum image applications (lines 3-5) guarantee the closest (i.e. smallest) value of r_{ij}^2 in line 6. The application of the cut-off criterion in line 7 operates on the squared value of r_{ij}, since taking the square root at this stage is an expensive and unnecessary operation. The square root is taken at line 8, which is inside the cut-off condition. The value of r_{ij} obtained is used to calculate the potential energy contribution to the system potential, Φ, (line 9) and the scalar value of the pair force, f_{ij}, (line 10). The latter is finally used to accumulate atomic forces $\vec{f}_i$ and $\vec{f}_j$, while exploiting Newton's third law as required (lines 11-16). The quantities Φ, $\vec{f}_i$ and $\vec{f}_j$ are of course zeroed before the procedure (8.26) is entered into.

The procedure (8.26) is suitable for all short ranged pair forces and the real space component of the Ewald sum. It is reasonably efficient and satisfactory in many circumstances but it can be improved. One feature that is not optimal for many computers is the structure of the double loop over i and j, which have different run lengths from one cycle to the next. Since most processors work best with constant data streams, (the longer the better,) this makes it hard to drive the processor with maximum efficiency. This can be particularly telling in the case of computers with vector processing units. The structure of the loop also makes it difficult to devise an efficient approach for parallel computers.

For these reasons a different structure was devised by Brode and Ahlrichs [79], which takes the form shown in (8.28).(In this scheme line 13 should be replaced by lines 3-16 from procedure (8.26).) The double loop structure here is radically different. The loops are governed by three *integer* variables N_0, N_1 and N_2, which are defined in lines 1-3. The first loop, starting at line 4 is over an auxiliary index, k, which runs from 1 to $N/2$ and the second loop, starting at line 8, runs over index i from 1 to N_0. In most cases N_0 equals the number of atoms, N, but if N is an *even number*, then it is required that during the last pass of the loop over k, N_0 should equal $N/2$. This is arranged by the conditional lines 5-7. When N is an *odd number*, N_0 retains the value N always. At line 9 an index j is defined from indices i and k. If it exceeds N it is reset by the conditional lines 10-12 to $j-N$. The indices i and j are the atomic indices required by the pair force calculations.

The basis of the Brode-Ahlrichs scheme is the conversion of the set of ij pairs into a matrix of $(N-1)/2$ *columns* of N pair force calculations. There is no problem with this is if N is odd, since then $(N-1)$ is divisible by 2 and all columns can have N entries. However, if N is even, then the last column must be half the length of the others. Overall then, the loop over k has a guaranteed fixed length, while for most of the time the loop over i also has a fixed length. This maximises the opportunity for efficient number-crunching by the processor. At the same time the scheme only

generates unique ij pairs. It is recommended over the "obvious" approach of procedure (8.26).

```
1   N_0 ← N
2   N_1 ← N/2
3   N_2 ← (N−1)/2
4   for k = 1 → N_1 {
5     if (k > N_2) {
6         N_0 ← N_1
7     }
8     for i = 1 → N_0 {
9         j ← i + k
10        if (j > N) {
11            j ← j − N
12        }
13        ...etc...
14    }
15  }
```

(8.28)

8.4.2 The Verlet Neighbour List

The direct approaches described in the previous section can, perhaps surprisingly, made more efficient. It is readily apparent that the double loop over i and j actually performs two distinct tasks: the search for atom pairs within the cut-off distance; and the calculation of the pair interaction (potential and force) for pairs within range. The cost of calculating the interaction cannot be avoided, but the cost of the search for interacting pairs is another matter. If we are using a cubic simulation cell the largest cut-off permitted by the minimum image convention is half the width of the cell. The volume of the cut-off sphere is the approximately half of the cell volume (section 8.2), so the search for interacting pairs is only 50% successful. If the cut-off is reduced from this maximum, the search yields interacting pairs with rapidly diminishing efficiency. This is unfortunate because the search for interacting pairs involves calculating the separation vector between atom pairs, shifting this to the minimum image and calculating the square of the distance (see (8.26)) which has real time costs. It was to deal with this issue that Verlet introduced the concept of a neighbour list [3].

The Verlet neighbour list is simply a list we associate with each atom that stores the index of every atom with which it interacts within the cut-off radius. The list is constructed so that, in use, it avoids a given ij pair arising more than once. Once such a list has been composed, the calculation of the pair interactions may use it directly to obtain interacting atom pairs. If such a list could be constructed once only, say at the start of a simulation, there would be no need to conduct the search for interacting pairs every subsequent time step and the costs associated with the operation would vanish for the remainder of the simulation. However, as the system evolves in time, atoms will change position and the initial list will no longer reflect the true neighbourhood of each atom. At this point it becomes necessary to *update* the

neighbour list, which means reconstructing it afresh. This raises questions about how to determine when the update is needed and how the usable lifetime of a given neighbour list can be extended, since clearly, the longer the time between updates, the greater will be the efficiency. How this is dealt with will become apparent below, where we describe how the neighbour list is constructed.

Implementation of a Verlet neighbour list requires the introduction of two auxiliary integer arrays. The first is a population counter, $N_{pop}(N),$ which records how many atoms are within the neighbour list for each of the N atoms in the system. The second is the neighbour list itself, $L_{list}(N_{max}),$ which records the indices of the atoms each atom interacts with. The dimension N_{max} is some number comfortably above the expected number of pair interactions in the system and is of the order $2\pi\rho N r_{cut}^3/3,$ where ρ is the mean atomic density of the system. These arrays are constructed by the procedure presented in (8.29).

```
1   for i=1→N{
2     N_pop(i)←0
3   }
4   d²_cut←(r_cut+Δr)²
5   k←0
6   for i=1→N−1{
7     for j=i+1→N{
8       x_ij←MI(x_j−x_i)
9       y_ij←MI(y_j−y_i)
10      z_ij←MI(z_j−z_i)
11      r²_ij←x²_ij+y²_ij+z²_ij                    (8.29)
12      if (r²_ij<d²_cut){
13        k←k+1
14        if (k>N_max){
15          Error Abort()
16        }
17        N_pop(i)←N_pop(i)+1
18        L_list(k)=j
19      }
20    }
21  }
```

Lines 1-3 in procedure (8.29) set the initial population counter to zero for all atoms. Line 4 defines a new real variable, d_{cut}^2 which is nominally the square of the cut-off radius, but with an increment, $\Delta r,$ added. This is used later. Lines 6-7 reproduce the double loop structure seen in (8.26) and lines 8-11 calculate the minimum image separation of the ij pair of atoms. Lines 12-19 construct the neighbour list array L_{list} and the population count N_{pop} for atom $i.$ Note the error abort condition in lines 14-16, which are a precaution against N_{max} being defined too small. This is a non-recoverable error that requires intervention. Since the scheme constructs unique

ij pairs, the final neighbour list contains no duplicates.

The use of d_{cut}^2 in place of r_{cut}^2 at line 12 increases the radius of search for interacting pairs of atoms and will have the effect of slightly increasing the size of the neighbour list for each atom. Though it now includes a small contingent of non-interacting pairs, the possibility exists that these will become interacting pairs as the simulation proceeds. In this way the usable life of the neighbour list is increased, if at the expense of a small inclusion of redundant pairs. In line 17, the population count for atom i is incremented and in line 18 the index of atom j is added to the neighbour list.

How the neighbour list is used in the calculation of the forces and potential is shown in procedure (8.30). (In this scheme the line 7 should be replaced by lines 3-16 from (8.26). (It is important to note that the cut-off condition in this case uses r_{cut}^2 and not d_{cut}^2.) The loop over index i starts at line 2. The second loop is over the index k, (line 5) which is governed by the population number of the neighbour list of i though the accumulators N_s (start) and N_e (end), which are updated in lines 3-4, after initialisation of N_e at line 1. The index of atom j is found from the neighbour list, L_{list}, at line 7.

```
1   N_e ← 0
2   for i = 1 → N {
3     N_s ← N_e + 1
4     N_e ← N_e + N_pop(i)
5     for k = N_s → N_e {
6       j ← L_list(k)
7         ...etc...
8       }
9     }
10  }
```

(8.30)

To complete this description of the Verlet neighbour list we need some means to determine when the neighbour list should be updated as the simulation proceeds. This is actually quite simple. It is done by following the squared displacement of each individual atom from the moment the neighbour list is constructed. If the maximum squared displacement of any atom exceeds a pre-specified tolerance, this means at least one atom has moved sufficiently far for the neighbour list to be unsafe. At this point in the simulation a new neighbour list must be constructed. An obvious tolerance for this determination is $(\Delta r/2)^2$, since this relates the list update to the parameter Δr, which is used to extend the neighbour list beyond the bare cut-off condition. The procedure is outlined in (8.31).

In (8.31), the function $MI(x)$ calculates the minimum image value of x, while $Max(x, y)$ returns the larger value of x and y. The arrays x^0, y^0, z^0 store the values of the atomic coordinates at the time step of the previous neighbour list update.

```
1   d²max ← 0
2   for i = 1 → N
3     δx ← MI(xi − x⁰i)
4     δy ← MI(yi − y⁰i)
5     δz ← MI(zi − z⁰i)
6     d²max ← Max(d²max, δx² + δy² + δz²)
7   }
8   if (d²max > (Δr/2)²){
9     Update Neighbour List
10  }
```
(8.31)

To get a rough idea of the performance of the force calculations (in the manner described in section 8.2) using the Verlet neighbour list, we can rewrite equation (8.1) in the modified form:

$$t_{force} = (f(\alpha + \alpha') + \chi\alpha')N(N-1)/2, \tag{8.32}$$

This is in a form appropriate for using the Verlet neighbour list, in which α' again represents the time needed to calculate the inter-particle separation and α represents the time to calculate the pair force and potential from the obtained inter-atomic separation. The parameter χ is the ratio

$$\chi = N_{updates}/N_{simulation}, \tag{8.33}$$

where $N_{simulation}$ is the number of time steps in the complete simulation and $N_{updates}$ is the number of times the neighbour list is reconstructed during the simulation. Hopefully, χ turns out to be much less than $1,$ which would significantly reduce the time spent searching for interacting pairs in the simulation overall. Unfortunately, it is difficult to specify what $N_{updates},$ and hence $\chi,$ will be *a priori* since it depends on the speed the atoms are diffusing in the system, which in turn depends in an unpredictable manner on the system density and temperature. However, it is usual to try to set the parameter Δr so that the neighbour list is updated every 20~30 time steps (which can be done with practice!) which means that $\chi \approx 0.03 - 0.05$ in most simulations. This represents a major time saving.

The Verlet neighbour list offers a major saving in the cost of molecular dynamics simulations, but it has an Achilles heel. Unfortunately it is evident from equation (8.32) that overall the method still scales as $O(N^2),$ which threatens to make the simulation of large systems prohibitively costly. But it pays to look more closely at (8.32) as it tells us something more. Separating out the force calculation and pair search from t_{force} gives two terms:

$$\begin{aligned} t^a_{force} &= f(\alpha + \alpha')N(N-1)/2, \\ t^b_{force} &= \chi\alpha' N(N-1)/2 \quad . \end{aligned} \tag{8.34}$$

The term t^b_{force} is unquestionably of order $O(N^2),$ no matter how small the factor

$\chi\alpha'$ may be in practice, but on the other hand t^a_{force} can be made to behave differently. In a system of reasonably uniform density ρ, volume V_0, and with a cut of r_{cut}, we can write f as

$$f = \frac{4\pi r_{cut}^3}{3V_0}, \tag{8.35}$$

where

$$V_0 = N/\rho. \tag{8.36}$$

It follows that t^a_{force} can be written as

$$t^a_{force} \approx \frac{2}{3}\pi r_{cut}^3(\alpha+\alpha')\rho N. \tag{8.37}$$

If the system size, N, is increased while ρ and r_{cut} remain fixed, then t^a_{force} evidently scales with order $O(N)$. Which means, under these circumstances, that the $O(N^2)$ scaling of simulations using the Verlet neighbour list is due entirely to t^b_{force} i.e. to the search for interacting pairs. To make further progress, we require some means to address this issue. This is achieved through the linked cells method, which is applicable to systems in which r_{cut} is small in relation to the system dimension.

8.4.3 The Linked Cells Method

The linked cells method was devised by Hockney and Eastwood [16]. We will describe the method as it is applied to a general triclinic simulation cell, of which the cubic and orthorhombic simulation cells are special cases.

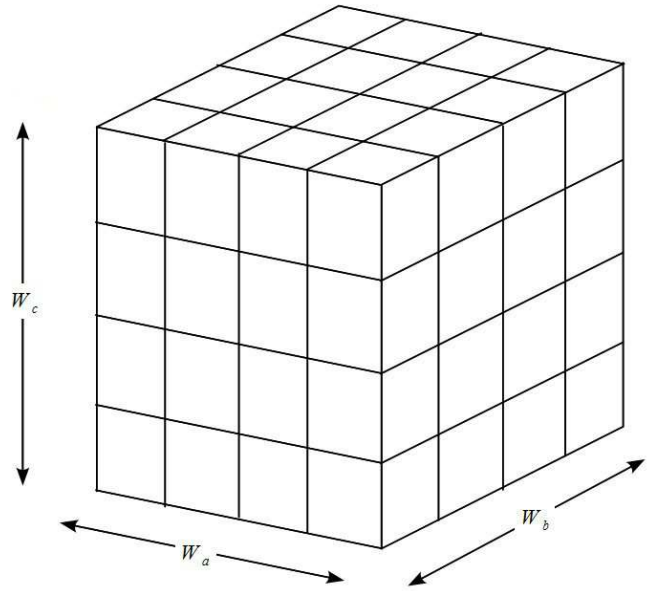

Figure 8.4: Division of Simulation Cell into Sub Cells

The linked cell method begins with the division of the simulation cell into smaller, uniform sub cells, as exemplified for a cubic simulation cell in figure 8.4. A key point of this division is that the *perpendicular width* between opposite faces of the sub cells must be larger than the cut-off distance r_{cut}, (but preferably not very much larger). To guarantee this we must use the perpendicular widths of the simulation cell when constructing the division of the cell. These are obtained from the formulae

$$W_a = \vec{A} \cdot \frac{(\vec{B} \times \vec{C})}{|\vec{B} \times \vec{C}|}, \qquad W_b = \vec{B} \cdot \frac{(\vec{C} \times \vec{A})}{|\vec{C} \times \vec{A}|}, \qquad W_c = \vec{C} \cdot \frac{(\vec{A} \times \vec{B})}{|\vec{A} \times \vec{B}|}, \tag{8.38}$$

in which $\vec{A}, \vec{B}, \vec{C}$ are the cell vectors defining the simulation cell. The origin of these formulae is shown in figure 8.5. (Should any of (8.38) return a negative result, the absolute value should be used.)

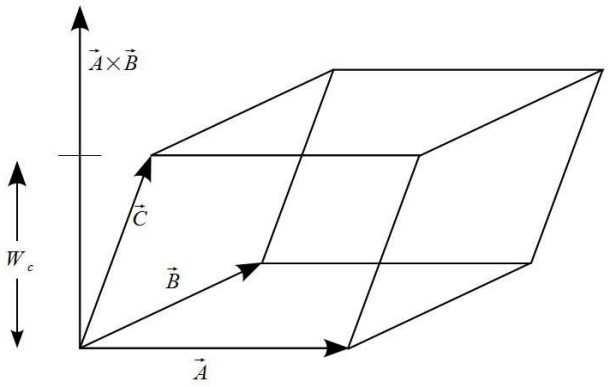

Figure 8.5: Perpendicular Width W_c of a Triclinic Simulation Cell

The number of sub cells, (N_a, N_b, N_c), in the directions perpendicular to the faces of the simulation cell is then given by

$$N_a = Int.\left(\frac{W_a}{(r_{cut} + \delta)}\right), \qquad N_b = Int.\left(\frac{W_b}{(r_{cut} + \delta)}\right), \qquad N_c = Int.\left(\frac{W_c}{(r_{cut} + \delta)}\right), \tag{8.39}$$

where δ is an arbitrary and very small number introduced to ensure the widths of the sub cells cannot exactly equal r_{cut}. The function $Int.(x)$ takes the integer part of x. The actual widths of the sub cell are

$$w_a = \frac{W_a}{N_a}, \qquad w_b = \frac{W_b}{N_b}, \qquad w_c = \frac{W_c}{N_c}. \tag{8.40}$$

The total number of sub cells, N_{cells}, is then the product

$$N_{cells} = N_a N_b N_c. \tag{8.41}$$

In a system composed of N atoms the average occupancy, $\bar{n}_{occ}$, of each sub cell is simply

$$\bar{n}_{occ} = N / N_{cells}. \tag{8.42}$$

If we consider an atom in a given sub cell, then it is clear from the dimensions of the sub cell that all the atoms that interact with it are to be found either in the same sub cell or one of its immediately neighbouring sub cells (of which there are 26). This at once removes the $O(N^2)$ dependency of the search for interacting pairs since it is longer over $N(N-1)/2$ possibilities, but over $N(27\bar{n}_{occ}-1)/2,$ and providing the system size, $N,$ is scaled up according to the relation

$$N = \bar{n}_{occ} N_a N_b N_c, \tag{8.43}$$

where $\bar{n}_{occ}$ remains constant and $N_a, N_b, N_c,$ are integers, the search procedure will scale as $O(N).$ The condition (8.43) ensures the sub cells are always commensurate with the system size.

We shall now implement these ideas. The algorithm presented in (8.44) is based on the concept of a linked list. In this context a linked list is a self referencing list which defines a "chain" of atoms that all belong to the same sub cell. This requires two integer arrays: $L_{head}(k),\ k=1\ldots N_{cells},$ which marks the start (or "head") of the linked list for a cell k; and $L_{link}(n),\ n=1\ldots N,$ which is the actual linked list. These arrays are initialised to zero in lines 1-6.

```
 1 for k=1→N_cells{
 2     (L_head)_k ← 0
 3 }
 4 for i=1→N{
 5     (L_link)_i ← 0
 6 }
 7 for i=1→N{
 8     s_a ← h_xx^-1 x_i + h_xy^-1 y_i + h_xz^-1 z_i
 9     s_b ← h_yx^-1 x_i + h_yy^-1 y_i + h_yz^-1 z_i
10     s_c ← h_zx^-1 x_i + h_zy^-1 y_i + h_zz^-1 z_i
11     L_a ← Min(Int.(N_a(s_a+0.5)), N_a-1)
12     L_b ← Min(Int.(N_b(s_b+0.5)), N_b-1)
13     L_c ← Min(Int.(N_c(s_c+0.5)), N_c-1)
14     k ← 1+L_a+N_a(L_b+N_b L_c)
15     if((L_head)_k=0){
16         (L_head)_k ← i
17     } else {
18         (L_link)_i ← (L_head)_k
19         (L_head)_k ← i
20     }
21 }
```

(8.44)

The arrays L_{head} and L_{link} are populated in the lines 7-21, which loop over the atom index i, starting at line 7. The first task, in lines 8-10, is to convert the atomic coordinates x_i, y_i, z_i to fractional coordinates s_a, s_b, s_c using the method described in section 8.3.1.1. This allows the calculation of three *integer coordinates*, L_a, L_b, L_c, which identify the sub cell location in the 3D array of sub cells. These are calculated in lines 11-13. Note that 0.5 is added to the fractional coordinate to offset the fact that the coordinate origin is at the simulation cell centre. The integer result is obtained by the function $Int.(x)$. Acceptable results for L_a, L_b, L_c lie between the limits $[0,N_a-1], [0,N_b-1], [0,N_c-1]$ respectively. The $Min(n,m)$ function, which returns the minimum of integers n and m, is used to force the assignment of atoms that are *exactly* on the highest coordinate faces of the simulation cell into the highest allowed sub cell. The coordinates L_a, L_b, L_c, are used to generate a unique index, k, for the sub cell, using the formula on line 14.

Having established that sub cell k contains atom i, the information is then used to construct the linked list. Firstly, a check of the content of array element $(L_{head})_k$ is made at line 15 to determine if this is the first found occupant of the sub cell k. If it is, then atom i is recorded as the head of the chain for sub cell k in line 16. If it is not the first, then the existing content of $(L_{head})_k$ is moved into the linked list at element $(L_{link})_i$ in line 18 and atom i becomes the new head of the chain, $(L_{head})_k$, in line 19. This procedure is repeated every time a new occupant of the sub cell is found.

The information stored in completed arrays L_{head} and L_{link} allows determination of all the occupants of a given sub cell, k, in the following manner. Firstly, the content of the head of the chain, $(L_{head})_k$, provides the index of the first atom, say i. The content of the linked list element $(L_{link})_i$ then provides the identity of the second atom, say j, after which the element $(L_{link})_j$ provides the third atom index and so on down the linked chain of atoms. The final atom in the chain returns a zero value for $(L_{link})_i$, which indicates there are no more atoms belonging to the sub cell k.

It should be noted that the reconstruction of the linked list occurs every time step in a simulation, but fortunately this is not too costly because the procedures involved are only of order $O(N)$. We should also mention an important requirement for procedure (8.44), which is that it assumes all atoms are already within the simulation cell. If any are not, the index k obtained on line 14 may reference a nonexistent sub cell! External atoms must be transferred to their image positions in the simulation cell using the usual minimum image procedures.

The linked list is used when calculating the pair interactions in the following manner. We start by defining the three auxiliary *integer* arrays, $t_a(n), t_b(n), t_c(n), n=1\ldots14$, which have the indicated content:

$$\begin{aligned} &data: (t_a)_n(n=1\ldots14)\leftarrow\{0,1,1,0,-1,0,1,1,0,-1,-1,-1,0,1\} \\ &data: (t_b)_n(n=1\ldots14)\leftarrow\{0,0,1,1,1,0,0,1,1,1,0,-1,-1,-1\} \\ &data: (t_c)_n(n=1\ldots14)\leftarrow\{0,0,0,0,0,1,1,1,1,1,1,1,1,1\} \end{aligned} \tag{8.45}$$

These arrays define the *integer displacements* of the sub cells neighbouring any given sub-cell, for example, from the data in (8.45) we see that $((t_a)_2, (t_b)_2, (t_c)_2)$ is the integer displacement $(1, 0, 0),$ which indicates a neighbouring sub cell in the positive A-direction. Just 14 elements are defined in (8.45) and they represent half of the neighbouring 26 sub cells surrounding a given sub cell, plus the sub cell itself with displacement $(0, 0, 0).$ Only half the sub cells are required, as the use of all of them would result in double counting of ij pairs in the final scheme. The exploitation of these arrays becomes clearer in (8.46), which is the procedure for calculating the pair interactions in the linked cells method.

```
1  for k=0→N_cells−1{
2      k_1←k+1
3      i_c←Int.(k/(N_a N_b))
4      i_b←Int.(k/N_a)−N_b*i_c
5      i_a←Mod(k, N_a)
6      for n=1→14
7          j_a←Bracket(i_a+(t_a)_n, N_a)
8          j_b←Bracket(i_b+(t_b)_n, N_b)
9          j_c←Bracket(i_c+(t_c)_n, N_c)
10         k_2←1+j_a+N_a(j_b+N_b j_c)
11         i←(L_head)_k1
12         while(i>0){                                  ,        (8.46)
13             j←(L_head)_k2
14             while(j>0){
15                 if(k_1≠k_2 || j>i){
16                     ...etc...
17                 }
18                 j←(L_link)_j
19             }
20             i←(L_link)_i
21         }
22     }
23 }
```

The first line of (8.46) is a loop over all the sub cells in the system $(N_{cells}),$ though the actual index of the sub cell is k_1 defined on line 2. Lines 3-5 define the three integer coordinates (i_a, i_b, i_c) of the sub cell k_1 in the system. The function $Int.(x)$ returns the integer part of x and $Mod(n, m)$ returns the remainder from the integer division of n by $m.$ The loop over $n,$ commencing at line 6, runs over sub cells that are immediate neighbours of $k_1.$ The integer coordinates of the neighbouring sub cell are generated in lines 7-9 and the index of the sub cell, $k_2,$ is calculated at line 10. To obtain an integer coordinate of a neighbouring sub cell, one of the integer displacements, $(t_\alpha)_n,$ (where $\alpha=a, b, c,$) is added to the corresponding integer coordinate i_α and the result is passed to the integer function $Bracket(i, n),$ which is

presented in procedure (8.47) below. This returns an integer that lies between the limits $[0,n-1],$ so that if sub cell k_1 is on an outer face of the simulation cell, the neighbour k_2 will be from the opposite face, in accordance with the periodic boundary condition.

Next the contents of sub cells k_1 and k_2 are processed. The loop over atoms i in sub cell k_1 is controlled by lines 11, 12 and 20. Line 11 takes the first value of i from the head of the linked list, $(L_{head})_{k_1},$ and subsequent values are taken from the linked list element $(L_{list})_i$ at line 20. The loop over atoms $i,$ which starts at line 12, runs until $(L_{list})_i$ returns a zero value. The loop over atoms j is controlled in the same way, by lines 13, 14 and 18. At line 15, inside the loop over $j,$ a check is made to determine if sub cells k_1 and k_2 are identical, in which case a method is necessary to prevent double counting of ij pairs. This takes the form of the additional condition on line 15, relevant only when $k_1=k_2,$ that index j must be greater than i if an interaction is to be calculated. Line 16 in (8.46) is where lines to calculate the the pair interaction must be inserted. These are identical to the lines 3-16 of procedure (8.26).

```
1 integer function Bracket(i,n){
2     if(i<0){
3         i←i+n
4     }else if(i≥n){
5         i←i-n
6     }
7     return i
8 }
```
(8.47)

This completes the description of the linked cell implementation. Let us look at its scaling properties.

The time to obtain the pair forces in a linked cell algorithm can be written as

$$t_{force}=\frac{N}{2}(\bar{n}_{cut}-1)\alpha+\frac{N}{2}(27\bar{n}_{occ}-1)\alpha', \tag{8.48}$$

in which the first term on the right is the time to calculate the pair interaction and the second term is the time to search over atom pairs. The quantity, $\bar{n}_{cut},$ is the average number of atoms within the cut-off sphere, which is given by

$$\bar{n}_{cut}=\frac{4}{3}\pi r_{cut}^3\rho, \tag{8.49}$$

Where ρ is the average atomic density, which is

$$\rho=\bar{n}_{occ}/v_{cell} \tag{8.50}$$

In which v_{cell} is the volume of a sub cell. Inserting (8.50) into (8.49) and the result into (8.48) gives

$$t_{force}=\frac{N}{2}\left[(f_{lc}\bar{n}_{occ}-1)\alpha+(27\bar{n}_{occ}-1)\alpha'\right], \tag{8.51}$$

In which f_{lc} is the volume ratio of the cut-off sphere to the sub cell. All the terms in the square brackets are constants, so evidently t_{force} scales linearly with N (subject to equation (8.43) holding as N increases).

It is revealing to compare (8.51) with its counterpart for the basic pair force method (8.1) described in section 8.2, when applied to the same physical system of Lennard-Jones atoms. For simplicity we assume the simulation cell is cubic and any scale up in system size follows the condition (8.43). We can write the ratio of the force time for the basic algorithm (8.1) to that for the linked cells (8.51) as

$$S=\frac{(f\alpha+\alpha')(N-1)}{\left[(f_{lc}\bar{n}_{occ}-1)\alpha+(27\bar{n}_{occ}-1)\alpha'\right]}. \tag{8.52}$$

This is also the factor by which the linked cells method will be faster than the basic method.

If, for simplicity, we assume the cut-off is very slightly less than $(1/m)th$ of the simulation cell width, where m is the number of sub cells in each dimension (i.e. $N_{cells}=m^3$), then it can easily be shown that $f\sim 4/N_{cells}$ while $f_{lc}\sim 4$ and $\bar{n}_{occ}=N/N_{cells}.$ Now if we further assume that $\bar{n}_{occ}$ is sufficiently large that $(4\bar{n}_{occ}-1)\sim 4\bar{n}_{occ}$ and $(N-1)\sim N,$ then (8.52) reduces to

$$S\sim\frac{(4\alpha+N_{cells}\alpha')}{(4\alpha+27\alpha')}. \tag{8.53}$$

This shows that the basic and linked cells methods spend the same amount of time calculating the pair interactions, but the two methods differ in the time spent searching for atom pairs. This formula implies that when $N_{cells}=3\times3\times3$ then $S\sim1$ and the two methods will hardly differ. If we make the (not unrealistic) assumption that $\alpha\sim4\alpha',$ in the Lennard-Jones system, then with $N_{cells}=5\times5\times5$ we have $S\sim3.3,$ so the linked cells method will be significantly faster. Likewise, if $N_{cells}=10\times10\times10,$ then the speed up factor is $S\sim24.$ Ultimately, as N_{cells} becomes the dominant factor in (8.53), we can write

$$S\sim N_{cells}/43=m^3/43, \tag{8.54}$$

which shows that the speed up grows linearly with the number of sub cells or as the cube of the number of sub cells in each dimension. Evidently, for very large systems the speed up is likely to astonishing.

The linked cell method is highly recommended. It is simple to implement and extremely powerful. Enormous systems can be simulated for relatively little cost. However it is not without its limitations. The most obvious of which is that it cannot be used efficiently if the cut-off is *not* small in relation to the system dimensions. If fewer than 3 sub cells can be constructed in each principal direction, it is not worth attempting. However things improve rapidly as the number of sub cells increases in each dimension. Also, it is a method very sensitive to correct use of numerical precision, such that careless specification of value (using 0.5 instead of 0.5d0 in Fortran for instance) can have dangerous consequences when the program is trying to

decide where atoms belong in the linked cell structure. Furthermore, the execution of the program can be critically dependent on implicit assumptions about the physical state of the system under investigation. For example, there is an implicit assumption that the system has a reasonably uniform density if it is to function efficiently. Also, in simulating systems under extreme conditions it sometimes happens that atoms end up in physical locations that cannot be mapped into the linked list. This means the program may break down in such a way as to imply an error in the linked list build, when it is actually something wrong with the specification of the control parameters. One needs to watch out for this.

8.5 Integrating the Equations of Motion

For completeness, we present explicit procedures for the Integration blocks for the leapfrog (8.55) and velocity Verlet (8.56)-(8.57) algorithms. These procedures are straightforward: r_i^x, v_i^x, f_i^x *etc.* are the components of the position, velocity and force vectors and m_i the mass of the $i'th$ atom. These must surely be among the simplest of integration algorithms! Note that, as presented, neither of these algorithms reset the atomic positions to their minimum image counterparts. Indeed it is not essential to do so, but experience shows it is to be recommended. In the case of velocity Verlet note also the division of the time step Δt by 2 in the velocity integration, since the velocity update always occurs over half a time step.

Leapfrog:

$$
\begin{aligned}
&1\ \textit{for } i=1\rightarrow N\{\\
&2\quad v_i^x \leftarrow v_i^x + f_i^x * \Delta t/m_i\\
&3\quad v_i^y \leftarrow v_i^y + f_i^y * \Delta t/m_i\\
&4\quad v_i^z \leftarrow v_i^z + f_i^z * \Delta t/m_i\\
&5\quad r_i^x \leftarrow r_i^x + v_i^x * \Delta t\\
&6\quad r_i^y \leftarrow r_i^y + v_i^y * \Delta t\\
&7\quad r_i^z \leftarrow r_i^z + v_i^z * \Delta t\\
&8\ \}
\end{aligned}
\tag{8.55}
$$

Here, the atomic forces are calculated afresh once the new atomic positions are known.

Velocity Verlet (stage 1):

$$
\begin{aligned}
&1\ \textit{for } i=1\rightarrow N\{\\
&2\quad v_i^x \leftarrow v_i^x + f_i^x * \Delta t/2m_i\\
&3\quad v_i^y \leftarrow v_i^y + f_i^y * \Delta t/2m_i\\
&4\quad v_i^z \leftarrow v_i^z + f_i^z * \Delta t/2m_i\\
&5\quad r_i^x \leftarrow r_i^x + v_i^x * \Delta t\\
&6\quad r_i^y \leftarrow r_i^y + v_i^y * \Delta t\\
&7\quad r_i^z \leftarrow r_i^z + v_i^z * \Delta t\\
&8\ \}
\end{aligned}
\tag{8.56}
$$

After stage 1 of the velocity Verlet, the new atomic positions are known and so the

forces can be recalculated ready for stage 2.

Velocity Verlet (stage 2):

$$\begin{array}{ll} 1 & for\ i=1\rightarrow N\{ \\ 2 & \quad v_i^x \leftarrow v_i^x + f_i^x * \Delta t/2\, m_i \\ 3 & \quad v_i^y \leftarrow v_i^y + f_i^y * \Delta t/2\, m_i \\ 4 & \quad v_i^z \leftarrow v_i^z + f_i^z * \Delta t/2\, m_i \\ 5 & \} \end{array} \tag{8.57}$$

Note there is no recalculation of the atomic forces after stage 2.

8.6 Calculating Time Correlation Functions

The calculation of time correlation functions is fundamental to the analysis of molecular dynamics data. As we saw in chapter 3 this amounts to calculating integrals of the form

$$C(t)=\frac{1}{T-t}\int_0^{T-t}\left(A(t+u)-\bar{A}\right)\left(B(u)-\bar{B}\right)du, \tag{8.58}$$

where $C(t)$ is the *auto-correlation* function of the time functions $A(t)$ and $B(t)$, when $A(t)=B(t)$ and the *cross-correlation* function when $A(t)\neq B(t)$. Quantities $\bar{A}$ and $\bar{B}$ represent the mean values of $A(t)$ and $B(t)$ respectively over the time interval $[0,T]$. The discrete form of this integral, suitable for numerical computation is

$$C_k=\frac{1}{N_S-k}\sum_{i=0}^{N_s-k-1}\left(A_{(i+k)}-\bar{A}\right)\left(B_i-\bar{B}\right) \qquad with \qquad \{k=0\ldots,N_s-1\}, \tag{8.59}$$

in which N_s is the number of consecutive time samples of A and B taken from the equilibrated simulation. Arrays A_i and B_i represent *time ordered data samples* i.e.

$$A_i\equiv A(i\Delta t), \qquad B_i\equiv B(i\Delta t), \tag{8.60}$$

etc. where Δt is the time interval between consecutive samples.

The form of (8.59) indicates that C_k is calculated as an average of N_s-k terms, which means that at one extreme C_0 is averaged over N_s terms and at the other extreme C_{N_s-1} is an average of one term! The quality of the average thus diminishes progressively as index k increases and it makes little sense to proceed too far. Correlation mostly persists in molecular dynamics on a time scale from 10 fs to 5 ps, which can be used to set a practical limit on k if some idea of the time scale is known beforehand. The simulation on the other hand should run over a time period at least an order of magnitude longer than the supposed correlation time.

A procedure for obtaining the discrete correlation function is shown in (8.61), in which $N_t \Delta t$ is the chosen practical time limit of $C(t)$. In this procedure lines 1 to 8 are concerned with calculating the averages $\bar{A}$ and $\bar{B}$ and lines 9 to 12, subtract these averages from the instantaneous values. The correlation calculation between lines 13 and 20 is a direct implementation of (8.59). Note the dimension of arrays $A(i)$ and $B(i)$ is N_s and the dimension of $C(k)$ is N_t, where ideally $N_s > 10 N_t$.

```
1   Ā ← 0
2   B̄ ← 0
3   for i = 0 → N_s − 1
4       Ā ← Ā + A_i / N_s
5       B̄ ← B̄ + B_i / N_s
6   }
7   for i = 0 → N_s − 1
8       A_i ← A_i − Ā
9       B_i ← B_i − B̄
10  }
11  for k = 0 → N_t − 1
12      C_k ← 0
13      for i = 0 → N_s − k − 1
14          C_k ← C_k + A_{i+k} B_i / (N_s − k)
15      }
16  }
```
(8.61)

This procedure is noteworthy for its simplicity, but when large volumes of data require processing, it is not the most efficient way to proceed. A surprisingly faster method is based on the discrete Fourier transform which is also known as the Fast Fourier Transform (FFT). This approach derives from the well known fact that the Fourier transform of a correlation function can be obtained from the product of the Fourier transforms of the functions being correlated (see chapter 3, section 3.8). Thus if functions $A(t)$ and $B(t)$ have the Fourier transforms $\tilde{A}(f)$ and $\tilde{B}(f)$ respectively, where f is the *frequency,* (t^{-1}), *and*

$$\tilde{A}(f) = \int_{-\infty}^{\infty} A(t) \exp(-2\pi i f t) dt \tag{8.62}$$

etc. then we can write

$$\tilde{C}(f) = \tilde{A}^*(f) \tilde{B}(f), \tag{8.63}$$

where $\tilde{C}(f)$ is the Fourier transform of the correlation function $C(t)$ and $\tilde{A}^*(f)$ is the complex conjugate of $\tilde{A}(f)$. In principle then, to obtain the correlation function we need only apply the FFT separately to arrays $A(i)$ and $B(i)$ to obtain $\tilde{A}(k)$ and $\tilde{B}(k)$, form the product $\tilde{C}(k) = \tilde{A}^*(k) \tilde{B}(k)$, and invert the FFT to obtain $C(i)$. The speed of the FFT is so great that this rather round-about way of proceeding obtains

the result extremely quickly.

However, as might be expected, there are some details that must be taken care of before this can be used as a practical scheme. These stem largely from the fact the the discrete Fourier transform possesses many subtle properties distinct from the integral form (8.62) which we cannot detail here, but are well explained in the book by E.O. Brigham [80], chapter 6. Here, in (8.64) we merely provide a recipe that takes account of these properties.

1. Subtract the average $\bar{A}$ from all $A(i)$;
2. Subtract the average $\bar{B}$ from all $B(i)$;
3. Double the lengths of arrays $A(i)$ and $B(i)$ from N_s to $2N_s$ by appending N_s zeroes.
4. Obtain the FFT, $(\tilde{A}(i))$, of array $A(i)$;
5. Obtain the FFT, $(\tilde{B}(i))$, of array $B(i)$;
6. Form the product array $\tilde{C}(i)=\tilde{A}^*(i)\tilde{B}(i)$;
7. Obtain the inverse FFT, $C(i)$, of $\tilde{C}(i)$;
8. Set $C(i)=C(i)/(2N_s(N_s-i))$, for $i=0\rightarrow N_t-1$, with $N_t\leq N_s$.

(8.64)

Note that in this recipe, the arrays $A(i)$ and $B(i)$ are *real*, while $\tilde{A}(i)$ and $\tilde{B}(i)$ are *complex*, so $A(i)$ and $B(i)$ need to be cast as complex arrays prior to the FFT application. This is somewhat wasteful in terms of memory usage, but it is interesting to note that Brigham [80], chapter 9, section 9.3, describes how two real arrays can be packed into a complex array and simultaneously Fourier transformed. This makes better use of the computer memory and reduces the number of FFT calls overall, so it is recommended. Note the division by (N_s-i) on line 8 above is required by the formula (8.59) and also includes a FFT normalisation factor: $1/2N_s$.

The procedures (8.61) and (8.64) are suitable for calculating the correlation function at the close of a simulation, when the arrays $A(i)$ and $B(i)$ are complete. Sometimes however it is desirable to calculate correlation on-the-fly as the simulation proceeds. This can be achieved by the following, which is called at regular intervals during the molecular dynamics cycle

```
1  Ā ← i_s * Ā/(i_s+1) + A_{i_s}/(i_s+1)
2  B̄ ← i_s * B̄/(i_s+1) + B_{i_s}/(i_s+1)
3  k ← Mod(i_s, N_t)
4  G_k ← B_{i_s}
5  for j=0→Min(i_s, N_t-1){
6     C_j ← M_j C_j/(M_j+1) + A_{i_s} * G(Mod(k-j+N_t, N_t))/(M_j+1)
7     M_j ← M_j+1
8  }
```

(8.65)

In (8.65) the current values of variables A and B are A_{i_s} and B_{i_s}, where the sampling index i_s ranges from 0 to N_s-1 and counts the N_s consecutive values used in the correlation calculation. Lines 1 and 2 accumulate the averages $\bar{A}$ and

$\bar{B}$ (where $\bar{A}$ and $\bar{B}$ have been zeroed at the start of the simulation). The procedure is called at regular intervals with new values A_{i_s} and B_{i_s}. Each value of B_{i_s} is stored consecutively in the array $G(k):(k=0\rightarrow N_t-1)$, up to a maximum of N_t values, after which older values are cyclically overwritten. The overwrite is controlled by the index k (calculated on line 3) which marks the location where the new value should be inserted. Index k thus also marks the element in G that is current with A_{i_s}. The array $C(j):(j=0\rightarrow N_t-1)$ (initialised to zero) accumulates the correlation function, which is calculated in lines 5 to 8. On line 6 the $j'th$ element of C accumulates the product of A_{i_s} and B_{i_s-j} as required. The integer array $M(j):(j=0\rightarrow N_t-1)$ (initialised to zero) counts the number of contributions made to C_j and increases by 1 on each pass (line 7). When all intended N_s values of A and B have been processed, the final correlation function is calculated using

$$\begin{array}{ll} 1 & for\, i=0\rightarrow N_t-1\{ \\ 2 & \quad C_i\leftarrow C_i-\bar{A}\,\bar{B} \quad , \\ 3 & \} \end{array} \tag{8.66}$$

which merely subtracts the mean values of A and B from C_i. Overall, the structure of procedure (8.65) is intended to maximise numerical accuracy.

In principle the final procedure (8.66) can be avoided if, in (8.65), the current averages $\bar{A}$ and $\bar{B}$ (obtained from lines 1 and 2) are subtracted from A_{i_s} and B_{i_s} respectively in lines 6 and 4. Though the current averages are relatively inaccurate to begin with, they quickly improve and any remaining error becomes insignificant if the overall number of samples taken (N_s) is high.

Chapter 9

Molecular Dynamics on Parallel Computers

9.1 Introduction

In the preceding chapters we have outlined some fundamental algorithms and methods but said little about the kind of computers on which they may be run. In truth all of them so far have assumed a serial computer with a single processor. This is excusable on the grounds that it makes it easier to explain and understand the principles behind the methods described, but it does not properly reflect molecular dynamics as an evolving methodology. The fact is, researchers today are running molecular dynamics simulations on parallel computers with many processors and it is essential to have some understanding of the subject. In this section we provide a brief overview of parallel computing as it applies to molecular dynamics and discuss the most important issues.

We begin with an idealised model of what a parallel computer consists of. This shown in figure 9.1. This represents a *distributed memory* computer, which is the most general kind. It consists of a number of processing nodes $\{P_0, \ldots, P_{L-1}\}$, linked together by a *communication network*. Each processor has its own *memory bank* $\{M_0, \ldots, M_{L-1}\}$, which is not directly accessible by any other processor.

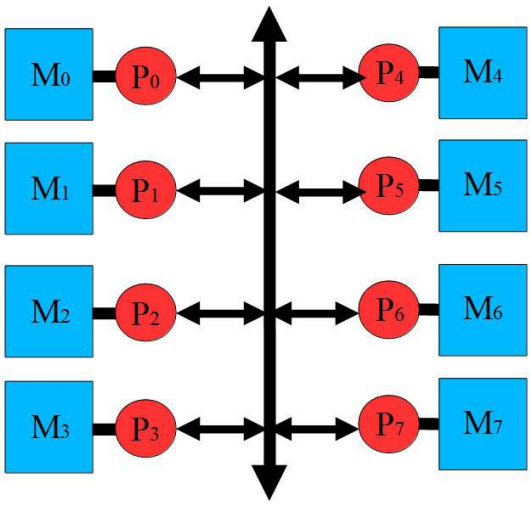

Figure 9.1. An Idealised Distributed Memory Parallel Computer

As a concept it is not that different from an ordinary computer network, except that in this case, all the processing nodes are usually the same – the so called *homogeneous architecture*. Most modern parallel computers depart from this simple picture in a

number of ways. The most common difference is for the distributed memory to be integrated to make one large memory accessible to all processors. This makes it a *shared memory* computer, which generally functions differently from the distributed kind, but nevertheless can work in the same way as a distributed memory machine if required. Another common difference is for the processing nodes to consist of more than one processor, so that each node is a kind of shared memory computer in its own right. Such "fat" nodes bring more power and versatility to the computer, but sometimes this bestows different capabilities to the nodes giving the overall machine a *heterogeneous* architecture. Such departures from the homogeneous architecture make it harder to produce efficient software without incorporating specialised features which reduce software portability, so we will confine this account to the homogeneous machines.

9.1.1 Communication and Parallel Computers

The distinguishing characteristic of distributed memory computing that is not encountered in serial programming is *message passing*. Since none of the processing nodes has access to all the data that resides in the machine's total memory, it is necessary for the data to be passed between nodes to get the compute task done. As a result a parallel program must manage these communications as well as the numerical processing and these operations are a key component of the software. Fortunately all parallel machines have software libraries to handle the message passing and the application program can access this capability through procedural calls. The most common library for this is the Message Passing Interface (MPI) which is callable via Fortran and C. In this library are procedures for *sending* and *receiving* messages between individual nodes, *broadcasting* information from one node to many and *gathering* information from many nodes to one. Which of these procedures is to be used and when is the choice of the programmer and the choice matters if the program is to be efficient and trouble free. This is the central concern in parallel program design.

To implement efficient programs it is important to realise that message passing between nodes is a two stage process. The first thing that must happen is that a connection must be made between the nodes, a form of "handshaking" that establishes contact and exchanges *meta*-information about the data message to be transferred - so that appropriate data buffers can be set up for example. While this is going on no actual user data is transferred. This represents a delay in the passing of every message which is called *latency*. The next stage is that the data message is sent over the communications network. The time taken for this to complete is a function of the physical characteristics of the network and of the size of the message (i.e. volume of data) being transferred. At its simplest level there is a linear relationship between message size and the time to completion, but in most computers there is a hierarchy in the communications network, such that physically near nodes communicate faster than more distant ones. The data transfer may not even be directly node to node, but proceed via intermediate nodes.

It appears difficult to formulate any principles of good practice in this area without a detailed knowledge of the machine construction and use of such knowledge would not necessarily be of benefit if the software was run on a different machine. Nevertheless

there are certain rules of thumb that are useful to programmers:

1. All communication should be kept to the minimum necessary, since all messages incur time penalties.

2. It is better to send one large message than many small ones; the latency of one message is much less than the accumulated latency of many.

3. It is better to use direct communication than global. Direct communication means passing specific data to a specific node, while global means broadcasting all the data to all or many nodes. Algorithms that overly rely on global communications are less efficient than ones that mostly use direct communication.

4. If possible use *asynchronous* communication rather than *synchronous*. Asynchronous communication, also called *unblocked* communication, passes control back to the user program before the message transfer is complete, while the synchronous (or *blocked*) communication retains control until the end of the transfer. Thus asynchronous communication, in principle, allows data processing (number crunching) to go on while the transfer takes place and it has the potential to save time.

The possibility of asynchronous communication is interesting, but it is not always practical. If the data to be processed has not yet arrived, a wait is inevitable. The potential is only there if different data can be processed in the meantime. Also, it has to be said that asynchronous communication does not always deliver the hoped for gains. The data transfer and the concurrent numerical processing may need to access the same areas of computer memory, creating a data acquisition conflict. Usually the memory access for communication takes precedent and this delays numerical processing, so numerical performance is unfortunately downgraded. Also the latency of message passing cannot be avoided by this approach; it is always a source of delay.

9.1.2 Load Balancing

The principle of parallel computing seems simple enough: using many nodes on a task will get the job done faster than one. But the bad news is that this does not always happen. One of the key factors in this is *load balancing*, by which we mean using all the machine nodes to *equal capacity*. This has a number of aspects to it:

1. The workload is shared equally between nodes.
2. The memory requirement is shared equally between nodes.
3. The nodes are working *concurrently*.

All of these must be satisfied at the same time. That the workload is shared equally is a necessary requirement but not sufficient on its own. It is quite possible for the nodes to do equal amounts of work, but not do it concurrently, so everything waits for the last node to complete. Concurrence is crucial. If the memory requirement is not equal for all nodes, this implies that the message passing between nodes is not optimal and more messages will be dealt with by some nodes than others, which

means delay. It also implies that different nodes must be handling tasks of a different kind, which makes workload balance difficult. It also follows that the largest simulation that can be performed will be limited by the most overloaded node, with spare capacity on the others left untapped.

Load balancing is not a feature of serial computing, so it can easily be overlooked when attempting parallel programming for the first time. It is important to give the matter due thought.

9.1.3 Parallel Scaling

Parallel computing adds extra dimensions to the issue of performance scaling, beyond scaling with the size of the physical system being simulated. In parallel computing, scaling is also affected by the number of nodes employed and the amount of message passing that takes place. These inevitably make things much more complicated, but it is important to be aware of the relationships between the these factors to understand how to use parallel programs efficiently. In parallel computing three different kinds of scaling can occur:

1. *Type 1 scaling*. This might also be called *zeroth* scaling from a parallel perspective, since it does not involve a change in the number of nodes. It addresses the question of how a parallel program responds when the workload is increased while keeping the number of nodes fixed. For serial programs we expect that doubling the workload means the program takes twice as long to run to completion, but this is not generally the case for parallel programs. Sometimes this makes things more efficient and sometimes less. Usually, increasing the workload implies increasing the amount of data that must be transferred between nodes and this has a deleterious effect. This can be serious and we will see examples of this later.

2. *Type 2 scaling*. This is more commonly called *strong* scaling and it concerns how the overall speed of a program increases with an increasing number of nodes while keeping the size of the computational problem fixed. It is the kind of scaling that most parallel programmers want to achieve. If the number of nodes is doubled, does this double the overall performance speed or, equivalently, halve the time to job completion? In practice, linear scaling of this kind is only possible if the program has been designed well, but even then ideal scaling is usually restricted to certain *regimes* of operation. It is therefore important to understand the architecture of the program to ensure that it is run under the most favourable regime and is therefore operating efficiently.

3. *Type 3 scaling*. This is also called *weak* scaling. It addresses the question of how performance changes when the workload *and* the number of nodes are increased in proportion. Ideally one would like to see the time to completion remain constant under such conditions, but as with Type 1 scaling the outcome may be affected by the additional communication implied. The name weak scaling comes from the notion that it should be easier to satisfy expectations with this kind of scaling than with Type 2, and this often seems to be the case. Not surprisingly therefore it is the form of scaling that is most often reported. It

is arguable however is that it is better than Type 2 scaling at reflecting how parallel computers are actually used in practice. Researchers are usually more interested in tackling bigger simulations than they are with accelerating smaller ones.

When designing, reviewing or testing programs these three kinds of scaling should always be borne in mind. Programs that show bad Type 1 scaling are bad programs and should be avoided. Good type 3 scaling is acceptable, particularly if large scale simulations are intended. The ideal is good Type 2 scaling since it means that the parallel algorithm is effective. However, it must be said that good scaling does not necessarily mean the program is highly efficient, as we shall see in the following section. In later sections we describe some parallel molecular dynamics methods and comment on the quality of their scaling.

9.1.4 Numerical Processing, Communication and Parallel Efficiency

All parallel programs involve numerical processing and communication between the nodes. Here we consider the relative importance of these in constructing efficient programs.

If we consider only synchronous (blocked) communications, we can say that the time, t_s^L, required per time step for a simulation on L nodes is a sum of the processing time, t_p^L, and the communication time, t_c^L,

$$t_s^L = t_p^L + t_c^L, \tag{9.1}$$

which can be written as

$$t_s^L = t_p^L(1 + R_{cp}^L), \tag{9.2}$$

Where R_{cp}^L is the ratio of the communications time to the processing time:

$$R_{cp}^L = t_c^L / t_p^L. \tag{9.3}$$

R_{cp}^L is of fundamental importance in determining how a parallel program will behave with regard to efficiency and scaling, as we will show.

If we could assume that for a given parallel program R_{cp}^L is a constant then equation (9.2) reveals that if we increase the number of nodes allotted to the task, then we can reasonably expect that (in a properly load balanced program), t_p^L will be inversely proportional to the number of nodes, L. Thus, for example, doubling L will halve t_p^L. This behaviour is typical of Type 2 scaling. So it appears that if ever R_{cp}^L could be considered a constant, ideal scaling with the number of nodes will result.

Unfortunately, experience shows that R_{cp}^L is seldom, if ever, a constant. It usually

varies with the number of nodes and also with the size of the simulation. However, if an operating *regime* can be found where $R_{cp}^{L} \ll 1,$ then the factor $(1+R_{cp}^{L})$ will be *practically* constant and ideal scaling behaviour will be observed while the program remains within that regime. Users of parallel programs should therefore always strive to find the regime where the condition $R_{cp}^{L} \ll 1$ holds for the program and always to use the program under that regime. To obtain this knowledge some understanding of how the program has been parallelised is essential.

Something else equation (9.2) reveals is that because $R_{cp}^{L} > 0,$ the time per time step, $t_s,$ is always greater than the processing time, $t_p.$ This is obviously because the running program requires communication, but since this is not something inherent in serial processing, where we have $t_s^1 = t_p^1,$ this indicates that parallel processing inevitably introduces a degree of inefficiency. In fact, we can use R_{cp}^{L} to define the *parallel efficiency,* $\epsilon_{par},$ of a molecular dynamics program in the following way:

$$\epsilon_{par} = \frac{t_s^1}{L t_s^L} = \frac{t_p^1}{L t_p^L (1+R_{cp}^{L})} = \frac{1}{(1+R_{cp}^{L})}, \tag{9.4}$$

where we make the reasonable assumption that $t_p^1 = L t_p^L.$ Equation (9.4) shows that the parallel efficiency can only approach 100% as $R_{cp}^{L} \to 0.$ It is therefore apparent that ideal scaling and maximum parallel efficiency both require that R_{cp}^{L} be as small as possible, which means keeping all communications to a minimum.

Equation (9.4) also reveals that R_{cp}^{L} can be made smaller (and the parallel efficiency higher) by increasing t_p as well as by reducing $t_c.$ This means that when the numerical processing is made *less* efficient, the parallel efficiency *improves*! This is a clear indication that ϵ_{par} defines the effectiveness of the parallelisation and not the effectiveness of the programming overall. Parallel programs with a high ϵ_{par} are therefore not necessarily maximally efficient. However, what we can say is that on a given number of nodes a more optimised numerical code will always reduce the simulation time, even though it may cause the program to scale less well. So this is not an excuse for bad programming.

With regard to asynchronous communications it should be evident that the above analysis represents a worst case scenario for parallel programming. Using asynchronous communications will help to keep R_{cp}^{L} small, but cannot make parallel programs 100% efficient. It is even arguable that constructing a program to ensure the minimum number of messages passed is a better strategy for making programs efficient, than making all communications asynchronous. The latter option pays off most when the former has been achieved.

9.2 Parallel Algorithms for Molecular Dynamics

There are at least five distinct approaches to doing molecular dynamics on a distributed memory parallel computer. They are very different in kind, though it is

quite possible to combine them to produce hybrid algorithms. They are not all equally successful nor equally widely used, but have all appeared in the literature at some time. These are as follows

1. Parallel replication;
2. Task farming;
3. Systolic loops;
4. Replicated data;
5. Domain decomposition.

We shall discuss each of these in turn.

9.2.1 Parallel Replication

Parallel replication is a rather naive method of achieving parallelisation, but it has practical application. The idea is a simple one (see figure 9.2): if there are L processing nodes, why not run L simulations concurrently? This is an obvious application if many simulations are required. Unfortunately it effectively exploits the parallel computer as a mere throughput engine rather than a true parallel device so the full benefits of parallel computing are not realised – one cannot attempt simulations of large systems for example. It works best when the simulations are all of the same system, so that there is a good chance they will run more or less concurrently and finish at the same time. For example the simulations can be set at different state points to explore the phase properties of one system, or at the same state point, but with different initial configurations, to achieve enhanced statistical sampling overall. This approach is sometimes called *Cloning*.

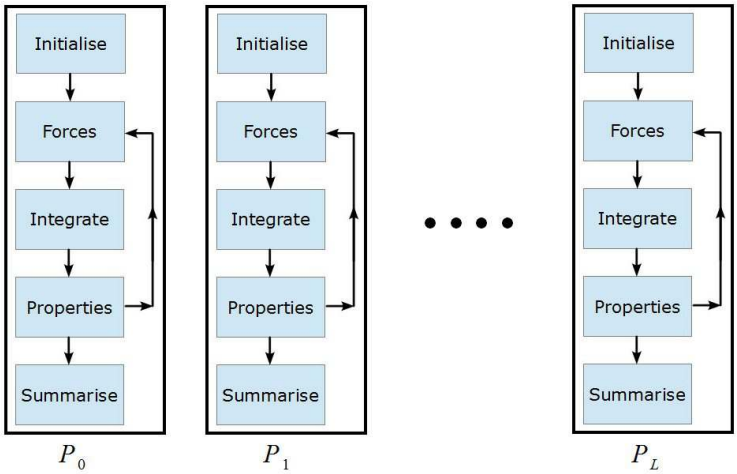

Figure 9.2. The Parallel Replication of many Simulations

This form of parallel computing has several real advantages[22]. No communication is required between the nodes, so the efficiency is high. Good scaling of types 1, 2 and 3 is guaranteed, provided the workload is defined as the number of simulations to be run *or* the size of the simulations, as appropriate. However the management of so many concurrent simulations brings its own problems in terms of data output and storage. It is therefore advisable to calculate all the required system properties *on the fly* rather than output the raw data for later calculation. Even then, downloading and saving the configuration data at the end of a simulation to provide restart capability can itself be a considerable challenge.

Initially it was thought that this approach offered no advantage to researchers wanting to achieve long time scales in their simulations, since the individual simulations on each node progress at the same rate as on a serial processor. This effectively meant no new science could be done with it. However, it later emerged that a combination of parallel replication with the principles of chemical kinetics produces two viable methods which really can address the long time scale issue. These methods, known as Parallel Tempering [81] and Parallel Replica Dynamics [82], are described in detail in Chapter 10. Apart from these interesting applications however, it has to be admitted that parallel replication leaves something to be desired. It is not a general solution to the problem of parallel molecular dynamics and its use is quite narrowly defined.

22 It should also be said that the parallel replica approach offers a convenient way to run Monte Carlo simulations on parallel computers.

9.2.2 Task Farming

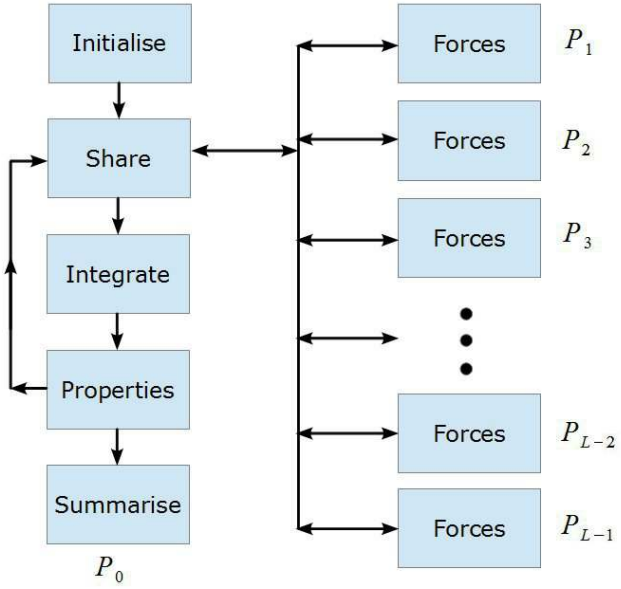

Figure 9.3. A Task Farming Approach to Forces Calculations

Task farming is an inherently hierarchical approach to parallel computing and is therefore sometimes known as the *Master-Slave* method. It is very often the first considered approach to parallelising a computational procedure, since it has an immediate intuitive appeal, which is a great pity, since it is fraught with severe difficulties. In fact, with regard to molecular dynamics, it should be avoided at all costs. The basic assumption of the approach is that the computation to be parallelised needs to be decomposed into manageable pieces so that each processing node can be allotted a proportionate amount. This unfortunately makes decomposing the data into manageable pieces the key issue of the computational task, rather than the actual calculations using the data. The result is that an undue amount of time is lost in peripheral activity related to managing the data . Furthermore there is a heavy price to be paid in the inter-node communication implicit in this approach.

A typical task farming approach to molecular dynamics is shown in figure 9.3. On the right is the basic program running on the node P_0. It is essentially a standard, serial molecular dynamics program, except that the Forces block has now been replaced by a *Share* block, the function of which is to divide up the pair forces and pass them in "packets" to the processing nodes P_1 to P_{L-1}, where they will be calculated and the results returned to P_0. Node P_0 is therefore the "master" in this scheme, while nodes P_1 to P_{L-1} are the "slaves". This is, admittedly, a rather simplistic attempt at parallel molecular dynamics but it is one which is often tried because it makes a real attempt to tackle the central problem, which is how to speed up the calculation of the pair forces. Unfortunately it is not a good idea.

The first issue associated with this kind of approach is that, while P_0 is dividing up the pair forces, the nodes P_1 to P_{L-1} are idle, which corresponds to a significant

loss of processing power. Also, because node P_0 has to send different information to the slave nodes the data cannot simply be broadcast to all at the same time, so each must queue for its allotted data to arrive. This inevitably entails delay. Furthermore it is hard to avoid the circumstance in which processors finish their allotted work before others have even started, a situation that gets more likely as the number of nodes increases and the share of work each slave node receives gets smaller. It is practically impossible to achieve load balancing under such circumstances. Things can get worse however, since each slave node must return its results back to the master, P_0, it most likely has to deal with an avalanche of incoming messages, which threaten to overload its capacity to deal with them (a situation known as "deadlock"). This clearly gets worse as the number of nodes increases. Indeed, all of these problems can get worse by merely increasing the size of the simulation while keeping the number of nodes fixed.

On the face of it then, task farming fails on a number of levels as a parallel strategy: it does not load balance easily; it does not scale well with the number of processors or with the size of the simulation (failing Type 1, 2 and 3 scaling conditions!); it is prone to deadlocking; and compared with other approaches it is difficult to code up into a practical program. No doubt it is possible by clever programming to deal these issues, employing sophisticated queuing, multiple hierarchies of masters and slaves, dynamic load balancing schemes *etc.* Some of the deficiencies can also be ameliorated if the master node is considerably more capable than the slaves. This implies a different kind of computer architecture from that shown in figure 9.1; a special purpose computer perhaps, or one that is very heterogeneous in design. Such machines are not common. Alas, all of these issues are a major distraction to the research scientist seeking to understand molecular processes.

9.2.3 Systolic Loops

Systolic loop methods have much to commend them in molecular dynamics and it is regrettable that they are not more widely used in the leading molecular dynamics programs. Unlike the more common Replicated Data and Domain Decomposition approaches described later, systolic loops have no counterpart in serial processing and are thus uniquely parallel in concept. There is a wide number of possible schemes for exploiting systolic loops, but we shall describe only one: the SLG-G algorithm [83], which is known to be extremely efficient for systems involving pair forces.

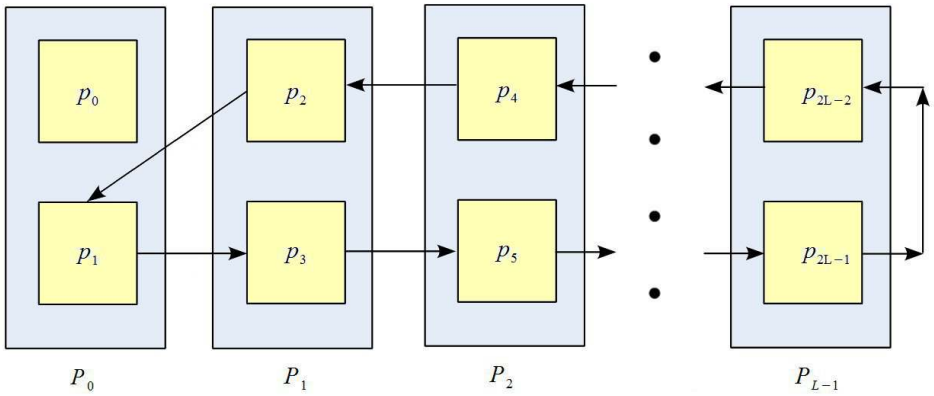

Figure 9.4. The SLS-G Systolic Loop Algorithm

The SLS-G algorithm is shown in figure 9.4. It is assumed we have L processing nodes, $(P_{0,}\ldots, P_{L-1}),$ arranged in a linear "chain" with *bi-directional* communication links between each pair along the chain. (This configuration is within the capability of the idealised parallel computer shown in figure 9.1.) The complete, original configuration data arrays, $(\vec{r}_i, \vec{v}_i, \vec{f}_i),$ for all atoms $i=1,\ldots, N,$ are divided into $2L$ "packets," $(p_{0,}\ldots, p_{2L-1}),$ of size $n=N/2L$ atoms and allocated to the nodes as shown. Then, for each molecular dynamics time step the following procedure is followed:

1. On each node, the pair interactions for all the atoms *within* each packet are calculated. (Subject to any cut-off condition that may apply.)

2. Next, all the pair forces between atoms in *different* packets are calculated.

3. Each data packet is then passed in the direction of the arrows shown in figure 9.4. This usually means the data are moved to the next node in the chain (either forward or back), but in the case of node $P_{L-1},$ the packet p_{2L-1} merely replaces packet p_{2L-2} on the same node. Meanwhile on node $P_{0,}$ packet p_0 remains in place. The movement of each packet in this way is referred to as a *systolic pulse*.

4. Step 2 and 3 are repeated a further $2L-2$ times. Each time calculating a new set of pair interactions on each processor. After a total of $2L-1$ systolic pulses, each packet returns to its original, "home" node with a complete set of force arrays, $\vec{f}_i,$ appropriate for the packets it hosts. These can be used with $\vec{r}_i$ and $\vec{v}_i$ to compute the atomic motion for one time step. (Note that in this algorithm it is not necessary to move the velocity data, $\{\vec{v}_i\},$ since it is not needed until the motion is calculated and can be left on the home node.)The data packets are then ready for the next time step.

The SLS-G algorithm neatly arranges for each packet of data to pass by every other packet in the system during the complete systolic loop cycle. (This may not be immediately obvious, but it can be proven mathematically [83].) This allows opportunity for every pair force in the system to be calculated. Furthermore it provides a complete and uniform distribution of the data across the nodes, so it fulfils the requirements for good load balancing. There is even some scope for asynchronous communication, in which the positional data is passed on to the next processor while the interaction calculations are being done. However, there is likely to be some memory contention while this takes place, which will reduce the hoped for gains. Overall however, this is a very favourable approach to parallel molecular dynamics.

What are the drawbacks? These appear when the cut-off is short in relation to the system size. There is no avoiding the need to pass atomic data around, despite the fact that many possible pair interactions are to be disregarded. Thus the communication requirement remains fixed even though the processing requirement

has diminished. It also has to be admitted that systolic loop schemes for many-body interactions are more difficult to construct. For methods like the Ewald sum however there is no difficulty. The real space part can be calculated directly using the SLS-G algorithm and the reciprocal space part can be obtained as a global sum of contributions calculated for the groups of atoms on each processor.

The viability of the SLS-G algorithm as a method for parallel molecular dynamics makes it worthwhile to look at its scaling properties in some detail. Bearing in mind that the number of atoms in each packet is $n=N/2L,$ the processing time per node can be written as

$$t_p=n(2Ln-1)(f\alpha+\alpha')+2ng\beta, \tag{9.5}$$

in which α, α' and β respectively represent the times for calculating one pair interaction, one inter-particle distance and the update of the position and velocity of one atom. Also f is the time reduction factor due to application of the cut-off to the pair interaction calculations. The constant g is 1 when the leapfrog algorithm is used and 1.5 for the velocity Verlet algorithm. The quantity $n(2Ln-1)$ is number of all possible pair interactions each node must process, obtained by summing the contributions from *intra*- and *inter*-packet interactions. Multiplying this by the number of nodes $L,$ gives the expected result for the whole system: $N(N-1)/2.$ We can therefore write (9.5) equally well as

$$t_p=\frac{N}{2L}\left((N-1)(f\alpha+\alpha')+2g\beta\right). \tag{9.6}$$

With regard to communication, there are $2L-1$ systolic pulses to consider, each of which transports $2/3$ of the configuration data of n atoms (leaving out the velocity data). Thus the communication time per time step is

$$t_c=(2L-1)\left(\frac{2}{3}n\gamma+\delta\right), \tag{9.7}$$

in which γ represents the time to send the (full) configuration data for one atom and δ is the latency of the message. Equation (9.7) can also be written as

$$t_c=\frac{(2L-1)}{3L}\left(N\gamma+3L\delta\right). \tag{9.8}$$

It follows from (9.6) and (9.8) that the fundamental ratio R_{cp}^L is

$$R_{cp}^L=\frac{2(2L-1)(N\gamma+3L\delta)}{3N\left((N-1)(f\alpha+\alpha')+2g\beta\right)} \tag{9.9}$$

Having obtained this expression, our next concern is to find the regime under which the algorithm will demonstrate good scaling, which means determining when $R_{cp}^L\to 0.$ If we consider the circumstances where $N\gg L,$ i.e. the number of atoms vastly exceeds the number of nodes and rewrite (9.9) as

$$R_{cp}^{L}=\frac{2(2L-1)(\gamma+3L\delta/N)}{3(N-1)\left((f\alpha+\alpha')+2g\beta/(N-1)\right)}, \tag{9.10}$$

then we can see that as N increases, the dependence of R_{cp}^{L} on the communication latency, δ, becomes progressively less significant, as does the integration constant β. If these are neglected we obtain

$$R_{cp}^{L}\approx\frac{2(2L-1)\gamma}{3N(f\alpha+\alpha')}, \tag{9.11}$$

which shows that R_{cp}^{L} diminishes approximately as $1/N$. In other words the requirement that $R_{cp}^{L}\to 0$ (for maximum parallel efficiency) becomes fulfilled when N is large enough. This helps to define the working regime of the program. We also note that, in this extreme, R_{cp}^{L} increases linearly with L, so we also expect that as the number of nodes increases, the parallel efficiency will fall and departure from linear scaling will result. Quite when this occurs however, is determined by the parameters α, α', γ as well as the number N.

Unfortunately, if we are employing the SLS-G algorithm at all, it is unlikely that N will ever be so large that this regime can be achieved unambiguously. SLS-G is basically an all-pairs algorithm, which means is is not well suited to very large systems. In which case it is necessary to obtain empirical values for the parameters $\alpha, \alpha', \beta, \gamma$ and δ and calculate R_{cp}^{L} for various combinations of L and N from (9.9). Low values of R_{cp}^{L} will identify the regime of linear scaling. Some feel for this can also be obtained through experience of using the program and this is often enough to avoid a catastrophic misuse of the computing resource. The objective is always to ensure the program is running as efficiently as it can.

A qualitative idea of how the SLS-G algorithm scales can be obtained by substituting speculative values for the parameters in (9.9) and calculating the parallel efficiency, ϵ_{par}, in (9.4) as a function of L and N. The result is a plot like that shown in figure 9.5, which was obtained using the parameters: $\alpha=1, \alpha'=0.25, \beta=1, \gamma=0.1, \delta=10, g=1$ and $f=0.5$. The number of nodes, L, ranges from 8 to 16384 and the number of atoms, N, from 10 to 100,000.

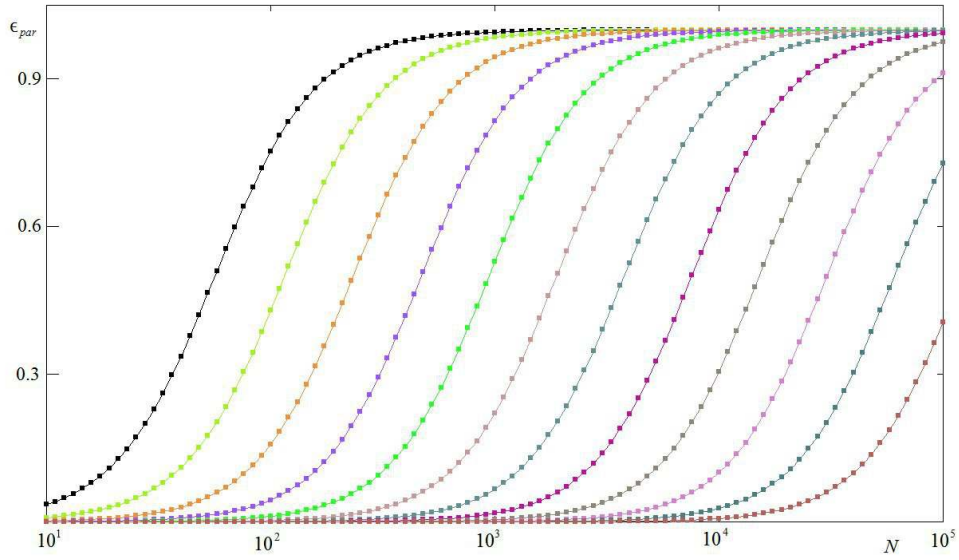

Figure 9.5. The Parallel Efficiency of the SLS-G Algorithm.

The parallel efficiency for 8 nodes is plotted (in black) on the far left of figure 9.5 and the number of nodes doubles with each plot towards the right. With the adopted parameters the algorithm becomes fully efficient on 8 nodes just below 1000 atoms, but as the number of nodes increases, this is progressively delayed. With 64 nodes (blue line) this is not achieved until N approaches 10,000 atoms, and pushing on to 1024 processors (red line) we would need to simulate 100,000 atoms to recover maximum efficiency. Somewhere around 10^4 atoms is probably the largest system one would want to simulate with this algorithm. Plots like figure 9.5 are useful in deciding the correct number of nodes for optimal efficiently.

With regard to performance of the SLS-G algorithm, figure 9.6 plots the time per time step, t_s, (measured in units of the parameter α) as a function of the number of atoms, N, for different numbers of nodes, L, ranging from 2 to 128, doubling in number each time. On the right hand side the number of nodes increases as we descend the graph. Each plot has two clearly distinct regions: a horizontal region associated with smaller values of N and a linear, rising region associated with larger values of N. The first of these regions corresponds to a regime in which the algorithm is communications dominated and where increasing N has little effect on the execution time t_s, which implies that the processing nodes are under-utilised. Eventually however, increasing N results in the domination by numerical processing and the increasing workload causes a rise in t_s. The slope of these (logarithmic) plots at this extreme is ~2.0, which is consistent with this being an $O(N^2)$ algorithm.

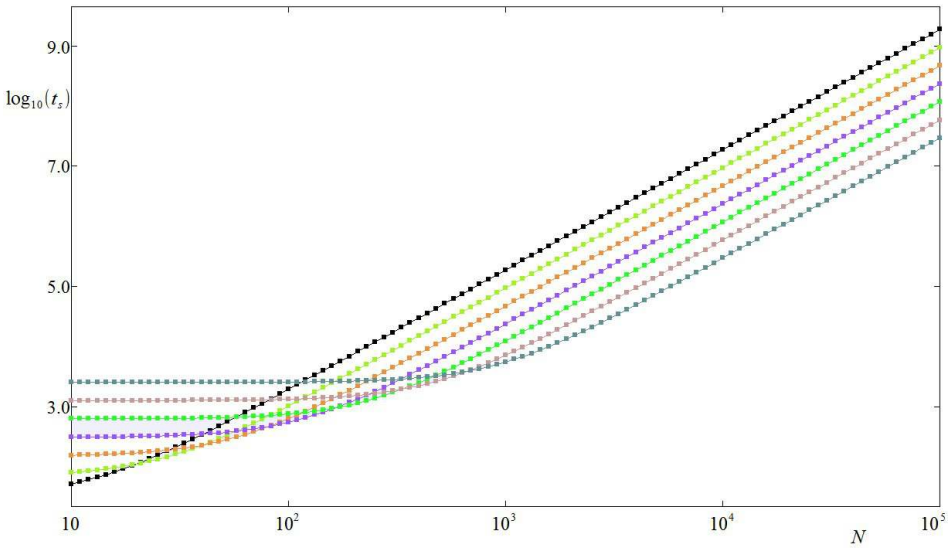

Figure 9.6. The Performance of the SLS-G Algorithm

In the communications dominated regime we see that using larger numbers of nodes *increases* the execution time, since the communications become even more dominant. The effect of this is to progressively delay the onset of the processing dominant regime. However, in the processing dominant regime, increasing numbers of nodes works to *reduce* the execution time. This behaviour is in keeping with what we know about the parallel efficiency, since the algorithm is most efficient for large $N.$ We also note that, for large N the plots become parallel lines with an equal vertical distance between them. Since the number of nodes doubles as we descend the plots, this pattern indicates that good type 2 (strong) scaling is occurring. In fact it is easily shown that the expected vertical (downwards) distance between the plots is $\approx -\log_{10}(2),$ which is obtained here. Equally we may argue that in this region, the equal horizontal distances between the plots, is indicative of good type 3 (weak) scaling, since both the workload and the number of nodes is increasing without increasing $t_s.$ In this case the horizontal distance between the plots is expected to be $\log_{10}(\sqrt{2}),$ as is observed here.

9.2.4 Replicated Data

Replicated Data (RD) is a valuable approach to parallel computation. Though it does have some significant limitations, a number of major programs in the public domain have adopted this approach, at least for the first generation of parallel programs. It is an all-pairs approach, like the SLS-G algorithm above, and is perfectly adequate for parallel computers with small numbers of processor nodes (<100) and for simulations with less than 10,000 atoms. A wide range of useful science can be done within these limits.

The RD molecular dynamics algorithm is presented in figure 9.7. Superficially it resembles the Parallel Replication algorithm shown in figure 9.2, in that each node apparently runs a full simulation. However, in this case each node is running the *same* simulation. To achieve this, each node has a *complete replica* of all the arrays defining

the atomic configuration i.e. $\{\vec{r}_i\}, \{\vec{v}_i\}$ and $\{\vec{f}_i\}$. But whenever a significant body of calculation is required, the nodes arrange to do a different share of the work. The opportunities for this work sharing most obviously occur in the Forces, Integration and Properties blocks of the program.

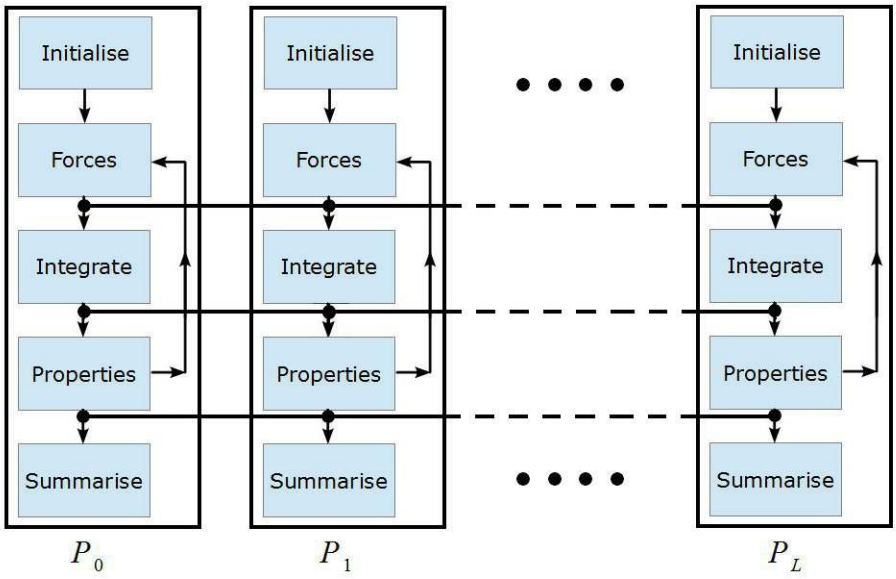

Figure 9.7. The Replicated Data Molecular Dynamics Algorithm

We will leave aside the Properties block for a moment and deal with the others first. In the case of the Forces block, there are $N(N-1)/2$ pairs of atoms to consider. This means that each of the L nodes must handle, as near as possible, $N(N-1)/2\mathrm{L}$ pairs in a load balanced algorithm. In the Integration block, each node must handle the integration for close to N/L atoms. Since each node has a full specification of the atomic positions, the calculation of the pair forces and potential can proceed without communication between the nodes. All that is required for this is a simple procedure to identify the set of pair interactions each node is to handle. For this purpose a simple modification of the Brode-Ahlrichs scheme from chapter 8 is needed.

The scheme (9.12) differs from the Brode-Ahlrichs scheme presented in (8.28) on just one line. On line 8, the index identifying the node on which the scheme is running (i_{node}) is used to offset the start of the count over atoms, and the stride through the atoms is of step $L,$ (the number of nodes) rather than the default of $1.$ This is all that is needed to ensure each node calculates a different set of pair interactions.
It is evident that after these calculations have finished, the forces acting on individual atoms are unlikely to be complete and we have to restore the full replication of the force arrays on each node. To achieve this we make use of a *global array summation*, which is a common parallel operation in which the individual elements of an array that is present on all nodes becomes the sum of all corresponding array elements on all

```
1   N_0 ← N
2   N_1 ← N/2
3   N_2 ← (N−1)/2
4   for k = 1 → N_1 {
5     if (k > N_2) {
6       N_0 ← N_1
7     }
8     for i = i_node + 1 → N_0 step L {
9       j ← i + k
10      if (j > N) {
11        j ← j − N
12      }
13      ...etc...
14    }
15  }
```
(9.12)

nodes. This is represented by the formula

$$A_i^G = \sum_{p=0}^{L-1} A_i^p , \tag{9.13}$$

in which A_i^G represents the $i'th$ element of the global array $\boldsymbol{A}^G$, and A_i^p is the $i'th$ element of array A on individual processor p. After the summation, the *local* array A becomes A^G on all nodes. This operation is applied to all the atomic force arrays $\boldsymbol{f}^x, \boldsymbol{f}^y, \boldsymbol{f}^z$. In figure 9.7, this global sum is indicated by the horizontal line beneath the Forces block, linking all nodes. Global sum procedures are generally available as part of the communications libraries of the computer, for example the widely used Message Passing Interface, or MPI.

Once the full force arrays have been restored on every processor, the integration of the equations of motion can take place. Once again, independent sets of atoms are processed on each node. A loop structure for achieving this is as follows

```
1   n_s ← Int.(N * i_node)/L
2   n_e ← Int.(N * (i_node + 1))/L
3   for i = n_s → n_e {
4     ...etc...
5   }
```
(9.14)

in which $...etc...$ represents lines to be inserted to perform the atomic integrations, as seen for example in equations (8.55)-(8.57), and $Int.(x)$ is a function that returns the integer part of x. The limits n_s and n_e define a unique subset of atoms on the node i_{node} of minimum size $Int.(N/L)$ and maximum size $Int.((N+L-1)/L)$, depending on the identity of the node. This procedure only integrates the motion for the specified set of atoms. Thus to restore replication of data, the new atomic positions and velocities must be passed from each node to all others. This can be

conveniently done using a global sum procedure, after first setting the array entries that have *not* been integrated on each node to zero. Alternatively, the updated atom data can be passed around using a systolic loop procedure. This global communication is indicated by the horizontal line below the Integrate blocks in figure 9.7. It should be noted that in case of the velocity Verlet algorithm two such global communications are required in each time step, since the integration has two stages. These are additional to the global communication for the forces.

The Forces and Integrate blocks and their associated global communications are executed every time step. The Properties block is also executed every time step, but in this case no global communication is required during the molecular dynamics cycle. This is because the Properties block is chiefly concerned with the accumulation statistical data, which does not need to be converted into system properties until the very end of the simulation. Parallelisation is therefore trivial; it is an easy matter to set each processor the independent task of accumulating a separate subset of the statistical data and only one global sum is needed at the end of the molecular dynamics cycle to bring it all together. Some exception may be made to this for some system properties to enable proper monitoring of the simulation, such as the system total energy, but these can be incorporated with the existing global sums for little extra cost.

On the whole the RD approach does a good job of load balancing. But there is an issue concerning the use of a cut-off. Like the SLS-G algorithm it is essentially an all pairs algorithm and if a cut-off is applied then it is possible for different nodes to end up calculating different numbers of in-range interactions. In practice, providing the cut-off is large in relation to the dimensions of the simulated system, this seems not to have too detrimental an impact. If the cut-off is small however, this could well be a serious issue. Under these circumstances an alternative approach, using linked cells, is preferable (see below). Otherwise the effect can be ameliorated by using fewer nodes for the simulation; the job time may not be significantly worse (it can even be better!) but at least it will be using the compute resource more effectively.

The scaling of the RD algorithm can be estimated using the approach undertaken for the SLS-G case. The processing time per time step on L nodes can be written as

$$t_p = \frac{N}{2L}\left((N-1)(f\alpha+\alpha')+2g\beta\right), \tag{9.15}$$

in which α, α' and β again represent the times for calculating one pair interaction, one inter-particle distance and the update of the position and velocity of one atom respectively. Furthermore, f is the fraction of interactions expected within the prescribed cut-off. The constant factor g is 1 if the integration algorithm is in the Verlet leapfrog form and 1.5 if in the velocity Verlet form. Note that (9.15) is the same as (9.6), which follows from the fact that both represent the same effective decomposition of the pair interactions.

With regard to the communication time in this algorithm, we are confronted with many possible ways of managing the message passing. So we will make the simplifying assumption that both the final accumulation of pair forces into atomic forces and the global exchange of coordinates after integration of the atomic motion

are handled by global sum operations. This has the benefit of making the issues somewhat clearer if not optimally efficient. If we further assume that the global sum is based on a systolic loop approach (again with some sacrifice of efficiency), we can write

$$t_c = 3(2L-1)\left(N\gamma/3+\delta\right), \tag{9.16}$$

in which δ and γ are the communication latency and the time to pass (and in this case sum) the configuration data of one atom. In this naïve description, the global sum handles the position, velocity and force separately, so three global sums are required. This explains the appearance of the number 3 on the right of (9.16) , both as a factor and a divisor. The fundamental ratio is therefore

$$R_{cp}^{L} = \frac{2L(2L-1)\left(\gamma+3\delta/N\right)}{(N-1)\left((f\alpha+\alpha')+2g\beta/(N-1)\right)}. \tag{9.17}$$

As before we need to find the circumstances in which $R_{cp}^{L} \to 0.$ We consider the case $N \gg L$ once more and see that, as before, as N rises the terms involving δ and β become progressively less important and we can approximate (9.17) by

$$R_{cp}^{L} \approx \frac{2L(2L-1)\gamma}{N(f\alpha+\alpha')}. \tag{9.18}$$

This form resembles that for the SLS-G case, but there is an important difference. Like the SLS-G algorithm, the RD algorithm becomes more efficient (i.e. $R_{cp} \to 0,$ as N increases). However, we also see that R_{cp} scales as the square of $L,$ not $L,$ as before. This means that the RD algorithm will hit problems faster as the number of nodes increases.

Using the parameters: $\alpha=1,\ \alpha'=0.25,\ \beta=1,\ \gamma=0.1,\ \delta=10,\ g=1$ and $f=0.5.$ We can use (9.17) to obtain an idea of how the efficiency behaves over the number of nodes, $L,$ rises from 8 to 16384 and the number of atoms, $N,$ from 10 to 100,000. The result is shown in figure 9.8.

The parallel efficiency for 8 nodes is plotted (in black) on the far left of figure 9.8 and the number of nodes doubles with each plot towards the right. Superficially this resembles figure 9.5, which referred to the SLS-G algorithm, but shows significant differences. The main feature in common is the progression towards maximum parallel efficiency as the number of atoms increases – behaviour expected from equation (9.18). However, achievement of maximum efficiency is much delayed. The SLS-G algorithm was capable of running a 10,000 atom simulation on 512 nodes ~90% efficiency, but here the efficiency on 64 processors is already as low as ~80%. The important point here is that the RD algorithm we have described here has a much higher communications overhead and lower efficiency is the consequence of that. However, the specific details of this comparison are very much dependent on the form of the global summation used and it is fair to say that versions offered by the message passing software on any real machine are likely to be significantly more efficient than that assumed here. Nevertheless, concern about the communications cost of the RD algorithm remains valid and for this reason is is not recommended for parallel computers with more than 64 processors. Within this limitation it is a robust

and practical approach to parallel molecular dynamics.

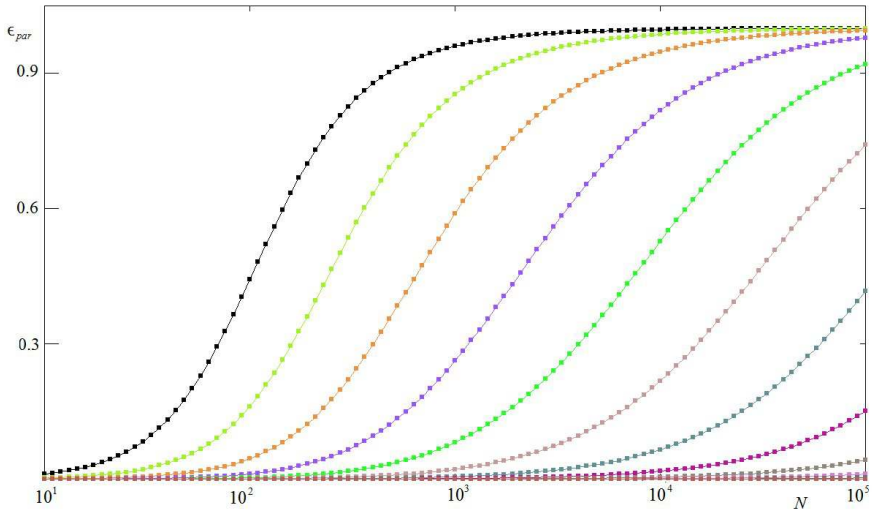

Figure 9.8. The Parallel Efficiency of the Replicated Data Algorithm

The performance of the RD algorithm is shown in figure 9.9 which plots the time per time step, t_s, (in units of the parameter α) as a function of the number of atoms, N, for different numbers of nodes, L, starting from 2 and doubling in number until 128 nodes. (On the far right of the figure, the number of nodes increases as we move down the plots.) This figure resembles figure 9.6 corresponding to the SLS-G algorithm. On the left of the graph we again have a communications dominated

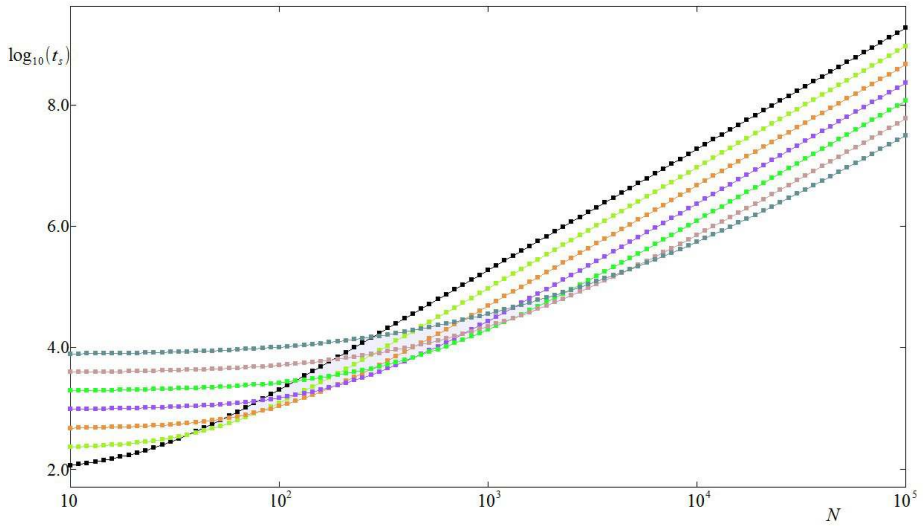

Figure 9.9. The Performance of the Replicated Data Algorithm

region, (which becomes worse as the number of nodes increases) and on the right a processing dominated region. There are differences however and these are mostly due to the characteristics of the communications employed. There are no horizontal lines in the communications dominated region, instead there is a continuous rise that eventually becomes linear, with slope ~2.0 as the expected $O(N^2)$ behaviour emerges. Only at the extreme right do all the plots become parallel with equidistant vertical separation and the delay in achieving this outcome increases with increasing node numbers. The algorithm therefore struggles to maintain type 2 (strong) scaling for larger numbers of nodes, in comparison with SLS-G. Clearly the algorithm does eventually start scaling well as the number of atoms increases, but the time required per time step may become unacceptable before this ideal behaviour arises. For this reason the algorithm is best confined to ~64 processing nodes or fewer.

The main advantage of the RD approach is that it is relatively easy to implement and existing serial programs can be quite easily modified to run on parallel machines. Much of the work of parallelisation can be handled by intrinsic global sum routines, which hide practically all of the communication details. In contrast the complicated communication structures in the SLS-G algorithm are not available as an intrinsic feature and have to be constructed from lower level utilities.

9.2.5 Domain Decomposition: Parallel linked cells

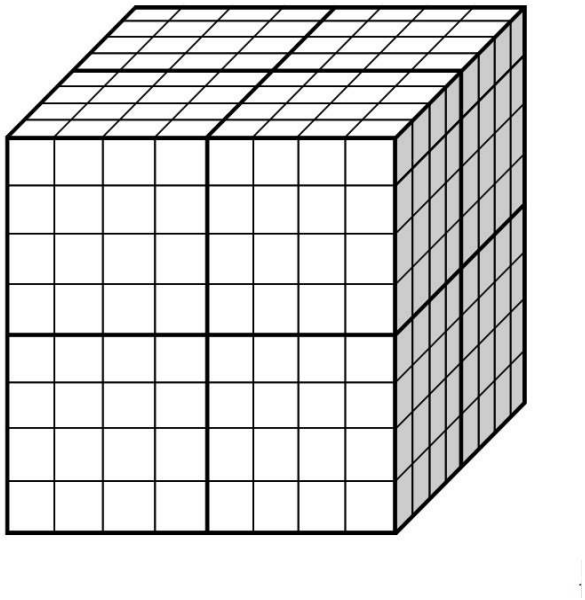

Figure 9.10. The Parallel linked cells Algorithm

The parallel algorithms described so far are based on *algorithmic decomposition*, which means they achieve parallelism by exploiting inherently parallel attributes within the molecular dynamics algorithms. Domain decomposition on the other hand achieves parallelism by exploiting the geometrical properties of the simulated system. In essence this involves dividing the physical system into distinct spatial regions and allocating each region to a specific node of the parallel computer. It is of course essential that all relevant regions of space are accounted for and that collectively the regions constitute a contiguous whole. It is also important, for reasons of efficiency, that the decomposition of system space is such that load balancing of the processor nodes is achieved. Domain decomposition can be approached in many ways, but in the context of molecular dynamics the most widely used approach is the Parallel linked cells (PLC) algorithm, which is based on the serial algorithm described in chapter 8.

The essence of the PLC algorithm is shown in figure 9.10, which in this example represents a cubic simulation cell divided into eight equal domains, each of which is allocated to a single node. There is thus a one to one correspondence between domains and nodes. The nodes are themselves logically mapped to an orthorhombic grid in a manner consistent with the physical location of each domain in space. Subsequently, each domain is divided into sub-cells, as is done for the serial linked cells algorithm, and once again the widths of the sub-cells are determined by the cut-off that is applied to the pair interactions (i.e. they are as small as possible but greater than r_{cut}.) Once the system is divided up in this way each node is able to simulate its allocated domain largely independently of the other nodes. Adaptation of this idea to large simulations on very many processors is straightforward.

In the PLC construction each domain may be located by a vector, $\vec{L}_D$, which defines the centre of the domain D in the simulation cell. If we suppose that the simulation cell is decomposed into $L=M_a\times M_b\times M_c$ domains, where L is the number of nodes and M_a, M_b, M_c represent the integer numbers of domains in the principal directions of the cell (which is $2\times2\times2$ in the figure 9.10), we can write $\vec{L}_D$ as

$$\vec{L}_D=(I+1/2)\vec{A}/M_a+(J+1/2)\vec{B}/M_b+(K+1/2)\vec{C}/M_c, \tag{9.19}$$

where I, J, K are the integer coordinates for the domain D and $\vec{A}, \vec{B}, \vec{C}$ are the simulation cell vectors. This vector allows the construction of a *local* coordinate, $\vec{u}_i$, for each atom in the domain from its *global* coordinate, $\vec{r}_i$, as in

$$\vec{u}_i=\vec{r}_i-\vec{L}_D. \tag{9.20}$$

The vector $\vec{u}_i$ is used in place of $\vec{r}_i$ in subsequent procedures.

The major issue in PLC, as in all domain decomposition methods, is *contiguity*. This is the maintenance of the connectedness of the simulated system even though it is broken up into largely independent pieces. This requires the introduction of the concept of a *data halo* around each domain which is a space or "boundary zone" around the domain containing atomic data from the neighbouring domains. The node uses this halo to maintain physical contact with its neighbours. In the case of PLC the content of the halo comes from the outermost regions of the neighbouring domains (corresponding to "border" sub cells), which is "parcelled up" and transported to the neighbour node that requires it. The incoming sub-cells are packed around the node's domain in the appropriate position, so creating the "halo" which is one sub-cell thick (see figure 9.11.) It need be no larger than this because this meets the limiting cut-off criterion for the inter-atomic interaction. The periodicity of the simulation cell boundary is automatically allowed for in the mapping of nodes to the orthorhombic grid.

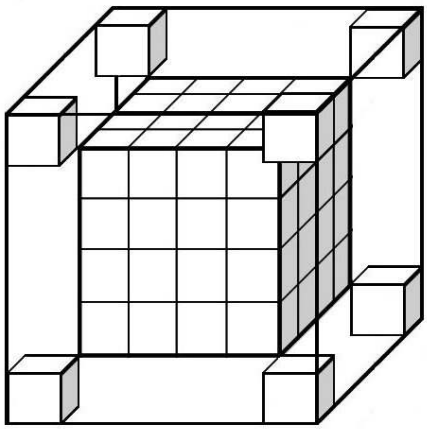

Figure 9.11. The Size of the Halo Region Surrounding a Domain

Note that, as a result of the halo, the *working* number of sub-cells each domain has is

$$N^{w}_{cells}=(N_a/M_a+2)(N_b/M_b+2)(N_c/M_c+2), \tag{9.21}$$

where N_a, N_b, N_c are the sub-cell numbers for the principal directions of the full simulation cell, whereas the formal number of sub-cells on each domain is

$$N_{cells}=(N_a/M_a)(N_b/M_b)(N_c/M_c). \tag{9.22}$$

It is N^{w}_{cells} that is required for the construction of the link cells on the halo enlarged domain.

The transport of the halo data minimally requires data exchange between nodes in directions perpendicular to each face of the domain, which implies six two-way transfers. We may conveniently label them as North (N), South (S), East (E), West (W), Up (U) and Down (D). There are however, 26 neighbours to each domain, so the exchanges must also transfer data from nodes that do not share a domain face and this needs to be factored in. The sequence of transport events on each domain must therefore be:

1. Atoms within r_{cut} of the extreme positive x-coordinate are sent in positive x-direction.
2. Atoms within r_{cut} of the extreme negative x-coordinate are sent in negative x-direction.
3. Atoms within r_{cut} of the extreme positive y-coordinate (including any halo atoms at this extreme) are sent in positive y-direction.
4. Atoms within r_{cut} of the extreme negative y-coordinate (including any halo atoms at this extreme) are sent in negative y-direction.
5. Atoms within r_{cut} of the extreme positive z-coordinate (including any halo atoms at this extreme) are sent in positive z-direction.
6. Atoms within r_{cut} of the extreme negative z-coordinate (including any halo atoms at this extreme) are sent in negative z-direction.

The inclusion of transferred halo data for consideration for further transfer in steps 3 to 6 guarantees that every domain has a complete halo. The transfers are generally easy to implement but there are some hidden subtleties. It should be clear that what is sent to the neighbouring nodes is a *copy* of the *global* atomic coordinates, $\vec{r}_i$, of the atoms concerned, while the original data remains in place. Also, on receipt of the halo data, the receiving node must convert $\vec{r}_i$, to the appropriate local coordinate, $\vec{u}_i$, while also allowing for any periodic boundary shifting.

Note also that it is sometimes possible (if the domains happen to stretch across the full width of the simulation cell) that after the halo construction a specific atom is represented more than once on a node - perhaps several times, in the halo as well as within the domain. This will be apparent from the occurrence of several atoms with the same global identity on a node. In such cases the different images of the same atom will distinguishable by their coordinates, which will have been modified by periodic boundary and domain transfer adjustments. The important point is that, as far as the domain is concerned, these replica atoms are geometrically different and

should not be purged as erroneous. There is no danger of the images interacting with each other.

When all the halo data has been transferred, the linked cells can be constructed on each domain using scheme (9.23), which is based on the scheme for serial linked cells (8.44), but has some differences.

$$
\begin{array}{ll}
1 & for\ k=1\rightarrow N_{cells}\{ \\
2 & \quad (L_{head})_k \leftarrow 0 \\
3 & \} \\
4 & for\ i=1\rightarrow N\{ \\
5 & \quad (L_{link})_i \leftarrow 0 \\
6 & \} \\
7 & for\ i=1\rightarrow N\{ \\
8 & \quad s_a \leftarrow h_{xx}^{-1}u_i^x + h_{xy}^{-1}u_i^y + h_{xz}^{-1}u_i^z + w_a \\
9 & \quad s_b \leftarrow h_{yx}^{-1}u_i^x + h_{yy}^{-1}u_i^y + h_{yz}^{-1}u_i^z + w_b \\
10 & \quad s_c \leftarrow h_{zx}^{-1}u_i^x + h_{zy}^{-1}u_i^y + h_{zz}^{-1}u_i^z + w_c \\
11 & \quad L_a \leftarrow Min\left(Int.\left((n_a+2)(s_a+0.5)\right), n_a+1\right) \\
12 & \quad L_b \leftarrow Min\left(Int.\left((n_b+2)(s_b+0.5)\right), n_b+1\right) \\
13 & \quad L_c \leftarrow Min\left(Int.\left((n_c+2)(s_c+0.5)\right), n_c+1\right) \\
14 & \quad k \leftarrow 1 + L_a + (n_a+2)\left(L_b + (n_b+2)L_c\right) \\
15 & \quad if\left((L_{head})_k = 0\right)\{ \\
16 & \quad\quad (L_{head})_k \leftarrow i \\
17 & \quad \} else \{ \\
18 & \quad\quad (L_{link})_i \leftarrow (L_{head})_k \\
19 & \quad\quad (L_{head})_k \leftarrow i \\
20 & \quad \} \\
21 & \}
\end{array}
\qquad (9.23)
$$

In (9.23) the matrix, $\boldsymbol{h}$, is the domain counterpart to the usual cell matrix and is defined as

$$\boldsymbol{h} = \left[\vec{A}/M_a, \vec{B}/M_b, \vec{C}/M_c\right]. \qquad (9.24)$$

Also the integers n_a, n_b, n_c are defined as

$$n_a = \frac{N_a}{M_a}, \qquad n_b = \frac{N_b}{M_b}, \qquad n_c = \frac{N_c}{M_c}, \qquad (9.25)$$

and the parameters w_a, w_b, w_c are the displacements

$$w_a = \frac{M_a}{N_a}, \qquad w_b = \frac{M_b}{N_b}, \qquad w_c = \frac{M_c}{N_c}, \qquad (9.26)$$

which are the widths of a sub cell in the space of the fractional coordinates.

Once the linked-lists have been constructed, the calculation of the atomic forces

proceeds as for the serial algorithm. The procedure (9.27) can be used for this.

```
1  for i_c=1→n_c
2      for i_b=1→n_b
3          for i_a=1→n_a
4              k_1←1+i_a+(n_a+2)(i_b+(n_b+2)i_c)
5              for n=1→14
6                  j_a←Bracket(i_a+(t_a)_n, n_a+2)
7                  j_b←Bracket(i_b+(t_b)_n, n_b+2)
8                  j_c←Bracket(i_c+(t_c)_n, n_c+2)
9                  k_2←1+j_a+(n_a+2)(j_b+(n_b+2)j_c)
10                 i←(L_head)_{k_1}
11                 while(i>0){
12                     j←(L_head)_{k_2}                    ,          (9.27)
13                     while(j>0){
14                         if(k_1≠k_2 || j>i){
15                             ...etc...
16                         }
17                         j←(L_link)_j
18                     }
19                     i←(L_link)_i
20                 }
21             }
22         }
23 }
```

In which n_a, n_b, n_c are defined in (9.25) and the function $Bracket(i,n)$ is defined in (8.47). This is similar to the procedure (8.46), but automatically excludes interactions between atom pairs when both atoms are in the halo. This requires the set of nested loops commencing at lines 1 – 3 and the modified sub-cell index calculations at lines 4 and 9. The arrays of sub-cell increments, $t_a(n),\ t_b(n),\ t_c(n),\ n=1\ldots14,$ are defined in (8.45). Line 15 $"...etc..."$ is the point at which the formulae for calculating the pair interaction (potential and force) are inserted. With this construction all the pair interactions for the atoms in a domain can be calculated and the atomic forces obtained independent of every other node. After which the halo data can be discarded. A global sum operation is suitable to obtain the total system potential energy and virial.

In the normal molecular dynamics procedure, following the calculation of the atomic forces, the next step is the integration of the equations of motion. Here again this can take place independently on each processor and is simple to perform. However, this must be followed by an important procedure known as *deportation*. Under which any atom that no longer belongs geometrically to the domain of the node it is on, must be moved to the appropriate neighbouring node. Which atoms are to be deported is easy to decide – their position coordinates sit outside the domain. A simple calculation then determines which domain it rightly belongs to, subject to the periodic boundary condition. The deportation procedure operates in a similar way to the halo

construction. Once again, six stages are involved:

1. Atoms beyond the extreme positive x-coordinate of the domain are sent in the positive x-direction.
2. Atoms beyond the extreme negative x-coordinate of the domain are sent in the negative x-direction.
3. Atoms beyond the extreme positive y-coordinate of the domain (including newly transferred atoms at this extreme) are sent in the positive y-direction.
4. Atoms beyond the extreme negative y-coordinate of the domain (including newly transferred atoms at this extreme) are sent in the negative y-direction.
5. Atoms beyond the extreme positive z-coordinate of the domain (including newly transferred atoms at this extreme) are sent in the positive z-direction.
6. Atoms beyond the extreme negative z-coordinate of the domain (including newly transferred atoms at this extreme) are sent in negative z-direction.

Note that the atomic data being transferred includes the atom identity, position coordinates, velocities and forces as a minimum. In more advanced programs, additional information may need to be sent, such as bond connectivities and force field data, depending on how the force field description is also decomposed across the nodes, if at all.

We now assess the scaling performance of this algorithm. For simplicity we will assume that the domains mapped onto the nodes are cubic and that the simulation cell is generally orthorhombic. It is then decomposed over L processing nodes, where $L=p_1 p_2 p_3$ and p_1, p_2, p_3 are integers. This means we have a simple, orthorhombic arrangement of nodes onto which the simulation cell maps neatly, with p_1 domains in one principal direction, p_2 in another *etc.*

We start by writing the processing time per time step on L nodes as

$$t_p=\frac{N}{2L}\left[(f_{lc}\bar{n}_{occ}-1)\alpha+(27\bar{n}_{occ}-1)\alpha'\right]+\frac{N}{L}g\beta, \tag{9.28}$$

which is in part derived from equation (8.51) and includes the time cost of the integration of the equations of motion for N/L atoms. Parameters α, α' and β are respectively the costs of calculating a pair force; applying the cut-off condition; and integrating the equations of motion for one atom. The constant g is either 1 or $3/2,$ depending on the choice of the leapfrog or velocity Verlet algorithm. $\bar{n}_{occ}$ is the mean number of atoms in a sub-cell and f_{lc} is the volume ratio of the cut-off sphere to a sub-cell (which is $\approx 4\pi/3\approx 4$ if the sub-cell width and r_{cut} are similar). Since the domains are cubic, we can assume that the sub-cells are also cubic and that there are η^3 sub cells in each domain, where η is integer. Then in order to map the domains on to the L nodes, we must have

$$N_{sc}=p_1 p_2 p_3 \eta^3=L\eta^3, \tag{9.29}$$

where N_{sc} is the number of sub-cells in the whole simulation cell. From this we can see that the number of atoms in the simulation cell must be

$$N = \bar{n}_{occ} N_{sc} = \bar{n}_{occ} L \eta^3. \tag{9.30}$$

This shows that for a given number of nodes, L, an increase in the number of atoms that is compatible with mapping the simulation onto the parallel computer is achievable by increasing the number of atoms per sub-cell, $\bar{n}_{occ}$, or by increasing the parameter η in integer steps. Changing $\bar{n}_{occ}$ implies changing the system density, which is not what we usually want to do when scaling up the system size, so we choose the latter option, which means that N scales with order $O(\eta^3)$. Substituting (9.30) into (9.28) gives

$$t_p = \frac{\bar{n}_{occ} \eta^3}{2} \left[(f_{lc} \bar{n}_{occ} - 1) \alpha + (27 \bar{n}_{occ} - 1) \alpha' + 2 g \beta \right]. \tag{9.31}$$

The communication time per node, t_c, may be estimated from

$$t_c = 6 (\bar{n}_{occ} \eta^2 \gamma / 3 + \delta), \tag{9.32}$$

in which $\bar{n}_{occ} \eta^2$ represents the number of atoms occupying the sub-cells on the outer faces of a domain, which to a first approximation is the number of atoms that must be transported to construct the halo data. Parameter γ is the time required to transfer the positions, velocities and forces of one atom, so $\gamma/3$ is the time for the coordinates alone. Lastly, parameter δ is the communication latency of the message containing the data and, of course, six messages are required in the data transfer scheme shown above.

Thus we can write the fundamental ratio, R_{cp}^L, as

$$R_{cp}^L = \frac{4 \bar{n}_{occ} \eta^2 \gamma + 12 \delta}{\bar{n}_{occ} \eta^3 \left[(f_{lc} \bar{n}_{occ} - 1) \alpha + (27 \bar{n}_{occ} - 1) \alpha' + 2 g \beta \right]}. \tag{9.33}$$

We can now investigate how the parallel efficiency of the algorithm behaves as the numbers of atoms, N, or nodes, L, are varied while keeping $\bar{n}_{occ}$ constant. We note first of all that if we require $N \gg L$, this can be achieved by making η large in (9.30). In which case the δ term drops away and (9.33) can be written as

$$R_{cp}^L \approx \frac{4 \gamma \bar{n}_{occ}^{1/3}}{\left[(f_{lc} \bar{n}_{occ} - 1) \alpha + (27 \bar{n}_{occ} - 1) \alpha' + 2 g \beta \right]} \left(\frac{L}{N} \right)^{1/3}. \tag{9.34}$$

This formula shows that as N becomes increasingly large, for a fixed number of nodes, R_{cp}^L will tend towards zero and thus eventually, high efficiency and good type 2 scaling is possible. However, the dependence is only $N^{-1/3}$, which is considerably slower than for the SLS-G or RD algorithms, which have the dependence N^{-1}. The implication is that the PLC algorithm is best exploited on very large simulations. What is also interesting about (9.34) is that the dependence on L is also weak, which means it is relatively insensitive to the number of nodes used, provided $N \gg L$.

The other extreme range of application of PLC is when $N \approx \bar{n}_{occ} L$ and so $\eta \approx 1.$ Inserting this condition into (9.33) shows that in this case it is not possible to discount the impact of the parameter δ (unless $\bar{n}_{occ}$ is abnormally large!). In fact, on most parallel machines δ is now likely to be the dominant factor. R_{cp}^{L} will be large under these circumstances and the parallel efficiency correspondingly low.

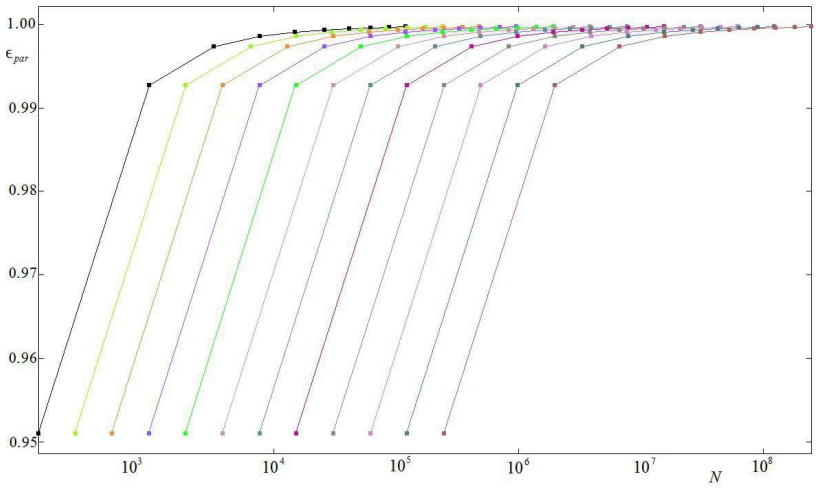

Figure 9.12. The Parallel Efficiency of the Parallel linked cells Algorithm

In figure 9.12 we present a plot of the parallel efficiency, ϵ_{par}, (9.4) as a function of the number of atoms, N, for a range of node numbers L, rising from 8 to 16384. For this purpose R_{cp}^{L} has been obtained from (9.33) using the parameters: $\alpha=1,\ \alpha'=0.25,\ \beta=1,\ \gamma=0.1,\ \delta=10,\ g=1,\ f=4.0$ and $\bar{n}_{occ}=15.$ The largest simulation presented in this plot is of order 246 million atoms on 16384 nodes (on the far right), and the smallest is 120 atoms on 8 nodes (far left). The number of nodes doubles with each plot, moving to the right of the figure. The plots shown do not progress as smoothly as the previous cases, due to the fact that increasing N in a manner commensurate with the mapping of the domains onto the nodes requires incrementing η according to (9.30), which produces rather large increases in $N.$

It is evident in figure 9.12 that, with the given parameters, the parallel efficiency is never less than 0.95. This is impressive, but is based on an especially favourable construction of the domains and it is certainly possible to arrange things so that a lower level of efficiency results. For example, if $\bar{n}_{occ}$ is reduced significantly from the assumed value of 15 (which is of the order occurring in Lennard-Jones systems), the dominance of t_p over t_c can be overturned. The same can happen if the assumed value of $\delta=10$ for the communication latency turns out to be much larger (which is certainly possible). Such occurrences would lead to a communications dominated execution, which would not have anything like the presented level of efficiency. This is always something to be wary of and ideally should be investigated for any particular application on a given platform. Nevertheless, Figure 9.12 is a typical representation and shows that the algorithm becomes maximally efficient as N tends to some large number, as we expect. It is also evident on close inspection that, if the number of nodes is doubled, the original efficiency can be recovered by doubling the number of atoms in the simulation. This is seen in the horizontal rows of dots in the figure.

The performance of the PLC algorithm is shown in figure 9.13, which plots the time

per time step, t_s, (in units of the parameter α) as a function of the number of atoms, N, for different numbers of nodes, L, ranging from 8 to 16384, doubling in number as we move across the figure from left to right. All the plots are are remarkably linear on a log-log scale with slope ~1, which shows the algorithm is scaling $O(N)$, as a linked cells algorithm should.

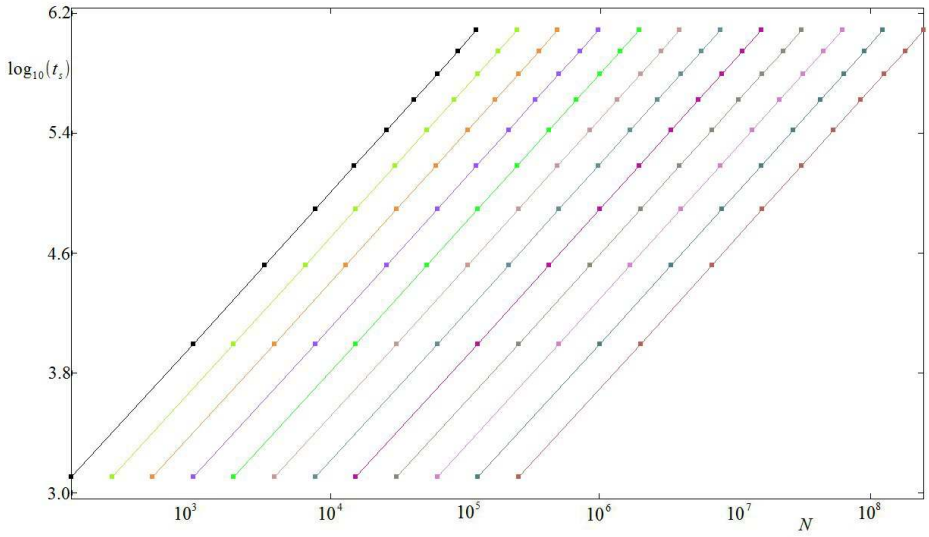

Figure 9.13. The Performance of the Parallel linked cells Algorithm

That the plots are parallel and an equal distance apart on this log-log scale demonstrates two important properties. Firstly the constant vertical separation for fixed N shows that doubling the number of processors halves the execution time t_s. Theoretically the vertical (downwards) gap between consecutive plots should be $\approx -\log_{10}(2)$, which is indicative of good type 2 (strong) scaling. The fixed horizontal gap between plots indicates good type 3 (weak) scaling, since the number of atoms, N, doubles with the number of nodes, L, to give a theoretical gap of $\approx \log_{10}(2)$.

What is most remarkable about figure 9.13 is that it clearly shows that if it is practical to simulate a system of N atoms on 8 nodes, it is equally practical to simulate 16384 x N atoms on 16384 nodes. On this basis the algorithm is clearly ideal for massively parallel applications. However it is necessary to recall that physical relaxation times will be large for multi-million atom systems, so a great deal of equilibration may be necessary before useful results can be obtained.

The PLC algorithm is evidently extremely powerful, but there are a couple of aspects that need mentioning. The first of these is load-balancing and the second is data input/output. These are rather complex subjects somewhat beyond the scope of this book, but it is useful to give some account of them, however brief.

Load balancing, as described in section 9.1.2 , arises naturally in the PLC approach provided the simulated system is reasonably homogeneous for the duration of the simulation. Homogeneity in this context means the system has a uniform density and the molecular components of the system are scattered reasonably uniformly throughout. These conditions often arise in practice and in such cases simulations are efficiently executed. However, many large systems are naturally inhomogeneous.

Examples include systems with surfaces or boundaries, systems with voids, systems that exhibit phase separation or micro-segregation and systems in which large entities, such as biomolecules or micro-crystals, are embedded in otherwise uniform media (e.g. solid or liquid solutions).

In such cases it not safe to assume load balancing will occur without some form of intervention. In practice the solution to this is to construct a load-balancing scheme, which monitors the work load of each node and re-allocates it from the excessively used nodes to other nodes less loaded. The scheme can be either static (i.e. performed at infrequent intervals during the simulation) or dynamic (i.e. constantly updating the work load as the simulation proceeds). Such a scheme must be over-arching and therefore global in parallel terms. The monitoring of loading and the reallocation of work load are potentially expensive operations that add to the overall cost of simulation. Achieving an efficient load balancing scheme is therefore a very difficult task and is likely to add a heavy layer of complexity to a simulation program for relatively small gains in performance in most applications. Nevertheless there are systems for which some form of load balancing is essential. Examples of successful load balancing schemes can be found in [84] and [85].

The other issue in large scale simulations is the practical matter of managing the huge volume of data associated with them [86]. Simply loading the atomic coordinates into the distributed memory of a massively parallel system can be an expensive task. Added to this is the problem of writing out the configuration data to disc storage at intervals during the simulation and the depositing of checkpoint data to restart and continue the simulation at a later date. Leaving aside the potentially vast amount of disc space required to store the data (which should always be estimated and allowed for before commencing on a given study!) the matter of simply reading and writing the data requires careful attention. The common default strategy of passing the data from each node, p_n to node p_0 in sequence and writing from there to disc (or the reverse when reading data) is not adequate in these cases. Fortunately large scale parallel machines have many so-called I/O nodes which can be used concurrently to facilitate an efficient I/O process. Groups of processing nodes can be associated with one specific I/O node, which handles all the data traffic for the group [87]. This divide-and-conquer strategy can be highly efficient and it effectively removes the bottleneck in data transfer. However, implicit in this strategy is the need to sort and store the data in a sensible manner on the disc system, so that it can subsequently be used by both the simulation and analysis software.

9.3 Parallel Methods for Complex Force Fields

The parallel algorithms we have described so far have dealt exclusively with systems that employ pair interactions, such as Lennard-Jones and other van der Waals forms. We may also extend their application to Coulombic systems if we are bold enough to use truncation schemes (though this is not advised!) But the more complex kinds of interactions, such as intra-molecular interactions (bonds, bond angles, dihedral angles *etc.*) or realistic methods for Coulombic interactions, such as the Ewald summation or the Smoothed Particle Mesh described in chapter 6, require special attention.

9.3.1 Parallel Treatment of Intra-molecular Bonding

Implementing a parallel scheme for intra-molecular interactions turns out to be straightforward if it is based on an appropriate data structure. Most software designed to calculate the intra-molecular interactions is based on the use of bookkeeping arrays, which record the indices of the atoms participating in the bonding potential. This makes it easy to gather the data for the atoms concerned (i.e. the atomic positions and atomic types *etc.*) and proceed to calculate the required interaction. It is not necessary therefore to search for interacting atoms in the manner required for van der Waals interactions. The bookkeeping arrays used in this approach are easily adapted for parallel calculation.

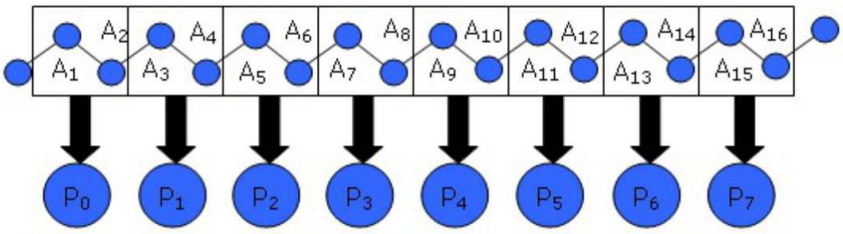

Figure 9.14. Parallel Decomposition of Bonded Potentials

Consider figure 9.14, in which we have a molecular structure consisting of many atoms linked by chemical bonds, which define bond angles A_1 to A_{16}. It is a simple matter to distribute the set of bond angles over the processing nodes p_0 to $p_{7,}$ so that each node becomes exclusively responsible for evaluating the bond angle interactions allocated to it. In terms of the bookkeeping arrays defining the bond angles, this simply means that each node acts on the allocated subset of the bookkeeping arrays. Each node can thus calculate the allotted bond angle interactions concurrently with all the others. This is a simple and easily implemented parallel scheme that works in conjunction with all the parallel molecular dynamics schemes described above for the non-bonded interactions.

In the case of the RD method, the replication of the bookkeeping arrays make this scheme trivially easy to implement: each node is simply assigned an equal sized portion of the arrays to process. For the SLS-G approach, it is evident that the bookkeeping arrays defining the bonding terms are necessarily decomposed into separate pieces across the nodes. Once allocated, they remain fixed on the allotted node. As the systolic "pulses" move the configuration data around the nodes, the coordinates of atoms relevant to the terms assigned to the node must be stored until a full set is obtained to permit the calculation of the bonded term. Since all atoms meet with all others as they are circulated around the nodes, completing the required atom sets is guaranteed. Once the bonded interaction has been calculated, the associated atomic forces are sent to the "home nodes" of the atoms concerned, to be added to the non-bonded forces prior to the integration of the equations of motion.

In the PLC approach, each node holds the configuration data for complete and partial

molecules, which are split across domains. The bookkeeping arrays for the bonded interactions are inevitably distributed across nodes, but the description of each bonded term is always complete, even if some of the atoms required are not resident on the node[23]. There is an implicit assumption that the complete interaction can be calculated from atoms that belong to the domain of the node and if any atoms are missing, they will be found in the halo surrounding the domain. The truth of this assumption can be guaranteed if the potential cut-off that defines the sub-cell width (and thus the width of the halo) is large enough. (In practice this is easy to arrange, but it is wise precaution to build in checks the ensure the required atoms are indeed present.) If this assumption holds, any bonded term that has component atoms on different domains, can and will be calculated on all the relevant domains. This replication ensures that all the atomic forces on all the atoms of a domain include the bonded as well as the non-bonded interactions and are therefore complete. Replication of the potential energy calculation is avoided if it is only calculated when the bonded atom with the lowest identification number resides of the domain. Bonded interactions that are not split across domains are unique to a single node and therefore not replicated. In the course of the simulation when an atom is "deported" to another node's domain, all the associated bookkeeping information pertaining to the atom must be deported with it. After which the existing bookkeeping arrays on the deporting and importing nodes must be reconstructed to allow for the change. This evolving relationship between the bookkeeping arrays and the nodes marks a major difference between the PLC and the RD and SLS-G algorithms and makes the PLC approach considerably more complex to program.

9.3.2 Parallel Treatment of the Ewald Summation

The parallel treatment of the basic Ewald Summation, which was described in chapter 6, section 6.4 , turns out to be fairly straightforward [88]. The basic Ewald formula (reproduced from equation (6.18)) is

$$\Phi^E(\{\vec{r}_i\}) = \frac{1}{2\epsilon_0 V}\sum_{\vec{k}\neq\vec{0}}^{\vec{\infty}} \frac{\exp(-k^2/4\alpha^2)}{k^2}\left|\sum_{j=1}^{N} q_j \exp(i\vec{k}\cdot\vec{r}_j)\right|^2 + \frac{1}{2}\frac{1}{4\pi\epsilon_0}\sum_{\vec{L}=\vec{0}}^{\vec{\infty}}\sum_{j=1}^{N}\sum_{n=1}^{N}{}' \frac{q_j q_n \, erfc(\alpha r_{jn}^L)}{r_{jn}^L} - \frac{\alpha}{4\pi^{3/2}\epsilon_0}\sum_{j=1}^{n} q_j^2. \quad (9.35)$$

We note firstly that the self interaction correction term (last on the right) is a constant that needs to be calculated once only in a simulation and therefore presents no difficulty for parallel execution. The *real space* term (second on the right) is a modified Coulomb sum augmented by the convergence factor $erfc(\alpha r_{jn}^L)$, which has the effect of changing the original Coulombic potential into a short ranged one. On this basis we may use, without modification, any of the SLS-G, RD or PLC approaches to calculate the potential and forces arising from this. This leaves only the *reciprocal space* term (first on the right) requiring special attention. To help us do this we first write the reciprocal space term as

23 The number of nodes splitting a bonding interaction in this way may equal the number of atoms in the bond. For example a dihedral potential can be split over as many as 4 nodes.

$$\Phi_1^E(\{\vec{r}_i\}) = \frac{1}{2\epsilon_0 V} \sum_{\vec{k}\neq\vec{0}}^{\vec{\infty}} A(k) S(k), \tag{9.36}$$

where

$$A(k) = \frac{\exp(-k^2/4\alpha^2)}{k^2}, \tag{9.37}$$

and

$$S(k) = Q^*(k) Q(k), \tag{9.38}$$

is the *structure factor,* with

$$Q(k) = \sum_{j=1}^{N} q_j \exp(i\vec{k}\cdot\vec{r}_j). \tag{9.39}$$

The asterisk $(*)$ in (9.38) indicates a complex conjugate. From these equations we see we can parallelise the reciprocal space calculations either by distributing the sum over the $\vec{k}$ vectors in (9.36) or by distributing the sum over atoms in (9.39), over the nodes. Both of these are viable options for the RD algorithm, but for the SLS-G and PLC algorithms it is the distribution of atoms that is best. We shall therefore focus on the second option.

The principle we employ is that each node calculates the contributions to the Coulombic interactions for a specific set of charged atoms. If the are N charged atoms and L nodes, each node p handles $n_p \approx N/L$ atoms. The first step is to calculate the sub-sum $Q_n^p(k)$ for each $\vec{k}$ vector on each node p:

$$Q_n^p(k) = \sum_{j=1}^{n_p} q_j \exp(i\vec{k}\cdot\vec{r}_j), \tag{9.40}$$

where index j runs over all the n_p atoms the node is accountable for. These sub-sums must be combined to create $Q(k),$ which is achieved using a global sum procedure:

$$Q(k) = \sum_{p=0}^{L-1} Q_n^p(k), \tag{9.41}$$

From $Q(k)$ the structure factor $S(k)$ can be calculated using (9.38). (Since we are using an expensive global sum operation in (9.41) it is advisable to calculate $Q(k)$ for *all* the $\vec{k}$ vectors simultaneously, by summing as an array of $Q_n^p(k)$ values rather

than for individual $Q_n^p(k)$ terms.) Once $S(k)$ is obtained, the *complete* reciprocal space potential $\Phi_1^E(\{\vec{r}_i\})$ can be calculated on every node and with it the atomic forces on the atoms allocated to the node:

$$\vec{f}_m^{\,1}=\frac{q_m}{\epsilon_0 V}\sum_{\vec{k}\neq\vec{0}}^{\vec{\infty}} A(k)\Im\left\{\exp(i\vec{k}\cdot\vec{r}_m)Q^*(k)\right\}\vec{k}. \tag{9.42}$$

Equation (9.42) is obtained in the usual way from the derivative of $\Phi_1^E(\{\vec{r}_i\})$ in (9.36) with respect to $\vec{r}_m$. The atomic force $\vec{f}_m^{\,1}$ is that due to the reciprocal space terms only.

9.3.3 Parallel Treatment of the Smoothed Particle-Mesh Ewald

It is fair to say that the SPME method does not present itself well for parallel implementation. The reason is that its reciprocal space term[24] depends crucially on the 3D discrete Fourier transform, which is a *global* operation incorporating data from all parts of the simulated system. This implies we must either gather all the data together on one node to perform the Fourier transform, or we must devise a *distributed* Fourier transform that can handle data scattered across the nodes. The first option places a potentially excessive demand on the available memory of one processor, while simultaneously discarding the possibility of parallel computation. The second option risks disrupting the extreme efficiency of the fast Fourier transform (FFT) by embedding communications calls within its structure and thus robbing the SPME approach of its greatest asset, which is speed. What then, if anything, can be done to implement a parallel SPME?

We begin by recalling from chapter 6, section 6.5 , that in the SPME method the reciprocal space term of the Ewald Sum is written as

$$\Phi^F(\{\vec{r}_i\})=\frac{1}{2\epsilon_0 V}\sum_{k_1,k_2,k_3}\check{\boldsymbol{G}}(k_1,k_2,k_3)\boldsymbol{Q}(k_1,k_2,k_3), \tag{9.43}$$

where $\boldsymbol{Q}$ and $\boldsymbol{G}$ are 3D arrays, with $\check{\boldsymbol{G}}$ being the 3D Fourier transform of $\boldsymbol{G}$. Array $\boldsymbol{Q}$ is the *interpolated charge array* and $\boldsymbol{G}$ is obtained from the 3D Fourier transform of $\boldsymbol{Q}$. So the two arrays are closely related (see chapter 6 for details). The primary issue in parallel computing is how to manage the 3D Fourier transforms associated with these arrays.

We will first consider the RD and SLS-G approaches. Since these are $O(N^2)$ algorithms, we may assume that the simulate systems are not extravagantly huge, perhaps of the order of 10,000 atoms at most. In which case, we hope, we need not worry too much about memory requirements and our first option of performing the Fourier transforms completely on a node is a possibility. That way at least we preserve

24 We need not be concerned with the real space component, since it is perfectly well handled by any of the parallel methods discussed so far.

the superlative performance of the 3D FFT algorithm. Our proposed scheme (on a platform of L nodes) is as follows

1. Allocate to each node a different subset of $n_p \approx N/L$ atoms. For RD this will be a subset of all the atoms already on the node. For SLS-G this will be all the atoms allocated to the node.
2. Construct the $\boldsymbol{Q}$ array independently on each node, using the n_p atoms the node has been allocated.
3. Perform a *global sum* of the array $\boldsymbol{Q}$ to establish a complete copy of the array on all nodes.
4. Perform the 3D FFT on $\boldsymbol{Q}$ (in its entirety) on all nodes.
5. Construct array $\boldsymbol{G}$ from the array $\check{\boldsymbol{Q}}$.
6. Perform the 3D FFT of $\boldsymbol{G}$ on all nodes.
7. Calculate $\Phi^F(\{\vec{r}_i\})$ independently on all nodes and the forces acting on the n_p atoms of the node.

The reader will rightly object that apart from the construction of the $\boldsymbol{Q}$ array, this is not a parallel algorithm. However it does not claim to be, it is merely making the best of a bad situation. It's main merit is that it allows adaptation of an SPME code to run on a parallel machine, where the benefit of parallel computing pays off is in the real space term and the calculations of other forces. Experience shows that, for modest numbers of nodes, this approach can still be faster than the parallel version of the basic Ewald method described above, which is undoubtedly due to the efficiency of the serial FFT. For that reason this method worth implementing.

In the case of the PLC algorithm, we would not contemplate loading all the data of a massively parallel computation onto a single processor, so we take an entirely different approach. The array $\boldsymbol{Q}$ is distributed over the nodes in much the same way as the simulation cell and is built in separate pieces using the atomic data that is present on each node. In order to calculate the Fourier transform of $\boldsymbol{Q}$ we must develop a distributed version the 3D FFT. There are a number of ways this can be done, which are discussed in [53], but here we shall focus on a particular approach based on the 3D FFT of Bush [59], that has proven to be very efficient.

A 3D FFT applied to an array of dimensions $M \times M \times M$ in fact performs $M \times M$ 1D Fourier transforms of dimension M in each of the 3 principal directions, so we need only look at the 1D FFT algorithm to see how it can be adapted for distributed use.

To see where the difficulty lies, we need to know something about the way the FFT works, but fortunately we will not need to dig deep. (A full description of the FFT is given in the book by Brigham [80], which is highly recommended.) The data flow of the commonly used radix 2 FFT is shown in figure 9.15. The left side this figure represents an example 16 element array $\boldsymbol{A}$, which we assume is distributed over four nodes p_0 to p_3. The FFT operates in four stages to produce the final Fourier transformed result on the far right. The arrows show how individual array elements are processed by the FFT at each stage. It is seen that pairs of elements some distance apart in the array are combined at each stage.

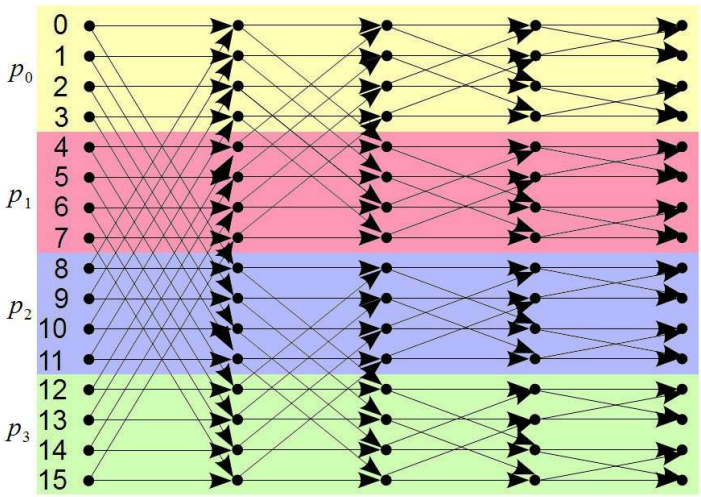

Figure 9.15. Data Flow in the Radix 2 Fast Fourier Transform

On a single processor the efficiency of this procedure is extremely high, an array of dimension 2^n requires only n stages to produce the final result. Clearly, for this scheme to work best, the data array needs to be contiguous in the memory of a single node. However this cannot be the case for the data in the distributed $\boldsymbol{A}$ array. Figure 9.15 shows that we are *obliged* to pass data between nodes at certain stages in the FFT. We note however, that as we advance through the stages, the required element pairs become progressively more local, so data transfer is not necessary for all stages. A distributed FFT therefore can proceed in the same way as the serial version, except that for some stages data must be transferred between nodes. This inevitably means some efficiency is lost due to communication between the nodes and the question is what damage this does to the overall performance of the algorithm. Fortunately, we note, the geometry of the PLC method implies that if we perform a distributed 3D FFT on L nodes, then *in each principal direction*, the 1D FFT required is only distributed over $\approx L^{1/3}$ nodes. This means, for example, that a 3D FFT distributed over 512 nodes requires a 1D distributed FFT that performs well on only 8 nodes. Because of this $L^{1/3}$ dependency, the impact of inter-node communication therefore turns out to be less than we might imagine. An acceptable implementation of SPME in the PLC algorithm is therefore:

1. Each node handles a different subset n_p atoms, where $n_p \approx N/L$.
2. The $\boldsymbol{Q}$ array is constructed in separate parts on each node, based on the n_p atoms the node handles.
3. A *distributed* 3D FFT is performed on $\boldsymbol{Q}$ and the result is the distributed array $\check{\boldsymbol{Q}}$.
4. Distributed array $\boldsymbol{G}$ is obtained from the array $\check{\boldsymbol{Q}}$.
5. A distributed 3D FFT is performed on $\boldsymbol{G}$.
6. The potential energy $\Phi^F(\{\vec{r}_i\})$ is calculated in parts on all nodes and globally summed.
7. The forces acting on the n_p atoms associated with each node are calculated.

In this approach it is assumed that the 3D FFT contains all the communication operations required, which is logically the right place for them to be, so the programmer does not need to code these explicitly. A freely available distributed 3D FFT (known as DAFT) has been described by Bush [59]. The procedure described here has been implemented in the simulation program DL_POLY 4, where it has proven to be highly efficient [53].

Chapter 10

Free Energy Calculations Using Mixed Hamiltonians

10.1 Introduction

The calculation of free energies by molecular dynamics simulations is a huge subject. There are many methods available and new ones frequently appear in the literature. The main thing that connects them all is ingenuity. This is inevitable because the basic molecular dynamics method is not able to calculate free energies directly and some special tricks are required to obtain them. Faced with a wide choice of possible techniques, we have decided to present just one: the mixed Hamiltonian, which is arguably the most widely used method among molecular dynamics practitioners. Focusing on one method allows us to go into more detail on some tricky issues, since the devil is in the detail when it comes to free energy calculations!

10.2 The Mixed Hamiltonian System

We begin by repeating some of section 3.19.2 where the idea of the mixed Hamiltonian was introduced. However we shall make one key modification because it helps to clarify some issues – we shall assume that the two Hamiltonians in the mixed Hamiltonian can have different numbers of atoms.

For two systems, A and B, with atom numbers N_A and N_B, represented by the Hamiltonians $H_A^{N_A}(\vec{\Gamma}^{N_A})$ and $H_B^{N_B}(\vec{\Gamma}^{N_B})$, the mixed system Hamiltonian is

$$H_\lambda^{\tilde{N}}(\vec{\Gamma}^{\tilde{N}}) = (1-\lambda) H_A^{N_A}(\vec{\Gamma}^{N_A}) + \lambda H_B^{N_B}(\vec{\Gamma}^{N_B}), \tag{10.1}$$

in which $\tilde{N} \le N_A + N_B$ is the number of independent atoms in the mixed system and the parameter λ is the so-called mixing parameter, which may take any value between 0 and 1. For our purposes here we will regard systems A and B as *subsystems* of the mixed system. It was shown in section 3.19.2 that the derivative of the free energy with respect to the parameter λ is

$$\frac{\partial F}{\partial \lambda} = \langle \Delta H_{AB}^{\tilde{N}}(\vec{\Gamma}^{\tilde{N}}) \rangle_\lambda = \langle H_B^{N_B}(\vec{\Gamma}^{N_B}) - H_A^{N_A}(\vec{\Gamma}^{N_A}) \rangle_\lambda \tag{10.2}$$

which is an ensemble average obtainable by molecular dynamics simulation. The equation (10.2) can be integrated numerically over the interval $0 \leqslant \lambda \leqslant 1$ to give the free energy difference between the systems A and B, as

$$\Delta F_{AB} = \int_{\lambda=0}^{\lambda=1} \langle \Delta H_{AB}^{\tilde{N}}(\vec{\Gamma}^{\tilde{N}}) \rangle_{\lambda} \, d\lambda . \tag{10.3}$$

This is an example of thermodynamic integration, where the free energy difference is calculated as an integral along some "path", which can be a real physical path or, as in this case, an abstract one.

As we previously indicated, equations (10.1) to (10.3) do not assume that the number of atoms is the same in both subsystems A and B. When they are not, then the thermodynamic integration (10.3) implies the creation or annihilation of degrees of freedom as λ changes from 0 to 1. This fact raises issues of its own, which we shall deal with at the appropriate places. But in practice there are other serious issues that arise when using the mixed Hamiltonian approach, even in its simplest form. The issues are both practical and theoretical and can easily give rise to confusion and even disaster if the method is used uncritically. Hopefully if we raise awareness of these issues here, they will not confound us later.

(Note: To ease our notational burden in what follows we shall henceforth drop obvious arguments such as $(\vec{\Gamma}^{N_A})$ from our formulae and leave them implied.)

10.3 Thermodynamic Integration

Our intention is to calculate the Helmholtz free energy in a canonical system. This means that the system is supposed to be embedded in a heat bath permitting fluctuations in the overall system energy, while maintaining a constant average temperature. It follows that the Hamiltonian (10.1) is not a fully adequate basis for a molecular dynamics simulation of the system. We need to add a thermostat to ensure canonical dynamics. As written, there is nothing necessarily connecting the subsystems A and B and they could indeed be fully independent of each other. If that really was the case we could not establish a thermodynamic equilibrium between them as required by the average in equation (10.2). Equilibrium is only guaranteed if the two subsystems share components in common, such as a heat bath, to ensure an exchange of energy. The use of a canonical NVT molecular dynamics algorithm would therefore be advisable to simulate these systems. This does not rule out the use of NVE molecular dynamics, but it does mean we should be careful to ensure that the subsystems have common components through which a shared equilibrium can be established. In effect the two subsystems must make physical contact with each other.

The most obvious example of a common component is an explicit solvent. Interestingly it turns out that, since we propose to calculate free energy differences, the presence of solvent molecules in both subsystems A and B means that no terms involving the solvent-solvent interactions or the solvent kinetic energy can arise in the average in equation (10.2). (By the same token we would not expect any terms involving the thermostat to appear either.) It is therefore not unreasonable to regard the solvent as an effective heat sink or "canonical thermostat" in partial justification of the use of NVE molecular dynamics in these calculations. In what follows in this section, it is assumed for simplicity that subsystems A and B are equilibrated using a thermostat. The conclusions drawn are the same as those for a system in which a

solvent is common to both subsystems.

The most difficult problems associated with thermodynamic integration arise from the presence of the factors $(1-\lambda)$ and λ in the mixed Hamiltonian. These difficulties are particularly serious at the extremes of the range of λ, (i.e. at values approaching either 0 or 1). For example, when $\lambda=0$ then $H_\lambda^{\tilde{N}}=H_A^{N_A}$ and $H_B^{N_B}$ vanishes from the Hamiltonian. This is unfortunate because we are required to calculate the average quantity $\langle H_B^{N_B}-H_A^{N_A}\rangle_\lambda$ even though $H_B^{N_B}$, being no longer a component of the system, cannot be in equilibrium with it. Indeed trouble may arise well before the extremes are reached, as the vanishing component becomes progressively more detached from the system. This raises practical problems about how to calculate the required averages and also theoretical concerns about how to handle these diminishing components in a physically sensible way.

These problems become clearer if we decompose the mixed Hamiltonian into kinetic and potential energy contributions. It is natural to write the Hamiltonians $H_A^{N_A}$ and $H_B^{N_B}$ as

$$\begin{aligned} H_A^{N_A}&=K_A^{N_A}+\Phi_A^{N_A},\\ H_B^{N_B}&=K_B^{N_B}+\Phi_B^{N_B}. \end{aligned} \tag{10.4}$$

where $K_A^{N_A},\Phi_A^{N_A}$ and $K_B^{N_B},\Phi_B^{N_B}$ are the kinetic and potential energies of subsystems A and B respectively. The mixed Hamiltonian (10.1) now becomes

$$H_\lambda^{\tilde{N}}=(1-\lambda)K_A^{N_A}+\lambda K_B^{N_B}+(1-\lambda)\Phi_A^{N_A}+\lambda\Phi_B^{N_B} \tag{10.5}$$

and equation (10.3) becomes accordingly

$$F_{AB}=\int_0^1\langle K_B^{N_B}-K_A^{N_A}\rangle_\lambda d\lambda+\int_0^1\langle\Phi_B^{N_B}-\Phi_A^{N_A}\rangle_\lambda d\lambda. \tag{10.6}$$

It is easily seen that in the Hamiltonian (10.5) the terms $(1-\lambda)\Phi_A^{N_A}$ and $\lambda\Phi_B^{N_A}$ are equivalent to ordinary potential energy functions that behave like the original, forms but happen to have weaker interaction strengths. Likewise the terms $(1-\lambda)K_A^{N_A}$ and $\lambda K_B^{N_B}$ are equivalent to the kinetic energies of molecules with masses reduced by the same scaling factors. The mixed Hamiltonian is thus equivalent to

$$H_\lambda^{\tilde{N}}=\tilde{K}_A^{N_A}+\tilde{K}_B^{N_B}+\tilde{\Phi}_A^{N_A}+\tilde{\Phi}_B^{N_B}. \tag{10.7}$$

where we have the identities

$$\begin{aligned}\tilde{K}_A^{N_A}&=(1-\lambda)K_A^{N_A},\\ \tilde{K}_B^{N_B}&=\lambda K_B^{N_B},\\ \tilde{\Phi}_A^{N_A}&=(1-\lambda)\Phi_A^{N_A},\\ \tilde{\Phi}_B^{N_B}&=\lambda \Phi_B^{N_B}.\end{aligned} \tag{10.8}$$

We may now consider the kinetic and potential components of (10.7) separately.

10.3.1 The Kinetic Energy

It may now be understood that a molecular dynamics simulation based on (10.7) can be equilibrated to a specified temperature T so that[25]

$$\begin{aligned}\langle \tilde{K}_A^{N_A}\rangle_\lambda&=\frac{3}{2}N_B k_B T,\\ \langle \tilde{K}_B^{N_B}\rangle_\lambda&=\frac{3}{2}N_A k_B T.\end{aligned} \tag{10.9}$$

It follows from equations (10.8) and (10.9) that

$$\begin{aligned}\langle K_A^{N_A}\rangle_\lambda&=\frac{3}{2(1-\lambda)}N_A k_B T,\\ \langle K_B^{N_B}\rangle_\lambda&=\frac{3}{2\lambda}N_B k_B T,\end{aligned} \tag{10.10}$$

and we immediately see that the kinetic energies, $K_A^{N_A}$ and $K_B^{N_B}$, of subsystems A and B are significantly larger in the mixed system than they would be in the corresponding pure state systems. In practice this means that the particle velocities are higher than the simulation temperature implies. Furthermore, at the extremes of the scaling range, when λ approaches either 0 or 1, these kinetic energies, and their associated atomic velocities approach infinity! It would seem from this that a practical simulation is impossible and all simulations at the extremes of the λ scale are doomed to break down. However this only happens if the equations of motion are integrated using the atomic velocities (as is often the case in simulation programs). The Hamiltonian momentum, on the other hand, retains a finite value for all values of λ. Unfortunately this will not prevent the unlimited rise of $\langle K_A^{N_A}\rangle_\lambda$ and $\langle K_B^{N_B}\rangle_\lambda$ at the opposing ends of the λ scale, but this need not to be a problem as we now show.

From (10.10) we find that

$$\langle K_B^{N_B}-K_A^{N_A}\rangle_\lambda=\frac{3}{2}k_B T\left\{\frac{N_B}{\lambda}-\frac{N_A}{(1-\lambda)}\right\}, \tag{10.11}$$

which evidently approaches infinity at both extremes of the λ scale. This implies that the kinetic energy contribution to the integration of equation (10.6) is impossible to obtain. However it pays to look at the actual integral which is

25 Hopefully the suffix B in Boltzmann's constant k_B causes no confusion here!

$$\int_0^1 \langle K_B^{N_B} - K_A^{N_A} \rangle_\lambda \, d\lambda = \frac{3}{2} N_B k_B T \int_0^1 \frac{1}{\lambda} d\lambda - \frac{3}{2} N_A k_B T \int_0^1 \frac{1}{(1-\lambda)} d\lambda. \qquad (10.12)$$

The substitution $\mu \to (1-\lambda)$ into the second integral on the right of (10.12) shows that it is the negative of the first and we can write

$$\int_0^1 \langle K_B^{N_B} - K_A^{N_A} \rangle_\lambda = \frac{3}{2} (N_B - N_A) k_B T \int_0^1 \frac{1}{\lambda} d\lambda. \qquad (10.13)$$

The integral on the right is of course infinite, but we note the presence of the factor $(N_B - N_A)$ which is zero when N_A and N_B are equal. The conclusion to be drawn is that when subsystems A and B have the same number of degrees of freedom, there is no kinetic energy contribution to the free energy integral. This observation is useful for designing thermodynamic integration schemes that are free of infinities.

However, it is also useful to note that a change in the number of degrees of freedom is associated with the creation or destruction of atoms or molecules and sometimes this merely a side effect of Hamiltonians constructed to obtain some particular property. In such cases the creation or destruction of atoms is not scientifically relevant. It should therefore be possible to extract the problematic terms (if they can be identified) and obtain a free energy integral that includes only terms relevant to the property of interest. Such an integral will be free of the infinities associated with changes in the number of degrees of freedom. We return to this issue later, when explicit examples will be presented.

10.3.2 The Potential Energy

Problems with potential energy arise from the fact that the scaled potentials of the mixed Hamiltonian represent weakened interactions. Weaker interactions allow atoms to approach each other much more closely that they would do under the normal potentials. However, when the averages of the potential terms of the integral (10.6) are computed, these represent normal strength interactions. In consequence, when atoms have approached to very short ranges, the normal interaction energy is extremely large. As a result of this, the potential energies obtained near the ends of the λ scale are subject to enormous fluctuations, which makes computing reliable averages more difficult.

Another way in which the close approach of atoms is problematic is when the configuration obtained in a simulation that employed a particular value of λ is used to initiate another simulation where the value of λ is different. It is often found that extremely large forces can arise, which immediately destabilize the simulation. This can usually be put right by careful amendment of the starting structure and likelihood of it occurring is reduced by progressing through small changes in λ from one simulation to another. It is a practical problem, not an intrinsic one like obtaining good averages for the potential energy, but one needs to be aware of it.

To help with the averaging problem we now consider how we may benefit from using a function of λ (i.e. $f(\lambda)$) in place of λ in equation (10.1). By this substitution we may impose additional conditions to help alleviate problems at the extremes of λ.

This turns out to be a helpful approach, as we shall see.

In place of equation (10.1) we write

$$H_{\lambda}^{\tilde{N}}=(1-f(\lambda))H_{A}^{N_A}+f(\lambda)H_{B}^{N_B} \tag{10.14}$$

and following the same procedures that gave us equation (10.2) we find that

$$\frac{\partial F}{\partial \lambda}=\left\langle \Delta H_{AB}^{\tilde{N}}\left(\frac{\partial f(\lambda)}{\partial \lambda}\right)\right\rangle_{\lambda}. \tag{10.15}$$

This equation is no more difficult to integrate numerically than equation (10.2) and to help overcome the problem of convergence of the average potential energy terms encountered, we may propose that $f(\lambda)$ has the following properties:

1. $f(\lambda)=0$ when $\lambda=0$ and $f(\lambda)=1,$ when $\lambda=1,$ as required by the mixing equation;
2. $\partial f(\lambda)/\partial\lambda=0$ when $\lambda=0$ or $\lambda=1$;
3. $\partial f(\lambda)/\partial\lambda\to 0$ in a continuous manner when $\lambda\to 0$ and $\lambda\to 1,$ so as to suppress the contribution of the term $\Delta H_{AB}^{N_{AB}}$ at these extremes.

The significance of point 2 is that if we can ensure the derivative is zero at the extremes, the problem contribution of the missing component of the Hamiltonian becomes void. Furthermore, if the derivative function goes smoothly to zero at the extremes, as demanded by point 3, large fluctuations (and hence large errors) in the averages can be dampened down. It is not difficult to propose functional forms for $f(\lambda)$ that meet these requirements. Some examples are:

- Trigonometric mixing:

$$f(\lambda)=\frac{1}{2}(1+\sin(\pi(\lambda-1/2))) \tag{10.16}$$

- Spline kernel mixing:

$$f(\lambda)=-1/2+2\lambda-8(1-|(\lambda-1/2)|)(\lambda-1/2)^3 \tag{10.17}$$

- Error function mixing:

$$f(\lambda)=\frac{\alpha}{\sqrt{\pi}}\int_{0}^{\lambda}\exp(-\alpha^2(x-1/2))\,dx \tag{10.18}$$

  where α is a parameter of order 10~11;

- Polynomial mixing:

$$f(\lambda)=1-(1-\lambda)^k\sum_{i=0}^{k-1}\frac{(k-1+i)!}{(k-1)!}\lambda^i, \tag{10.19}$$

where k is an integer exponent.

All these functions have the properties listed under items 1 and 2 above. (Error function mixing does not quite fulfil the requirement that the derivative becomes zero at the extremes, but it is clear that we can easily make it sufficiently small to produce negligible error.) Not all these functional forms are equally good at suppressing large fluctuations at the extremes of range.

Thus using these functions we can eliminate the extreme range problem involving the vanishing component and at the same time do something to reduce the importance of the statistical errors in the evaluation of the ensemble average. This represents a practical solution to the problem of obtaining reliable potential energy averages at the extremes of the λ scale.

It should be noted that using $f(\lambda)$ rather than λ (with the properties described above) would not help in the treatment of the kinetic energy. In place of equation (10.13) we would obtain

$$\int_0^1 \langle K_B^{N_B} - K_A^{N_A} \rangle_\lambda = \frac{3}{2}(N_B - N_A) k_B T \int_0^1 \frac{1}{f(\lambda)} \left(\frac{\partial f(\lambda)}{\partial \lambda} \right) d\lambda . \tag{10.20}$$

Once again the (logarithmic) integral on the right is infinite and we can only produce a sensible answer (zero) if N_A and N_B are equal.

10.4 The Calculation of Gibbs Free Energies

The thermodynamic integration approach can also be used to determine Gibbs free energy differences. For this purpose we need to perform NPT simulations to ensure the integration under constant pressure conditions. Firstly however we need a means to determine the pressure in the mixed system.

Formally the pressure $P_\lambda^{\tilde{N}}$ in the mixed system is obtained from thermodynamics, as was described in section 3.19.3 . In this case we re-write equation (3.123) as:

$$P_\lambda^{\tilde{N}} = \left\langle -\left(\frac{\partial H_\lambda^{\tilde{N}}}{\partial V} \right)_T \right\rangle_\lambda . \tag{10.21}$$

where V is the volume of the mixed system. Using the mixed Hamiltonian (10.1) this gives

$$P_\lambda^{\tilde{N}} = (1-\lambda) \left\langle -\left(\frac{\partial (K_A^{N_A} + \Phi_A^{N_A})}{\partial V} \right)_T \right\rangle + \lambda \left\langle -\left(\frac{\partial (K_B^{N_B} + \Phi_B^{N_B})}{\partial V} \right)_T \right\rangle_\lambda , \tag{10.22}$$

which can be written as

$$P_\lambda^{\tilde{N}} = (1-\lambda) P_A^{N_A} + \lambda P_B^{N_B} , \tag{10.23}$$

where $P_A^{N_A}$ and $P_B^{N_B}$ are the independent pressures of the subsystems A and B in the volume V.

The Gibbs free energy associated with a mixed Hamiltonian system of the kind given in equation (10.1) is given by

$$G_{AB} = -\beta^{-1} \log\left(Q_\lambda^{\tilde{N}}\right), \tag{10.24}$$

where the partition function $Q_\lambda^{\tilde{N}}$ is given by

$$Q_\lambda^{\tilde{N}} = \frac{h^{-3\tilde{N}}}{V\,\tilde{N}!} \int dV \int\int \exp\left(-\beta\left[H_\lambda^{\tilde{N}} + P_\lambda^{\tilde{N}} V\right]\right) d\,\vec{r}^{\,\tilde{N}}\, d\,\vec{p}^{\,\tilde{N}}. \tag{10.25}$$

$H_\lambda^{\tilde{N}}$ and $P_\lambda^{\tilde{N}}$ are as given in equations (10.1) and (10.23) respectively, while V represents the system volume. Differentiation of (10.25) with respect to λ leads to the result

$$\frac{\partial G_{AB}}{\partial \lambda} = \left\langle \left(H_B^{N_B} - H_A^{N_A}\right) + \left(P_B^{N_B} - P_A^{N_A}\right) V \right\rangle_\lambda. \tag{10.26}$$

This is the Gibbs analogue of equation (10.2), where it is understood that the ensemble average is to be evaluated in the NPT ensemble, with $P^{\tilde{N}}$ as the target pressure. It should be clear from equations (10.1) and (10.23) that when λ is 0 or 1 the simulated system becomes one or the other pure subsystem at the same temperature and pressure and at the corresponding equilibrium volume. Integration of (10.26) with respect to λ will therefore yield the Gibbs free energy difference between the states.

Of course this is the formal statement only. Practically, we have similar problems to the Helmholtz case where at the extreme values of λ it becomes problematic to calculate the ensemble average in (10.26) and there will be large fluctuations as the extremes are approached. However we may again employ the functions $f(\lambda)$ like those shown in equations (10.16) to alleviate the problem. In this case (10.26) becomes

$$\frac{\partial G_{AB}}{\partial \lambda} = \left\langle \left(\left(H_B^{N_B} - H_A^{N_A}\right) + \left(P_B^{N_B} - P_A^{N_A}\right) V\right)\left(\frac{\partial f(\lambda)}{\partial \lambda}\right) \right\rangle_\lambda, \tag{10.27}$$

where it is no longer essential to evaluate the ensemble average at the extremes of λ. The properties of $f(\lambda)$ found convenient in the Helmholtz case will be advantageous here also.

Equation (10.27) can be reduced to a form that is more convenient for computational purposes. First we note that, after performing the differentiation with respect to the volume V in equation (10.22), the pressures $P_A^{N_A}$ and $P_B^{N_B}$ can be written as

$$P_A^{N_A} = \left(2K_A^{N_A} - \Psi_A^{N_A}\right)/3V$$
$$P_B^{N_B} = \left(2K_B^{N_B} - \Psi_B^{N_B}\right)/3V \quad , \tag{10.28}$$

where $\Psi_A^{N_A}$ and $\Psi_B^{N_B}$ represent the virials of subsystems A and B respectively. Substituting (10.28) and (10.4) into (10.27) gives

$$\frac{\partial G_{AB}}{\partial \lambda} = \left\langle \left(\frac{5}{3}\left(K_B^{N_B} - K_A^{N_A}\right) + \left(\Phi_B^{N_B} - \Phi_A^{N_A}\right) - \frac{1}{3}\left(\Psi_B^{N_B} - \Psi_A^{N_A}\right) \right) \left(\frac{\partial f(\lambda)}{\partial \lambda} \right) \right\rangle_{\lambda}, \tag{10.29}$$

The kinetic energy and potential terms are expected to behave much as they do in the Helmholtz case, while the virial terms are expected to behave similarly to the potential terms. We also note that, once again, the kinetic energy terms will cancel if the numbers of atoms in subsystems A and B are the same.

In the following section we give some examples of how thermodynamic integration can be used to obtain free energies.

10.5 Some Examples of Free Energy Calculations

10.5.1 Example 1: The Helmholtz Free Energy of Solvation.

In this example we have a solvent S and a solute X together in the same simulation and by increasing the mixing parameter λ we gradually produce the conditions for a solution. In order to calculate the free energy of solvation we use the Hamiltonians

$$H_A^{N_{XS}} = K_S^{N_S} + \Phi_S^{N_S} + K_X^{N_X} + \Phi_X^{N_X}$$
$$H_B^{N_{XS}} = K_S^{N_S} + \Phi_S^{N_S} + K_X^{N_X} + \Phi_X^{N_X} + \Phi_{XS}^{N_{XS}} \tag{10.30}$$

In which N_S, N_X are the numbers of solvent and solute atoms respectively, with $N_{XS} = N_S + N_X$. Also $K_S^{N_S}, K_X^{N_X}$ are the kinetic energies of the solvent and solute molecules, $\Phi_S^{N_S}, \Phi_X^{N_X}$ are the solvent-solvent and solute-solute potential energies and $\Phi_{XS}^{N_{XS}}$ is the solvent-solute potential energy. What is noticeable about these Hamiltonians is that they are based on the same set of atoms, and only differ in that $H_B^{N_{XS}}$ contains an explicit term for solvent-solute interactions. Combining these Hamiltonians into a mixed Hamiltonian as required gives

$$H_\lambda^{N_{XS}} = K_S^{N_S} + \Phi_S^{N_S} + K_X^{N_X} + \Phi_X^{N_X} + f(\lambda)\Phi_{XS}^{N_{XS}}. \tag{10.31}$$

In which we see that $\tilde{N}$ from equation (10.1) is represented by N^{XS} in this equation. Significantly, the kinetic energies of the solvent and the solute are not scaled by the mixing parameter, which means that, during simulation, the velocities of solvent and solute molecules will not become unduly large at the extremes of the λ scale. We should also mention that coupling between the solvent and solute Hamiltonians is weak when $f(\lambda)$ is small, so proper equilibration between the system components could be a problem unless we are using a canonical (i.e. thermostatted) ensemble.

Following from equation (10.15) the relevant free energy equation is

$$\frac{\partial F}{\partial \lambda} = \left\langle \Phi_{XS}^{N_{XS}} \partial_\lambda f(\lambda) \right\rangle_\lambda, \tag{10.32}$$

where we have used the symbol ∂_λ to indicate a partial derivative with respect to λ (i.e. $\partial_\lambda = \partial / \partial \lambda$). Since the Hamiltonians $H_A^{N_{XS}}$ and $H_B^{N_{XS}}$ have the same number of degrees of freedom, this result confirms what was first pointed out in section 10.3: the kinetic energy makes no contribution to the free energy difference between the two systems. It is also evident in this case that the solvent-solvent and solute-solute interactions similarly make no contribution.

The integration of (10.32) provides the free energy of solvation, $\Theta_{X,S}$, of the solute X in solvent S as

$$\Theta_{X,S} = \int_0^1 \left\langle \Phi_{XS}^{N_{XS}} \partial_\lambda f(\lambda) \right\rangle_\lambda d\lambda. \tag{10.33}$$

This is the Helmholtz free energy difference between the solute in solution and the solute alone at the same volume. If a single solute molecule is used we may say that this is also the chemical potential. In practice the equation (10.33) is integrated numerically.

10.5.2 Example 2. The Comparative Solubility of Two Solutes

In this example a solute X is gradually transformed into the solute Y while in a solvent S and we wish to determine if X more, or less, soluble than Y. There are potentially some complications in this. Firstly, changing one solute into another implies that the number of atoms may change. Secondly, the change also means that we are effectively destroying one species while creating another and this also has an associated change in free energy. This *free energy of formation and destruction* is not a component of the solvation free energy and must be extracted from the calculation. We must therefore design a simulation protocol that avoids these complications.

In the previous example the solute X retained a presence on both Hamiltonians (thus satisfying the conservation of degrees of freedom) and the solute-solvent interaction was "phased in" as λ increased. In his case we combine this with a procedure in which solute Y is also present in both Hamiltonians, but its solute-solvent interaction is "phased out" as λ increases. At the same time neither of the solutes interacts with the other and effectively remain "invisible" to each other throughout. As with the previous example, this approach requires a canonical ensemble simulation to generate a properly equilibrated system.

The start and end Hamiltonians we propose are

$$\begin{aligned} H_A^{N_{XYS}} &= K_S^{N_S} + \Phi_S^{N_S} + K_X^{N_X} + \Phi_X^{N_X} + K_Y^{N_Y} + \Phi_Y^{N_Y} + \Phi_{XS}^{N_{XS}}, \\ H_B^{N_{XYS}} &= K_S^{N_S} + \Phi_S^{N_S} + K_X^{N_X} + \Phi_X^{N_X} + K_Y^{N_Y} + \Phi_Y^{N_Y} + \Phi_{YS}^{N_{YS}}, \end{aligned} \tag{10.34}$$

and the corresponding mixed Hamiltonian is

$$H_{\lambda}^{N_{XYS}} = K_S^{N_S} + \Phi_S^{N_S} + K_X^{N_X} + \Phi_X^{N_X} + K_Y^{N_Y} + \Phi_Y^{N_Y} + \left(1 - f(\lambda)\right)\Phi_{XS}^{N_{XS}} + f(\lambda)\Phi_{YS}^{N_{YS}}. \tag{10.35}$$

The Helmholtz free energy equation derived from this, following (10.15) , is

$$\frac{\partial F}{\partial \lambda} = \left\langle \left(\Phi_{YS}^{N_{YS}} - \Phi_{XS}^{N_{XS}}\right)\partial_{\lambda} f(\lambda) \right\rangle_{\lambda}. \tag{10.36}$$

The difference in the Helmholtz free energy of solvation, $\Delta\Theta_{XY,S}$, is then given by the integral

$$\Delta\Theta_{XY,S} = \int_0^1 \left\langle \left(\Phi_{YS}^{N_{YS}} - \Phi_{XS}^{N_{XS}}\right)\partial_{\lambda} f(\lambda) \right\rangle_{\lambda} d\lambda. \tag{10.37}$$

The full simulation procedure thus requires that we have both solutes present in the system at the same time. Both may interact with the solvent (to degrees determined by the mixing parameter), but neither interacts with the other. The numerical integration of (10.37) provides the Helmholtz free energy difference in solubility of solutes X and Y in the solvent S.

10.5.3 Example 3: Gibbs Free Energy of Solvation.

This calculation is similar to example 1, but there are important differences. We have a solvent S and a solute X and by increasing the mixing parameter λ we gradually produce the solution. In the molecular dynamics simulation the pressure is fixed and it follows that the system volume must be allowed to vary. The obvious methodology for this is to use NPT molecular dynamics (i.e. employing both a thermostat and barostat). The NPT method guarantees that the start and end systems will be at the same pressure. The Hamiltonians we use are

$$\begin{aligned} H_A^{N_S} &= K_S^{N_S} + \Phi_S^{N_S}, \\ H_B^{N_{XS}} &= K_S^{N_S} + \Phi_S^{N_S} + K_X^{N_X} + \Phi_X^{N_X} + \Phi_{XS}^{N_{XS}}, \end{aligned} \tag{10.38}$$

from which we obtain the mixed Hamiltonian

$$H_{\lambda}^{N_{XS}} = K_S^{N_S} + \Phi_S^{N_S} + f(\lambda)\left(K_X^{N_X} + \Phi_X^{N_X} + \Phi_{XS}^{N_{XS}}\right). \tag{10.39}$$

In contrast to example 1, this Hamiltonian progressively creates the solute in solution as λ increases. In consequence the free energy difference between the two Hamiltonians in equations (10.38) includes the free energy of creation of the solute as well as the free energy of solvation. We therefore need to modify the procedure somehow if we require only the solvation free energy.

The pressure resulting from the mixed Hamiltonian is obtained by differentiating (10.39) with respect to the system volume (see equation (7.1)), which gives

$$P_{\lambda}^{N_{XS}} = P_{S}^{N_{S}} + f(\lambda)\left(P_{X}^{N_{X}} - \partial_{V}\Phi_{XS}^{N_{XS}}\right) \tag{10.40}$$

where $P_{S}^{N_{S}}$ and $P_{X}^{N_{X}}$ are the pressures of the pure solvent and pure solute respectively at the equilibrium volume V, which is the volume of the mixed system in an NPT simulation based on the Hamiltonian (10.39). The term $\partial_{V}\Phi_{XS}^{N_{XS}}$ is the derivative of the solute-solvent interaction with respect to the mixed system volume.

Following section 10.4 the Gibbs free energy equation for the mixed system is

$$\frac{\partial G}{\partial \lambda} = \left\langle \left(K_{X}^{N_{X}} + \Phi_{X}^{N_{X}} + \Phi_{XS}^{N_{XS}} + V\left(P_{X}^{N_{X}} - \partial_{V}\Phi_{XS}^{N_{XS}}\right)\right)\partial_{\lambda} f(\lambda)\right\rangle_{\lambda}. \tag{10.41}$$

In this equation we can identify terms $\phi_{XS}^{N_{XS}}$ and $V\partial_{V}\phi_{XS}^{N_{XS}}$ that contribute exclusively to the free energy of solvation and terms $K_{X}^{N_{X}}$, $\Phi_{X}^{N_{X}}$ and $P_{X}^{N_{X}}V$ that contribute exclusively to the free energy of solute creation. The latter we discard, since they are not relevant to the issue of solvation. Thus equation (10.41) may be rewritten as

$$\frac{\partial G}{\partial \lambda} = \left\langle \left(\Phi_{XS}^{N_{XS}} - V\partial_{V}\Phi_{XS}^{N_{XS}}\right)\partial_{\lambda} f(\lambda)\right\rangle_{\lambda}, \tag{10.42}$$

which contains only terms relevant to solvation. The integration of (10.42) gives the Gibbs free energy of solvation as

$$\Xi_{X,S} = \int_{0}^{1} \left\langle \left(\Phi_{XS}^{N_{XS}} - \Psi_{XS}^{N_{XS}}/3\right)\partial_{\lambda} f(\lambda)\right\rangle_{\lambda} d\lambda, \tag{10.43}$$

where $\Psi_{XS}^{N_{XS}}$ is a virial obtained from

$$\Psi_{XS}^{N_{XS}} = 3V\partial_{V}\Phi_{XS}^{N_{XS}}. \tag{10.44}$$

Thus provided we can obtain the terms $\phi_{XS}^{N_{XS}}$ and $V\partial_{V}\phi_{XS}^{N_{XS}}$ through simulation, there is no difficulty in principle in evaluation the integral (10.43). This is despite the fact that the initial and final systems may have a different number of degrees of freedom.

10.5.4 Example 4: Comparative Gibbs Free Energy of Solvation

The requirement in this example is to determine the Gibbs free energy difference between solutions of solutes X and Y in the solvent S at the same pressure. It is thus the Gibbs counterpart to Example 2 above. Once again we require a thermostatted and barostatted NPT simulation and must perform a thermodynamic integration along a path that converts the solution of X into a solution of Y in the solvent S. We can not re-use the Hamiltonians (10.34) and (10.35) for our study, however, because the start and end systems of the path over λ will contain *both* solutes and the system pressure and volume will not correspond to the required "pure" solutions. For this reason the start and end Hamiltonians we use for this

purpose are

$$\begin{aligned} H_A^{N_{XS}} &= K_S^{N_S} + \Phi_S^{N_S} + K_X^{N_X} + \Phi_X^{N_X} + \Phi_{XS}^{N_{XS}}, \\ H_B^{N_{YS}} &= K_S^{N_S} + \Phi_S^{N_S} + K_Y^{N_Y} + \Phi_Y^{N_Y} + \Phi_{YS}^{N_{YS}} \end{aligned} \tag{10.45}$$

and the mixed Hamiltonian is then

$$\begin{aligned} H_\lambda^{N_{XYS}} = K_S^{N_S} + \Phi_S^{N_S} + (1-f(\lambda))(K_X^{N_X} + \Phi_X^{N_X} + \Phi_{XS}^{N_{XS}}) + \\ f(\lambda)(K_Y^{N_Y} + \Phi_Y^{N_Y} + \Phi_{YS}^{N_{YS}}) \end{aligned} \tag{10.46}$$

This guarantees that as the parameter λ increases the mixed system evolves from the pure solution of X to a pure solution of Y in S. The pressure in the mixed system is given by differentiating (10.46) with respect to volume (see equation (7.1)) and is

$$P_\lambda^{N_{XYS}} = P_S^{N_S} + (1-f(\lambda))(P_X^{N_X} - \partial_V \Phi_{XS}^{N_{XS}}) + f(\lambda)(P_Y^{N_Y} - \partial_V \Phi_{YS}^{N_{YS}}) \tag{10.47}$$

where $P_X^{N_X}$ and $P_Y^{N_Y}$ are the independent pressures of the pure solutes in the equilibrium volume V of mixed system under the pressure $P_\lambda^{N_{XYS}}$.

This choice of Hamiltonians has the required property of producing the pure solutions at opposite ends of the λ path, but it is not suitable in other ways. Unlike the choice made in Example 2, it does not automatically guarantee the conservation of the number of degrees of freedom along the path, which follows from the creation and destruction of the solute molecules. We therefore anticipate some remedial intervention will be necessary at a later stage.

Following section 10.4 the Gibbs free energy equation for the mixed system is

$$\frac{\partial G}{\partial \lambda} = \left\langle \left(\{\Delta K_{XY}^{N_{XY}} + \Delta \Phi_{XY}^{N_{XY}} + \Delta \Phi_{XYS}^{N_{XYS}}\} + \{\Delta P_{XY}^{N_{XY}} - \partial_V \Delta \Phi_{XYS}^{N_{XYS}}\} V \right) \partial_\lambda f(\lambda) \right\rangle_\lambda, \tag{10.48}$$

where

$$\begin{aligned} \Delta K_{XY}^{N_{XY}} &= K_Y^{N_Y} - K_X^{N_X}, \\ \Delta \Phi_{XY}^{N_{XY}} &= \Phi_Y^{N_Y} - \Phi_X^{N_X}, \\ \Delta P_{XY}^{N_{XY}} &= P_Y^{N_Y} - P_X^{N_X}, \\ \Delta \Phi_{XYS}^{N_{XYS}} &= \Phi_{YS}^{N_{YS}} - \Phi_{XS}^{N_{XS}}. \end{aligned} \tag{10.49}$$

As in the previous example, we can now identify the terms $\Delta K_{XY}^{N_{XY}}$, $\Delta \Phi_{XY}^{N_{XY}}$ and $\Delta P_{XY}^{N_{XY}} V$ which are exclusively associated with the creation or destruction of solute molecules and the terms $\Delta \Phi_{XYS}^{N_{XY}}$ and $V \partial_V \Delta \Phi_{XYS}^{N_{XY}}$ which are exclusively associated with solvation. It is only the latter terms that we require to determine the free energy difference of solvation and we can discard the others. Thus we can now write the required thermodynamic integral as

$$\Delta\Xi_{XY,S}=\int_0^1\left\langle\left(\Delta\Phi_{XYS}^{N_{XYS}}-\Delta\Psi_{XYS}^{N_{XYS}}/3\right)\partial_\lambda f(\lambda)\right\rangle_\lambda d\lambda. \tag{10.50}$$

where $\Delta\Xi_{XY,S}$ is the difference in the free energy of solubility between the solutes X and Y in the solvent S. In (10.50) we have also used the virial quantity $\Delta\Psi_{XYS}^{N_{XYS}}$ defined by

$$\Delta\Psi_{XYS}^{N_{XYS}}=3V\partial_V\Delta\Phi_{XYS}^{N_{XYS}}. \tag{10.51}$$

Thus once again we have obtained a result free from the infinities associated with changing the number of degrees of freedom along the thermodynamic path.

10.6 The Properties of Mixed Hamiltonian Systems

A discussion of this topic is necessary since it underpins every molecular dynamics simulation that uses a mixed Hamiltonian. As we shall see, there are some aspects of the mechanics that raise unexpected issues and we need to be aware of them.

10.6.1 Dynamics

We shall take as our example the Hamiltonians appropriate for two solutes X,Y with a common solvent S for which the separate Hamiltonians are

$$\begin{aligned}H_A^{N_{XS}}&=K_S^{N_S}+\Phi_S^{N_S}+K_X^{N_X}+\Phi_X^{N_X}+\Phi_{XS}^{N_{XS}},\\ H_B^{N_{YS}}&=K_S^{N_S}+\Phi_S^{N_S}+K_Y^{N_Y}+\Phi_Y^{N_Y}+\Phi_{YS}^{N_{YS}},\end{aligned} \tag{10.52}$$

where $K_X^{N_X},K_Y^{N_Y}$ and $K_S^{N_S}$ are respectively the kinetic energies of the solutes X,Y and the solvent S and $\Phi_X^{N_X},\Phi_Y^{N_Y},\Phi_S^{N_S},\Phi_{XS}^{N_{XS}}$ and $\Phi_{YS}^{N_{YS}}$ are the interaction energies within and between these components.

The mixed Hamiltonian in this case is

$$H_\lambda^{\tilde{N}}=K_S^{N_S}+\Phi_S^{N_S}+(1-\lambda)\left(K_X^{N_X}+\Phi_X^{N_X}+\Phi_{XS}^{N_{XS}}\right)+\lambda\left(K_Y^{N_Y}+\Phi_Y^{N_Y}+\Phi_{YS}^{N_{YS}}\right) \tag{10.53}$$

where

$$\tilde{N}=N_X+N_Y+N_S \tag{10.54}$$

and the free energy equation is

$$\frac{\partial F}{\partial\lambda}=\left\langle\left(K_Y^{N_Y}+\Phi_Y^{N_Y}+\Phi_{YS}^{N_{YS}}\right)-\left(K_X^{N_X}+\Phi_X^{N_X}+\Phi_{XS}^{N_{XS}}\right)\right\rangle_\lambda. \tag{10.55}$$

Following Hamilton (section 2.5), the momentum of an individual atom is obtained from the kinetic energy

$$p_i = \frac{\partial K}{\partial \dot{x}_i} = \frac{\partial K_S^{N_S}}{\partial \dot{x}_i} + (1-\lambda)\frac{\partial K_X^{N_X}}{\partial \dot{x}_i} + \lambda \frac{\partial K_Y^{N_Y}}{\partial \dot{x}_i}, \tag{10.56}$$

where p_i represents a momentum component and $\dot{x}_i$ is the corresponding velocity for the coordinate component x_i. This gives

$$\begin{array}{lll} p_i = m_i x_i & \text{if } i \text{ is a solvent atom;} & \text{(a)} \\ p_i = (1-\lambda) m_i x_i & \text{if } i \text{ is an atom of solute } X, \text{ or} & \text{(b)} \\ p_i = \lambda m_i x_i & \text{if } i \text{ is an atom of solute } Y, & \text{(c)} \end{array} \tag{10.57}$$

or in general

$$p_i = \alpha_i m_i \dot{x}_i, \tag{10.58}$$

where

$$\alpha_i = 1, (1-\lambda), \lambda, \tag{10.59}$$

as appropriate. Hamilton's equations of motion are therefore

$$\begin{array}{ll} \dot{p}_i = -\dfrac{\partial H_\lambda^{\tilde{N}}}{\partial x_i} = -\dfrac{\partial}{\partial x_i}\left(\Phi_S^{N_S} + (1-\lambda)(\Phi_X^{N_X} + \Phi_{XS}^{N_{XS}}) + \lambda(\Phi_Y^{N_Y} + \Phi_{YS}^{N_{YS}})\right), & \text{(a)} \\ \dot{x}_i = \dfrac{\partial H_\lambda^{\tilde{N}}}{\partial p_i} = \dfrac{\partial}{\partial p_i}\left(K_S^{N_S} + (1-\lambda) K_X^{N_X} + \lambda K_Y^{N_Y}\right). & \text{(b)} \end{array} \tag{10.60}$$

From equation (10.60)(a) we obtain

$$\begin{array}{lll} \dot{p}_i = -\dfrac{\partial}{\partial x_i}\left(\Phi_S^{N_S} + (1-\lambda)\Phi_{XS}^{N_{XS}} + \lambda \Phi_{YS}^{N_{YS}}\right), & \text{for } i \text{ in } S & \text{(a)} \\ \dot{p}_i = -(1-\lambda)\dfrac{\partial}{\partial x_i}(\Phi_X^{N_X} + \Phi_{XS}^{N_{XS}}), & \text{for } i \text{ in } X & \text{(b)} \\ \dot{p}_i = -\lambda \dfrac{\partial}{\partial x_i}\left(\Phi_Y^{N_Y} + \Phi_{YS}^{N_{YS}}\right), & \text{for } i \text{ in } Y. & \text{(c)} \end{array} \tag{10.61}$$

While equation (10.60)(b) leads to

$$\dot{x}_i = \frac{p_i}{\alpha_i m_i}, \tag{10.62}$$

which is a re-arrangement of (10.58).

We note in passing that at the absolute extremes of λ the atomic momentum of one of the solutes defined in (10.57)(b) or (10.57)(c) will vanish and so will the corresponding equation of motion (10.61)(b) or (10.61)(c). This implies that there is no practical problem in using the Hamiltonian equations at these extremes – we simply obtain equations corresponding to either pure state A or pure state B. This is a potentially useful observation but it does not mean there are no problems evaluating the averages in (10.55) when λ is either 0 or 1, as will become

evident below.

Many molecular dynamics programs are designed to solve the equations of motion based on acceleration rather than rate of change of momentum, so we might recast equations (10.61) and (10.62) into an alternative form. Differentiating the equation (10.58) with respect to time and substituting the result into (10.61) gives, after dividing out the common factor α_i defined in (10.59):

$$\begin{aligned}
\ddot{x}_i &= -\frac{1}{m_i}\frac{\partial}{\partial x_i}\left(\Phi_S^{N_S} + (1-\lambda)\Phi_{XS}^{N_{XS}} + \lambda\Phi_{YS}^{N_{YS}}\right), && \text{for } i \text{ in } S && \text{(a)} \\
\ddot{x}_i &= -\frac{1}{m_i}\frac{\partial}{\partial x_i}\left(\Phi_X^{N_X} + \Phi_{XS}^{N_{XS}}\right), && \text{for } i \text{ in } X && \text{(b)} \\
\ddot{x}_i &= -\frac{1}{m_i}\frac{\partial}{\partial x_i}\left(\Phi_Y^{N_Y} + \Phi_{YS}^{N_{YS}}\right), && \text{for } i \text{ in } Y. && \text{(c)}
\end{aligned} \qquad (10.63)$$

Note that in the cases where λ is either 0 or 1, dividing out the common factor α_i to produce either equation (10.63)(b) or (10.63)(c) is not possible. This means these equations are not justified in these circumstances. However this is not the worst problem with these equations. Despite their apparent simplicity and the fact that they should generate the same atomic trajectories as Hamilton's equations, they cannot easily be used to adapt an existing molecular dynamics program (one which normally integrates the equations of motion via the acceleration). This is because the standard Newtonian definition of momentum $(m_i \dot{x}_i)$ and the Hamiltonian definition that applies here $(\alpha_i m_i \dot{x}_i)$ are manifestly not the same. This difference means that many basic computations that concern momentum such as: temperature initialisation; temperature scaling; momentum initialisation; and conservation of momentum, will need to be restructured, as we show in the following sections. The potential for confusion here is considerable and the equations (10.63) are not recommended.

A conceptually better approach to adapting a Newtonian program is as follows. We first redefine the masses of the atoms in the mixed Hamiltonian in the following way

$$\tilde{m}_i = \alpha_i m_i, \qquad (10.64)$$

where α_i is defined in (10.59). Substituting this into equation (10.58) allows us to write *all* the atomic momenta as

$$p_i = \tilde{m}_i \dot{x}_i. \qquad (10.65)$$

Differentiating this with respect to time and substituting the result into Hamilton's equations (10.61) gives an alternative form to (10.63):

$$\ddot{x}_i = -\frac{1}{\tilde{m}_i}\frac{\partial}{\partial x_i}\left(\Phi_S^{N_S} + (1-\lambda)\Phi_{XS}^{N_{XS}} + \lambda\Phi_{YS}^{N_{YS}}\right), \qquad \text{for } i \text{ in } S \qquad \text{(a)}$$

$$\ddot{x}_i = -\frac{(1-\lambda)}{\tilde{m}_i}\frac{\partial}{\partial x_i}\left(\Phi_X^{N_X} + \Phi_{XS}^{N_{XS}}\right), \qquad \text{for } i \text{ in } X \qquad \text{(b)} \qquad (10.66)$$

$$\ddot{x}_i = -\frac{\lambda}{\tilde{m}_i}\frac{\partial}{\partial x_i}\left(\Phi_Y^{N_Y} + \Phi_{YS}^{N_{YS}}\right), \qquad \text{for } i \text{ in } Y. \qquad \text{(c)}$$

These equations are completely free from the confusion that can occur when using (10.60) and do not require changes to the temperature scaling or momentum initialisation routines of an existing molecular dynamics program. In one important respect however, the equations (10.66) are no better than equations (10.63). When λ is either 0 or 1, some modified particle masses will be zero and the equations (10.66) have no meaning. However, as we have seen in section 10.3 , the use of an appropriate function $f(\lambda)$ in place of λ avoids the need ever to deal with the extremes of the λ scale.

In the following sections it is useful to describe equations (10.61) and (10.62) as the *Hamiltonian representation*, the equations (10.63) as the *pseudo-Newtonian representation* and the equations (10.66) as the *scaled mass representation*. We now explore the properties of these equations and show that the Hamiltonian and scaled mass representations are directly equivalent while the pseudo-Newtonian representation is not.

10.6.2 Temperature

To be sure what temperature means in the mixed Hamiltonian system (10.53), we recast the Hamiltonian into the following form

$$H_\lambda^{\tilde{N}} = K_S^{N_S} + \Phi_S^{N_S} + \tilde{K}_X^{N_X} + \tilde{\Phi}_X^{N_X} + \tilde{\Phi}_{XS}^{N_{XS}} + \tilde{K}_Y^{N_Y} + \tilde{\Phi}_Y^{N_Y} + \tilde{\Phi}_{YS}^{N_{YS}}, \qquad (10.67)$$

where

$$\begin{aligned} \tilde{K}_X^{N_X} &= (1-\lambda)K_X^{N_X}, \\ \tilde{K}_Y^{N_Y} &= \lambda K_Y^{N_Y}, \end{aligned} \qquad (10.68)$$

are scaled kinetic energies (which are equivalent to the kinetic energies of the scaled mass representation) and

$$\begin{aligned} \tilde{\Phi}_X^{N_X} &= (1-\lambda)\Phi_X^{N_X}, \\ \tilde{\Phi}_{XS}^{N_{XS}} &= (1-\lambda)\Phi_{XS}^{N_{XS}}, \\ \tilde{\Phi}_Y^{N_Y} &= \lambda\Phi_Y^{N_Y}, \\ \tilde{\Phi}_{YS}^{N_{YS}} &= \lambda\Phi_{YS}^{N_{YS}}. \end{aligned} \qquad (10.69)$$

are scaled potential energies of the mixed system. There is no need to redefine the solvent terms $K_S^{N_S}$ and $\Phi_S^{N_S}$, since they are not preceded by scaling factors in (10.53).

The Hamiltonian (10.67) is evidently perfectly normal and under Hamiltonian

dynamics (10.61) will therefore evolve as expected to a state of equilibrium with eventual energy equipartition between the degrees of freedom of the system. It follows that the system temperature is given, as expected, by:

$$\frac{3}{2}(\tilde{N}-1)k_B T=\left\langle K_S^{N_S}+\tilde{K}_X^{N_X}+\tilde{K}_Y^{N_Y}\right\rangle, \tag{10.70}$$

where $\tilde{N}$ is the number of atoms in the system (equation (10.54)).

On the assumption of equipartition we may write

$$\begin{aligned}
\frac{3}{2}N_S\left(1-\frac{1}{\tilde{N}}\right)k_B T&=\left\langle K_S^{N_S}\right\rangle,\\
\frac{3}{2}N_X\left(1-\frac{1}{\tilde{N}}\right)k_B T&=\left\langle \tilde{K}_X^{N_X}\right\rangle,\\
\frac{3}{2}N_Y\left(1-\frac{1}{\tilde{N}}\right)k_B T&=\left\langle \tilde{K}_Y^{N_Y}\right\rangle.
\end{aligned} \tag{10.71}$$

The sum of these three equations returns equation (10.70). These equations hold for both the Hamiltonian and scaled mass representations and are the correct temperature relations for the system. In the pseudo-Newtonian representation, we would not naturally write the temperature equation (10.70) using $\tilde{K}_X^{N_X}$ and $\tilde{K}_Y^{N_Y}$, but $K_X^{N_X}$ and $K_Y^{N_Y}$, which would manifestly be wrong. For example, considering solute Y we have from (10.69) and (10.70) that

$$\tilde{K}_Y^{N_Y}=\lambda K_Y^{N_Y}=\frac{3}{2}N_Y\left(1-\frac{1}{\tilde{N}}\right)k_B T. \tag{10.72}$$

From this it is apparent that using $K_Y^{N_Y}$ in place of $\tilde{K}_Y^{N_Y}$ in (10.70) cannot lead to the right definition of temperature. This would make the system thermodynamics and the canonical ensemble dynamics incorrect.

Equation (10.72) also usefully shows that, *at a fixed temperature*, the smaller the factor λ the greater is $K_Y^{N_Y}$. From this we deduce that, for finite λ, the molecules of type Y move more quickly in the mixed system than they would in the unmixed system at the same temperature. Indeed as λ approaches zero, the speed approaches infinity. Similarly as λ increases it follows that the molecules of type X will also speed up. These outcomes demonstrate that at a given temperature in a mixed Hamiltonian system the velocities of some of the atoms exceeds what might normally be expected, so instabilities in the numerical integration are possible. The best escape from this is to revert to the Hamiltonian representation, since the atomic momentum always remains within sensible bounds.

10.6.3 Momentum

Conservation of momentum in the Hamiltonian representation means conservation of the sum of all of the momenta $p_{i.}$ Specifically, it is required that:

$$\sum_{i=1}^{\tilde{N}} p_i = 0 . \tag{10.73}$$

this is a property of the system Hamiltonian and is thus guaranteed. With the aid of equation (10.65) we may write this as

$$\sum_{i=1}^{N_X} \tilde{m}_i \dot{x}_i + \sum_{i=1}^{N_Y} \tilde{m}_i \dot{x}_i + \sum_{i=1}^{N_S} \tilde{m}_i \dot{x}_i = 0 . \tag{10.74}$$

Clearly this result would not be true if the real atomic masses m_i were used in place of the redefined masses $\tilde{m}_i$. This shows once again how the scaled mass representation is superior to the pseudo-Newtonian representation.

We may also define the corresponding centre of mass of the system as

$$MX = \sum_{i=1}^{N_X} \tilde{m}_i x_i + \sum_{i=1}^{N_Y} \tilde{m}_i x_i + \sum_{i=1}^{N_S} \tilde{m}_i x_i = 0 , \tag{10.75}$$

with

$$M = \sum_{i=1}^{\tilde{N}} \tilde{m}_i . \tag{10.76}$$

Then, differentiating (10.75) with respect to time, it is clear from (10.74) that

$$M \dot{X} = 0 , \tag{10.77}$$

thus showing that the Hamiltonian centre of mass remains fixed. Once again, this would not be the case if the true atomic masses m_i were used instead of the redefined masses $\tilde{m}_i$.

10.6.4 The Mixed Hamiltonian without Kinetic Energy Mixing

It is apparent from all the above discussion that the kinetic energy components of the Hamiltonians in the mixed system that are the main difficulty in calculating free energy differences. We have seen that these can be avoided if steps are taken to ensure the kinetic terms vanish from the final integral. This can be done by adjusting the numbers of atoms in the various components so that the numbers of degrees of freedom are constant throughout the integration. Nevertheless we are left with the problem that in some components of the system the atomic velocities are larger than that implied by the temperature and sometimes may require small (and therefore inefficient) time steps for a stable molecular dynamics simulation.

However, if we make the assumption that it is *always* possible to ensure that the number of degrees of freedom is preserved across the λ scale, then another approach becomes possible. For example, in place of the mixed Hamiltonian (10.53) we could propose

$$H_{\lambda}^{\tilde{N}} = K_X^{N_X} + K_Y^{N_Y} + K_S^{N_S} + \Phi_S^{N_S} + (1-\lambda)\left(\Phi_X^{N_X} + \Phi_{XS}^{N_{XS}}\right) + \lambda\left(\Phi_Y^{N_Y} + \Phi_{YS}^{N_{YS}}\right), \tag{10.78}$$

where it is apparent that none of the kinetic energies are scaled by the mixing factor. The Hamiltonian momentum for this system is

$$p_i = \frac{\partial K}{\partial \dot{x}_i} = \frac{\partial K_X^{N_X}}{\partial \dot{x}_i} + \frac{\partial K_Y^{N_Y}}{\partial \dot{x}_i} + \frac{\partial K_S^{N_S}}{\partial \dot{x}_i}, \tag{10.79}$$

from which we obtain

$$p_i = m_i \dot{x}_i \qquad \text{for } i \text{ in } X, Y \text{ or } S. \tag{10.80}$$

Immediately we see there is no distinction between the Hamiltonian and Newtonian momentum, which removes main source of difficulty.

The resulting equations of motion (10.81) for this system resemble (10.66) but are different in that the masses appearing are the normal atomic masses.

$$\begin{aligned}
\ddot{x}_i &= -\frac{1}{m_i}\frac{\partial}{\partial x_i}\left(\Phi_S^{N_S} + (1-\lambda)\Phi_{XS}^{N_{XS}} + \lambda\Phi_{YS}^{N_{YS}}\right), && \text{for } i \text{ in } S \quad \text{(a)} \\
\ddot{x}_i &= -\frac{(1-\lambda)}{m_i}\frac{\partial}{\partial x_i}\left(\Phi_X^{N_X} + \Phi_{XS}^{N_{XS}}\right), && \text{for } i \text{ in } X \quad \text{(b)} \\
\ddot{x}_i &= -\frac{\lambda}{m_i}\frac{\partial}{\partial x_i}\left(\Phi_Y^{N_Y} + \Phi_{YS}^{N_{YS}}\right), && \text{for } i \text{ in } Y. \quad \text{(c)}
\end{aligned} \tag{10.81}$$

In this case the Newtonian equations do not have the difficulties with the temperature or total momentum we encountered earlier; they have the meaning and behaviour we expect for a normal Newtonian system.

The free energy equation for this case is

$$\frac{\partial F}{\partial \lambda} = \left\langle \left(\Phi_Y^{N_Y} + \Phi_{YS}^{N_{YS}}\right) - \left(\Phi_X^{N_X} + \Phi_{XS}^{N_{XS}}\right) \right\rangle_{\lambda}. \tag{10.82}$$

Integration of (10.82) is straightforward. Note however that the approach does nothing to dampen the large fluctuations in the potential energy at the extremes of the λ scale, so it is still advisable to employ a scaling function $f(\lambda)$ to help with this.

A disadvantage of this approach is that, at the extremes of the λ scale molecules of both types X and Y retain a presence through the kinetic energy term, when ideally only one of them should be present. This is not a problem for calculations of Helmholtz free energy differences, since we are assuming that the kinetic energy contribution to the free energy difference vanishes. However the retention of both

X and Y molecules at the extremes means that the system pressure is not that of a pure state and consequently calculation of Gibbs free energy differences cannot be done directly.

Chapter 11

Extending Molecular Dynamics Time Scales

11.1 Introduction

The molecular phenomena that can be tackled by the molecular dynamics methods described so far in this book are generally confined to a time scale that is very short in experimental terms. The vast majority of simulations are concerned with time scales of less than 1 nanosecond, which represents a simulation of order one million time steps. While there is a great deal of interesting science that can be explored on this time scale, there are many molecular processes that occur on much longer time scales, for which the insight provided by molecular dynamics would be extremely beneficial. Diffusion in solids is an example, as is the configurational dynamics of proteins and polymers. These processes are underpinned by structural changes, or transitions, in the configurational structure which happen only on very rare occasions and it is practically impossible to capture a sufficient number of these events in normal molecular dynamics to give a statistically meaningful account of them. Methods other than molecular dynamics can be applied to such systems, for example Dissipative Particle Dynamics (DPD) [14] and Lattice Boltzmann [88] and these have their uses and adherents, but for investigations where understanding the *mechanism* of the process in terms of the motion of individual atoms is the objective, some variant of molecular dynamics is essential. This chapter is devoted to the techniques by which molecular dynamics can be extended to meet this objective.

All the techniques described here are based on the idea of structural transition, which is described by Transition State Theory [92]. The molecular system is assumed to be confined to a structure by some form of *energy barrier* that is high in comparison with the system thermal energy. As a result, a transition from this structure to another is a statistically rare event requiring an exceptional thermodynamic fluctuation to make it happen. Transition state theory defines an *activation energy*, which is the critical amount of energy required to overcome the barrier that inhibits the transition. When the transition occurs, the system follows a particular trajectory through configuration space known as the *reaction path*. The reaction path identifies the *reaction coordinate* along which the progress of the transition can be quantified. In this view the quest to uncover the mechanisms of long time scale phenomena becomes one of discovering the sequence of rare events in question, determining the reaction paths and quantifying the activation energies.

In order to study such systems by molecular dynamics, our primary need is to *accelerate* the occurrence of the key rare events, but in a way that does not modify their natural order or lose contact with the true time scale on which these events naturally occur. We also need suitable tools for identifying structural changes when they happen and as accurately as possible quantify them so that their statistical

significance can be established. The methods described here are designed to meet these requirements.

However, before we proceed with our description, we must outline a very important computational tool for elucidating the details of the structural transitions in molecular systems: the Nudged Elastic Band method [90].

11.2 The Nudged Elastic Band Method

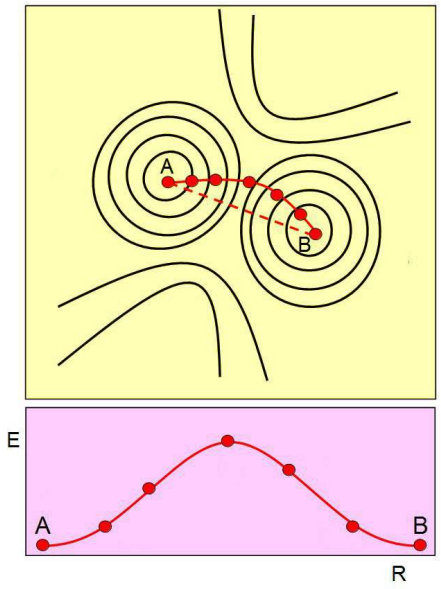

Figure 11.1: The Nudged Elastic Band Method.
(Top: interpolated reaction path [dotted] and the optimised path.
Bottom: Energy of intermediate structures along the reaction coordinate.)

The Nudged Elastic Band method [90] is a method for determining the reaction path for a transition and its associated activation energy E^*, which we may define as the energy difference

$$E^* = \Phi(\vec{R}_N^*) - \Phi_0, \tag{11.1}$$

where Φ_0 is the minimum configuration energy of the initial structure (defined at zero Kelvin), $\Phi(\vec{R}_N^*)$ is the configuration energy at the the top of the free energy barrier and $\vec{R}_N^*$ is its location in the configuration space $\vec{R}^N$. The position $\vec{R}_N^*$ marks an energy *saddle point* in configuration space, though along the reaction coordinate it represents a maximum.

In practice the identification of a transition from a structure A to another structure B is easily accomplished by comparing structure A with structure B, both at zero Kelvin. Both of these structures can be regarded as an energy minimum in configuration space. (In this context a structure is defined as the set of $3N$ coordinates, $(\vec{R}^N)$, locating all N atoms in the system.) The comparison between two structures must pay due regard to any periodic boundary conditions that may be

operating. The zero Kelvin structures are generally obtained by energy minimisation. However these merely mark the start and end points of a transition (see figure 11.1). The NEB method is designed to reconstruct the reaction path and determine the activation energy using the start and end structures as input.

We describe here the NEB method of Henkelman and Jonsson [90] which is based on the following scheme:

1. The start and end points of the NEB construction are the energy minimised (zero Kelvin) structures A and B.

2. A series of structures is constructed by linear interpolation between the structures A and B (see figure 11.1 top) i.e. a series of configurations $\vec{R}_n^N$ with $n=0,\ldots,N_{NEB}$ is generated such that $n=0$ indicates structure A, $n=N_{NEB}$ indicates structure B and

$$\vec{R}_n^N=\vec{R}_0^N+\frac{n}{N_{NEB}}(\vec{R}_{N_{NEB}}^N-\vec{R}_0^N) \tag{11.2}$$

   Each structure has a configuration energy which may be written as $\Phi_c(\vec{R}_n^N)$. Differences of the kind $(\vec{R}_{N_{NEB}}^N-\vec{R}_0^N)$ are to be interpreted as the *minimum image* distances.

3. Each structure in the NEB chain is then connected to its two nearest neighbours by a harmonic spring (except for the end beads which have only one neighbour each). In effect the structures make a chain strung from structure A to structure B, (see figure 11.1 top). The spring energy of the whole chain is then defined as

$$\Phi_s\left(\left\{\vec{R}_n^N\right\}\right)=\frac{1}{2}k_{NEB}\sum_{n=1}^{N_{NEB}}(\vec{R}_n^N-\vec{R}_{n-1}^N)^2, \tag{11.3}$$

   where k_{NEB} is the spring force constant.

4. The objective is now to minimise the energy function $E\left(\left\{\vec{R}_n^N\right\}\right)$ where

$$E\left(\left\{\vec{R}_n^N\right\}\right)=\Phi_s\left(\left\{\vec{R}_n^N\right\}\right)+\sum_{n=1}^{N_{NEB}-1}\Phi_c(\vec{R}_n^N), \tag{11.4}$$

   in which the adjustable variables are the configurations $\vec{R}_n^N : n=1,\ldots,N_{NEB}-1$ i.e. the atomic coordinates in each structure, with the chain end structures at $\vec{R}_0^N$ and $\vec{R}_{N_{NEB}}^N$ fixed.

5. Convergence of the minimisation yields an optimised series of configurations which individually characterise a stage on the reaction path from structure A to structure B. We may define the reaction coordinate R_i as the accumulated coordinate distance in configuration space, where

$$R_i = \sum_{n=1}^{i} |\vec{R}_n^N - \vec{R}_{n-1}^N|. \quad (11.5)$$

Each R_i is a point along the reaction coordinate marking the structures $\vec{R}_i^N$ which have configuration energies $E_i = E(\vec{R}_i^N)$. A plot of R_i *versus* E_i has at least one maximum (E_{max}) which may be located by interpolation (see figure 11.1, bottom). The activation energy then corresponds to

$$E^* = E_{max} - E(\vec{R}_0^N). \quad (11.6)$$

The principles behind the NEB method are simple. The unconstrained structures $\vec{R}_n^N$ would normally relax into the nearest local minimum, which for some would be structure A and for the others, structure B. But this cannot happen if they are sufficiently constrained by the harmonic springs (i.e. if k_{NEB} is large enough). Thus the minimisation of the chain will tend to locate each structure in a position along a path between structures A and B like a stretched necklace, which approximates the minimum energy path between the start and end structures.

In practice this simple idea needs refining (or "nudging"). Thus care is taken to ensure that the spring forces acting on the structures and the forces optimising structural configurations are approximately orthogonal. This means that the atomic forces are zeroed in directions parallel to the path of the chain and the spring forces are zeroed in directions normal to the chain. This is what the method of Henkelman and Jonsson [90] is designed to do. If the NEB optimisation works correctly, the result will be that beads are evenly spaced along the minimum energy path (as in figure 11.1 top). The rôle NEB plays in extended time scale molecular dynamics is in quantifying structural transitions, as will become apparent in later sections.

In the next section we start describing some of the methods available for extending time scales.

11.3 Parallel Replica Methods

The idea of parallel replication was introduced in chapter 9, where it was described as a means to improve statistical sampling of the system of interest. In its basic form it does nothing to access long time scales, but by injecting some ideas from statistical mechanics and reaction kinetics, the approach can yield some useful procedures. We shall describe two: Parallel Replica Dynamics and Parallel Tempering.

11.3.1 Parallel Replica Dynamics

Parallel Replica Dynamics was devised by Voter [82] and is most suited to parallel computer systems with large numbers of processing nodes. High numbers of processors are best because the accessible time scale provided by the method is directly proportional to the number of processors used. The basic idea is a simple one: if we simulate M replicas of a system, all representing the same thermodynamic state but commencing from different initial velocities and therefore following different

trajectories, then over a chosen interval of time the probability of observing a rare event during a simulation is directly proportional to the number of replica systems.

According to first order kinetics theory, the probability of a transforming system being found in its original structure in a time interval dt at a time t is given by

$$P(t)\,dt = k\exp(-kt)\,dt, \tag{11.7}$$

where k is the *first order rate constant*. In the collective system of M replicas, the rate constant is effectively $M\,k,$ so this equation becomes

$$P_M(t)\,dt = M\,k\exp(-M\,k\,t)\,dt. \tag{11.8}$$

Defining a new time scale, $t_{tot} = M\,t,$ means we can write

$$P_M(t)\,dt = k\exp(-k\,t_{tot})\,dt_{tot} = P(t_{tot})\,dt_{tot}, \tag{11.9}$$

which shows that the kinetics of the $M-fold$ replicated system is equivalent to that of a single system over a time scale that is M times longer. Based on this relation, Voter outlined the following procedure:

1. Replicate the system on all M processors. Energy minimise the original structure to provide a reference structure.
2. Equilibrate the system at the required temperature, starting from different random velocities on each processor. At the end of equilibration, minimise the structures and compare with the reference to make sure all replicas are still in the same structure. Set the start time $t_{tot} = 0.$
3. Run the simulation on each processor for a time interval $\Delta t_{block},$ then minimise the structures and check against the reference, looking for a structural transition. The accumulated time $t_{tot} = t_{tot} + M\,\Delta t_{block},$ is recorded.
4. If no structural transition is found, repeat step 3.
5. If a structural transition is found on (say) processor $p_j,$ continue the simulation for an additional period of time $\Delta t_{corr},$ the *decorrelation* period. The accumulated time $t_{tot} = t_{tot} + \Delta t_{corr}$ is recorded.
6. Now take the configuration from processor p_j as a new structure and proceed again from step 1.

This procedure is presented pictorially in figure 11.2.

In this scheme Δt_{block} is an interval chosen by the user as a compromise between cost of performing the energy minimisations that occur once in each interval and the accuracy with which the actual time of transition can be determined. If the transition time is given as half way through the interval Δt_{block} the maximum error is evidently $\pm\Delta t_{block}/2.$ (This can be found more accurately by back-tracking through past configurations, but this is slow and expensive in memory usage.) The decorrelation time Δt_{corr} is an interval during which any apparent transition may revert back to the original structure. Inclusion of this stage is a recognition that not all barrier-crossing events give rise to a permanent change in structure. Those that do

not cannot be considered as valid transitions and the simulation must revert back to the end of the last Δt_{block} interval and continued. The choice of Δt_{corr} depends on independent knowledge of the time it takes for the transition to revert.

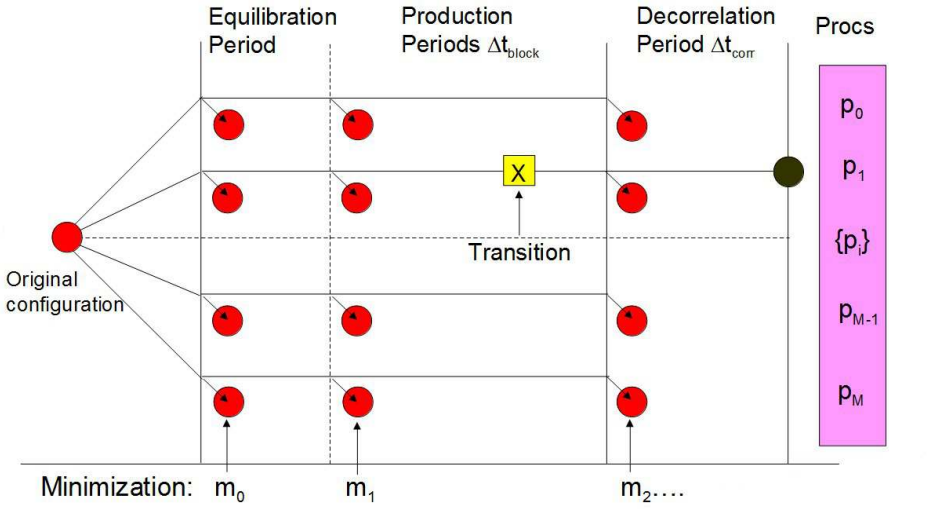

Figure 11.2: Parallel Replica Dynamics

The purpose of this procedure is to find the sequence of transitions that underlie the long time scale evolution of the system at a rate that is accelerated compared with normal molecular dynamics. From the information gained, the evolution of the system and the true time scale on which the change takes place are also determined. Clearly, access to a 1000 processor computer, means the history can be determined at 1000 times the rate at which the simulation of a single system could provide. The procedure is well suited for studies of diffusion in solids (provided they are on an accessible time scale). It can also be used, for example, to study conformational changes in polymeric molecules in solution, though it is more difficult to identify structural transitions unambiguously. In crystalline solids transitions can be identified by checking the squared differences in atomic positions for all atoms between the current configuration (energy minimised) and the reference configuration – allowing for periodic boundaries. In molecular systems the squared differences between inter-atomic distances for the same molecule in the two configurations can be used.

11.3.2 Parallel Tempering

Parallel Tempering was devised by Swendsen and Wang [81] for Monte Carlo simulations, but it is equally appropriate for molecular dynamics. This is not actually a method for studying the evolution of long time scale phenomena, but it does address an important issue in simulation: how to properly sample the configurations in a system where relaxation times are long. For example, some systems like amorphous polymers or silicate glasses often spend long intervals trapped in a small part of the configuration space that is thermodynamically accessible to them, thus skewing the calculation of average properties. Parallel tempering helps to overcome this difficulty and obtain a more representative average. The method can also be used to obtain an initial configuration for a system that is representative of the thermodynamic state required. There is a resemblance to parallel replica dynamics in this scheme, as the following procedure shows

1. Start M simulations $(n=0,\ldots,M-1)$ of the system, (one per processor,) at

different temperatures $T = n\Delta T + T_0$. Equilibrate the simulations for N_{equil} time steps.

2. After equilibration, at intervals of N_{sample} time steps, attempt a Monte Carlo controlled swap of the configurations from two adjacent processors, randomly chosen.
3. Continue the simulation until the distribution of configuration energies in the system at the lowest temperature, T_0, is a Boltzmann distribution.
4. Use the configurations of the lowest temperature system to calculate the average properties.
5. Save all replica configurations for a possible restart to improve sampling.

This procedure is presented pictorially in figure 11.3.

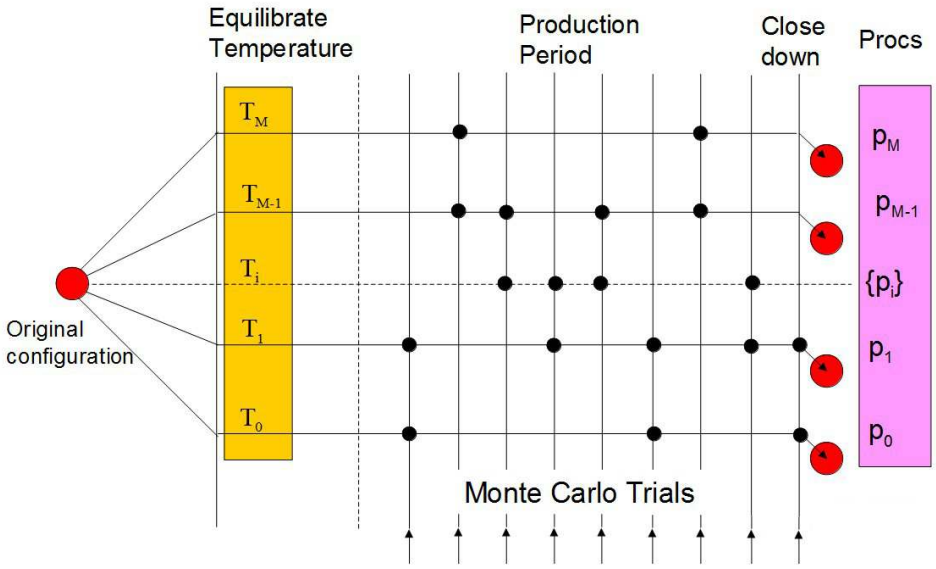

Figure 11.3: Parallel Tempering

The principles behind this procedure are clear. Low temperature systems have difficulty in sampling the configuration space, but this diminishes as temperature increases. The procedure allows the lower temperature simulations to sample from the high temperature ones and thereby access configurations denied to it by slow kinetics. The Monte Carlo selection procedure ensures that these configurations are introduced with a proper Boltzmann probability. Since a large temperature difference between two simulations makes a successful swap unlikely, having a range of temperatures with a small temperature difference between each pair, makes swaps between adjacent pairs more likely. Gradually, as the simulation proceeds, the lower temperature systems will gain access to configurations generated at high temperature and can select them with the correct Boltzmann weighting.

The Monte Carlo selection criterion for swapping configurations is as follows:

- For processors p_i and p_j with temperatures T_i and T_j and configurations energies E_i and E_j the configurations are swapped if a random number R, chosen on interval $(0,1)$ satisfies the condition

$$R \leq \left(\frac{(E_j - E_i)}{k_B} \left(\frac{1}{T_j} - \frac{1}{T_i} \right) \right). \qquad (11.10)$$

It is advisable to monitor the simulation as it proceeds to ensure the rate of configuration swaps is reasonable. Too low a swap rate implies that the temperature difference between systems is too large and a larger number of processors is required, with a smaller temperature difference between them.

Deciding that the lowest temperature simulation is properly Boltzmann can be accomplished by generating a histogram of the configuration energies and seeing if this has the exponential form expected. If at the end of the simulation the results are not satisfactory, the simulation can be continued for a longer period. It is important to note that this check is only valid if there is evidence for an efficient rate of configuration swaps over the full range of temperatures of the replica systems. For this purpose it is valuable to keep a record of all configuration swaps made to ensure that configurations that were initially in high temperature systems really do make their way down to the lowest temperature system.

11.4 Bias Potential Dynamics

Bias Potential Dynamics (BPD), which is due to Voter [91], was designed to study diffusion in solids. It is an example of a class of methods known as *hyperdynamics.* These are methods that modify the dynamics of the system to accelerate its time evolution while maintaining the correct sequence of events from the natural time scale. The accelerated time scale can be related to the natural time scale by a determinable factor. As its name suggests, BPD achieves this by biasing the natural *background* potential of the system. Thus the working potential for the dynamical system becomes

$$\Phi_{mod}(\vec{R}^N) = \Phi(\vec{R}^N) + \Phi_b(\vec{R}^N), \qquad (11.11)$$

where $\Phi(\vec{R}^N)$ is the background potential $\Phi_b(\vec{R}^N)$ is the bias potential and $\Phi_{mod}(\vec{R}^N)$ is the modified, working potential.

The effect of a bias potential is shown in figure 11.4 in which the normal background potential is represented by a double peaked function with a minimum between. In the natural system this potential drives the dynamics. In terms of the Transition State Theory [92] this plot represents the configurational energy variation along the reaction coordinate, which is the path in the many dimensions of configuration space along which a configurational change takes place. The peaks (which are of course *saddle* points in the many dimensional configuration space) represent energy barriers and minima represent particular configurations or states of the system. In order to explore configuration space adequately (i.e. with the appropriate Boltzmann weighting), it is necessary for the system to acquire the *activation energy* to overcome such barriers, which usually happens through natural energy fluctuations. However, if the barrier between structures is high, the probability of a large enough

fluctuation occurring is exceedingly small. Voter's idea was that a suitable bias potential added to the background potential would reduce the activation energy required and make a transition between structures far more probable. In figure 11.4 it is clear that adding a suitable bias potential can drastically reduce the depth of the minimum between saddle points and make it less likely that the system remains trapped in one structure.

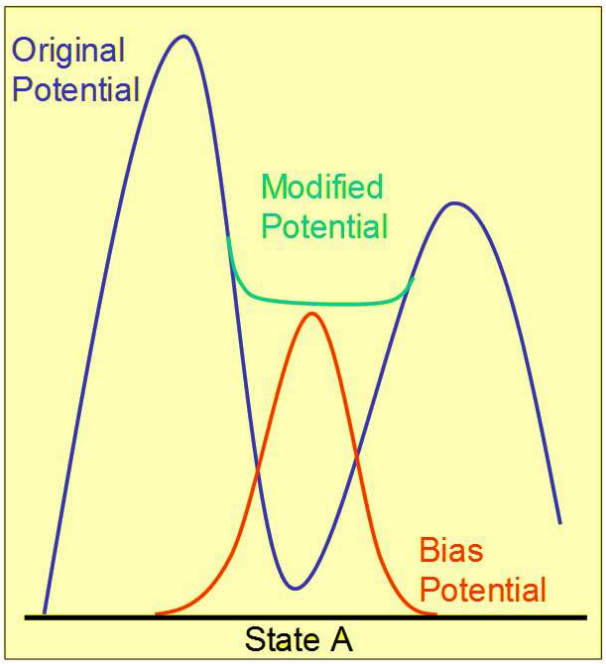

Figure 11.4: The Bias Potential Method

The amount by which the bias potential boosts the rate of escape from the minimum can be determined from statistical mechanics. In chapter 3, section 3.19.4 , it was shown how an average property of a system, $\langle A \rangle$, could be extracted from a similar system augmented by the addition of a bias potential $\Phi_b(\vec{R}^N)$:

$$\langle A \rangle = \frac{\left\langle A(\vec{R}^N) \exp(\beta \Phi_b(\vec{R}^N)) \right\rangle_b}{\langle \exp(\beta \Phi_b(\vec{R}^N)) \rangle_b}, \tag{11.12}$$

where the subscript b on the angular brackets signifies an average calculated in the biased system. The property that is of concern here is the Transition State Theory rate constant k_{TST}, which is the rate at which a free energy barrier between structures is crossed. In the form appropriate to first order kinetics it is defined by the relation

$$k_{TST} = \left\langle |V^N| \delta(\vec{R}^*_N) \right\rangle, \tag{11.13}$$

where $|V^N|$ is the configurational velocity in the positive direction of the reaction coordinate and $\delta(\vec{R}^*_N)$ is the probability density for the configuration occurring at the top of the free energy barrier between the two structures concerned (i.e. at the saddle point). This point is located at $\vec{R}^N = \vec{R}^*_N$. According to (11.12), in the presence of a

bias potential, (11.13) becomes

$$k_{TST}=\frac{\left\langle |V^N|\delta(\vec{R}_N^*)\exp\left(\beta\Phi_b(\vec{R}^N)\right)\right\rangle_b}{\left\langle \exp\left(\beta\Phi_b(\vec{R}^N)\right)\right\rangle_b}. \tag{11.14}$$

The key requirement of the bias potential [91] is that it must be *zero* at the location of the maximum in the energy barrier. This means we can write

$$k_{TST}=\frac{\left\langle |V^N|\delta(\vec{R}_N^*)\right\rangle_b}{\left\langle \exp\left(\beta\Phi_b(\vec{R}^N)\right)\right\rangle_b}=\frac{k_{TST}^b}{\left\langle \exp\left(\beta\Phi_b(\vec{R}^N)\right)\right\rangle_b}, \tag{11.15}$$

where k_{TST}^b is the modified rate constant calculated in the biased system. An important property follows from this, if a given structure A has adjacent structures B and C with different energy barriers between then, for a given bias potential, the following relationship between the rate constants is easily seen to be true:

$$\frac{k_{A\to B}}{k_{A\to C}}=\frac{k_{A\to B}^b}{k_{A\to C}^b}. \tag{11.16}$$

i.e. the ratio of rates of transition from A to B and from A to C are the same in the original and the biased systems. Such ratios hold for all possible escapes from a given structure. This means that the sequence of transitions between structures in the course of a simulation is the same in the biased and unbiased systems. Thus we can thus directly relate the system kinetics in the biased system to that in the original system and exploit a much faster system evolution. Voter refers to the term $\left\langle \exp\left(\beta\Phi_b(\vec{R}^N)\right)\right\rangle_b$ in (11.15) as the *boost factor*, since it defines the "speed up" of the original system kinetics. It can often be as large as 10^5 or 10^6.

The critical issue in BPD is the choice of the bias potential. It is not immediately obvious how one can be constructed at the precise location needed and with the important requirement that it be zero at the saddle points. Fortunately a useful approach to this problem has been devised by Hamelberg et al. [93] who proposed a bias potential of the following form

$$\begin{aligned}&\Phi_b(\vec{R}^N)=\frac{\left(E-\Phi(\vec{R}^N)\right)^2}{\left(\lambda+E-\Phi(\vec{R}^N)\right)}, \quad if \quad E\geq\Phi(\vec{R}^N),\\ &\Phi_b(\vec{R}^N)=0 \qquad\qquad\qquad otherwise.\end{aligned} \tag{11.17}$$

in which $\Phi(\vec{R}^N)$ is the background potential for the system. The parameters E and λ are constants to be specified. E defines a *working energy level* and for a given structure of the system has the requirement that

$$\Phi(\vec{R}_N^*) \;>\; E \;>\; \Phi(\vec{R}_{min}^N), \tag{11.18}$$

where $\Phi(\vec{R}_N^*)$ is the energy of the lowest saddle point for escape from the structure and $\Phi(\vec{R}_{min}^N)$ is the lowest potential energy for the structure. Note that whenever

$\Phi(\vec{R}^N) = E$ or higher, the bias potential is zero, which means the saddle point is unaffected by the bias as required. Also, provided the constant λ is assigned a positive value the bias potential is always positive within the bounds of the structure concerned, so accelerated dynamics are guaranteed.

The parameter λ must have a positive value, but what is its rôle? Setting $\lambda=0$ means $\Phi_b(\vec{R}^N)=E-\Phi(\vec{R}^N)$ which, on insertion into (11.11), gives $\Phi_{mod}(\vec{R}^N)=E.$ In other words, the modified potential is flat within the bounds of the structure, and where this joins with the background potential near the saddle point, there is a discontinuity in the slope of the potential. Such discontinuities are not good for dynamical simulation, since it can give rise to poor energy conservation. A non zero value of λ helps to smooth out the discontinuity. This is seen by inserting (11.17) into (11.11), differentiating with respect to $\vec{R}^N$ within the domain confined by the condition $\Phi(\vec{R}^N)\leq E$ and rearranging the result to give

$$\frac{d\,\Phi_{mod}(\vec{R}^N)}{d\,\vec{R}^N}=\frac{d\,\Phi(\vec{R}^N)}{d\,\vec{R}^N}\left(\frac{\lambda}{(\lambda+E-\Phi(\vec{R}^N))}\right)^2. \tag{11.19}$$

Provided $\lambda>0$ equation (11.19) yields the following result when $\Phi(\vec{R}^N)=E$:

$$\frac{d\,\Phi_{mod}(\vec{R}^N)}{d\,\vec{R}^N}=\frac{d\,\Phi(\vec{R}^N)}{d\,\vec{R}^N}. \tag{11.20}$$

which shows that the gradient of the modified potential is the same as that of the background potential at the domain boundary. The modified potential is therefore continuous when $\lambda>0.$ However when $\lambda=0$ the term in large brackets right of (11.19) is always zero within the domain boundary and there has to be a discontinuity at the boundary. Nevertheless a small value of λ still means that the change in force at the domain boundary can be large. This diminishes with larger values of $\lambda.$ However it is also the case that as λ increases, the modified potential within the domain boundary becomes more deeply "bowl shaped" which has a tendency to push the modified potential minimum back towards the background minimum, which works against the acceleration of the dynamics. Clearly a compromise is required here and Hamelberg et al. [93] have proposed a scheme for obtaining this.

It is apparent from(11.19) that, for a given structure, the minimum of the background potential $\Phi(\vec{R}^N)$ is in the same location, $\vec{R}^N_{min},$ as for the modified potential $\Phi_{mod}(\vec{R}^N).$ We thus define the parameters

$$\Phi_0=\Phi(\vec{R}^N_{min}) \quad \text{and} \quad \Phi_0^{mod}=\Phi_{mod}(\vec{R}^N_{min}). \tag{11.21}$$

So from equations (11.11) and (11.17), within the domain confined by $\Phi(\vec{R}^N)\leq E,$ we can write

$$\Phi_0^{mod}=\Phi_0+\frac{(E-\Phi_0)^2}{(\lambda+E-\Phi_0)}. \tag{11.22}$$

This can be rearranged to give

$$\lambda = \frac{(E-\Phi_0)(E-\Phi_0^{mod})}{(\Phi_0^{mod}-\Phi_0)}. \tag{11.23}$$

With foreknowledge of E and Φ_0 this formula allows us to set λ so that Φ_0^{mod} is at some predefined level between E and Φ_0. The energy Φ_0 is obtained by energy minimising the system structure, which is also required to define a reference structure for each structure. The energy E is more difficult to choose beforehand. For this reason it can be helpful to define it in terms of an associated temperature T_E as follows

$$E = \frac{3}{2} N k_B T_E + \Phi_0. \tag{11.24}$$

We can also define a temperature T_M associated with Φ_0^{mod}:

$$\Phi_0^{mod} = \frac{3}{2} N k_B T_M + \Phi_0, \tag{11.25}$$

where $T_E > T_M$. In terms of which (11.23) becomes

$$\lambda = \frac{3 N k_B T_E (T_E - T_M)}{2 T_M}. \tag{11.26}$$

The advantage of these conversions is that temperature is arguably more intuitive as a control variable than configuration energy. Furthermore the chosen temperatures are likely to be useful in more than one structure of the system.

The forces of the modified potential are of course different from the background potential, but this is easily accounted for in the case of the bias potential (11.17). The result is easily shown to be

$$\begin{aligned} \vec{f}_j^{mod} &= \vec{f}_j \left(\frac{\lambda}{\lambda + E - \Phi(\vec{R}^N)} \right)^2, \quad if \quad E > \Phi(\vec{R}^N), \\ \vec{f}_j^{mod} &= \vec{f}_j \qquad\qquad\qquad\qquad otherwise. \end{aligned} \quad 0 \tag{11.27}$$

where $\vec{f}_j$ is the force on the $j'th$ atom in the absence of the bias potential.

BPD is best exploited when it is used to provide full path kinetics, which means a thorough exploration of the possible transitions that can occur in a system and a systematic gathering of associated data (activation energies, reaction coordinates and the meta-stable structures comprising the structures of the system *etc*.) to permit reconstruction of the long time scale processes. The method is as follows.

1. Construct a reference structure for the current structure by energy minimisation. The simulation then proceeds with the bias potential (11.17) in the same manner as a normal simulation and during which a running estimate

of the boost factor $\langle \exp(\beta \Phi_b(\vec{R}^N)) \rangle_b$ is calculated.

2. After a defined interval Δt_{block} the simulation is halted and the structure energy minimised to compare with the original reference structure to determine if a transition has occurred. A transition is deemed to have occurred if one or more atoms are displaced by more than a pre-set distance R_{catch}. If no transition is detected, the simulation continues for another interval Δt_{block} and tested again. When a transition is eventually detected, a NEB calculation should ideally be performed to find the activation energy (E^*) and ensure the chosen bias parameter E does not exceed this.

3. A determination of the occurrence time of the transition t_{occ} is made. This can be done by checking back from the detection of the transition through past configurations saved at regular intervals during the period t_{block}. Each saved configuration must be energy minimised and compared with the reference structure structure until the first occurrence of the new structure is found. This provides a reasonable accuracy on the transition time, somewhat better than $\pm t_{block}/2$. The transition time is then corrected for *true* time using the current boost factor.

4. The new found structure becomes the reference structure for the next stage of the simulation, which is a repeat of stage 2 using the new structure.

5. The simulation is continued until, from inspection, it is apparent that all significant transitions have been observed. This determination is relatively straightforward in simple, ordered systems, but is much more difficult in more complicated systems. Clearly some knowledge of the system, gained from other sources, is invaluable here.

6. The end result of the simulation is a detailed knowledge of all transitions that have taken place with associated times at which they occurred corrected for the effect of the bias potential. With all the information gathered it should be possible to determine the full kinetics for the process of interest.

Another way of using BPD is for *configurational sampling,* which is simply a means for exploring configurational space, and has a number of possible applications. For example, it can be used to help attain equilibration at a given temperature more quickly so that thermodynamic averages can be obtained with greater reliability (as with parallel tempering, above). It can be used to observe configurations which are difficult to obtain under normal conditions, perhaps because they are far from the starting structure and the system has a slow relaxation time. Such configurations may be important from a mechanistic viewpoint. Since the trajectory of the system evolves faster, it means that the simulation can show the motions of the system on a reasonable time scale, which is useful for demonstration purposes.

11.5 Temperature Accelerated Dynamics

Temperature Accelerated Dynamics (TAD) was devised by Sorensen and Voter [94]. This is based on the simple principle that change happens faster in systems at higher temperature. However, simply increasing the temperature does not guarantee that the time evolution of a system follows the same mechanistic path or the same kinetics as in the lower temperature system. Nevertheless, the authors showed that by appropriate intervention in a high temperature simulation it is possible to recover the low temperature kinetics with a high degree of reliability. As with bias potential dynamics, the method was primarily devised for investigating diffusion in solids, in which rare displacements of atoms (transitions) underlie the diffusion process. These processes are described by first order kinetics.

For a system in a structure A at time zero, from which it may escape to structure B in due course, the probability of it being found in the same structure at a later time t is given by

$$P(t)dt = k\ \exp(-k\,t)\ dt, \tag{11.28}$$

where $P(t)$ is a probability distribution and k is the first order rate constant. It follows from this that the mean lifetime, τ, of the system in structure A is

$$\tau = 1/k, \tag{11.29}$$

from which we have the property, universal for first order systems, that

$$\tau\ k = 1. \tag{11.30}$$

i.e. the rate constant is inversely proportional to the lifetime of the initial structure.

According to transition state theory the rate constant exhibits a temperature dependence given by the Arrhenius law

$$k = \nu\ \exp(-\beta\, E^*), \tag{11.31}$$

where ν is the pre-exponential factor (in units of frequency) and $\beta = 1/k_B T$, as usual. E^* is the *activation energy* of the process, which is the energy barrier between the bottom of the potential basin of structure A and the saddle point on the configuration energy surface that provides the escape route to structure B. This equation shows that, at different temperatures, T_1 and T_2, the same escape route from structure A has different rate constants, k_1 and k_2 respectively. Nevertheless, the universal property of equation (11.30) means that

$$\tau_1\ k_1 = \tau_2\ k_2, \tag{11.32}$$

which is an important relation underpinning the TAD method, showing how the time scale for a barrier crossing event at one temperature is related to the time scale for the same event at another temperature.

In most practical systems structure A is likely to have more than one escape route (to distinct structures: B, C, D, etc.) each with its own activation energy, pre-exponential factor and temperature-dependent rate constant. At any given temperature, escape from structure A may occur via any one of these routes, but is most probable via the route that has the highest rate constant and therefore, by equation (11.29), the lowest associated residence time. A normal molecular dynamics simulation commencing from structure A will undergo a transition to a neighbouring structure via the first encountered route and never sample the alternatives. Since the different routes have different temperature dependent rates, it follows that at different temperatures, the system may evolve along completely different paths. The TAD method avoids this possibility at high temperature by returning the system to structure A after every transition, so that all the important escape routes at this temperature may be discovered. From the calculated properties of these escape routes the true low temperature escape route may be determined by extrapolation. Thus TAD provides a high temperature method for identifying the transitions that mark out the low temperature diffusion pathway.

An outline of the TAD method is as follows, where it is assumed that the kinetic properties of a system at the temperature T_{low} are required.

1. The starting structure (A) is energy minimised to provide a reference structure against which later structures may be compared to determine any structural transitions.

2. The system is simulated at high temperature (T_{high}) and halted at regular intervals Δt_{block} (the *TAD block*) to energy minimise the structure for comparison with the existing reference structure and determine if a structural transition (to structure B) has occurred. A transition is deemed to have occurred if one or more atoms are displaced by more than a pre-set distance, R_{catch}. If a transition is detected, a NEB calculation is initiated, using the current and reference structures, to find the activation energy E^*. If no transition is found the simulation continues for another interval Δt_{block} and tested again – and so on until a transition is found.

3. Next a determination of the transition time (t_{occ}^{high}) is made. As with BPD t_{occ}^{high} is determined by checking back from the detection of the transition through past configurations saved at regular intervals (much less than a TAD block). Each saved configuration is energy minimised and compared with the reference structure until the first occurrence of the new structure is found. This provides a reasonable accuracy on the transition time, somewhat better than using the middle time of the TAD block in which the transition occurred.

4. The time t_{occ}^{high} is extrapolated to the corresponding time of occurrence (t_{occ}^{low}) at temperature T_{low}. This is done by combining equations (11.31) and (11.32) and taking the logarithm:

$$\log\left\{\frac{t_{occ}^{low}}{t_{occ}^{high}}\right\}=\log\left\{\frac{k^{high}}{k^{low}}\right\}=-\frac{E^*}{k_B}\left\{\frac{1}{T_{high}}-\frac{1}{T_{low}}\right\}. \qquad (11.33)$$

This extrapolation is shown in figure 11.5.

5. The system is then returned to structure A and the simulation recommenced. Returning the system to its original structure means resetting the atomic coordinates to those taken just before the energy minimisation at the end of the last TAD block and resetting the velocities according to a Boltzmann distribution at temperature T_{high}, while retaining the total system energy of the original structure. The simulation is continued to obtain information on other transitions (to structures C, D, E *etc*) that may occur from structure A. (Note the difference here from what happens in the BPD method.)

6. A determination of the simulation "stopping time" (t_{stop}) is made, based on current knowledge of the system (see below). When the simulation reaches the calculated stopping time, it is terminated.

7. When the simulation has ended, the transition with the shortest determined occurrence time (t_{occ}^{low}) at T_{low} indicates the structure to which the system would have transformed in a molecular dynamics simulation at that temperature. This new structure becomes the starting point for a new high temperature simulation of the system, exploring transitions from this structure to further, new structures. By this procedure, after sufficient sampling of structures, the true low temperature evolution of the system may be determined.

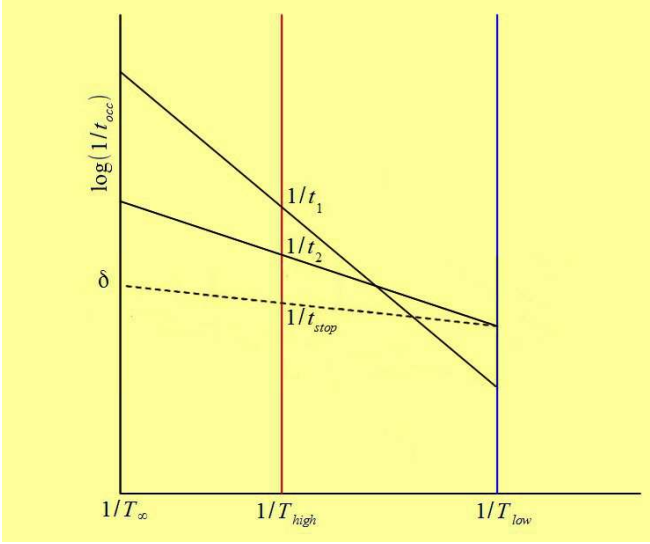

Figure 11.5: Theory of Temperature Accelerated Dynamics

The theory of TAD is presented in figure 11.5, in the form of a plot of $\log(1/t)$ *versus* $1/T$. Simulations at high temperature (T_{high}) locate different transitions

indicated as t_1 and t_2 (for example), with t_1 occurring first (time increases in a downward direction on this plot). Extrapolation to low temperature using equation (11.33) shows that these transitions would have occurred in reverse order at the lower temperature (T_{low}). The gradient of each line is determined by the respective activation energy, E^*. If no other transitions occur, the extrapolated t_2 time represents the earliest transition time in a low temperature molecular dynamics simulation and thus represents the first step in the series of diffusion events at T_{low}.

The dotted line in figure 11.5 indicates a possible hypothetical transition that just precedes t_2 when extrapolated to low temperature. Its high temperature intercept $(1/t_{stop})$ gives the estimated stopping time for the simulation. Ideally the pping time should be defined with a high probability that no more significant transitions will be found. This is determined from the history of the TAD simulation itself. Sorensen and Voter provided a prescription of this [94]. It begins by defining, for a supposed undiscovered escape route, a very small probability (δ) that after the time t_{stop} the system is still in structure A. This probability must chosen small enough to give confidence that the anticipated transition has had sufficient time to occur. δ may be defined as

$$\delta = \int_{t_{stop}}^{\infty} k \, \exp(-kt) \, dt \tag{11.34}$$

from which it follows that

$$\log\left(\frac{1}{\delta}\right) = k \; t_{stop}, \tag{11.35}$$

and hence combining this with (11.31):

$$\log\left(\frac{1}{\delta}\right) = t_{stop} \nu_{min} \exp\left(\frac{-E^*_{min}}{k_B \; T_{high}}\right), \tag{11.36}$$

where ν_{min} and E^*_{min} are the pre-factor and activation energy respectively of the supposed undiscovered escape route. Rearranging this gives

$$T_{high} \log\left(\frac{\log(1/\delta)}{t_{stop} \nu_{min}}\right) = -\frac{E^*_{min}}{k_B}. \tag{11.37}$$

Any undiscovered escape route is one which may possesses a low temperature occurrence time that is less than the current working minimum (t^{min}_{occ}). The right side of (11.37) may be approximately determined using equation (11.33) if it assumed that the largest observed value of t^{high}_{occ} is close to t_{stop} and the lowest possible low temperature time is close to t^{min}_{occ} (see figure 11.5). Combining the two equations and rearranging gives

$$t_{stop} = \left(\frac{\log(1/\delta)}{\nu_{min}} \right) \left(\frac{t_{occ}^{min} \nu_{min}}{\log(1/\delta)} \right)^{T_{low}/T_{high}} . \quad (11.38)$$

Sorensen and Voter [94] argue that ν_{min} is commonly of the order $10^{12} \approx 10^{13}\ s^{-1}$ and suggests $\delta = 0.001$ as a working value.

There are some things to be aware of when running a TAD simulation. Firstly the radius R_{catch} needs to be chosen carefully, where possible basing it on nearest neighbour distances in the parent crystal. A consequence of using too large a catch radius is that transitions that require a short "hop" in atom positions may be missed. Such misses make it difficult to reconstruct the reaction path and, in particular, cause the NEB calculation to crash, since there will be no simple path between the reference structures. Secondly, sometimes successive transitions into the same structure are observed. Such repeated transitions are normal but if they occur in succession it implies that there is some correlation creeping into the resetting of the system back into the starting structure. This however is harmless as the accumulated simulation time is reset back to the restart structure after each transition and so does not affect the time of the later transition to a new structure. Thirdly, note that in a TAD simulation the reference structure is always the same; it does not "follow" the diffusion path as it does in BPD.

11.6 Metadynamics

Metadynamics was devised by Laio and Parrinello [95] and like the BPD method described above achieves accelerated dynamics by using a bias potential. However the methodology and objectives are somewhat different. It is used mostly in the study of phase transitions, from solid to solid and solid to liquid, and has the facility to calculate free energy differences between the phases.

In metadynamics the bias potential that augments the Hamiltonian governing the system dynamics is *time dependent* and is defined in terms of appropriate order parameters which characterise the structure of the system, rather than the coordinates of the atomic configuration. The Hamiltonian is thus

$$H\left(\vec{R}^N, t\right) = \sum_{i=1}^{N} \frac{p_i}{2m_i} + \Phi\left(\vec{R}^N\right) + \Phi_b\left(\vec{S}^M\left(\vec{R}^N\right), t\right), \quad (11.39)$$

in which $\Phi\left(\vec{R}^N\right)$ is the usual potential energy function describing the interactions between, and within, the molecules, $\vec{p}_i$ is the momentum of the $i'th$ atom and m_i its mass. The novel term $\Phi_b\left(\vec{S}(\vec{R}^N), t\right)$ is the time dependent bias potential written as a function of a vector $\vec{S}^M$, which is an ordered set of M order parameters, $S_j\left(\vec{R}^N\right)$, each of which is defined in terms of the instantaneous positions $\left\{\vec{r}_i\right\}$ of the atoms in system. The bias potential is time dependent in the sense that it is "grown" by adding, at periodic intervals of time, τ_G, a Gaussian term of weight w and width δh, as in

$$\Phi_b\left(\vec{S}^M(\vec{R}^N),t\right)=w\sum_{k=1}^{N_G}\exp\left(\frac{-\left|\vec{S}^M(k\,\tau_G)-\vec{S}^M(t)\right|^2}{2\,\delta\,h^2}\right), \tag{11.40}$$

where k runs over all *previously deposited* Gaussians and N_G is the integer part of t/τ_G i.e. $N_G=\lfloor t/\tau_G \rfloor$. The values of τ_G and the Gaussian height w and width δh must be chosen to ensure accuracy for the free energy calculation. Schemes for this have been described by Laio *et al* [96] and Quigley and Rodger [97].

The force, $\vec{f}_i$, on each atom derived from the Hamiltonian (11.39) is formally

$$\vec{f}_i=-\vec{\nabla}_i\Phi(\vec{R}^N)-\sum_{j=1}^{M}\frac{\partial V}{\partial S_j}\vec{\nabla}_i S_j(\vec{R}^N). \tag{11.41}$$

If the deposition rate w/τ_G is slow enough the motion of the order parameters $\vec{S}^M$ is adiabatically separated from the atomic motion of the system. After a sufficiently long simulation, the bias potential cancels out or "fills" the free energy minima of the potential $\Phi(\vec{R}^N)$ and permits an accelerated dynamics. Meanwhile the bias potential becomes a measure of the free energy surface i.e.

$$F_G(\vec{S}^M)=-\lim_{t\to\infty}\ \Phi_b\left(\vec{S}^M(\vec{R}^N),t\right). \tag{11.42}$$

The accuracy of this estimate of the free energy surface is dependent on the deposition rate and the effective diffusion constant of the order parameters. Typically the error is of order w, the Gaussian weight. A discussion of these issues is given by Laio *et al.* [96] and Quigley and Rodger [97].

The importance of using order parameters in the Hamiltonian (11.39) is that they are a direct measure of the structure of a particular phase. Increasing the bias potential therefore has the effect of destabilising phases that are characterised by these parameters, forcing the simulation to move to alternative structures with lower free energy. In general several different order parameters can be used at the same time, to improve the control of the selectivity of the various phases and the pathways between them. However, this must be weighed against the additional computational cost, which grows exponentially with the number of order parameters.

Quigley and Rodger have described a protocol for deciding which order parameters to use [97]. Firstly a set of simulations of the disordered state and any accessible crystalline polymorph are performed and the equilibrium distribution functions for the candidate order parameters obtained. Any sets of parameters for which the distributions overlap are discarded until the sets remaining describe the known states with minimum ambiguity. This ensures that the realisable structures are distinct in the collective space of the order parameters. However this approach does not guarantee that pathways between states will match those that occur in the unbiased system, though it does set an upper bound for the corresponding free energy barrier. An alternative method devised by Peters and Trout [98] offers a better description of

pathways.

Popular order parameters used in metadynamics work are the system potential energy [99], the Steinhardt Q4 and Q6 parameters [100] and the Chau and Hardwick tetrahedral parameter [101]. These order parameters are described below.

The use of potential energy as an order parameter was pioneered by Donadio et al [99]. The configuration energy is a well behaved function that takes on distinct values for different structures. It has the additional advantage that it requires no additional computation time, since it is normally calculated automatically during any molecular dynamics simulation. It is also straightforward to calculate the associated biasing forces and stress tensor contributions:

$$\begin{aligned} \vec{f}_i &\rightarrow \vec{f}_i\left(1+\frac{\partial \Phi_b}{\partial \Phi}\right), \\ \boldsymbol{\sigma} &\rightarrow \boldsymbol{\sigma}\left(1+\frac{\partial \Phi_b}{\partial \Phi}\right). \end{aligned} \tag{11.43}$$

In addition to using potential energy as a *global* parameter it can also be used as a *local* parameter [97] specific to a subset of atoms in the system, such as those that form the central atoms in the definitions of the Steinhardt or tetrahedral order parameters described below. This approach has the advantage that it allows the driving of structural changes in parts of the system that are of greatest interest and not (for example) the solvent or substrate.

The parameters of Steinhardt, Nelson and Ronchetti [100] employ spherical harmonics to describe the local order of atoms of type β surrounding an atom of type α, as in

$$Q_l^{\alpha\beta}=\left(\frac{4\pi}{2l+1}\sum_{m=-l}^{l}\left|\frac{1}{N_c N_\alpha}\bar{Q}_{lm}^{\alpha\beta}\right|^2\right)^{1/2}, \tag{11.44}$$

in which N_α and N_c are the numbers of atoms of type α and type β that are near neighbours of a central atom of type α and

$$\bar{Q}_{lm}^{\alpha\beta}=\sum_{b=1}^{N_b} f_c(r_b)Y_{lm}(\theta_b,\phi_b). \tag{11.45}$$

The summation in (11.45) runs over all N_b atoms of type β within a prescribed cut-off surrounding an atom of type α and r_b is the scalar distance between the α and β atoms. The function $f_c(r_b)$ is a switching function that sets the cut-off range at the required separation in a continuous (and therefore differentiable) manner. It has the form:

$$f_c(r)=\begin{cases} 1 & : r\le r_1 \\ \frac{1}{2}\left\{\cos\left[\frac{\pi(r-r_1)}{(r_2-r_1)}\right]+1\right\} & : r_1<r\le r_2 \\ 0 & : r>r_2 \end{cases}. \tag{11.46}$$

The parameters r_1 and r_2 define a range over which the β atoms gradually cease to count towards the overall sum. These must be chosen so that r_2 absolutely excludes near-neighbouring atoms that are not considered part of the sum in equation (11.45). Parameter r_1 should not be so short that it sometimes does not include atoms that should be fully counted. The range $r_1 \rightarrow r_2$ should be set to correspond to the minimum in the appropriate pair correlation functions in the relevant system states. This choice minimises spurious forces that can arise from order parameters that have different ranges. Note that the numbers N_c and N_b in the above equations are formally expected to be the same in a perfect crystal. However, while N_c remains fixed, N_b may fluctuate according to circumstance. (In fact the switching function replaces the strict cut-off in the original definition by Steinhardt *et al* in which N_c would be equivalent to:

$$N_c \equiv \sum_{b=1}^{N_b} f_c(r_b) \tag{11.47}$$

rather than a constant.) Quigley and Rodger also note that the order parameter is not scale invariant between systems of different numbers of atoms [97], however this does not matter for simulation where the numbers are fixed. The spherical harmonic parameter l is confined to the values 4 and 6 in practice, so the order parameters $Q_4^{\alpha\beta}$ and $Q_6^{\alpha\beta}$ are most common.

The forces arising from the Steinhardt parameters are given by:

$$\begin{aligned}\vec{f}_{ij} = -\frac{\vec{r}_{ij}}{r_{ij}} \frac{\partial \Phi_b}{\partial Q_l^{\alpha\beta}} \frac{1}{Q_l^{\alpha\beta}} \frac{4\pi}{2l+1} \left(\frac{1}{N_c N_\alpha} \right)^2 \sum_{m=-l}^{l} \Big\{ \Re(\bar{Q}_{lm}^{\alpha\beta}) \frac{d}{dr_{ij}} [f_c(r_{ij}) \Re(Y_{lm}(\theta_{ij}, \phi_{ij}))] \\ + \Im(\bar{Q}_{lm}^{\alpha\beta}) \frac{d}{dr_{ij}} [f_c(r_{ij}) \Im(Y_{lm}(\theta_{ij}, \phi_{ij}))] \Big\}.\end{aligned} \tag{11.48}$$

where $\Re$ and $\Im$ indicate the *Real* and *Imaginary* parts of complex quantities.

The stress tensor contributions arising from these forces are given by

$$\sigma_{\alpha\beta} \rightarrow \sigma_{\alpha\beta} - f_{ij}^{\alpha} r_{ij}^{\beta}. \tag{11.49}$$

The tetrahedral order parameter of Chau and Hardwick [101] is also trigonometric in nature and quantifies the degree to which atoms surrounding a chosen atom are arranged tetrahedrally. When the chosen atom and its surrounding neighbours are of the same type (α) the parameter, ζ_α, is defined by the formula

$$\zeta_\alpha = \frac{1}{N_c N_\alpha} \sum_{i=1}^{N_\alpha} \sum_{j \neq I}^{N_\alpha} \sum_{k>j}^{N_\alpha} f_c(r_{ij}) f_c(r_{ik}) (\cos\theta_{jik} + 1/3)^2, \tag{11.50}$$

where indices i, j and k run up to N_α atoms of type α. Integer N_c and function $f_c(r)$ are the same as defined in the Steinhardt parameters (i.e. f_c once again replaces a fixed cut-off and N_c is a fixed constant). This order parameter is

maximal for tetrahedral atomic arrangements. The atomic forces that arise from this order parameter can be expressed in terms of pair forces between atoms i and j and between i and k which are given by

$$\begin{aligned}\vec{f}_{ij} &= -\frac{\partial \Phi_b}{\partial \zeta_\alpha}\left\{\frac{2}{r_{ij}} f_c(r_{ij}) f_c(r_{ik})\left(\cos\theta_{jik}+\frac{1}{3}\right)\left(\frac{\vec{r}_{ik}}{r_{ik}}-\frac{\vec{r}_{ij}}{r_{ij}}\cos\theta_{jik}\right)+\right.\\ &\left.\left(\cos\theta_{jik}+\frac{1}{3}\right)^2 \frac{df_c(r_{ij})}{dr_{ij}} f_c(r_{ik})\frac{\vec{r}_{ij}}{r_{ij}}\right\},\\ \vec{f}_{ik} &= -\frac{\partial \Phi_b}{\partial \zeta_\alpha}\left\{\frac{2}{r_{ik}} f_c(r_{ij}) f_c(r_{ik})\left(\cos\theta_{jik}+\frac{1}{3}\right)\left(\frac{\vec{r}_{ij}}{r_{ij}}-\frac{\vec{r}_{ik}}{r_{ij}}\cos\theta_{jik}\right)+\right.\\ &\left.\left(\cos\theta_{jik}+\frac{1}{3}\right)^2 \frac{df_c(r_{ik})}{dr_{ik}} f_c(r_{ij})\frac{\vec{r}_{ik}}{r_{ij}}\right\}.\end{aligned} \tag{11.51}$$

The stress tensor contributions can be described in terms of these forces:

$$\sigma_{\alpha\beta} \to \sigma_{\alpha\beta} - f_{ik}^{\alpha} r_{ik}^{\beta} - f_{ij}^{\alpha} r_{ij}^{\beta}. \tag{11.52}$$

As mentioned above, the order parameter vector $\vec{S}^M$ consists of an ordered set of different order parameters and it is not generally the case that all of them return numbers of the same order of magnitude. This is particularly true for the potential energy. It is therefore sensible that when using the order parameters collectively to define the state of a system, that they should be scaled to give numbers of similar magnitudes. So when specifying order parameters to define the metadynamics it is advisable to include a scale factor in the definition.

Different schemes have been described by Quigley and Rodger for managing the addition of the Gaussian terms (11.40) [97]. The first is *simple addition* in which Gaussians with fixed height and width parameters are simply added to the bias potential. The second is *Wang-Landau recursion*, which starts with a given Gaussian height and with each Gaussian addition a histogram is accumulated which records the visits to each zone of the order parameter space. Once this histogram is approximately (~80%) flat, the Gaussian height is halved and the histogram is reset to zero. Then the process continues until a higher degree of flatness has been achieved, and so on. The procedure is meant to ensure that the added Gaussian terms make progressively finer contributions as convergence is approached. Finally, there is *well-tempered dynamics*, which uses a maximum energy criterion. A threshold energy V_{max} is set above the largest expected energy barrier and at each step the Gaussian deposition height is given by $w = w_0 \exp[-V_{aug}(\vec{s}^M)/V_{max}]$, where V_{aug} is the current value of the bias energy.

An important application of metadynamics is the calculation of free energy. The free energy is a function of the M order parameters in the vector $S^M(\vec{R}^N)$. This information can be used to determine the free energy of activation and free energy differences between states in the following manner. Firstly the free energy is reduced down to a smaller subset of order parameters (usually about 2) by integrating the

function $\exp(-F/k_BT)$ over the other order parameters and then Boltzmann-inverting. Once the free energy is mapped onto fewer dimensions the free energy barrier heights and free energy differences can be read off directly.

In other contexts it is often useful to track the trajectory of the system in the space of the order parameters to see how well the simulation is exploring that space. For this purpose is is possible to plot the accumulated Gaussians graphically in a selection of 2D sections. Relatively simple graphics are sufficient for this purpose. Also, as with the TAD and BPD methods, the accelerated dynamics can be employed for demonstration purposed, to reveal the evolution of the system graphically.

Chapter 12

Path Integral Molecular Dynamics

12.1 Introduction

There are occasions when the classical physics underpinning molecular simulation is not enough. Systems containing light atoms, such as hydrogen, lithium and neon, display some degree of quantum behaviour, particularly at low temperatures. This happens because they have a sufficiently large thermal de Broglie wavelength, as given by the formula

$$\Lambda = \sqrt{\frac{\beta h^2}{2\pi m}} \tag{12.1}$$

For example, a hydrogen atom at $100\ K$ has a thermal wavelength of $\approx 1.74\ \mathring{A}$ and may reasonably be expected to show quantum effects on the atomic scale. This manifests itself in a broadening of the primary peak of the radial distribution function, which signifies a degree of *quantum tunnelling* into the hard sphere radius of the atom. In order to simulate such systems we therefore require some means for including quantum effects. We shall focus on one particularly important method known as the path integral method, which is based on the Feynman description of quantum mechanics [102] This offers a very practical approach, which has an appealing directness and yields much insight into how quantum mechanics can be incorporated into the classical approach we are most familiar with.

The reasons for this choice are several. Not least is the difficulty in solving the time dependent Schroedinger Equation for many particle systems. Methods for doing this are impractical when the number of degrees of freedom in the system is large and are restricted to very short time scales. There are also conceptual difficulties. In most condensed phase atomistic systems the atoms behave semi-classically and it is therefore useful to retain the discrete particle description, which the wave function approach wholly abandons. Path integral methods are much more helpful in this regard. An easy facility to alternate between classical and quantum descriptions is perhaps the greatest strength of the path integral approach. It also handles the quantum phenomenon of de-localisation in a pleasingly intuitive way.

We shall discuss the technique known as Path Integral Molecular Dynamics (PIMD). In truth this is much the same as the Path Integral Monte Carlo (PIMC) method in that both are simply methods for exploring the *configuration space* of the quantum system. This is distinct from the *phase space* of the system, which includes the momentum. From the perspective of molecular dynamics, this is a considerable loss of utility as no time dependent phenomena can be investigated directly. Nevertheless the addition of quantum capability represents an important extension of the methodology of molecular dynamics, which is why it is included in this book.

The purpose of this chapter is to provide an introduction to PIMD, starting from an outline of the basic path integral theory, deriving the important equations and giving a description of how it is implemented. Throughout we will assume the system under investigation is a single component system, such as a noble gas. The extension to multi-component systems is not difficult.

12.2 The Propagator

We begin with the simple case of a single particle moving in one dimension and consider the concept of a propagator. Suppose we wish to solve the Time Dependent Schroedinger Equation (TDSE):

$$\hat{H}\,\Psi(x,t)=i\hbar\frac{\partial\Psi(x,t)}{\partial t}, \tag{12.2}$$

in which the Hamiltonian operator $\hat{H}$ is *not* an explicit function of time and $\hbar=h/2\pi$ is Planck's constant. It is given that the solution at time $t=0$ is known and is represented by $\Psi(x,0).$ A possible solution is provided by the following integral:

$$\Psi(x,t)=\int G(\xi,0;x,t)\Psi(\xi,0)\,d\xi, \tag{12.3}$$

In which the function G (which is known as the *Green's function*) is a so-called *propagator*, since it allows us to propagate the original wavefunction, $\Psi(\xi,0),$ through to a later time $t,$ which is what the solution $\Psi(x,t)$ represents. Obtaining the form of the function G is therefore formally equivalent to solving the TDSE. Clearly, this is not the only way the equation could be solved, but the concept of a propagator is central to the Feynman interpretation of quantum mechanics, which will guide us to the correct form for G in more difficult cases. For the simple case being considered here however, we write the function G directly as (see Appendix 4):

$$G(\xi,0;x,t)=\sum_n \phi_n^*(\xi)\phi_n(x)\exp(-i\hbar^{-1}E_n t), \tag{12.4}$$

in which the functions ϕ_n are the eigenfunctions of the operator, $\hat{H},$ and E_n is the corresponding eigenvalue of $\phi_n.$ Thus, provided we can solve the time independent Schroedinger equation for the operator $\hat{H}$ we have, through $G,$ the solution to the more difficult problem of the propagation of the wavefunction Ψ through time.

A simple example of the use of a Green's function, and one which is of particular relevance to PIMD (see below), is the propagation of a free particle wavefunction moving in a constant potential Φ_o. The eigenfunctions for the Hamiltonian appropriate to this case have the form:

$$\phi_p(x)=W^{-1/2}\exp(i\hbar^{-1}px), \tag{12.5}$$

in which W is the system size (i.e. the width) and

$$p^2 = 2m(E_p - \Phi_o), \tag{12.6}$$

where p is the momentum and E_p is the energy. Using this form of eigenfunction, the Green's function may be written as:

$$G(\xi, 0; x, t) = (2\pi\hbar)^{-1} \int \exp\left(-i\hbar^{-1} \left[\left(\frac{p^2}{2m} \right) t + p(\xi - x) + t\Phi_o \right] \right) dp, \tag{12.7}$$

in which the sum in (12.4) has been replaced by an integral in recognition of the infinitesimal differences between the possible eigenvalues. Performing the integration leads to:

$$G(\xi, 0; x, t) = \left(\frac{m}{2\pi i \hbar t} \right)^{1/2} \exp\left(\left(\frac{i m}{2\hbar t} \right) \left[(\xi - x)^2 - \frac{2t^2}{m} \Phi_o \right] \right). \tag{12.8}$$

It is important to note that the Green's function is a Gaussian function of the spatial coordinates. In principle one may now use this function to propagate a given starting wavefunction and derive its time evolution. However it is not the purpose of PIMD to solve problems of this nature, as we shall see. What is of interest however is the case where the potential is a function of the position rather than a simple constant. In this circumstance the Green's function cannot, in general, be obtained in closed form, but Feynman's path integral approach allows us to derive an acceptable approximate solution, which in fact can be made as accurate as desired.

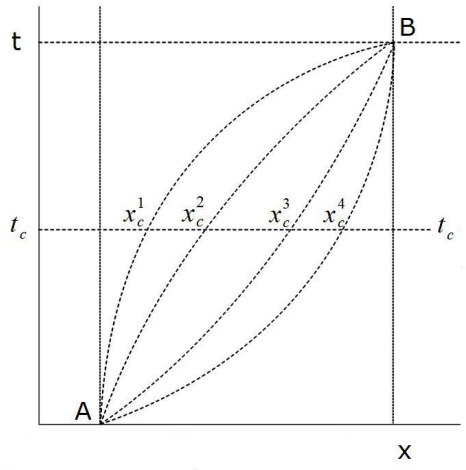

Figure 12.1: The Path Integral

Consider the space-time diagram (figure 12.1) for the propagation of a particle in one-dimensional space. The particle moves from point A at time zero to point B at time t. According to the tenets of quantum theory we cannot know what path the particle follows with certainty; we can only make probabilistic statements about where the particle is likely to be at a given time. We cannot perform any experiment to reveal its

position, since this will disrupt the process of propagation. However, the path integral theory provides a way of analysing the propagation from point $(A;0)$ to point $(B;t)$ through consideration of the intermediate points of the trajectory. Basically, the theory postulates that all possible trajectories starting at $(A;0)$ and ending at $(B;t)$ are equally admissible. According to Feynman the *probability amplitude* of arrival at $(B;t)$ arises from a sum (or integral) over all the possible paths. Most of the possible paths cancel each other through their different phase factors (i.e. destructive interference); only those paths close to the classical trajectory will effectively contribute to the propagation process. In the classical limit there is no significant contribution from any but the classical trajectory. On the atomic scale however we cannot specify the path followed with the same certainty.

If we consider a point in time t_c between 0 and t (figure 12.1) then we understand that the moving particle has a finite probability of crossing the line t_c at *any* position (e.g. $x_c^1, x_c^2, x_c^3, \ldots$ *etc.*). If we choose a point along this line, x_c^1 say, then we can describe the propagator from point A to x_c^1 quite simply as $G(A,0;x_c^1,t_c)$. Similarly the propagator from x_c^1 to B is $G(x_c^1,t_c;B,t)$. In Feynman's formulation, the propagator from A to B *via* x_c^1 is the product $G(A,0;x_c^1,t_c)G(x_c^1,t_c;B,t)$. (This is because the arrival of the particle at x_c^1 from A has a probability amplitude proportional to $G(A,0;x_c^1,t_c)$ and the arrival at B from x_c^1 has a probability amplitude proportional to $G(x_c^1,t_c;B,t)$, thus the probability amplitude of both events occurring in sequence is proportional to the product of the two [102]. To get the correct description of the complete propagator from A to B we must reconstruct all the possible paths in between; that is we must consider all arrivals along the line t_c from A and all the departures from t_c to B. In other words the propagator G(A,0;B,t) is given by:

$$G(A,0;B,t)=\int G(A,0;x_c,t_c)G(x_c,t_c;B,t)dx_c. \tag{12.9}$$

In this decomposition of the propagator we have considered only one partition of the time interval $[0,t]$. Clearly, we may consider any number of partitions and by a similar process expand the propagator to:

$$G(A,0;B,t)=\int\cdots\int[\Pi_{k=0}^{L-1}G(x_k,t_k;x_{k+1},t_{k+1}-t_k)]\Pi_{k=1}^{L-1}dx_k, \tag{12.10}$$

where $x_0=A$, $x_L=B$, $t_0=0$ and $t_L=t$. This equation describes the partition of the interval $[0,t]$ into L sub-intervals. (Note the symbol $\Pi_{k=0}^{L-1}$ indicates a product of L Green's functions in the above expression.)

The purpose of performing such a decomposition of the propagator is to be able to deal with more complicated potential energy functions, because we may now assume that the division of the time interval $[0,t]$ is such that over each sub-interval, the potential energy function hardly varies and we may treat it as a constant (i.e. $\Phi(x)\approx\Phi(x_i)$ over the interval $[x_i,x_{i+1}]$). Using these assumptions we can write the propagator for the cases with more complicated potential functions as:

$$G(A,0;B,t)=\left(\frac{m}{2\pi i\hbar(t/L)}\right)^{L/2}\int\cdots\int\exp\left(\left(\frac{im}{2\hbar(t/L)}\right)\right.$$
$$\left.\sum_{k=1}^{L}\left[(x_k-x_{k-1})^2-\frac{2}{m}(t/L)^2\Phi(x_k)\right]\right)dx_1\cdots dx_{L-1}. \tag{12.11}$$

In which the Gaussian propagator (12.8) appropriate to the case of a constant potential has been used for each sub-interval. This description is formally exact in the limit as $L\to\infty$. Thus we see how the application of Feynman's path integral method has allowed us sufficient insight into this rather difficult problem to provide a workable description of the propagator. Below, we make use of this approximation of the trajectory by finite sub-intervals. It is commonly called the *short time approximation*.

12.3 Statistical Mechanics

According to the formal laws of quantum mechanics, when a particle is in a quantum state ϕ_n, the probability density of the particle at a point x is given by the product $\phi_n^*(x)\phi_n(x)$. However, in a system at a finite temperature, this is merely one of the many states thermally accessible to the particle. Therefore the probability density for the particle in the state ϕ_n is given by a Boltzmann weighting of the state thus:

$$\frac{1}{Z}\phi_n^*(x)\phi_n(x)\exp(-\beta E_n), \tag{12.12}$$

in which E_n represents the energy eigenvalue of the state ϕ_n, β is $1/k_BT$ and Z is the normalising factor, which is given by

$$Z=\int\sum_n\phi_n^*(x)\phi_n(x)\exp(-\beta E_n)dx. \tag{12.13}$$

This is of course the partition function for the states thermally accessible to the particle. Therefore the average of a physical quantity A can be obtained in a way that is analogous to the classical case i.e.

$$\langle A\rangle=\frac{1}{Z}\int\sum_n\phi_n^*(x)\hat{A}\phi_n(x)\exp(-\beta E_n)dx, \tag{12.14}$$

in which $\hat{A}$ is the quantum mechanical operator associated with the physical property A.

In quantum mechanics, a more general way of expressing the partition function is by means of the density function $\rho(x,x';\beta)$, which is defined as:

$$\rho(x,x',\beta)=\sum_n\phi_n^*(x)\phi_n(x')\exp(-\beta E_n). \tag{12.15}$$

In terms of this function, the partition function may be expressed as:

$$Z = \int \rho(x, x'; \beta) dx = Tr[\rho] \tag{12.16}$$

and the average of a physical quantity A as:

$$\langle A \rangle = \frac{1}{Z} Tr[\hat{A} \rho(x, x'; \beta)], \tag{12.17}$$

in which the operation Tr (known as the *trace*) implies that the operator $\hat{A}$ acts first on functions of the coordinate x' and then x' is set equal to x for the final integration. (In the case of the integral for Z the operator is unity.) These equations serve to show the central importance of the density function $\rho(x, x'; \beta)$ in the statistical mechanics of quantum systems.

The PIMD and PIMC methods follow directly from this. We note that the form of the density function (12.15), which is expressed in terms of the wavefunctions ϕ_n, is analogous to the Green's function propagator (12.4). This suggests at once that any method that we can apply to solving the time dependent Schroedinger equation may also be applied to obtaining the partition function. All that is required is the formal substitution of β in place of the quantity $i\hbar^{-1}t$. This insight means that we may use the short time approximation described above and other properties of path integrals to calculate the properties of a quantum mechanical ensemble. Thus the density function $\rho(x, x'; \beta)$ can be written directly as:

$$\rho(x, x'; \beta) = \left(\frac{mL}{2\pi\hbar^2\beta} \right)^{L/2} \int \cdots \int \exp\left(-\beta \sum_{k=1}^{L} \left[\frac{mL}{2(\hbar\beta)^2} (x_k - x_{k-1})^2 + \frac{1}{L} \Phi(x_k) \right] \right) dx_1 \cdots dx_{L-1}, \tag{12.18}$$

which is obtained directly from the expression for the Green's function $G(A, 0; B, t)$ given in equation (12.11), and we assume $x = x_0$ and $x' = x_L$.

The extension of this formula to three dimensional problems is trivial; we merely replace the coordinates x_k by the vectors $\vec{r}_k$ and the pre-exponential factor becomes $(mL/[2\pi\hbar^2\beta])^{3L/2}$. To evaluate the partition function Z in (12.16) it is necessary to calculate the trace of the density function, which means setting $\vec{r} = \vec{r}'$ (or equivalently, $\vec{r}_0 = \vec{r}_L$,) and integrating over $\vec{r}$.

Since we are still dealing with the partition function of a single particle, we now assume that the partition function for N particles is the product of N single-particle partition functions. This leads to the following form for Z.

$$Z = \left(\frac{mL}{2\pi\hbar^2\beta} \right)^{3LN/2} \int \cdots \int \exp\left(-\beta \sum_{j=1}^{N} \sum_{k=1}^{L} \left[\frac{\kappa}{2} (\vec{r}_{j,k} - \vec{r}_{j,k-1})^2 + \frac{1}{L} \Phi(\vec{r}_{j,k}) \right] \right) \Pi_j^N \Pi_k^L d\vec{r}_{j,k}, \tag{12.19}$$

with

$$\kappa = \frac{m\,L}{(\beta\hbar)^2} \tag{12.20}$$

where index j refers to the $j'th$ particle and we have set $\vec{r}_{j,0} = \vec{r}_{j,L}$.

In this form the partition function reveals a surprising isomorphism: it is the same as the classical partition function for a system of ring polymers. To be more explicit: the quadratic term within the exponential corresponds to a ring necklace of L *beads*, each of which is coupled to two neighbours via a harmonic spring with force constant κ, given by (12.20). In addition each bead in the ring experiences the potential $\Phi(\vec{r}_{j,k})/L$, which arises from the interactions between the different polymer rings. (It is worth pointing out that the reason a ring polymer results from this treatment, is that the application of the trace operation Tr forces the closure of the ring.) Note that (12.19) is an integral over configuration space and does not involve momentum.

The most appealing aspect of this isomorphism is the fact that we can use classical molecular dynamics or Monte Carlo methods to explore the configuration space of the ring polymers and from them calculate the properties of quantum mechanical system. This is not particularly difficult, provided care is taken to ensure the systems are properly sampled in the statistical sense. We should also point out (again) that when molecular dynamics is used to sample the configuration space, it must not be thought that this implies that time propagation of the polymer system is relevant to the time propagation of the quantum system; the theory of the method does not support it.

12.4 The Properties of the Isomorphic Ring Polymer

The system of isomorphic polymers is one in which the *effective* potential energy may be written as:

$$\Phi_{eff}^{N} = \sum_{j=1}^{N} \sum_{k=1}^{L} \left[\frac{\kappa}{2} \left(\vec{r}_{j,k} - \vec{r}_{j,k-1} \right)^2 + \frac{1}{L} \Phi(\vec{r}_{j,k}) \right]. \tag{12.21}$$

We have already remarked on the nature of the quadratic term; classically it is equivalent to the intra-molecular interactions of a harmonic ring polymer. What is its physical significance in the quantum mechanical case? Since we have replaced a classical *point* particle by a *de-localised* polymer, we may guess that it reflects the *Heisenberg uncertainty* in the position of the particle. This intuitively appealing idea is supported by the properties of the spring constant κ, defined in equation (12.20). For a fixed number of beads, L, when the mass, m, of the particle is large, or when the temperature is high (i.e. β is small), κ is large and the ring polymer will tend to be tightly bound into a small volume. These conditions correspond to the classical limit. Conversely, a small mass or a low temperature will weaken the spring constant and the polymer will de-localise to a much greater extent.

A feature of the ring polymers we should discuss is the number of beads L. In

practice a suitable choice of L has to be determined by experience. There are however some facts which guide the choice. Firstly, the short time approximation inherent in the derivation of the partition function requires that the number of beads be sufficiently large to justify the approximation. Thus a rapidly varying potential function implies a need for a large number of beads. Secondly, the delocalisation of the ring polymer associated with quantum behaviour also implies an increase in L since the beads will sample a wider range of potential energy values. Increased quantum behaviour is associated with reducing the mass of the particles and/or the temperature of the system, so both of these influence the choice of $L.$ Thus we find that simulations of argon below the triple point require about 5 beads [103] and liquid neon requires 20-40 beads [104]. Simulations of the electron in molten potassium chloride have required about 200 beads [105] and the electron in liquid ammonia several thousand beads [106].

The second term of the effective potential Φ_{eff}^{N} is the $\Phi(\vec{r}_{j,k})/L$ term, which corresponds to the interaction between different polymer rings. We notice the presence of the bead number L in the denominator. This means that the interaction between beads on different rings is effectively reduced by the factor $1/L$. As a result of this, we find that beads on different polymers are allowed to approach each other more closely than the original classical particles. In other words, the beads have a greater ability to *tunnel* into classically forbidden regions. This is exactly the kind of behaviour we expect for quantum systems.

As for the potential $\Phi(\vec{r}_{j,k}),$ there are one or two points to mention. In principle, it may be treated as the sum of pair interactions between sites (beads) on different polymers, as is normally done in classical simulations. However we should point out an important difference. Namely each bead on a given polymer ring can only interact with *one* bead on a different polymer ring. It does *not* interact with them all (see figure 12.2). The reason for this lies in the correspondence between the time dependent Green's function and the temperature dependent density function. The former is constructed from a partition of a time interval into L sub-intervals. The potential energy function is evaluated once for each sub-interval and the contributions

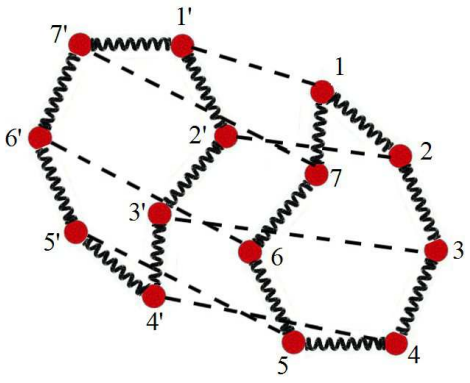

Figure 12.2: Bead-bead Interactions in the Effective Potential Energy

all derive from that time interval (i.e. all contributions are evaluated at the same instant in time). Translation of this feature into the calculation of the density function

shows that, since each bead corresponds to a time sub-interval in the original Green's function, a bead on a given polymer can only interact with the corresponding bead on another. Another way of saying this is that corresponding beads on different polymers exist at the same moment in imaginary time and can only interact with beads existing at the same moment. Thus we write the potential term as:

$$\Phi(\vec{r}_{j,k}) = \frac{1}{2}\sum_{j' \neq j}^{N} \Phi_{pair}(|\vec{r}_{j,k} - \vec{r}_{j',k}|). \tag{12.22}$$

Where $\Phi_{pair}(|\vec{r}_{j,k} - \vec{r}_{j',k}|)$ is a suitable pair potential, such as the Lennard-Jones potential. The factor ½ appears to allow for double counting in the double sum over j and j'.

This feature of the inter-polymer potential has two useful consequences. Firstly, the number of bead-bead interactions that must be calculated is considerably less than in a simulation of a real classical polymer. This represents a great reduction in computational cost. Secondly, the straightforward empirical (e.g. Lennard-Jones) potential can be used in the simulation without modification. This advantage only becomes apparent when we try to simulate quantum systems with wavepackets, since there the basic potential must be modified to take into account the delocalisation of the particle over the wavepacket [107]. In the path integral case, we are restricted to dealing with corresponding beads on each ring and the delocalisation of the rest of the ring does not enter into consideration.

In order to permit molecular dynamics with the effective potential (12.21) we need to construct a suitable Hamiltonian. This is easily obtained by the addition of a suitable kinetic energy term:

$$H^{N,L}(\{\vec{r}_{i,n}\},\{\vec{p}_{i,n}\}) = \sum_{j=1}^{N}\sum_{k=1}^{L}\left[\frac{p_{j,k}^2}{2m'} + \frac{\kappa}{2}\left(\vec{r}_{j,k} - \vec{r}_{j,k-1}\right)^2 + \frac{1}{L}\Phi(\vec{r}_{j,k})\right], \tag{12.23}$$

where $\vec{p}_{j,k}$ is the bead momentum. Note the use of the *dynamical mass* m', which need not be the same as the particle mass m.

12.5 Thermodynamic and Structural Calculations

In classical molecular dynamics or Monte Carlo simulations one of the simplest quantities to calculate is the energy of the system. The expression for this is obtained from the partition function in the form of a derivative:

$$\langle E \rangle = -\frac{\partial}{\partial \beta} \ln Z. \tag{12.24}$$

Application of this rule to the partition function (12.19), (while recognising that the spring constant κ is also a function of β,) gives an energy estimator of the form

$$\langle E\rangle = \frac{3LN}{2\beta} - \left\langle \sum_{j=1}^{N}\sum_{k=1}^{L}\left[\frac{\kappa}{2}\left(\vec{r}_{j,k}-\vec{r}_{j,k-1}\right)^2 - \frac{1}{L}\Phi\left(\vec{r}_{j,k}\right)\right]\right\rangle. \tag{12.25}$$

The first term on the right and the harmonic term with the force constant κ represent the kinetic energy of the system. The last term right represents the potential energy. This expression for the energy of the quantum system (which should *not* be confused with the Hamiltonian (12.23) that drives the molecular dynamics) is subject to fluctuations that grow linearly with the number of beads in the polymer rings [108] and consequently may lead to poor estimates of the energy. For this reason a new form of the energy estimator based on the virial was proposed by Herman *et al.* [108]. In the many-particle systems we are concerned with here, the estimator takes the form

$$\langle E\rangle = \frac{3N}{2\beta} + \left\langle \frac{1}{L}\sum_{j=1}^{N}\sum_{k=1}^{L}\left[\frac{1}{2}\left(\vec{r}_{j,k}-\vec{R}_j\right)\cdot\frac{\partial}{\partial\vec{r}_{j,k}}\Phi\left(\vec{r}_{j,k}\right)+\Phi\left(\vec{r}_{j,k}\right)\right]\right\rangle, \tag{12.26}$$

where $\vec{R}_j$ is the centre of mass of the $j'th$ ring polymer given by

$$\vec{R}_j = \frac{1}{L}\sum_{k=1}^{L}\vec{r}_{j,k}. \tag{12.27}$$

The virial estimator is considered to be more accurate than (12.25) in some applications, though not universally. It is apparently less accurate for condensed phase Lennard-Jones systems [104].

A formula for the pressure can be obtained by differentiating the Hamiltonian (12.23) with respect to the volume, as described in chapter 7. The result is

$$PV = \frac{LN}{\beta} - \frac{1}{3}\left\langle \sum_{j=1}^{N}\sum_{k=1}^{L}\left[\kappa\left(\vec{r}_{j,k}-\vec{r}_{j,k-1}\right)^2 + \left(\frac{1}{L}\frac{\partial}{\partial\vec{r}_{j,k}}\Phi\left(\vec{r}_{j,k}\right)\right)\cdot\vec{r}_{j,k}\right]\right\rangle. \tag{12.28}$$

In the case where $\Phi(\vec{r}_{j,k})$ is a sum of pair potentials, as in (12.22) this becomes

$$PV = \frac{LN}{\beta} - \frac{1}{3}\left\langle \sum_{j=1}^{N}\sum_{k=1}^{L}\left[\kappa\left(\vec{r}_{j,k}-\vec{r}_{j,k-1}\right)^2 + \frac{1}{2L}\sum_{j'\neq j}^{N}\left(\frac{\partial}{\partial\vec{r}_{jj',k}}\Phi_{pair}\left(\vec{r}_{jj',k}\right)\right)\cdot\vec{r}_{jj',k}\right]\right\rangle, \tag{12.29}$$

where

$$\vec{r}_{jj',k} = \vec{r}_{j',k} - \vec{r}_{j,k}. \tag{12.30}$$

A useful quantity to calculate in these quantum systems is the mean-square radius, $\langle R^2\rangle$, of the polymer rings. This is given quite simply by the formula:

$$\langle R^2 \rangle = \left\langle \frac{1}{LN} \sum_{j=1}^{N} \sum_{k=1}^{L} |\vec{r}_{j,k} - \vec{R}_j|^2 \right\rangle, \tag{12.31}$$

where $\vec{R}_j$ is the centre of mass of the polymer ring given by (12.27). The usefulness of this quantity is that it provides a ready measure of the delocalisation of the quantum particle. In this respect it rather resembles the mean-square width parameter $\langle \xi^2 \rangle$, encountered in semi-classical simulations of Gaussian wavepackets and which roughly corresponds to the mean-squared-width of the wavepackets [109]. It is of the same order of magnitude as the square of the thermal de Broglie wavelength (12.1).

The calculation of the radial distribution function for a many-particle system presents no difficulty in PIMD if we remember to consider only pairs of corresponding beads on different ring polymers. Thus we may write:

$$g(r) = \frac{1}{LN\rho} \left\langle \sum_{k=1}^{L} \sum_{j=1}^{N} \sum_{j' \neq j}^{N} \delta(|\vec{r} - \vec{r}_{j',k} + \vec{r}_{j,k}|) \right\rangle. \tag{12.32}$$

12.6 Implementation

PIMD allows us to calculate quantum properties through an analogy of the quantum system with a classical ring polymer. So in principle a classical molecular dynamics program can be exploited to do the job with relatively minor modification. This does not mean however, that the exercise is trivial. The harmonic springs of the ring polymers require special techniques to ensure ergodic dynamics – meaning that energy is exchanged between the degrees of freedom of the polymers with sufficient efficiency to guarantee equipartition and good sampling of the system configuration space. It is fair to say that almost all the difficulty of doing PIMD simulations lies in this issue.

The stiffness of the springs is an important factor in this. In general the problem of ergodicity is made worse as the stiffness increases. Stiffness is governed by the spring constant κ, which through (12.20) is evidently dependent on the bead number L, the mass m, and β (i.e. temperature). Increasing the mass, temperature or L will increase spring stiffness, but note that reducing mass or temperature increases quantum behaviour, which requires increasing L in response. This makes the problem of spring stiffness hard to avoid.

To deal with the ergodicity problem there are two issues that need to be resolved. Firstly, we must address the range of time scales that arise in the harmonic vibrations of the springs and slows the sampling of the configuration space. Secondly we must apply an efficient thermostatting scheme to ensure ergodicity. There are many approaches that can be taken, for example switching to normal modes for a vibrating system allows the different time scales to be addressed separately and stochastic methods can be used to thermostat the different modes. Here however we will describe the method of Tuckerman *et al.* [110], which is highly efficient and widely

used.

Rather than use normal modes Tuckerman *et al.* define a coordinate transformation of the form

$$\begin{aligned}\vec{u}_{j,1} &= \vec{r}_{j,1}, \\ \vec{u}_{j,k} &= \vec{r}_{j,k} - \frac{(k-1)\vec{r}_{j,k+1} + \vec{r}_{j,1}}{k}, \qquad (k = 2, \ldots, L).\end{aligned} \tag{12.33}$$

(Note that $\vec{r}_{j,L+1} = \vec{r}_{j,1}$ in these formulae.) The inverse of this transformation is

$$\begin{aligned}\vec{r}_{j,1} &= \vec{u}_{j,1}, \\ \vec{r}_{j,k} &= \vec{u}_{j,k} + \sum_{l=k}^{L} \frac{k-1}{l-1}\vec{u}_{j,l}, \qquad (k = 2, \ldots, L),\end{aligned} \tag{12.34}$$

for which there is also a useful recursive expression

$$\begin{aligned}\vec{r}_{j,1} &= \vec{u}_{j,1}, \\ \vec{r}_{j,k} &= \vec{u}_{j,k} + \frac{k-1}{k}\vec{r}_{j,k+1} + \frac{1}{k}\vec{r}_{j,1}, \qquad (k = L, \ldots, 2),\end{aligned} \tag{12.35}$$

which can be started at $k = L.$

We also introduce additional parameters μ_k defined as

$$\begin{aligned}\mu_1 &= 0, \\ \mu_k &= \frac{k}{k-1}, \qquad (k = 2, \ldots, L).\end{aligned} \tag{12.36}$$

With these new variables the driving Hamiltonian (12.23) becomes

$$H^{N,L}(\{\vec{u}_{i,n}\}, \{\vec{p}_{i,n}\}) = \sum_{j=1}^{N}\sum_{k=1}^{L}\left[\frac{p_{j,k}^2}{2m_k'} + \mu_k \frac{\kappa}{2}\left(\vec{u}_{j,k}\cdot\vec{u}_{j,k}\right) + \frac{1}{L}\Phi\left(\vec{r}_{j,k}(\{\vec{u}_{j',k}\})\right)\right], \tag{12.37}$$

in which a mass specific to each vector $\vec{u}_{j,k}$, namely m_k', has been introduced into the momentum term. (A suitable choice for m_k' will be given below.) This Hamiltonian has a form that permits separate treatment of the different time scales associated with each coordinate.

The next step is to derive a set of equations of motion based on (12.37). This is where the choice of thermostat procedure is made. The choice recommended by Tuckerman *et al.* is to use the Nosê-Hoover chains encountered in chapter 4 (section 4.8.7), but it is also worth considering the "Gentle" thermostat described in section 4.8.8 , since it requires far less bookkeeping. Here we follow Tuckerman *et al.* and describe the Nosé-Hoover chain approach, which has been fully tested for PIMD. The equations of motion including the thermostats are

$$
\begin{aligned}
&\dot{p}^{\alpha}_{jk} = -\mu_k \kappa u^{\alpha}_{jk} - \frac{1}{L}\frac{\partial}{\partial u^{\alpha}_{j,k}}\Phi\left(\vec{r}_{j,k}\left(\{\vec{u}_{j',k}\}\right)\right) - \zeta^{\alpha}_{j,k,1}u^{\alpha}_{j,k}, \qquad (\alpha = x,\ y,\ z),\\
&\dot{\vec{u}}_{jk} = \frac{\vec{p}_{j,k}}{m_k'},\\
&\dot{\zeta}^{\alpha}_{j,k,1} = \frac{k_B}{Q_{k,1}}\left(T - T_0\right) - \zeta^{\alpha}_{j,k,1}\zeta^{\alpha}_{j,k,2},\\
&\dot{\zeta}^{\alpha}_{j,k,n} = \frac{1}{Q_{k,n}}\left\{Q_{k,n-1}\left(\zeta^{\alpha}_{j,k,n-1}\right)^2 - k_B T_0\right\} - \zeta^{\alpha}_{j,k,n}\zeta^{\alpha}_{j,k,n+1}, \qquad (n = 2,\ldots, M-1),\\
&\dot{\zeta}^{\alpha}_{j,k,M} = \frac{1}{Q_{k,M}}\left\{Q_{k,M-1}\left(\zeta^{\alpha}_{j,k,M-1}\right)^2 - k_B T_0\right\}.
\end{aligned}
\tag{12.38}
$$

The variables $\zeta^{\alpha}_{j,k,n}$ represent the thermostat chain variables, ranging from $n=1$ to $n=M$, the order of the chain. Indices k and j identify variables pertaining to the transform variable $\vec{u}_{j,k}$, and α identifies the Cartesian component. Note that every individual degree of freedom in the system has its own thermostat chain! This apparent over-kill has been shown to be necessary by extensive use. These equations are readily integrated using the velocity Verlet algorithm. The forces required by the first equation of (12.38) are obtained using the chain rule and the recursion formula(12.35), which gives

$$
\begin{aligned}
&\frac{1}{L}\frac{\partial}{\partial u^{\alpha}_{j,1}}\Phi\left(\vec{r}_{j,k}\left(\{\vec{u}_{j,k}\}\right)\right) = \frac{1}{L}\sum_{k=1}^{L}\frac{\partial}{\partial r^{\alpha}_{j,k}}\Phi\left(\vec{r}_{j,k}\right),\\
&\frac{1}{L}\frac{\partial}{\partial u^{\alpha}_{j,k}}\Phi\left(\vec{r}_{j,k}\left(\{\vec{u}_{j,k}\}\right)\right) = \frac{1}{L}\left(\frac{(k-2)}{(k-1)}\frac{\partial}{\partial u^{\alpha}_{j,k-1}} + \frac{\partial}{\partial r^{\alpha}_{j,k}}\right)\Phi\left(\vec{r}_{j,k}\left(\{\vec{u}_{j,k}\}\right)\right).
\end{aligned}
\tag{12.39}
$$

All that remains is to specify the masses m_k'. These can be obtained from the requirement that the time scale for the evolution of each variable $\vec{u}_{j,k}$ should be approximately the same for all. This can be arranged by setting $m_k' = a\mu_k m$, where a is a suitable constant, μ_k is given by (12.36) and m is the physical mass of the quantum particle.

In implementing these equations in a simulation, it is useful to note that the following quantity is conserved and can be used to check the reliability of the integration.

$$
\begin{aligned}
E_{cons} = \sum_{j=1}^{N}\sum_{k=1}^{L}\Bigg[&\frac{p^2_{j,k}}{2m_k'} + \mu_k\frac{\kappa}{2}\left(\vec{u}_{j,k}\cdot\vec{u}_{j,k}\right) + \frac{1}{L}\Phi\left(\vec{r}_{j,k}\left(\{\vec{u}_{j',k}\}\right)\right) +\\
&\sum_{n=1}^{M}\sum_{\alpha}\left\{\frac{1}{2}Q_{k,n}\left(\dot{\zeta}^{\alpha}_{j,k,n}\right)^2 + \frac{1}{\beta}\zeta^{\alpha}_{j,k,n}\right\}\Bigg].
\end{aligned}
\tag{12.40}
$$

It would also be useful to verify the Gaussian distribution of sample thermostats using the technique described in chapter 4, section 4.8.9 .

Appendix 1. The Gaussian Distribution

The Gaussian distribution is an important distribution function in statistics and statistical physics. Its mathematical form is given by

$$g(x)=\frac{1}{\sigma\sqrt{2\pi}}\exp\left(-\frac{x^2}{2\sigma^2}\right). \tag{A.1.1}$$

This is the normalized form, so that

$$\int_{-\infty}^{\infty} g(x)\,dx=1. \tag{A.1.2}$$

This function has the well-known "bell-distribution" form shown in figure A.1.1, which is symmetric about the origin along the x-axis.

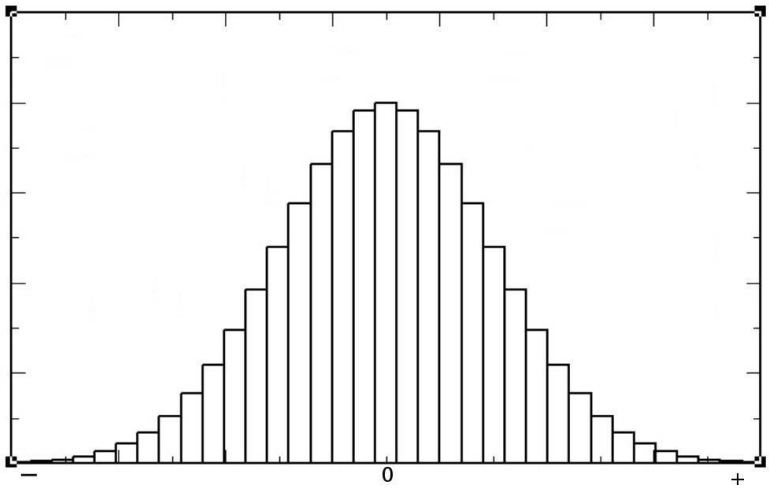

Figure A.1.1: The Normal Distribution

This function also describes the distribution of atomic velocities in molecular systems:

$$f(v_x)=\sqrt{\frac{m}{2\pi k_B T}}\exp\left(-\frac{mv_x^2}{2k_B T}\right), \tag{A.1.3}$$

which is the distribution of the x-component of the atomic velocity. The y- and z-components have similar forms.

Useful properties of the Normal distribution includes the zeroth, first, second and fourth moments, defined as follows:

$$
\begin{aligned}
I_0 &= \int_{-\infty}^{\infty} g(x)\,dx = 1, && \text{(a)}\\
I_1 &= \int_{-\infty}^{\infty} x\ g(x)\,dx = 0, && \text{(b)}\\
I_2 &= \int_{-\infty}^{\infty} x^2 g(x)\,dx = \sigma^2, && \text{(c)}\\
I_4 &= \int_{-\infty}^{\infty} x^4 g(x)\,dx = 3\sigma^4. && \text{(d)}
\end{aligned}
\qquad \text{(A.1.4)}
$$

The first of these, (A.1.4)(a), is obviously the normalisation condition (A.1.2). The second, (A.1.4)(b), is the first of the *odd* moments, and like all higher odd moments, is zero. The third, (A.1.4)(c), gives the mean-squared width of the Normal curve, which is an important parameter characterising the distribution. The fourth, (A.1.4)(d) and all higher *even* moments are characteristic properties of the normal distribution and are useful for checking that a suspected Gaussian distribution, is in fact a true Gaussian (see chapter 4, section 4.8.9 and [4].)

It is interesting to note that the even moments can be obtained in a surprising way. Consider the differentiation of equation (A.1.1) with respect to the parameter σ. This we can write as

$$
\frac{d}{d\sigma}\int_{-\infty}^{\infty}\frac{1}{\sigma\sqrt{2\pi}}\exp\left(-\frac{x^2}{2\sigma^2}\right)dx = 0, \qquad \text{(A.1.5)}
$$

which can be easily expanded into

$$
-\frac{1}{\sigma}\int_{-\infty}^{\infty}\frac{1}{\sigma\sqrt{2\pi}}\exp\left(-\frac{x^2}{2\sigma^2}\right)dx + \frac{1}{\sigma^3}\int_{-\infty}^{\infty}\frac{1}{\sigma\sqrt{2\pi}}\exp\left(-\frac{x^2}{2\sigma^2}\right)x^2\,dx = 0. \qquad \text{(A.1.6)}
$$

This is the same as

$$
-\frac{1}{\sigma}I_0 + \frac{1}{\sigma^3}I_2 = 0, \qquad \text{(A.1.7)}
$$

from which it follows that

$$
I_2 = \sigma^2 I_0 = \sigma^2. \qquad (8)
$$

This is the same result as equation (A.1.4)(c). The (even) higher order moments can be obtained similarly. This trick of obtaining higher order Gaussian integrals by differentiation with respect to the width parameter is useful and worth remembering.

Appendix 2. The Dirac Delta Function

The Dirac delta function, $\delta(x),$ is defined (without much rigour) as a function that has unit area between the ranges $[-\epsilon/2, \epsilon/2],$ where $\epsilon \to 0,$ but is zero everywhere outside this range. Such a function is represented in figure A.2.1, in which the vertical rectangle has width ϵ and height $1/\epsilon.$

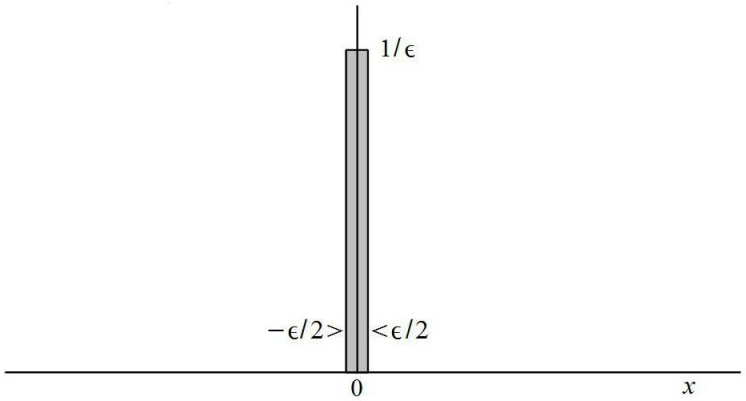

Figure A.2.1: The Dirac Delta Function

Mathematically, this can be written as

$$\int_{-\infty}^{\infty} \delta(x)\,dx = \int_{-\epsilon/2}^{\epsilon/2} \delta(x)\,dx = 1. \quad \text{with} \quad \epsilon \to 0. \tag{A.2.1}$$

Equation (A.2.1) provides a definition of the Heaviside function, $H(x),$ which has the properties

$$\begin{aligned} H(x) &= 0 \qquad when \quad x < 0, \\ H(x) &= 1 \qquad when \quad x \geq 0. \end{aligned} \tag{A.2.2}$$

and so we may define the Heaviside function as

$$H(x) = \int_{-\infty}^{x} \delta(u)\,du. \tag{A.2.3}$$

From which it follows that we may also define the derivative of $H(x)$ as

$$\frac{d}{dx} H(x) = \delta(x). \tag{A.2.4}$$

These formulae are intuitive rather than rigorous, but extremely useful.

The Dirac delta function has a number of interpretations. In physics it can be thought of as an impulse function - an instantaneous force, perhaps representing a collision between hard particles. In statistical mechanics it is a distribution function, for example defining the microcanonical or NVE ensemble, the partition function for which

is (see chapter 3):

$$Q^N(V,E)=\frac{h^{-3N}}{N!}\int \delta\left(H^N(\vec{\Gamma}^N)-E\right)d\vec{\Gamma}^N. \tag{A.2.5}$$

The form (A.2.5) guarantees the ensemble has the fixed energy $E.$ In numerical analysis, signals processing and statistics, it is used as a sampling function, as expressed in the equation

$$\int_{-\infty}^{\infty} f(x)\delta(x-u)dx=f(u), \tag{A.2.6}$$

in which the continuous function $f(x)$ is sampled at the point $x=u.$ The intuitive explanation of (A.2.6) is that, over the infinitesimal range $[u-\epsilon/2,u+\epsilon/2],$ the function $f(x)$ is indistinguishable from the constant value $f(u)$ i.e.

$$\int_{-\infty}^{\infty} f(x)\delta(x-u)dx=f(u)\int_{u-\epsilon/2}^{u+\epsilon/2}\delta(x-u)dx=f(u). \tag{A.2.7}$$

An important application of sampling is the *discretisation* of a continuous function $f(x)$:

$$\sum_{i=0}^{N-1} f(x_i)=\int_{-\infty}^{\infty}\sum_{i=0}^{N-1} f(x)\delta(x-i\Delta x)dx. \tag{A.2.8}$$

This procedure provides N samples of the function $f(x)$ at regular intervals of $\Delta x,$ and has innumerable applications. For example it is central to the development of the discrete (or *fast*) Fourier transform.

The Dirac delta function is also central to the concept of a *response function*, which describes how a system behaves when a sharp impulse[26] occurs in the system. In figure A.2.2 an applied impulse $\delta(x),$ produces the response, $h(x),$ which need not overlap with the impulse itself.

26 Variable x can be either time or distance.

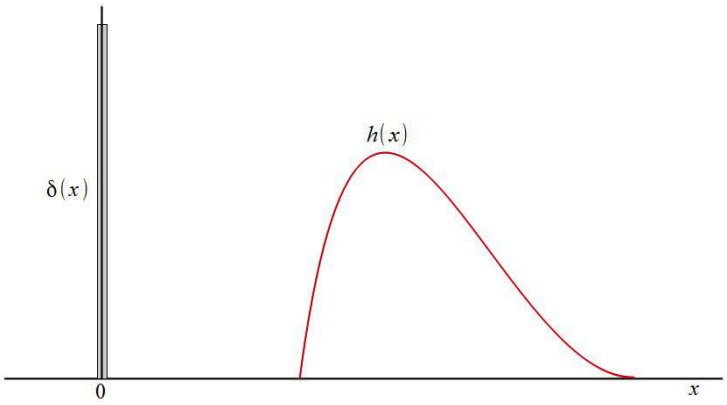

Figure A.2.2: The Impulse Function $\delta(x)$ and Response Function $h(x)$.

We now consider applying two different impulses, with magnitudes f_0 and f_1. From linear response theory we expect these to have responses proportional to their magnitudes, as in

$$\begin{aligned} h_0(x) &= f_0 h(x), \\ h_1(x) &= f_1 h(x). \end{aligned} \tag{A.2.9}$$

However we now propose that the second impulse is applied a short displacement, u_1, after the first. It has the same response but displaced along the x-coordinate by the distance u_1. If it happens that the two responses $h_0(x)$ and $h_1(x)$ overlap, they will produce a combined response as in figure A.2.3.

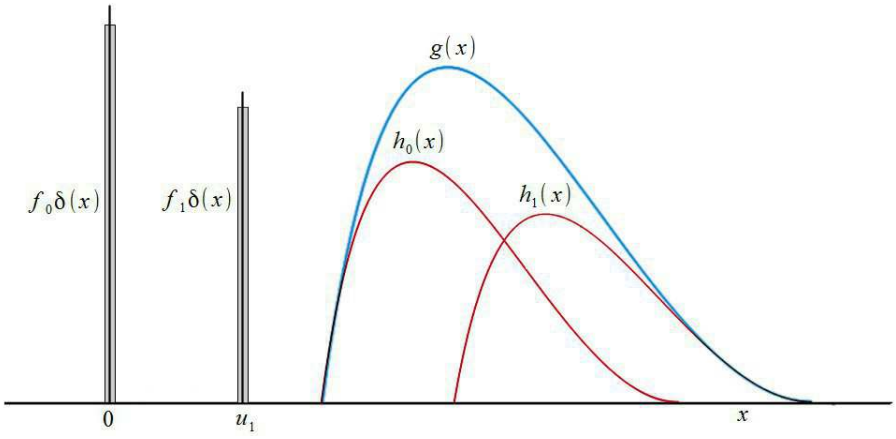

Figure A.2.3: The Combined Response of Two Impulses

Mathematically, the combined response, $g(x)$, shown in figure A.2.3 can be written as

$$g(x) = h_0(x) + h_1(x) = f_0 h(x) + f_1 h(x - u_1). \tag{A.2.10}$$

Generalising (A.2.10) to a large number of impulses, N, we write

$$g(x)=\sum_{i=0}^{N-1} f_i h(x-u_i), \tag{A.2.11}$$

in which we set $u_0=0.$ Now, if the factors f_i are obtained by sampling a continuous function, as in (A.2.8) we may write

$$g(x)=\int_{-\infty}^{\infty}\sum_{i=0}^{N-1} f(u)\delta(u-u_i)h(x-u_i)\,du. \tag{A.2.12}$$

We now further assume that the number of samples, $N,$ goes to infinity. In which case the sum in (A.2.12) can be replaced by an integral and u_i by a continuous variable, $w,$ to give

$$g(x)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(u)\delta(u-w)h(x-w)\,du\,dw. \tag{A.2.13}$$

We now integrate over the variable $w,$ which according to (A.2.6) gives

$$g(x)=\int_{-\infty}^{\infty} f(u)h(x-u)\,du. \tag{A.2.14}$$

The result is a convolution integral, which describes the response of the system, not to an impulse function, but to a continuous input function $f(x).$ For this reason the response function is sometimes called the *kernel function*, since it is at the heart of the system response to a range of input functions. It is also a *Green's function*, which can be used to solve differential equations in classical and quantum physics. (See Appendix 4).

Appendix 3. The Kronecker Delta

The Kronecker delta, δ_{ij}, is a simple but effective tool for selecting indexed variables from a vector (i.e. list) or tensor (i.e. array) components from a complete set. It is defined as follows

$$\delta_{ij}=\begin{cases}1 & if \quad i=j,\\ 0 & if \quad i\neq j.\end{cases} \tag{A.3.1}$$

Its use a great deal in this book to permit selection of particular combinations of indexed components from a general formula involving all possible components. It appears as a product of δ_{ij} with the components to be selected, as in the following, which selects the diagonal components from a matrix $\boldsymbol{A}$:

$$\sum_{i=1}^{N}\sum_{j=1}^{N}\delta_{ij}A_{ij}=\sum_{i=1}^{N}A_{ii}. \tag{A.3.2}$$

The off-diagonal components can be extracted using the form

$$\sum_{i=1}^{N}\sum_{j=1}^{N}(1-\delta_{ij})A_{ij}=\sum_{i=1}^{N}\sum_{j=1}^{N}{}'A_{ij}=\sum_{i=1}^{N}\sum_{j\neq i}^{N}A_{ij}. \tag{A.3.3}$$

Specific elements of a tensor can be extracted in the following manner.

$$\sum_{i=1}^{N}\delta_{in}v_i=v_n, \qquad \text{(Vector)} \tag{A.3.4}$$

$$\sum_{i=1}^{N}\sum_{j=1}^{N}\delta_{in}\delta_{jm}A_{ij}=A_{nm}, \qquad \text{(Matrix)} \tag{A.3.5}$$

$$\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{N}\delta_{il}\delta_{jm}\delta_{kn}A_{ijk}=A_{lmn}. \qquad \text{(Tensor)} \tag{A.3.6}$$

It is also possible to write expressions that are specific with respect to some indices, but general with regard to others, as in

$$\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{N}\delta_{il}\delta_{jm}A_{ijk}=\sum_{k=1}^{N}A_{lmk}. \tag{A.3.7}$$

The use of the Kronecker delta thus enables us to write general equations with specific

outcomes.

The Kronecker delta is often used in conjunction with the *Einstein Convention*, which applies to sums of the kind appearing in the above equations. The convention is that *repeated indices imply a summation*. So, for example the equations (A.3.4) to (A.3.7) can be written as

$$\delta_{in} v_i = v_n, \quad \text{implied sum over } i, \tag{A.3.8}$$

$$\delta_{in}\delta_{jm} A_{ij} = A_{nm}, \quad \text{implied sums over } i \text{ and } j, \tag{A.3.9}$$

$$\delta_{il}\delta_{jm}\delta_{kn} A_{ijk} = A_{lmn}, \quad \text{implied sums over } i,\ j \text{ and } k, \tag{A.3.10}$$

$$\sum_{k=1}^{N} \delta_{il}\delta_{jm} A_{ijk} = \sum_{k=1}^{N} A_{lmk}. \quad \text{implied sums over } i \text{ and } j. \tag{A.3.11}$$

The Kronecker delta allows complex formulae involving tensors to be written with great economy, particularly when allied with the Einstein Convention.

In chapter 5, where the inter-molecular forces are derived, particular use is made of the Kronecker delta. It may be useful to work through an example to see how it is employed. Let us work through the case for a three body potential, similar to that presented in equation (5.69). We write the system configuration energy as

$$\Phi(\vec{r}^{\,N}) = \sum_{i=1}^{N-2}\sum_{j>i}^{N-1}\sum_{k>j}^{N} \phi_{ijk}^{3\text{bdy}}(\vec{r}_i, \vec{r}_j, \vec{r}_k), \tag{A.3.12}$$

This form is slightly more general than that given in 5.3.2 in that $\phi_{ijk}^{3\text{bdy}}$ is dependent on nine variables: the components of the vectors $\vec{r}_i, \vec{r}_j, \vec{r}_k$. The force on an arbitrary atom n arising from three body terms is

$$\vec{f}_n = -\frac{\partial}{\partial \vec{r}_n}\Phi(\vec{r}^{\,N}) = -\sum_{i=1}^{N-2}\sum_{j>i}^{N-1}\sum_{k>j}^{N} \frac{\partial}{\partial \vec{r}_n}\phi_{ijk}^{3\text{bdy}}(\vec{r}_i, \vec{r}_j, \vec{r}_k), \tag{A.3.13}$$

or, in terms of individual components of the force:

$$f_n^{\alpha} = -\frac{\partial}{\partial r_n^{\alpha}}\Phi(\vec{r}^{\,N}) = -\sum_{i=1}^{N-2}\sum_{j>i}^{N-1}\sum_{k>j}^{N} \frac{\partial}{\partial r_n^{\alpha}}\phi_{ijk}^{3\text{bdy}}(\vec{r}_i, \vec{r}_j, \vec{r}_k). \tag{A.3.14}$$

Now, bearing in mind that $\phi_{ijk}^{3\text{bdy}}$ is a function of nine components i.e.

$$\Phi_{ijk}^{3\text{bdy}}(\vec{r}_i, \vec{r}_j, \vec{r}_k) \equiv \Phi_{ijk}^{3\text{bdy}}(r_i^x, r_i^y, r_i^z, r_j^x, r_j^y, r_j^z, r_k^x, r_k^y, r_k^z), \tag{A.3.15}$$

we can expand the differential furthest right of (A.3.14) to give

$$\frac{\partial}{\partial r_n^\alpha}\phi_{ijk}^{3\text{bdy}}(\vec{r}_i,\vec{r}_j,\vec{r}_k)=\sum_{\beta=1}^{3}\left(\frac{\partial\phi_{ijk}^{3\text{bdy}}}{\partial r_i^\beta}\frac{\partial r_i^\beta}{\partial r_n^\alpha}+\frac{\partial\phi_{ijk}^{3\text{bdy}}}{\partial r_j^\beta}\frac{\partial r_j^\beta}{\partial r_n^\alpha}+\frac{\partial\phi_{ijk}^{3\text{bdy}}}{\partial r_k^\beta}\frac{\partial r_k^\beta}{\partial r_n^\alpha}\right), \tag{A.3.16}$$

in which the sum over β identifies the components of each vector, and we have used the chain rule to obtain the full derivative. Regarding the derivatives with respect to r_n^α it is immediately obvious that these derivatives must be zero unless $\alpha=\beta$ and $n=i$, or $n=j$, or $n=k$, as appropriate. If these conditions are satisfied, the derivative is unity and otherwise zero. The Kronecker delta can be used to express this as follows.

$$\frac{\partial}{\partial r_n^\alpha}\phi_{ijk}^{3\text{bdy}}(\vec{r}_i,\vec{r}_j,\vec{r}_k)=\sum_{\beta=1}^{3}\delta_{\alpha\beta}\left(\frac{\partial\phi_{ijk}^{3\text{bdy}}}{\partial r_i^\beta}\delta_{in}+\frac{\partial\phi_{ijk}^{3\text{bdy}}}{\partial r_j^\beta}\delta_{jn}+\frac{\partial\phi_{ijk}^{3\text{bdy}}}{\partial r_k^\beta}\delta_{kn}\right), \tag{A.3.17}$$

or more simply as

$$\frac{\partial}{\partial r_n^\alpha}\phi_{ijk}^{3\text{bdy}}(\vec{r}_i,\vec{r}_j,\vec{r}_k)=\left(\frac{\partial\phi_{ijk}^{3\text{bdy}}}{\partial r_i^\alpha}\delta_{in}+\frac{\partial\phi_{ijk}^{3\text{bdy}}}{\partial r_j^\alpha}\delta_{jn}+\frac{\partial\phi_{ijk}^{3\text{bdy}}}{\partial r_k^\alpha}\delta_{kn}\right). \tag{A.3.18}$$

Inserting this result back into (A.3.14) gives

$$f_n^\alpha=-\sum_{i=1}^{N-2}\sum_{j>i}^{N-1}\sum_{k>j}^{N}\left(\frac{\partial\phi_{ijk}^{3\text{bdy}}}{\partial r_i^\alpha}\delta_{in}+\frac{\partial\phi_{ijk}^{3\text{bdy}}}{\partial r_j^\alpha}\delta_{jn}+\frac{\partial\phi_{ijk}^{3\text{bdy}}}{\partial r_k^\alpha}\delta_{kn}\right). \tag{A.3.19}$$

We can now separate out three sums:

$$f_n^\alpha=-\sum_{j>n}^{N-1}\sum_{k>j}^{N}\frac{\partial\phi_{njk}^{3\text{bdy}}}{\partial r_n^\alpha}-\sum_{i=1}^{n-1}\sum_{k>n}^{N}\frac{\partial\phi_{ink}^{3\text{bdy}}}{\partial r_n^\alpha}-\sum_{i=1}^{n-2}\sum_{j>i}^{n-1}\frac{\partial\phi_{ijn}^{3\text{bdy}}}{\partial r_n^\alpha}, \tag{A.3.20}$$

where the Kronecker deltas have fixed one of the i, j, k indices in each sum. (Note how this affects the sum limits in each case.) If, as is often the case, $\phi_{i,j,k}$ is unaffected by a permutation of the indices (as, for example when all three atoms are of the same type), then the sums in (A.3.20) can be recombined to give the formula:

$$f_n^\alpha=-\sum_{i=1}^{N-1}{}'\sum_{j>i}^{N}{}'\frac{\partial\phi_{nij}^{3\text{bdy}}}{\partial r_n^\alpha}. \tag{A.3.21}$$

The dash, $(')$, against the summation symbols in (A.3.21) indicate that neither i nor j can equal n.

In general, throughout this book, the Kronecker delta is used in this way, which is to identify specific terms within general formulae so the latter may be collapsed down easily into a specific form.

Appendix 4. The Green's Function Propagator

The Green's function propagator represents a formal solution of the time dependent Schroedinger equation:

$$\hat{H}\,\Psi(x,t)=i\hbar\frac{\partial\Psi(x,t)}{\partial t}, \tag{A.4.1}$$

in which we assume that the Hamiltonian operator $\hat{H}$ is *not* an explicit function of time and that the solution is known at time zero i.e. $\Psi(x)=\Psi(x,0)$ at $t=0.$ (We also assume here that the problem is one-dimensional). Since $\hat{H}$ does not contain the time explicitly (A.4.1) has solutions of the form

$$\phi_n(x,t)=\phi_n(x)\exp(-i\hbar^{-1}E_n t), \tag{A.4.2}$$

where $\phi_n(x)$ is an eigenfunction of the operator $\hat{H}$ and E_n is the corresponding eigenvalue i.e.

$$\hat{H}\,\phi_n(x)=E_n\phi_n(x). \tag{A.4.3}$$

The set of functions $\phi(x)$ are orthogonal, so the initial wavefunction $\Psi(x,0)$ may be expanded as a linear combination of eigenfunctions;

$$\Psi(x,0)=\sum_n a_n\phi_n(x), \tag{A.4.4}$$

in which the coefficients a_n are evaluated from the integrals

$$a_n=\int\Psi(x,0)\phi_n^*(x)\,dx, \tag{A.4.5}$$

(which is proved by multiplying both sides of (A.4.4) by $\phi_n^*(x)dx$ and integrating over $x.$ Orthogonality removes all but the $n'th$ term of the sum.)

Exploiting the relations (A.4.2) to (A.4.5), it is apparent that an acceptable solution of equation (A.4.1) would have the form

$$\Psi(x,t)=\sum_n a_n\phi_n(x)\exp(-i\hbar^{-1}E_n t), \tag{A.4.6}$$

which is easily seen to satisfy both equation (A.4.1) and the initial condition $\Psi(x)=\Psi(x,0)$ at $t=0.$

Substituting (A.4.5) into (A.4.6) gives the equation

$$\Psi(x,t)=\int \sum_n \phi_n^*(\xi)\phi_n(x)\exp(-i\hbar^{-1}E_n t)\,\Psi(\xi,0)\,d\xi, \tag{A.4.7}$$

which may be more economically written as

$$\Psi(x,t)=\int G(\xi,0;x,t)\,\Psi(\xi,0)\,d\xi \tag{A.4.8}$$

with

$$G(\xi,0;x,t)=\sum_n \phi_n^*(\xi)\phi_n(x)\exp(-i\hbar^{-1}E_n t) \tag{A.4.9}$$

which is recognisable as the Green's function described in chapter 12, equation (12.4). It is apparent from its role in the equation (A.4.8) that the Green's function represents a formal solution to (A.4.1), since everything that may be deduced about the wavefunction $\Psi(x,t)$ can be obtained from it. Extensions to problems of higher dimensionality may also be derived in this way.

Bibliography

[1] J. P. Hansen and I. R. MacDonald, *Theory of Simple Liquids*. New York: Academic Press, 1986.

[2] D. Chandler and P. G. Wolynes, "Exploiting the isomorphism between quantum theory and classical statistical mechanics of polyatomic fluids," *J Chem Phys*, vol. 74, p. 4078, 1981.

[3] L. Verlet, "Computer `experiments' on classical fluids. I. Thermodynamical properties of Lennard-Jones molecules," *Phys Rev*, vol. 159, p. 98, 1967.

[4] A. Rahman, "Correlation in the motions of atoms in liquid argon," *Phys Rev*, vol. 136, p. A405, 1964.

[5] H. Goldstein, *Classical Mechanics*, Second. Addison Wesley, 1980.

[6] J. P. Ryckaert, G. Ciccotti, and H. J. C. Berendsen, "Numerical integration of the Cartesian equations of motion of a system with constraints: molecular dynamics of n-alkanes," *J Comput Phys*, vol. 23, p. 327, 1977.

[7] D. J. Evans and S. Murad, "Singularity free algorithm for molecular dynamics simulations of rigid polyatomics," *Molec Phys*, vol. 34, p. 327, 1977.

[8] L. van Hove, "Correlations in Space and Time and Born Approximation Scattering in Systems of Interacting Particles," *Phys. Rev.*, vol. 95, p. 249, 1954.

[9] H. Flyvbjerg and H. G. Petersen, "Error estimates on averages of correlated data," *J Chem Phys*, vol. 91, p. 461, 1989.

[10] T. Cagin and B. M. Pettitt, "Grand molecular dynamics: a method for open systems," *Mol. Simul.*, vol. 6, pp. 5–26, 1991.

[11] C. H. Bennet, "Efficient estimation of free energy differences from Monte Carlo data," *J Comput Phys*, vol. 22, p. 245, 1976.

[12] D. J. Evans and G. P. Morriss, "Non-Newtonian molecular dynamics," *Comput. Phys. Rep.*, vol. 1, p. 297, 1984.

[13] B. L. Bhatnagar, E. R. Gross, and H. Krook, "A model for collision processes in gases. I Small amplitude processes in charged and neutral one-component systems.," *Phys. Rev.*, vol. 99, pp. 511–525, 1954.

[14] R. D. Groot and P. B. Warren, "Dissipative Particle Dynamics: Bridging the gap between atomistic and mesoscopic simulation," *J. Chem. Phys.*, vol. 107, pp. 4423–4435, 1997.

[15] D. Chandler, *Introduction to Modern Statistical Mechanics*, First. Oxford University Press, 1987.

[16] R. W. Hockney and J. W. Eastwood, *Computer Simulation Using Particles*. McGraw-Hill International, 1981.

[17] W. C. Swope, H. C. Andersen, P. H. Berens, and K. R. Wilson, "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, pp. 637–639, 1982.

[18] C. W. Gear, "The numerical integration of ordinary differential equations of various orders." Argonne National Laboratory, 1966.

[19] M. E. Tuckerman, B. J. Berne, and A. Rossi, "Molecular dynamics algorithms for multiple timescales: Systems with disparate masses," *J Chem Phys*, vol. 94, p. 1465, 1990.

[20] M. E. Tuckerman, B. J. Berne, and G. J. Martyna, "Reversible multiple time scale

molecular dynamics," *J. Chem. Phys.*, vol. 97, pp. 1990–2001.
[21] J. P. Ryckaert and A. Bellemans, "Molecular-dynamics of liquid normal-butane near its boiling-point," *Chem Phys Lett*, vol. 30, p. 123, 1975.
[22] H. C. Andersen, "Rattle: a velocity version of the SHAKE algorithm for molecular dynamics calculations," *J Comput Phys*, vol. 52, p. 24, 1983.
[23] R. Sonnenschein, "An improved algorithm for molecular dynamics simulation of rigid molecules.," *J. Compuational Phys.*, vol. 59, pp. 347–350, 1985.
[24] J. G. Powles, W. A. B. Evans, E. McGrath, K. E. Gubbins, and S. Murad, "A computer simulation for a simple model of liquid hydrogen chloride," *Mol. Phys.*, vol. 38, pp. 893–908, 1979.
[25] T. F. Miller, M. Eleftheriou, P. Pattnaik, A. Ndirango, D. Newns, and G. J. Martyna, "Symplectic quaternion scheme for biophysical molecular dynamics," *J Chem Phys*, vol. 116, p. 8649, 2002.
[26] J. G. Gay and B. J. Berne, "Modification of the overlap potential to mimic a linear site-site potential," *J. Chem. Phys.*, vol. 74, pp. 3316–3319.
[27] J. A. Purton and W. Smith, "Dissipative particle dynamics of non-spherical particles using a Gaussian density model," *Mol. Simul.*, vol. 36, pp. 796–800, 2010.
[28] T. Forester and W. Smith, "Shake, Rattle and Roll: Efficient constraint algorithms for linked rigid bodies," *J Comput. Chem.*, vol. 19, p. 102, 1998.
[29] 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, p. 3684, 1984.
[30] S. C. Harvey, R. K. Z. Tan, and T. E. Cheatham, "The flying ice cube: Velocity rescaling in molecular dynamics leads to violation of energy equipartition.," *J Comp Chem*, vol. 19, pp. 726–740, 1998.
[31] H. C. Andersen, "Molecular dynamics simulations at constant pressure and/or temperature," *J Chem Phys*, vol. 72, p. 2384, 1979.
[32] S. Nosé, "A unified formulation of the constant temperature molecular dynamics methods," *J Chem Phys*, vol. 81, p. 511, 1984.
[33] W. G. Hoover, "Canonical dynamics: Equilibrium phase-space distributions," *Phys Rev*, vol. A31, p. 1695, 1985.
[34] S. Melchionna, G. Ciccotti, and B. L. Holian, "Hoover NPT dynamics for systems varying in shape and size," *Molec Phys*, vol. 78, p. 533, 1993.
[35] G. J. Martyna, D. J. Tobias, and M. L. Klein, "Constant Pressure Molecular Dynamics Algorithms," *J. Chem. Phys.*, vol. 101, pp. 4177–4189, 1994.
[36] G. J. Martyna, M. L. Klein, and M. E. Tuckerman, "Nosé-Hoover chains: The canonical ensemble via continuous dynamics," *J. Chem. Phys.*, vol. 97, pp. 2635–2645, 1992.
[37] G. J. Martyna, M. E. Tuckerman, D. J. Tobias, and M. L. Klein, "Explicit reversible integrators for extended system dynamics," *Molec Phys*, vol. 87, p. 1117, 1996.
[38] B. Leimkuhler, E. Noorizadeh, and F. Thiel, "A Gentle Thermostat for Molecular Dynamics," *J. Stat. Phys.*, vol. 139, pp. 261–277, 2009.
[39] A. L. Rohl, K. Wright, and J. D. Gale, "Evidence from surface phonons for the (2 x 1) reconstruction of the (10-14) surface of calcite from computer simulation," *Amer Mineral.*, vol. 88, p. 921, 2003.
[40] J. D. Weeks, D. Chandler, and H. C. Anderson, "Roles of repulsive forces in determining the equilibrium structure of simple liquids," *J Chem Phys*, vol. 54, p. 5237, 1971.
[41] B. M. Axilrod and E. Teller, "Interaction of the van der Waals type between three

atoms," *J. Chem. Phys.*, vol. 11, p. 299, 1943.
[42] J. Tersoff, "Modelling solid state chemistry: Interaction potentials for multicomponent systems," *Phys Rev B*, vol. 39, p. 5566, 1989.
[43] M. S. Daw and M. I. Baskes, "Embedded-atom method: Derivation and application to impurities, surfaces, and other defects in metals," *Phys Rev B*, vol. 29, p. 6443, 1984.
[44] S. M. Foiles, M. I. Baskes, and M. S. Daw, "Embedded-atom-method functions for the fcc metals Cu, Ag, Au, Ni, Pd, Pt, and their alloys," *Chem Phys Lett*, vol. 33, p. 7983, 1986.
[45] M. W. Finnis and J. E. Sinclair, "A simple empirical N body potential for transition metals," *Philos Mag A*, vol. 50, p. 45, 1984.
[46] R. A. Johnson, "Alloy models with the embedded-atom method," *Phys Rev B*, vol. 39, p. 12556, 1989.
[47] H. Rafii-Tabar and A. P. Sutton, "Long range Finnis-Sinclair potentials for FCC metallic alloys," *Philos Mag Lett*, vol. 63, p. 217, 1991.
[48] A. P. Sutton and J. Chen, "Long range Finnis-Sinclair potentials," *Philos Mag Lett*, vol. 61, p. 139, 1990.
[49] M. P. Allen and G. Germano, "Expressions for forces and torques in molecular simulations using rigid bodies," *Mol. Phys.*, vol. 104, pp. 3225–3235, 2006.
[50] L. V. Woodcock and K. Singer, "Thermodynamic and structural properties of liquid ionic salts obtained by Monte Carlo Computation," *Trans Faraday Soc*, vol. 67, pp. 12–30, 1971.
[51] P. P. Ewald, "Die Berechnung optische und electrostatische Gitterpotentiale," *Ann Phys*, vol. 64, p. 253, 1921.
[52] U. Essmann, L. Perera, M. L. Berkowitz, T. Darden, H. Lee, and L. G. Pedersen, "A smooth particle mesh Ewald method," *J Chem Phys*, vol. 103, p. 8577, 1995.
[53] I. J. Bush, I. T. Todorov, and W. Smith, "A DAFT DL_POLY distributed memory adaptation of the Smoothed Particle Mesh Ewald method," *Comput. Phys. Commun.*, vol. 175, p. 323, 2006.
[54] L. Greengard and V. Rokhlin, "A fast algorithm for particle simulations," *J. Comput. Phys.*, vol. 73, pp. 325–348, 1987.
[55] C. J. Fennell and J. D. Gezelter, "Is the Ewald summation still necessary? Pairwise alternatives to the accepted standard for long-ranged electrostatics," *J Chem Phys*, vol. 124, p. 234104, 2006.
[56] D. Wolf, P. Keblinski, S. R. Phillpot, and J. Eggebrecht, "Exact method for the simulation of Coulombic systems for spherically truncated, pairwise 1/r summation.," *J Chem Phys*, vol. 110, pp. 8254–8282, 1999.
[57] M. Neumann, "The dielectric constant of water. Computer simulations with the MCY potential.," *J Chem Phys*, vol. 82, p. 5663, 1985.
[58] S. De Leeuw, J. W. Perram, and E. R. Smith, "Simulation of electrostatic systems in periodic boundary conditions. I. Lattice sums and dielctric constant.," *Proc Roy Soc Lond.*, vol. A373, pp. 27–56, 1980.
[59] I. J. Bush, "The Daresbury Advanced Fourier Transform (DAFT)," *Daresbury Lab.*, 2000.
[60] D. E. Parry, "The electrostatic potential in the surface region of an ionic crystal," *Surf. Sci.*, vol. 49, pp. 433–440, 1975.
[61] D. M. Heyes, "The surface potential of point charge and point dipole lattices," *Surf. Sci.*, vol. 110, p. L619, 1981.
[62] J. Hautman and M. L. Klein, "An Ewald summation method for planar surfaces and interfaces," *Molec Phys*, vol. 75, p. 379, 1992.

[63] S. L. Price, A. J. Stone, and M. Alderton, "Explicit formulas for the electrostatic energy, forces and torques between a pair of molecules of arbitrary symmetry," *Mol Phys*, vol. 52, pp. 987–1001, 1984.
[64] S. L. Price, M. Leslie, G. W. A. Welch, M. Hapgood, L. S. Price, P. G. Karamertzanis, and G. M. Day, "Modelling organic crystal structures using distributed multipole and polarizability-based model intermolecular potentials.Price, Sarah L. and Leslie, Maurice and Welch, Gareth W. A. and Habgood, Matthew and Price, Louise S. and Karamertzanis, Panagiotis G. and Day, Graeme M.," *Phys Chem Chem Phys*, vol. 12, pp. 8478–8490, 2010.
[65] B. G. Dick and A. W. Overhauser, "Theory of dielectric constants of alkali halide crystals," *Phys Rev B*, vol. 112, p. 90, 1958.
[66] D. Fincham and P. J. Mitchell, "Shell model simulations by adiabatic dynamics," *J Phys Condens Matter*, vol. 5, p. 1031, 1993.
[67] P. J. D. Lindan and M. J. Gillan, "Shell-model molecular-dynamics simulation of superionic conduction in CAF$_2,$," *J Phys Condens Matter*, vol. 5, p. 1019, 1993.
[68] M. P. Allen and D. J. Tildesley, *Computer Simulation of Liquids*. Oxford: Clarendon Press, 1989.
[69] D. Frenkel and B. Smit, *Understanding Molecular Simulation*, Second. Academic Press, 2002.
[70] B. Vessal, M. Amini, M. Leslie, and C. R. A. Catlow, "Potentials for molecular dynamics simulation of silicate glasses," *Mol. Simul.*, vol. 5, p. 1, 1990.
[71] S. L. Mayo, B. D. Olafson, and W. A. Goddard, "DREIDING: A generic force field for molecular simulations," *J Phys Chem*, vol. 94, p. 8897, 1990.
[72] B. R. Brooks, R. E. Bruccoleri, B. D. Olafson, D. J. States, Swaminathan, S., and M. Karplus, "CHARMM: A Program for Macromolecular Energy, Minimization, and Dynamics Calculations," *J Comp Chem*, vol. 4, p. 187, 1983.
[73] F. Cleri and F. Rosato, "Tight-binding potentials for transition metals and alloys," *Phys Rev B*, vol. 48, p. 22, 1993.
[74] A. W. Lees and S. F. Edwards, "The computer study of transport processes under extreme conditions," *J. Phys. C*, vol. 5, no. 15, p. 1921, 1972.
[75] W. G. Hoover and C. G. Hoover, "Nonequilibrium Molecular Dynamics," *Condens. Matter Phys.*, vol. 8, no. 2, pp. 247–260, 2005.
[76] D. J. Evans and G. P. Morriss, *Statistical mechanics of Nonequilibrium Liquids*, 2nd ed., 1 vols. ANU E Press, 2007.
[77] W. Smith and D. Fincham, "The Ewald Sum in Truncated Octahedral and Rhombic Dodecahedral Boundary Conditions," *Mol. Simul.*, vol. 10, p. 67, 1993.
[78] P. Smith, R. M. Lynden-Bell, and W. Smith, "Surfactant structure around DNA in aqueous solution," *Phys. Chem. Chem. Phys.*, vol. 2, pp. 1305–1310, 2000.
[79] S. Brode and R. Ahlrichs, "An optimised MD program for a vector computer Cyber 205," *Comput Phys Commun*, vol. 42, p. 41, 1986.
[80] E. O. Brigham, *The Fast Fourier Transform and its Applications*. Prentice Hall International, 1998.
[81] R. H. Swendsen and J. S. Wang, "Replica Monte Carlo simulation of spin glasses," *Phys. Rev. Lett.*, vol. 57, pp. 2607–2609, 1986.
[82] A. F. Voter, "Parallel replica method for dynamics of infrequent events," *Phys Rev B*, vol. 57, p. R13 985, 1998.
[83] A. R. C. Raine, D. Fincham, and W. Smith, "Systolic Loop Methods for Molecular Dynamics using Multiple Transputers," *Comput Phys Commun*, vol. 55, p. 13, 1989.

[84] J. E. Boillat, F. Bruge, and P. G. Kropf, "A dynamic load-balancing algorithm formolecular dynamics simulation on multi-processor systems," *J Comput Phys*, vol. 96, p. 1, 1991.
[85] L. V. Kale, M. Bhandarkar, and R. Brunner, "Load Balancing in Parallel Molecular Dynamics," in *Solving Irregularly Structured problems in Parallel*, vol. 1457, Springer, 1998, pp. 251–261.
[86] I. T. Todorov, I. J. Bush, and W. Smith, "The Need for Parallel I/O in Classical Molecular Dynamics." CUG2008 Meeting proceedings.
[87] I. T. Todorov, I. J. Bush, and A. R. Porter, "DL POLY 3 I/O: Analysis, Alternatives and Future Strategies," in *Proceedings of Performance Scientific Computing (International Networking for Young Scientists (February 2008, Lithuania)*, vol. 27, Lithuania: Springer, 2009.
[88] W. Smith, "A replicated data molecular dynamics strategy for the parallel Ewald sum," *Comput Phys Commun*, vol. 67, p. 392, 1992.
[89] S. Chen and G. D. Doolen, "Lattice-Boltzmann method for fluid flows.," *Ann Rev Fluid Mech*, vol. 30, pp. 329–364, 1998.
[90] G. Henkelman and H. Jonsson, "Improved tangent estimate in the nudged elastic band method for finding minimum energy paths and saddle points," *J Chem Phys*, vol. 113, p. 9978, 2000.
[91] A. F. Voter, "A method for accelerating the molecular dynamics simulation of infrequent events," *J Chem Phys*, vol. 106, p. 4665, 1997.
[92] A. A. Frost and R. G. Pearson, *Kinetics and Mechanism*, 2nd. ed. New York and London: John Wiley and Sons Inc., 1961.
[93] D. Hamelberg, J. Mongan, and J. A. McCammon, "Accelerated molecular dynamics: A promising and efficient simulation method for biomolecules," *J Chem Phys*, vol. 120, p. 11919, 2004.
[94] M. R. Sorensen and A. F. Voter, "Temperature accelerated dynamics for simulation of infrequent events," *J. Chem. Phys.*, vol. 112, pp. 9599–9607, 2000.
[95] A. Laio and M. Parrinello, "Escaping free energy minima," *Proc Natl Acad Sci*, vol. 99, p. 12562, 2002.
[96] A. Laio, A. Rordiguez-Fortea, F. L. Gervasio, M. Ceccarelli, and M. Parrinello, "Assessing the accuracy of metadynamics," *J Phys Chem B*, vol. 109, p. 6714, 2005.
[97] D. Quigley and P. M. Rodger, "A metadynamics-based approach to sampling crystallisation events," *Mol. Simul.*, vol. 35, p. 613, 2009.
[98] B. Peters and B. L. Trout, "Obtaining reaction coordinates by likelihood maximization," *J Chem Phys*, vol. 125, p. 054108, 2006.
[99] D. Donadio, P. Raiteri, and M. Parrinello, "Topological defects and bulk melting of hexagonal ice," *J Phys Chem B*, vol. 109, p. 5421, 2005.
[100] P. J. Steinhardt, D. R. Nelson, and M. Ronchetti, "Bond orientational order in liquids and glasses.," *Phys Rev B*, vol. 28, p. 784, 1983.
[101] P. L. Chau and A. J. Hardwick, "A new order parameter for tetrahedral configurations," *Molec Phys*, vol. 93, p. 511, 1998.
[102] R. P. Feynmann and A. R. Hibbs, *Quantum Mechanics and Path Integrals*. McGraw-Hill, 1965.
[103] K. Singer and W. Smith, "The Classical and Quantum Mechanical Free Energy of Solid (Lennard-Jones) Argon," *Chem Phys Lett*, vol. 140, p. 406, 1987.
[104] K. Singer and W. Smith, "Path Integral Simulations of Condensed Phase Lennard-Jones Systems," *Molec Phys*, vol. 64, p. 1215, 1988.

[105]M. Parrinello and A. Rahman, "Study of an F Center in Molten KCl," *J. Chem. Phys.*, vol. 80, pp. 860–867, 1985.
[106]M. Sprik, R. W. Impey, and M. L. Klein, "Study of electron solvation in liquid ammonia using quantum path integral Monte Carlo calculations," *J. Chem. Phys.*, vol. 83, p. 5802, 1985.
[107]K. Singer and W. Smith, "Semi-Classical Many-Particle Dynamics with Gaussian Wave Packets," in *Proceedings of the International Meeting on the Dynamics of Wave Packets in Molecular and Nuclear Physics, Belgium 2-4 July 1985.*, 1985.
[108]M. F. Herman, E. J. Bruskin, and B. J. Berne, "On Path Integral Monte Carlo Simulations," *J. Chem. Phys.*, vol. 76, p. 6155, 1982.
[109]K. Singer and W. Smith, "Semi-Classical Many Particle Dynamics with Spherical Gaussian Wave Packets," *Molec Phys*, vol. 57, p. 761, 1986.
[110]M. E. Tuckerman, B. J. Berne, G. J. Martyna, and M. L. Klein, "Efficient molecular dynamics and hybrid Monte Carlo algorithms for path integrals," *J. Chem. Phys.*, vol. 99, pp. 2796–2808, 1993.

Alphabetical Index