CHAPTER SIX

Computer Simulation Methods

6.1 Introduction

Energy minimisation generates individual minimum energy configurations of a system. In some cases the information provided by energy minimisation can be sufficient to predict accurately the properties of a system. If all minimum configurations on an energy surface can be identified then statistical mechanical formulae can be used to derive a partition function from which thermodynamic properties can be calculated. However, this is possible only for relatively small molecules or small molecular assemblies in the gas phase The molecular modeller more often wants to understand and to predict the properties of liquids, solutions and solids, to study complex processes such as the adsorption of molecules onto surfaces and into solids and to investigate the behaviour of macromolecules which have many closely separated minima. In such systems the experimental measurements are made on macroscopic samples that contain extremely large numbers of atoms or molecules, with an enormous number of minima on their energy surfaces. A full quantification of the energy surfaces of such systems is not possible, nor is it ever likely to be. Computer simulation methods enable us to study such systems and predict their properties through the use of techniques that consider small replications of the macroscopic system with manageable numbers of atoms or molecules. A simulation generates representative configurations of these small replications in such a way that accurate values of structural and thermodynamic properties can be obtained with a feasible amount of computation. Simulation techniques also enable the time-dependent behaviour of atomic and molecular systems to be determined, providing a detailed picture of the way in which a system changes from one conformation or configuration to another. Simulation techniques are also widely used in some experimental procedures, such as the determination of protein structures from X-ray crystallography.

In this chapter we shall discuss some of the general principles involved in the two most common simulation techniques used in molecular modelling: the molecular dynamics and the Monte Carlo methods. We shall also discuss several concepts that are common to both of these methods. A more detailed discussion of the two simulation methods can be found in Chapters 7 and 8.

6.1.1 Time Averages, Ensemble Averages and Some Historical Background

Suppose we wish to determine experimentally the value of a property of a system such as the pressure or the heat capacity. In general, such properties will depend upon the positions and

momenta of the N particles that comprise the system The instantaneous value of the property A can thus be written as $A(\mathbf{p}^N(t)\ \mathbf{r}^N(t))$, where $\mathbf{p}^N(t)$ and $\mathbf{r}^N(t)$ represent the N momenta and positions respectively at time t (i.e. $A(\mathbf{p}^N(t),\mathbf{r}^N(t)) \equiv A(p_{1x},p_{1y},p_{1z},p_{2x},\cdot\ ,$ $x_1,y_1,z_1,x_2,\ldots,t)$ where p_{1x} is the momentum of particle 1 in the x direction and x_1 is its x coordinate). Over time, the instantaneous value of the property A fluctuates as a result of interactions between the particles. The value that we measure experimentally is an average of A over the time of the measurement and is therefore known as a *time average* As the time over which the measurement is made increases to infinity, so the value of the following integral approaches the 'true' average value of the property:

$$A_{\text{ave}} = \lim_{\tau\to\infty}\frac{1}{\tau}\int_{t=0}^{\tau} A(\mathbf{p}^N(t),\mathbf{r}^N(t))\,dt \tag{6.1}$$

To calculate average values of the properties of the system, it would therefore appear to be necessary to simulate the dynamic behaviour of the system (i.e. to determine values of $A(\mathbf{p}^N(t),\mathbf{r}^N(t))$, based upon a model of the intra- and intermolecular interactions present). In principle, this is relatively straightforward to do. For any arrangement of the atoms in the system, the force acting on each atom due to interactions with other atoms can be calculated by differentiating the energy function. From the force on each atom it is possible to determine its acceleration via Newton's second law. Integration of the equations of motion should then yield a trajectory that describes how the positions, velocities and accelerations of the particles vary with time, and from which the average values of properties can be determined using the numerical equivalent of Equation (6.1). The difficulty is that for 'macroscopic' numbers of atoms or molecules (of the order of 10^{23}) it is not even feasible to determine an initial configuration of the system, let alone integrate the equations of motion and calculate a trajectory. Recognising this problem, Boltzmann and Gibbs developed statistical mechanics, in which a single system evolving in time is replaced by a large number of replications of the system that are considered simultaneously. The time average is then replaced by an *ensemble average*:

$$\langle A\rangle = \iint d\mathbf{p}^N\,d\mathbf{r}^N\,A(\mathbf{p}^N,\mathbf{r}^N)\rho(\mathbf{r}^N,\mathbf{r}^N) \tag{6.2}$$

The angle brackets $\langle\,\rangle$ indicate an ensemble average, or *expectation value*; that is, the average value of the property A over all replications of the ensemble generated by the simulation. Equation (6.2) is written as a double integral for convenience but in fact there should be $6N$ integral signs on the integral for the $6N$ positions and momenta of all the particles. $\rho(\mathbf{p}^N\mathbf{r}^N)$ is the *probability density* of the ensemble; that is, the probability of finding a configuration with momenta $\mathbf{p}^N$ and positions $\mathbf{r}^N$. The ensemble average of the property A is then determined by integrating over all possible configurations of the system. In accordance with the *ergodic hypothesis*, which is one of the fundamental axioms of statistical mechanics, the ensemble average is equal to the time average. Under conditions of constant number of particles, volume and temperature, the probability density is the familiar Boltzmann distribution:

$$\rho(\mathbf{p}^N\mathbf{r}^N) = \exp(-E(\mathbf{p}^N,\mathbf{r}^N)/k_{\text{B}}T)/Q \tag{6.3}$$

In Equation (6.3), $E(\mathbf{p}^N,\mathbf{r}^N)$ is the energy, Q is the partition function, k_{B} is Boltzmann's constant and T is the temperature. The partition function is more generally written in terms of the Hamiltonian, $\mathscr{H}$; for a system of N identical particles the partition function

for the canonical ensemble is as follows:

$$Q_{NVT} = \frac{1}{N!}\frac{1}{h^{3N}}\iint d\mathbf{p}^N\, d\mathbf{r}^N \exp\left[-\frac{\mathscr{H}(\mathbf{p}^N,\mathbf{r}^N)}{k_B T}\right] \quad (6\,4)$$

The canonical ensemble is the name given to an ensemble for constant temperature, number of particles and volume. For our purposes $\mathscr{H}$ can be considered the same as the total energy, $E(\mathbf{p}^N,\mathbf{r}^N)$, which equals the sum of the kinetic energy ($\mathscr{K}(\mathbf{p}^N)$) of the system, which depends upon the momenta of the particles, and the potential energy ($\mathscr{V}(\mathbf{r}^N)$), which depends upon the positions. The factor $N!$ arises from the indistinguishability of the particles and the factor $1/h^{3N}$ is required to ensure that the partition function is equal to the quantum mechanical result for a particle in a box. A short discussion of some of the key results of statistical mechanics is provided in Appendix 6.1 and further details can be found in standard textbooks.

The first computer simulations of fluids were performed in 1952 by Metropolis, Rosenbluth, Rosenbluth, Teller and Teller, who developed a scheme for sampling from the Boltzmann distribution to give ensemble averages. This gave rise to the Monte Carlo simulation method. Not long afterwards (in 1957) Alder recognised that it was, in fact, possible to integrate the equations of motion for a relatively small number of particles, and to mimic the behaviour of a real system using periodic boundary conditions. This led to the first molecular dynamics simulations of molecular systems.

6.1.2 A Brief Description of the Molecular Dynamics Method

Molecular dynamics calculates the 'real' dynamics of the system, from which time averages of properties can be calculated. Sets of atomic positions are derived in sequence by applying Newton's equations of motion. Molecular dynamics is a *deterministic* method, by which we mean that the state of the system at any future time can be predicted from its current state. The first molecular dynamics simulations were performed using very simple potentials such as the hard-sphere potential. The behaviour of the particles in this potential is similar to that of billiard or snooker balls. the particles move in straight lines at constant velocity between collisions. The collisions are perfectly elastic and occur when the separation between a pair of spheres equals the sum of their radii. After a collision, the new velocities of the colliding spheres are calculated using the principle of conservation of linear momentum. The hard-sphere model has provided many useful results but is obviously not ideal for simulating atomic or molecular systems. In potentials such as the Lennard-Jones potential the force between two atoms or molecules changes continuously with their separation. By contrast, in the hard-sphere model there is no force between particles until they collide. The continuous nature of the more realistic potentials requires the equations of motion to be integrated by breaking the calculation into a series of very short time steps (typically between 1 femtosecond and 10 femtoseconds; 10^{-15} s to 10^{-14} s). At each step, the forces on the atoms are computed and combined with the current positions and velocities to generate new positions and velocities a short time ahead The force acting on each atom is assumed to be constant during the time interval. The atoms are then moved to the new positions, an updated set of forces is computed, and so on. In this way a molecular dynamics simulation generates a

trajectory that describes how the dynamic variables change with time. Molecular dynamics simulations are typically run for tens or hundreds of picoseconds (a 100 ps simulation using a 1 fs time step requires 100 000 steps) Thermodynamic averages are obtained from molecular dynamics as time averages using numerical integration of Equation (6.2):

$$\langle A \rangle = \frac{1}{M} \sum_{i=1}^{M} A(\mathbf{p}^N, \mathbf{r}^N) \tag{6 5}$$

M is the number of time steps. Molecular dynamics is also extensively used to investigate the conformational properties of flexible molecules as will be discussed in Chapters 7 and 9.

6.1.3 The Basic Elements of the Monte Carlo Method

In a molecular dynamics simulation the successive configurations of the system are connected in time. This is not the case in a Monte Carlo simulation, where each configuration depends only upon its predecessor and not upon any other of the configurations previously visited. The Monte Carlo method generates configurations randomly and uses a special set of criteria to decide whether or not to accept each new configuration. These criteria ensure that the probability of obtaining a given configuration is equal to its Boltzmann factor, $\exp\{-\mathcal{V}(\mathbf{r}^N)/k_B T\}$, where $\mathcal{V}(\mathbf{r}^N)$ is calculated using the potential energy function. States with a low energy are thus generated with a higher probability than configurations with a higher energy. For each configuration that is accepted the values of the desired properties are calculated and at the end of the calculation the average of these properties is obtained by simply averaging over the number of values calculated, M:

$$\langle A \rangle = \frac{1}{M} \sum_{i=1}^{M} A(\mathbf{r}^N) \tag{6 6}$$

Most Monte Carlo simulations of molecular systems are more properly referred to as Metropolis Monte Carlo calculations after Metropolis and his colleagues, who reported the first such calculation. The distinction can be important because there are other ways in which an ensemble of configurations can be generated. As we shall see in Chapter 7, the Metropolis scheme is only one of a number of possibilities, though it is by far the most popular.

In a Monte Carlo simulation each new configuration of the system may be generated by randomly moving a single atom or molecule. In some cases new configurations may also be obtained by moving several atoms or molecules or by rotating about one or more bonds. The energy of the new configuration is then calculated using the potential energy function. If the energy of the new configuration is lower than the energy of its predecessor then the new configuration is accepted. If the energy of the new configuration is higher than the energy of its predecessor then the *Boltzmann factor* of the energy difference is calculated: $\exp[-(\mathcal{V}_{new}(\mathbf{r}^N) - \mathcal{V}_{old}(\mathbf{r}^N))/k_B T]$. A random number is then generated between 0 and 1 and compared with this Boltzmann factor. If the random number is higher than the Boltzmann factor then the move is rejected and the original configuration is retained for the next iteration; if the random number is lower then the move is accepted and the new

configuration becomes the next state. This procedure has the effect of permitting moves to states of higher energy. The smaller the uphill move (i.e. the smaller the value of $\mathscr{V}_{new}(\mathbf{r}^N) - \mathscr{V}_{old}(\mathbf{r}^N)$) the greater is the probability that the move will be accepted.

6.1.4 Differences Between the Molecular Dynamics and Monte Carlo Methods

The molecular dynamics and Monte Carlo simulation methods differ in a variety of ways. The most obvious difference is that molecular dynamics provides information about the time dependence of the properties of the system whereas there is no temporal relationship between successive Monte Carlo configurations. In a Monte Carlo simulation the outcome of each trial move depends only upon its immediate predecessor, whereas in molecular dynamics it is possible to predict the configuration of the system at any time in the future - or indeed at any time in the past. Molecular dynamics has a kinetic energy contribution to the total energy whereas in a Monte Carlo simulation the total energy is determined directly from the potential energy function. The two simulation methods also sample from different ensembles. Molecular dynamics is traditionally performed under conditions of constant number of particles (N), volume (V) and energy (E) (the microcanonical or constant NVE ensemble) whereas a traditional Monte Carlo simulation samples from the canonical ensemble (constant N, V and temperature, T). Both the molecular dynamics and Monte Carlo techniques can be modified to sample from other ensembles; for example, molecular dynamics can be adapted to simulate from the canonical ensemble. Two other ensembles are common:

isothermal–isobaric: fixed N, T, P (pressure)
grand canonical: fixed μ (chemical potential), V, T

In the canonical, microcanonical and isothermal–isobaric ensembles the number of particles is constant but in a grand canonical simulation the composition can change (i.e. the number of particles can increase or decrease). The equilibrium states of each of these ensembles are characterised as follows:

canonical ensemble: minimum Helmholtz free energy (A)
microcanonical ensemble: maximum entropy (S)
isothermal–isobaric ensemble: minimum Gibbs function (G)
grand canonical ensemble: maximum pressure × volume (PV)

6.2 Calculation of Simple Thermodynamic Properties

A wide variety of thermodynamic properties can be calculated from computer simulations; a comparison of experimental and calculated values for such properties is an important way in which the accuracy of the simulation and the underlying energy model can be quantified. Simulation methods also enable predictions to be made of the thermodynamic properties of systems for which there is no experimental data, or for which experimental data is difficult or impossible to obtain. Simulations can also provide structural information about the

conformational changes in molecules and the distributions of molecules in a system The emphasis in our discussion will be on those properties that are routinely calculated in computer simulations and on the way in which they are obtained. It is important to recognise that the results we derive are for the canonical ensemble. Sometimes the equivalent expressions in other ensembles are provided. The result obtained from one ensemble may also be transformed to another ensemble, though this is strictly only possible in the limit of an infinitely large system. The expressions follow from standard statistical mechanical relationships, which are given in standard texts and summarised in Appendix 6.1.

6.2.1 Energy

The internal energy is easily obtained from a simulation as the ensemble average of the energies of the states that are examined during the course of the simulation:

$$U = \langle E \rangle = \frac{1}{M} \sum_{i=1}^{M} E_i \tag{6 7}$$

6.2.2 Heat Capacity

At a phase transition the heat capacity will often show a characteristic dependence upon the temperature (a first-order phase transition is characterised by an infinite heat capacity at the transition but in a second-order phase transition the heat capacity changes discontinuously) Monitoring the heat capacity as a function of temperature may therefore enable phase transitions to be detected. Calculations of the heat capacity can also be compared with experimental results and so be used to check the energy model or the simulation protocol.

The heat capacity is formally defined as the partial derivative of the internal energy with respect to temperature:

$$C_V = \left(\frac{\partial U}{\partial T} \right)_V \tag{6.8}$$

The heat capacity can therefore be calculated by performing a series of simulations at different temperatures, and then differentiating the energy with respect to the temperature. The differentiation can be done numerically or by fitting a polynomial to the data and then analytically differentiating the fitted function. The heat capacity may also be calculated from a single simulation by considering the instantaneous fluctuations in the energy as follows:

$$C_V = \{\langle E^2 \rangle - \langle E \rangle^2\}/k_B T^2 \tag{6 9}$$

An alternative way to write this expression uses the relationship:

$$\langle (E - \langle E \rangle)^2 \rangle = \langle E^2 \rangle - \langle E \rangle^2 \tag{6.10}$$

giving:

$$C_V = \langle (E - \langle E \rangle)^2 \rangle / k_B T^2 \tag{6 11}$$

A derivation of this result is provided in Appendix 6 2.

The heat capacity can therefore be obtained by keeping a running count of E^2 and E during the simulation, from which their expectation values $\langle E^2 \rangle$ and $\langle E \rangle$ can be calculated at the end of the calculation. Alternatively, if the energies are stored during the simulation then the value of $\langle (E - \langle E \rangle)^2 \rangle$ can be calculated once the simulation has finished. This second approach may be more accurate due to round-off errors; $\langle E^2 \rangle$ and $\langle E \rangle^2$ are usually both large numbers and so there may be a large uncertainty in their difference.

6.2.3 Pressure

The pressure is usually calculated in a computer simulation via the virial theorem of Clausius. The *virial* is defined as the expectation value of the sum of the products of the coordinates of the particles and the forces acting on them. This is usually written $W = \sum x_i \dot{p}_{x_i}$ where x_i is a coordinate (e.g. the x or y coordinate of an atom) and $\dot{p}_{x_i}$ is the first derivative of the momentum along that coordinate ($\dot{p}_i$ is the force, by Newton's second law). The virial theorem states that the virial is equal to $-3Nk_BT$.

In an ideal gas, the only forces are those due to interactions between the gas and the container and it can be shown that the virial in this case equals $-3PV$. This result can also be obtained directly from $PV = Nk_BT$.

Forces between the particles in a real gas or liquid affect the virial, and thence the pressure. The total virial for a real system equals the sum of an ideal gas part ($-3PV$) and a contribution due to interactions between the particles. The result obtained is:

$$W = -3PV + \sum_{i=1}^{N} \sum_{j=i+1}^{N} r_{ij} \frac{dv(r_{ij})}{dr_{ij}} = -3Nk_BT \tag{6.12}$$

The real gas part is derived in Appendix 6.3. If $dv(r_{ij})/dr_{ij}$ is written as f_{ij}, the force acting between atoms i and j, then we have the following expression for the pressure:

$$P = \frac{1}{V}\left[Nk_BT - \frac{1}{3}\sum_{i=1}^{N} \sum_{j=i+1}^{N} r_{ij} f_{ij}\right] \tag{6.13}$$

The forces are calculated as part of a molecular dynamics simulation, and so little additional effort is required to calculate the virial and thus the pressure. The forces are not routinely calculated during a Monte Carlo simulation, and so some additional effort is required to determine the pressure by this route. When calculating the pressure it is also important to check that the components of the pressure in all three directions are equal.

6.2.4 Temperature

In a canonical ensemble the total temperature is constant. In the microcanonical ensemble, however, the temperature will fluctuate. The temperature is directly related to the kinetic energy of the system as follows:

$$\mathcal{K} = \sum_{i=1}^{N} \frac{|\mathbf{p}_i|^2}{2m_i} = \frac{k_BT}{2}(3N - N_c) \tag{6.14}$$

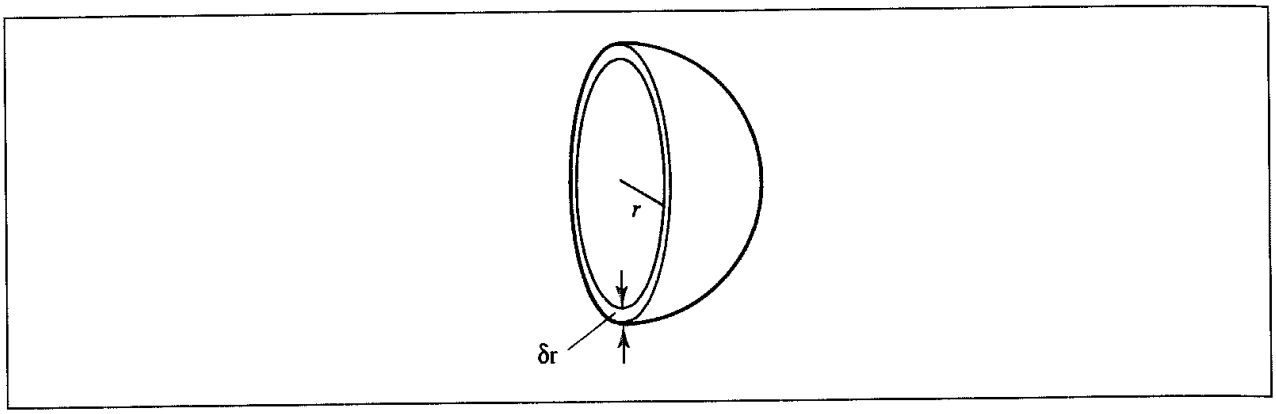

Fig 6 1· Radial distribution functions use a spherical shell of thickness δr.

In this equation, $\mathbf{p}_i$ is the total momentum of particle i and m_i is its mass. According to the theorem of the equipartition of energy each degree of freedom contributes $k_B T/2$. If there are N particles, each with three degrees of freedom, then the kinetic energy should equal $3Nk_B T/2$. N_c in Equation (6.14) is the number of constraints on the system. In a molecular dynamics simulation the total linear momentum of the system is often constrained to a value of zero, which has the effect of removing three degrees of freedom from the system and so N_c would be equal to 3. Other types of constraint are also possible as we shall discuss in Section 7.5.

6.2.5 Radial Distribution Functions

Radial distribution functions are a useful way to describe the structure of a system, particularly of liquids. Consider a spherical shell of thickness δr at a distance r from a chosen atom (Figure 6.1). The volume of the shell is given by:

$$\begin{aligned} V &= \tfrac{4}{3}\pi(r+\delta r)^3 - \tfrac{4}{3}\pi r^3 \\ &= 4\pi r^2 \delta r + 4\pi r \delta r^2 + \tfrac{4}{3}\pi \delta r^3 \approx 4\pi r^2 \delta r \end{aligned} \tag{6.15}$$

If the number of particles per unit volume is ρ, then the total number in the shell is $4\pi\rho r^2\,\delta r$ and so the number of atoms in the volume element varies as r^2.

The pair distribution function, $g(r)$, gives the probability of finding an atom (or molecule, if simulating a molecular fluid) a distance r from another atom (or molecule) compared to the ideal gas distribution. $g(r)$ is thus dimensionless. Higher radial distribution functions (e.g. the triplet radial distribution function) can also be defined but are rarely calculated and so references to the 'radial distribution function' are usually taken to mean the pairwise version. In a crystal, the radial distribution function has an infinite number of sharp peaks whose separations and heights are characteristic of the lattice structure.

The radial distribution function of a liquid is intermediate between the solid and the gas, with a small number of peaks as short distances, superimposed on a steady decay to a constant value at longer distances. The radial distribution function calculated from a molecular dynamics simulation of liquid argon (shown in Figure 6.2) is typical. For short distances (less

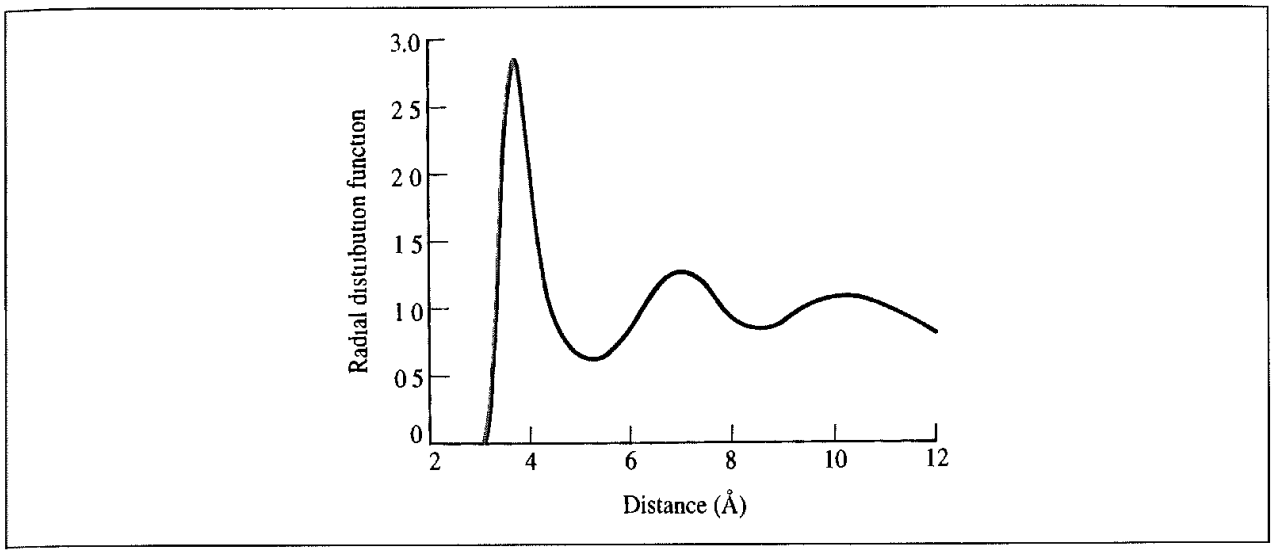

Fig 6.2· Radial distribution function determined from a 100 ps molecular dynamics simulation of liquid argon at a temperature of 100 K and a density of 1 396 g cm^{-3}

than the atomic diameter) $g(r)$ is zero. This is due to the strong repulsive forces. The first (and largest) peak occurs at $r \approx 3.7$ Å, with $g(r)$ having a value of about 3. This means that it is three times more likely that two molecules would have this separation than in the ideal gas. The radial distribution function then falls and passes through a minimum value around $r \approx 5.4$ Å. The chances of finding two atoms with this separation are less than for the ideal gas. At long distances, $g(r)$ tends to the ideal gas value, indicating that there is no long-range order.

To calculate the pair distribution function from a simulation, the neighbours around each atom or molecule are sorted into distance 'bins', or histograms. The number of neighbours in each bin is then averaged over the entire simulation. For example, a count is made of the number of neighbours between (say) 2.5 Å and 2.75 Å, 2.75 Å and 3.0 Å and so on for every atom or molecule in the simulation. This count can be performed during the simulation itself or by analysing the configurations that are generated.

Radial distribution functions can be measured experimentally using X-ray diffraction. The regular arrangement of the atoms in a crystal gives the characteristic X-ray diffraction pattern with bright, sharp spots. For liquids, the diffraction pattern has regions of high and low intensity but no sharp spots. The X-ray diffraction pattern can be analysed to calculate an experimental distribution function, which can then be compared with that obtained from the simulation.

Thermodynamic properties can be calculated using the radial distribution function, if pairwise additivity of the forces is assumed. These properties are usually given as an ideal gas part plus a real gas part. For example, to calculate the energy of a real gas, we consider the spherical shell of volume $4\pi r^2 \delta r$ that contains $4\pi r^2 \rho g(r) \delta r$ particles. If the pair potential at a distance r has a value $v(r)$ then the energy of interaction between the particles in the shell and the central particle is $4\pi r^2 \rho g(r) v(r) \delta r$. The total potential energy of the real gas is obtained by integrating this between 0 and ∞ and multiplying the result

by $N/2$ (the factor $1/2$ ensures that we only count each interaction once). The total energy is then given by:

$$E = \tfrac{3}{2}Nk_BT + 2\pi N\rho \int_0^\infty r^2 v(r)g(r)\,dr \tag{6.16}$$

In a similar way the following expression for the pressure can be derived:

$$PV = Nk_BT - \frac{2\pi N\rho}{3k_BT}\int_0^\infty r^2 r\frac{dv(r)}{dr}g(r)\,dr \tag{6.17}$$

It is usually more accurate to calculate such properties directly, partly because the radial distribution function is not obtained as a continuous function but is derived by dividing the space into small but discrete bins.

For molecules, the orientation must be taken into account if the true nature of the distribution is to be determined. The radial distribution function for molecules is usually measured between two fixed points, such as between the centres of mass. This may then be supplemented by an orientational distribution function. For linear molecules, the orientational distribution function may be calculated as the angle between the axes of the molecule, with values ranging from $-180°$ to $+180°$. For more complex molecules it is usual to calculate a number of site–site distribution functions. For example, for a three-site model of water, three functions can be defined (g(O–O), g(O–H) and g(H–H)) An advantage of the site–site models is that they can be directly related to information obtained from the X-ray scattering experiments. The O–O, O–H and H–H radial distribution functions have been particularly useful for refining the various potential models for simulating liquid water.

6.3 Phase Space

An important concept in computer simulation is that of the *phase space*. For a system containing N atoms, $6N$ values are required to define the state of the system (three coordinates per atom and three components of the momentum). Each combination of $3N$ positions and $3N$ momenta (usually denoted by Γ_N) defines a point in the $6N$-dimensional phase space; an ensemble can thus be considered to be a collection of points in phase space. The way in which the system moves through phase space is governed by Hamiltonian's equations:

$$\frac{d\mathbf{r}_i}{dt} = \frac{\partial \mathcal{H}}{\partial \mathbf{p}_i} \tag{6.18}$$

$$\frac{d\mathbf{p}_i}{dt} = -\frac{\partial \mathcal{H}}{\partial \mathbf{r}_i} \tag{6.19}$$

where i varies from 1 to N. Molecular dynamics generates a sequence of points in phase space that are connected in time. These points correspond to the successive configurations of the system generated by the simulation. A molecular dynamics simulation performed in the microcanonical (constant NVE) ensemble will sample phase space along a contour of constant energy. There is no momentum component in a Monte Carlo simulation and such simulations sample from the $3N$-dimensional space corresponding to the positions of

the atoms. It might seem odd that thermodynamic properties can be obtained from Monte Carlo simulations, given that there is no momentum contribution and so $3N$ degrees of freedom are not explored. In fact, all of the deviations from ideal gas behaviour are a consequence of interactions between the atoms and are encapsulated in the potential function, $\mathcal{V}(\mathbf{r}^N)$, which only depends upon the positions of the atoms. A Monte Carlo simulation does sample from the positional degrees of freedom and so can be used to provide the deviations of thermodynamic properties from ideal gas behaviour, which is what we want to calculate. We shall return to this point in Chapter 8.

If it were possible to visit all the points in phase space then the partition function could be calculated by summing the values of $\exp(-E/k_BT)$. The phase-space trajectory in such a case would be termed *ergodic* and the results would be independent of the initial configuration. For the systems that are typical of those studied using simulation methods the phase space is immense, and an ergodic trajectory is not achievable (indeed, even for relatively small systems with only a few tens of atoms the time that would be required to cycle round all of the points in phase space is longer than the age of the universe). A simulation can thus only ever provide an estimate of the 'true' energies and other thermodynamic properties and so a sequence of simulations using different starting conditions would be expected to give similar, but different, results.

The thermodynamic properties that we have considered so far, such as the internal energy, the pressure and the heat capacity are collectively known as the mechanical properties and can be routinely obtained from a Monte Carlo or molecular dynamics simulation. Other thermodynamic properties are difficult to determine accurately without resorting to special techniques. These are the so-called entropic or thermal properties: the free energy, the chemical potential and the entropy itself. The difference between the mechanical and thermal properties is that the mechanical properties are related to the derivative of the partition function whereas the thermal properties are directly related to the partition function itself. To illustrate the difference between these two classes of properties, let us consider the internal energy, U, and the Helmholtz free energy, A. These are related to the partition function by:

$$U = \frac{k_BT^2}{Q}\frac{\partial Q}{\partial T} \tag{6.20}$$

$$A = -k_BT\ln Q \tag{6.21}$$

Q is given by Equation (6.4) for a system of identical particles. We shall ignore any normalisation constants in our treatment here to enable us to concentrate on the basics, and so it does not matter whether the system consists of identical or distinguishable particles. We also replace the Hamiltonian by the energy, E. The internal energy is obtained via Equation (6.20):

$$\begin{aligned} U &= k_BT^2\frac{1}{Q}\iint d\mathbf{p}^N\,d\mathbf{r}^N\frac{E(\mathbf{p}^N,\mathbf{r}^N)}{k_BT^2}\exp(-E(\mathbf{p}^N,\mathbf{r}^N)/k_BT) \\ &= \iint d\mathbf{p}^N\,d\mathbf{r}^N E(\mathbf{p}^N,\mathbf{r}^N)\frac{\exp(-E(\mathbf{p}^N,\mathbf{r}^N)/k_BT)}{Q} \end{aligned} \tag{6.22}$$

Now consider the probability of the state with energy $E(\mathbf{p}^N, \mathbf{r}^N)$:

$$\frac{\exp(-E(\mathbf{p}^N, \mathbf{r}^N)/k_B T)}{Q} \tag{6.23}$$

This probability is written $\rho(\mathbf{p}^N, \mathbf{r}^N)$; the internal energy is thus given by

$$U = \iint d\mathbf{p}^N \, d\mathbf{r}^N E(\mathbf{p}^N, \mathbf{r}^N) \rho(\mathbf{p}^N, \mathbf{r}^N) \tag{6.24}$$

The crucial point about Equation (6.24) is that high values of $E(\mathbf{p}^N, \mathbf{r}^N)$ have a very low probability and make an insignificant contribution to the integral. The Monte Carlo and molecular dynamics methods preferentially generate states of low energy, which are the states that make a significant contribution to the integral in Equation (6.24). These methods sample from phase space in a way that is representative of the equilibrium state and are able to generate accurate estimates of properties such as the internal energy, heat capacity, and so on.

Let us now consider the problem of calculating the Helmholtz free energy of a molecular liquid. Our aim is to express the free energy in the same functional form as the internal energy, that is as an integral which incorporates the probability of a given state. First, we substitute for the partition function in Equation (6.21):

$$A = -k_B T \ln Q = k_B T \ln \left(\frac{N! h^{3N}}{\iint d\mathbf{p}^N \, d\mathbf{r}^N \exp(-E(\mathbf{p}^N, \mathbf{r}^N)/k_B T)} \right) \tag{6.25}$$

Next we recognise that the following integral is equal to 1:

$$1 = \frac{1}{(8\pi^2 V)^N} \iint d\mathbf{p}^N \, d\mathbf{r}^N \exp\left(-\frac{E(\mathbf{p}^N, \mathbf{r}^N)}{k_B T} \right) \exp\left(\frac{E(\mathbf{p}^N, \mathbf{r}^N)}{k_B T} \right) \tag{6.26}$$

Inserting this into the expression for the free energy and ignoring the constants (which act to change the zero point from which the free energy is calculated) gives:

$$A = k_B T \ln \left(\frac{\iint d\mathbf{p}^N \, d\mathbf{r}^N \exp\left(-\frac{E(\mathbf{p}^N, \mathbf{r}^N)}{k_B T} \right) \exp\left(+\frac{E(\mathbf{p}^N, \mathbf{r}^N)}{k_B T} \right)}{\iint d\mathbf{p}^N \, d\mathbf{r}^N \exp(-E(\mathbf{p}^N, \mathbf{r}^N)/k_B T)} \right) \tag{6.27}$$

We can now substitute for the probability density, $\rho(\mathbf{p}^N, \mathbf{r}^N)$ in this equation, leading to the final result (in which we have again ignored the normalisation factors):

$$A = k_B T \ln \left(\iint d\mathbf{p}^N \, d\mathbf{r}^N \exp\left(+\frac{E(\mathbf{p}^N, \mathbf{r}^N)}{k_B T} \right) \rho(\mathbf{p}^N, \mathbf{r}^N) \right) \tag{6.28}$$

The important feature of this result is that the configurations with a high energy make a significant contribution to the integral due to the presence of the exponential term $\exp(+E(\mathbf{p}^N, \mathbf{r}^N)/k_B T)$. A Monte Carlo or molecular dynamics simulation preferentially samples the *lower-energy* regions of phase space. An ergodic trajectory would, of course, visit all of these high-energy regions, but in practice these will never be adequately sampled

by a real simulation. The results for the free energy and other entropic properties will as a consequence be poorly converged and inaccurate.

To reiterate a point that we made earlier, these problems of accurately calculating the free energy and entropy do not arise for isolated molecules that have a small number of well-characterised minima which can all be enumerated. The partition function for such systems can be obtained by standard statistical mechanical methods involving a summation over the minimum energy states, taking care to include contributions from internal vibrational motion.

6.4 Practical Aspects of Computer Simulation

6.4.1 Setting Up and Running a Simulation

There are significant differences between the molecular dynamics and Monte Carlo simulation methods, but the same general strategies are used to set up and run either type of simulation. The first task is to decide which energy model is to be used to describe the interactions within the system. Simulations are usually performed with relatively large numbers of atoms over many iterations or time steps. The intra- and intermolecular interactions are therefore almost always described using an empirical (i.e. molecular mechanics) energy model. Faster computers and new theoretical techniques do now enable simulations to be performed using models based only on quantum mechanics or mixed models based on molecular mechanics/quantum mechanics as discussed in Section 11.13 Having chosen an energy model, the simulation itself can be broken into four distinct stages. First, an initial configuration for the system must be established. An *equilibration phase* is then performed, during which the system evolves from the initial configuration. Thermodynamic and structural properties are monitored during the equilibration until stability is achieved. Several distinct steps may be required during the equilibration, particularly for inhomogeneous systems. At the end of the equilibration the *production phase* commences. It is during the production phase that simple properties of the system are calculated At regular intervals the configuration of the system (i.e. the atomic coordinates) is output to a disk file. Finally, the simulation is analysed; properties not calculated during the simulation are determined and the configurations are examined, not only to discover how the structure of the system changed but also to check for any unusual behaviour that might indicate a problem with the simulation.

6.4.2 Choosing the Initial Configuration

Before a simulation can be performed it is obviously necessary to select an initial configuration of the system. This should be done with some care, as the initial arrangement can often determine the success or failure of a simulation. For simulations of systems at equilibrium (the most common sort) it is wise to choose an initial configuration that is close to the state which it is desired to simulate. For example, it would be unwise to initiate a simulation of a face-centred cubic solid from a body-centred cubic starting point. It is also good practice to ensure that the initial configuration does not contain any high-energy

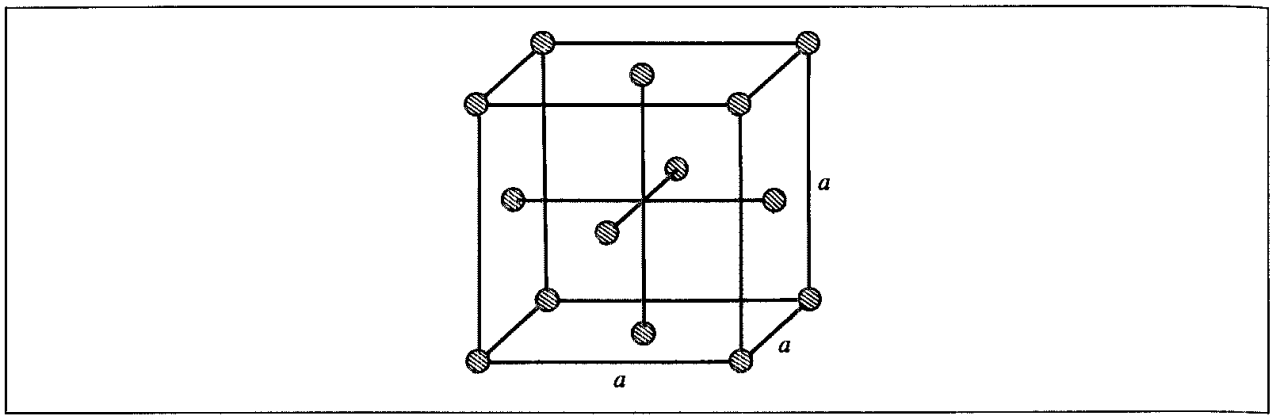

Fig 6 3 The face-centred cubic cell.

interactions as these may cause instabilities in the simulation. Such 'hot spots' can often be eradicated by performing energy minimisation prior to the simulation itself.

To simulate homogeneous liquids which contain large numbers of the same molecule, a standard lattice structure is often chosen as the starting configuration. If an experimentally determined arrangement is available (e.g an X-ray structure) then this could be used, provided that it was appropriate to the simulation being performed. When no experimental structure is available the initial configuration can be chosen from one of the common crystallographic lattices (simply placing molecules at random can often give rise to high-energy overlaps and instabilities). The most common lattice is the face-centred cubic lattice (fcc), shown in Figure 6.3. This structure contains $4M^3$ points ($M = 2, 3, 4, \ldots$) For this reason, simulations are often performed using 108, 256, 525, 784,..., atoms or molecules. The lattice size is chosen so that the density is appropriate to that of the system under study. For simulations of molecules it is also necessary to assign an orientation to each molecule. For small linear molecules, the solid structure of CO_2 is often chosen as the initial configuration. This is a face-centred cubic lattice with the molecules oriented in a regular fashion along the four diagonals of the unit cell. Alternatively, the orientations may be chosen completely at random or by making small random changes from the orientation in a regular lattice. At high densities non-physical overlaps may result, particularly if the molecules are large; in such cases it is more important to use an initial configuration that is close to the expected equilibrium distribution. For example, simulations of rod-shaped molecules such as liquid crystals are usually initiated from a configuration in which the molecules are all aligned approximately in the same direction.

For simulations of inhomogeneous systems comprising a solute molecule or intermolecular complex immersed in a solvent, the starting conformation of the solute may be obtained from an experimental technique such as X-ray crystallography or NMR, or may be generated by theoretical modelling. The coordinates of some solvent molecules may be known if the structure is obtained from X-ray crystallography, but it is usually necessary to add other solvent molecules to give the appropriate solvent density. A typical approach is to use the coordinates obtained from a previous simulation of the pure solvent The solute is immersed in the solvent 'bath' and any solvent molecules that are too close to the solute are then discarded before the calculation proceeds

6.5 Boundaries

The correct treatment of boundaries and boundary effects is crucial to simulation methods because it enables 'macroscopic' properties to be calculated from simulations using relatively small numbers of particles. The importance of boundary effects can be illustrated by considering the following simple example. Suppose we have a cube of volume 1 litre which is filled with water at room temperature. The cube contains approximately 3.3×10^{25} molecules. Interactions with the walls can extend up to 10 molecular diameters into the fluid. The diameter of the water molecule is approximately 2.8 Å and so the number of water molecules that are interacting with the boundary is about 2×10^{19}. So only about one in 1.5 million water molecules is influenced by interactions with the walls of the container. The number of particles in a Monte Carlo or molecular dynamics simulation is far fewer than 10^{25}–10^{26} and is frequently less than 1000. In a system of 1000 water molecules most, if not all of them, would be within the influence of the walls of the boundary Clearly, a simulation of 1000 water molecules in a vessel would not be an appropriate way to derive 'bulk' properties. The alternative is to dispense with the container altogether. Now, approximately three-quarters of the molecules would be at the surface of the sample rather than being in the bulk. Such a situation would be relevant to studies of liquid drops, but not to studies of bulk phenomena.

6.5.1 Periodic Boundary Conditions

Periodic boundary conditions enable a simulation to be performed using a relatively small number of particles, in such a way that the particles experience forces as if they were in bulk fluid. Imagine a cubic box of particles which is replicated in all directions to give a periodic array. A two-dimensional box is shown in Figure 6.4. In the two-dimensional example each box is surrounded by eight neighbours; in three dimensions each box would have 26 nearest neighbours. The coordinates of the particles in the image boxes can be computed simply by adding or subtracting integral multiples of the box sides. Should a particle leave the box during the simulation then it is replaced by an image particle that enters from the opposite side, as illustrated in Figure 6.4. The number of particles within the central box thus remains constant.

The cubic cell is the simplest periodic system to visualise and to program. However, a cell of a different shape might be more appropriate for a given simulation. This may be particularly important for simulations of systems which comprise a single molecule or intermolecular complex surrounded by solvent molecules. In such systems it is usually the behaviour of the central solute molecule that is of most interest and so it is desirable that as little of the computer time as possible is spent simulating the solvent far from the solute. In principle, any cell shape can be used provided it fills all of space by translation operations of the central box in three dimensions. Five shapes satisfy this condition: the cube (and its close relation, the parallelepiped), the hexagonal prism, the truncated octahedron, the rhombic dodecahedron and the 'elongated' dodecahedron (Figure 6.5) [Adams 1983]. It is often sensible to choose a periodic cell that reflects the underlying geometry of the system. For example, a rectangular cell is not the ideal choice to simulate an approximately spherical molecule.

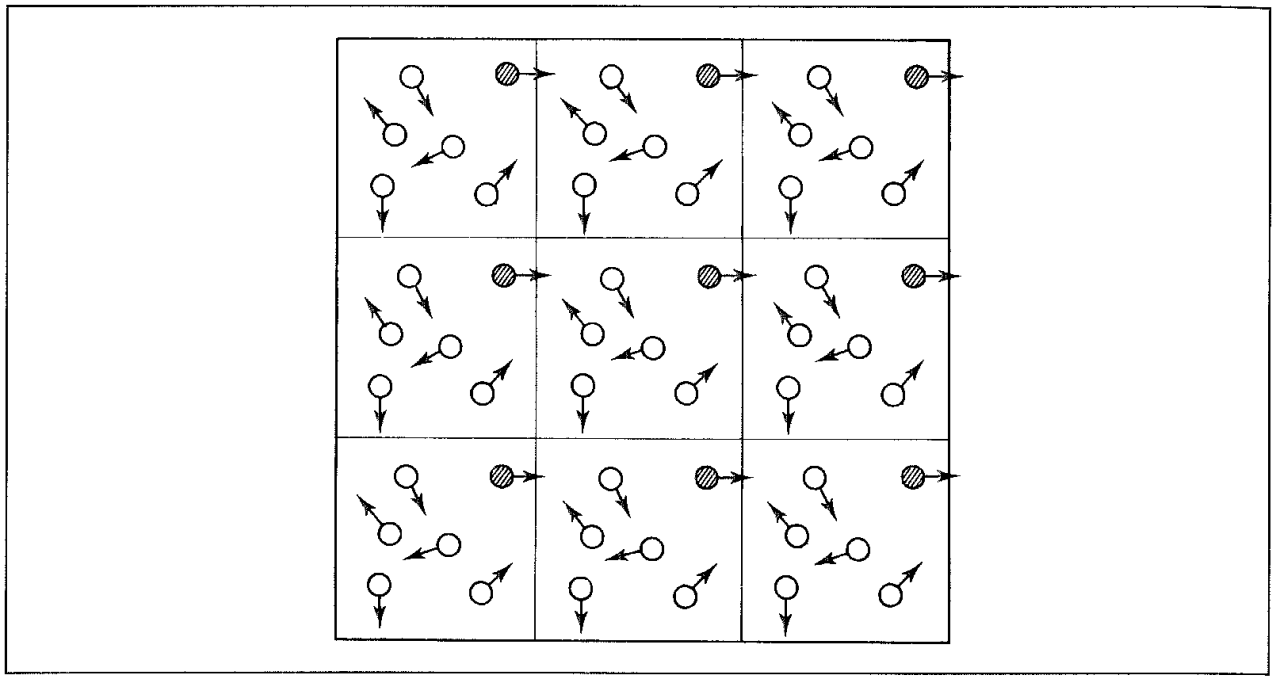

Fig. 6 4· Periodic boundary conditions in two dimensions

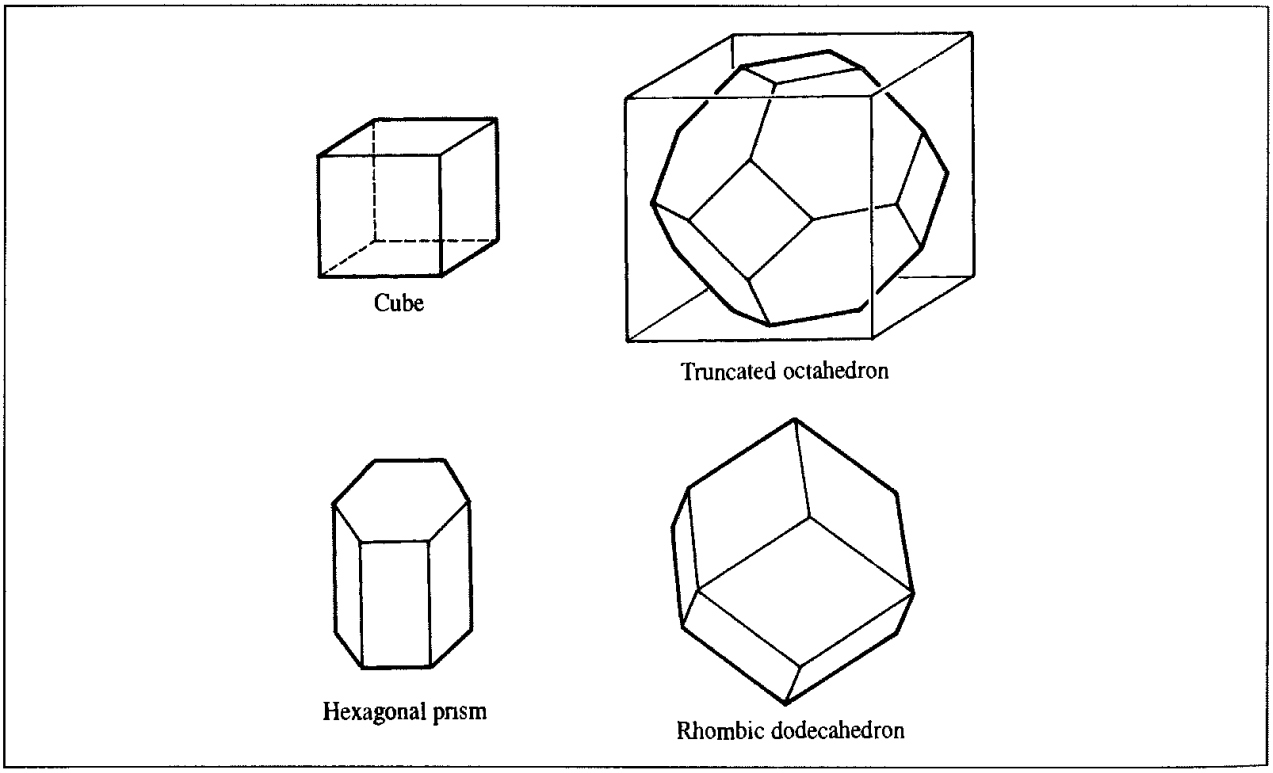

Fig 6 5. Periodic cells used in computer simulations. the cube, truncated octahedron, hexagonal prism and rhombic dodecahedron

The truncated octahedron and the rhombic dodecahedron provide periodic cells that are approximately spherical and so may be more appropriate for simulations of spherical molecules. The distance between adjacent cells in the truncated octahedron or the rhombic dodecahedron is larger than the conventional cube for a system with a given number of particles and so a simulation using one of the spherical cells will require fewer particles than a comparable simulation using a cubic cell. Of the two approximately spherical cells, the truncated octahedron is often preferred as it is somewhat easier to program. The hexagonal prism can be used to simulate molecules with a cylindrical shape such as DNA.

Of the five possible shapes, the cube/parallelepiped and the truncated octahedron have been most widely used, with some simulations in the hexagonal prism. The formulae used to translate a particle back into the central simulation box for these three shapes are given in Appendix 6.4. It may be preferable to use one of the more common periodic cells even if there are aesthetic reasons for using an alternative. This is because the expressions for calculating the images may be difficult and inefficient to implement, even though the simulation would use fewer atoms.

For some simulations it is inappropriate to use standard periodic boundary conditions in all directions. For example, when studying the adsorption of molecules onto a surface, it is clearly inappropriate to use the usual periodic boundary conditions for motion perpendicular to the surface. Rather, the surface is modelled as a true boundary, for example by explicitly including the atoms in the surface. The opposite side of the box must still be treated; when a molecule strays out of the top side of the box it is reflected back into the simulation cell, as indicated in Figure 6.6. Usual periodic boundary conditions apply to motion parallel to the surface.

Periodic boundaries are widely used in computer simulations, but they do have some drawbacks. A clear limitation of the periodic cell is that it is not possible to achieve fluctuations that have a wavelength greater than the length of the cell. This can cause problems in certain situations, such as near the liquid–gas critical point. The range of the interactions present in the system is also important; if the cell size is large compared with the range over which the interactions act then there should be no problems. For example, for the relatively short-range Lennard-Jones potential the cell should have a side greater than approximately 6σ, which corresponds to about 20 Å for argon. For longer-range electrostatic interactions the situation is more difficult and it is often necessary to accept that some long-range order will be imposed upon the system. The effects of imposing a periodic boundary can be evaluated empirically by comparing the results of simulations performed using a variety of cell shapes and sizes.

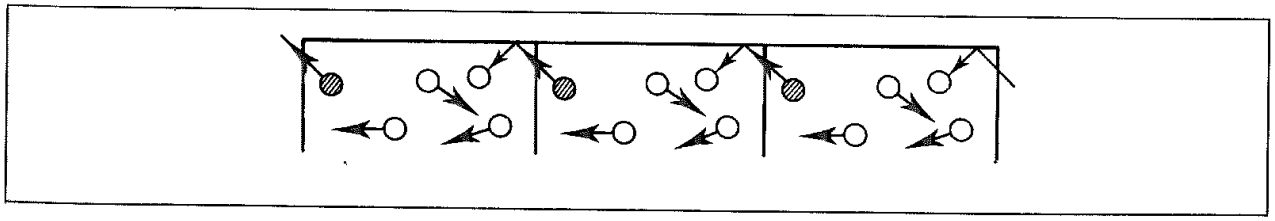

Fig 6 6 Periodic boundary conditions for surface simulations (Figure adapted from Allen M P and D J Tildesley 1987 Computer Simulation of Liquids. *Oxford, Oxford University Press)*

6.5.2 Non-periodic Boundary Methods

Periodic boundary conditions are not always used in computer simulations. Some systems, such as liquid droplets or van der Waals clusters, inherently contain a boundary. Periodic boundary conditions may also cause difficulties when simulating inhomogeneous systems or systems that are not at equilibrium. In other cases the use of periodic boundary conditions would require a prohibitive number of atoms to be included in the simulation. This particularly arises in the study of the structural and conformational behaviour of macromolecules such as proteins and protein–ligand complexes. The first simulations of such systems ignored all solvent molecules due to the limited computational resources then available. This corresponds to the unrealistic situation of simulating an isolated protein *in vacuo* and then comparing the results with experimental data obtained in solution. Vacuum calculations can lead to significant problems. A vacuum boundary tends to minimise the surface area and so may distort the shape of the system if it is non-spherical. Small molecules may adopt more compact conformations when simulated *in vacuo* due to favourable intramolecular electrostatic and van der Waals interactions, which would be dampened in the presence of a solvent.

As computer power has increased it has become possible to incorporate explicitly some solvent molecules and thereby simulate a more realistic system. The simplest way to do this is to surround the molecule with a 'skin' of solvent molecules. If the skin is sufficiently deep then the system is equivalent to a solute molecule inside a 'drop' of solvent. The number of solvent molecules in such cases is usually significantly fewer than would be required in the analogous periodic boundary simulation, where the solute molecule is positioned at the centre of the cell and the empty space is filled with solvent. Boundary effects should be transferred from the molecule–vacuum interface to the solvent–vacuum interface and so might be expected to result in a more realistic treatment of the solute. To illustrate these three situations, we can consider dihydrofolate reductase, which is a small enzyme that contains approximately 2500 atoms. If this enzyme is surrounded by water molecules in a cubic periodic system such that the surface of the protein is at least 10 Å from any side of the box, then the number of atoms rises to almost 20000. If a shell 10 Å thick is used then the number of atoms falls to 14700, and with a 5 Å shell the system contains 8900 atoms.

Sometimes we are only interested in a specific part of the solute, such as the active site of an enzyme. It has been common practice in such cases to divide the system into two regions (Figure 6.7). One region, often called the *reaction zone*, contains all atoms or groups within a given radius R_1 of the site of interest. The atoms in the reaction zone are subjected to the full simulation method. The second region (the reservoir region) contains all atoms outside the reaction zone but within a distance R_2 of the active site. The atoms in the reservoir region may be kept fixed in their initial positions, or may be restrained so that they stay within the shell defined by R_1 and R_2 Alternatively, they may be restrained to their initial positions using a harmonic potential. Any atoms further away from the active site than R_2 are discarded or may be kept fixed in their initial positions. It is important to be aware that restraining or fixing atoms in this way may prevent natural changes occurring and so lead to artificial behaviour. A variety of schemes for performing simulations using such *stochastic boundary conditions* have been proposed. However, such methods can be rather complicated to implement and if not used properly can give anomalous results.

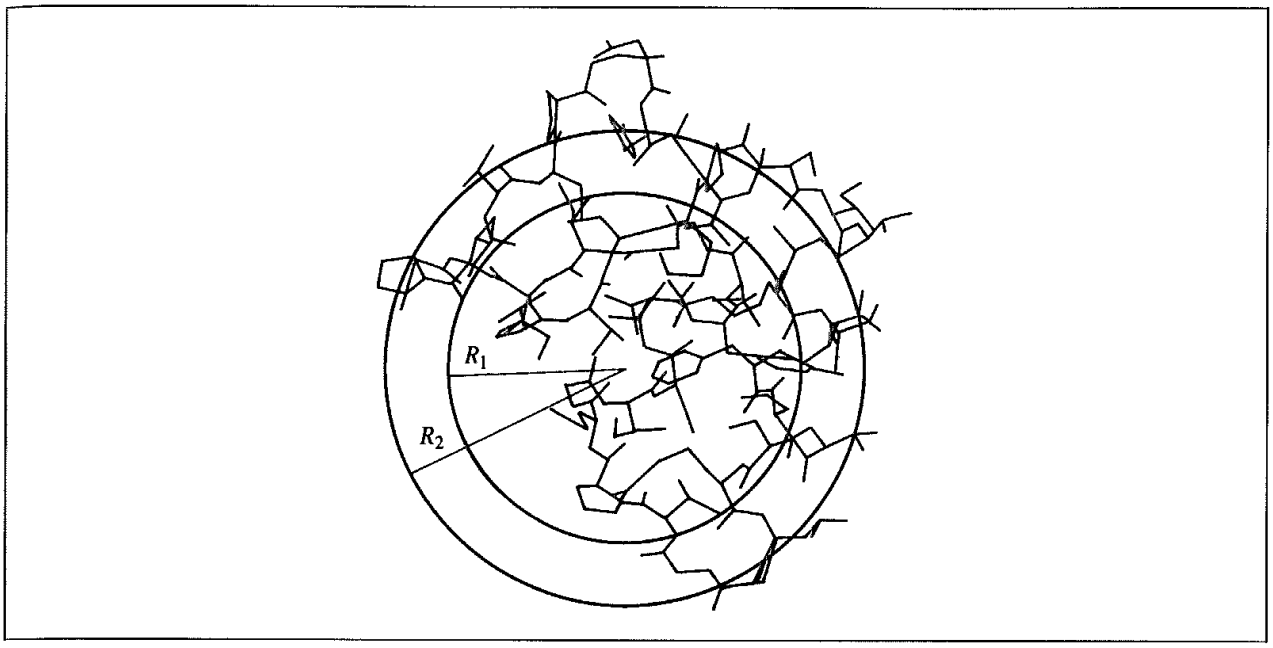

Fig 6 7· Division into reaction zone and reservoir regions in a simulation using stochastic boundary conditions

If at all possible, a periodic boundary is the 'safest' way to ensure that boundary effects are minimised, but sometimes an alternative may be the only practical course.

6.6 Monitoring the Equilibration

The purpose of the equilibration phase is to enable the system to evolve from the starting configuration to reach equilibrium. Equilibration should continue until the values of a set of monitored properties become stable. The properties to be monitored usually include thermodynamic quantities such as the energy, temperature and pressure and also structural properties. Many simulations of the liquid state involve a starting configuration that corresponds to a solid lattice. It is therefore important to establish that the lattice has 'melted' before the production phase begins. *Order parameters* can be used to determine that the liquid state has been reached. An order parameter is a measure of the degree of order (or, equivalently, disorder) in the system. During a simulation of a crystal lattice the atoms would be expected to remain in approximately the same positions throughout and thereby maintain a high degree of order In a liquid, however, we would expect considerable mobility of the species present, giving rise to translational disorder. One way to measure translational order in a system initially in a face-centred cubic lattice was suggested by Verlet, whose order parameter λ is:

$$\lambda = \tfrac{1}{3}[\lambda_x + \lambda_y + \lambda_z] \tag{6 29}$$

$$\lambda_x = \frac{1}{N}\sum_{i=1}^{N} \cos\left(\frac{4\pi x_i}{a}\right) \tag{6.30}$$

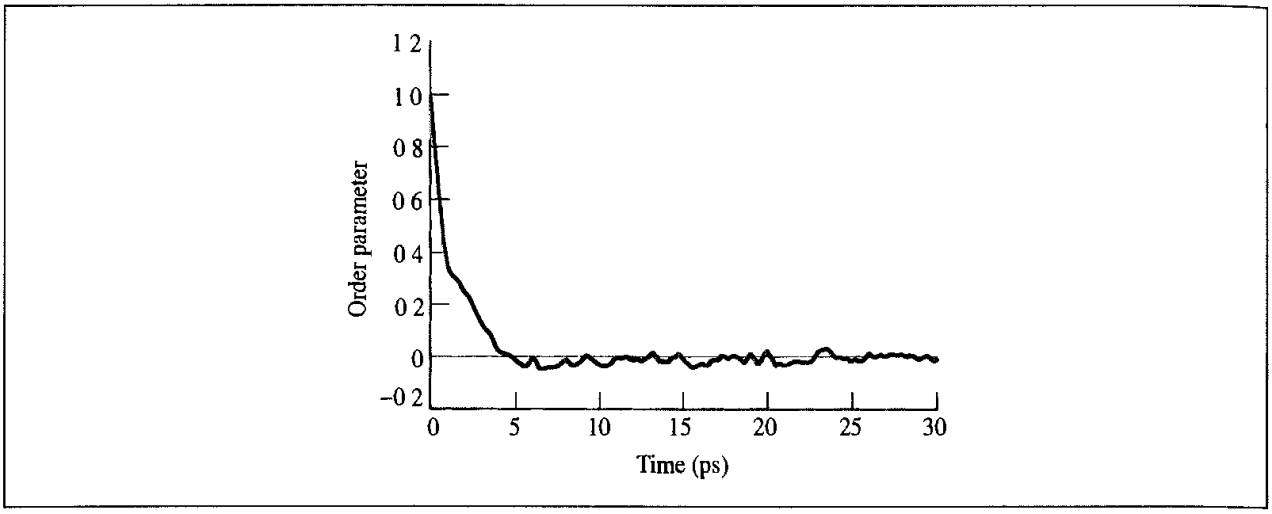

Fig. 6 8. Variation in Verlet order parameter during the equilibration phase of a molecular dynamics simulation of argon

where a is the length of one edge of the unit cell. Initially, all of the coordinates x_i, y_i and z_i are multiples of $a/2$ and so the order parameter has a value of 1. As the simulation proceeds the order parameter should gradually decrease to a value of zero, indicating that the atoms are distributed randomly. When equilibrium has been reached the fluctuations in the order parameter should be proportional to $1/\sqrt{N}$, where N is the size of the system. A typical result is shown in Figure 6.8 for an argon simulation.

For molecules, it is also necessary to consider their orientations, which can be monitored using a rotational order parameter. For some systems, such as carbon monoxide or water, complete disorder would be expected in the liquid state at equilibrium. However, if we were simulating a dense fluid of rod-shaped molecules which form a liquid crystalline phase then we might expect that, on average, the molecules would tend to line up in a common direction. The Viellard-Baron rotational order parameter for linear molecules is calculated using the following formula:

$$P_1 = \frac{1}{N}\sum_{i=1}^{N} \cos\gamma_i \tag{6 31}$$

where γ_i is the angle between the current and original direction of the molecular axis of molecule i. A value of 1 indicates that the molecules are perfectly aligned. Rotational disorder is indicated by a value of zero. The fluctuations about the average value should again be proportional to $1/\sqrt{N}$. For non-linear molecules, a number of rotational order parameters can be defined and each monitored.

The *mean squared displacement* also provides a means to establish whether a solid lattice has melted. The mean squared displacement is given by:

$$\Delta r^2(t) = \frac{1}{N}\sum_{i=1}^{N}[\mathbf{r}_i(t) - \mathbf{r}_i(0)]^2 \tag{6.32}$$

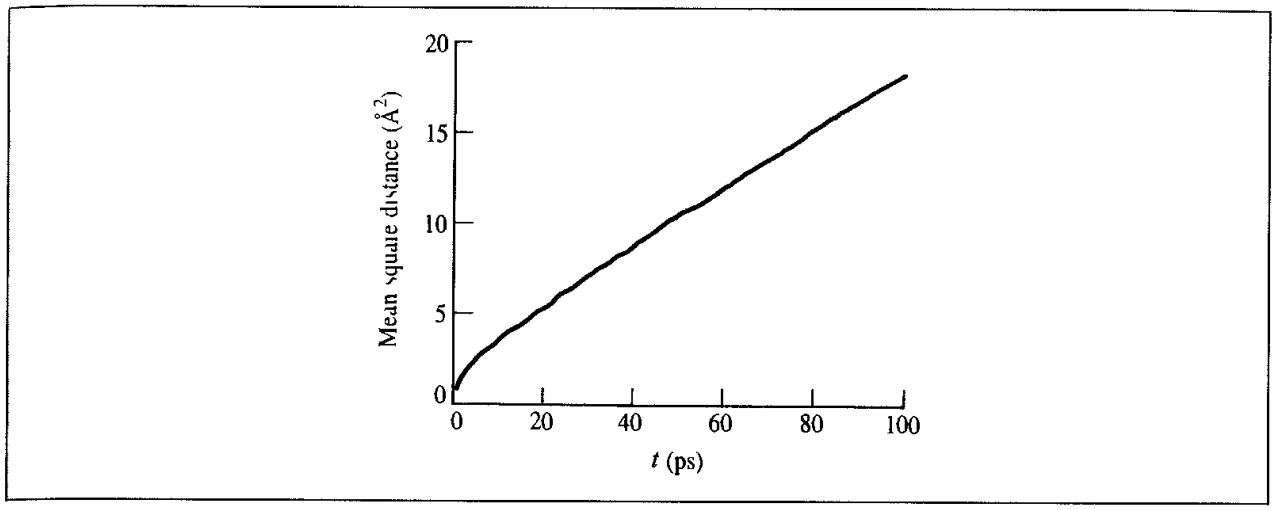

Fig 6 9: Variation in mean squared displacement during the initial steps of a molecular dynamics simulation of argon

For a fluid, with no underlying regular structure, the mean squared displacement gradually increases with time (Figure 6.9). For a solid, however, the mean squared displacement typically oscillates about a mean value. However, if there is diffusion within a solid then this can be detected from the mean squared displacement and may be restricted to fewer than three dimensions. For example, Figure 6.10 shows the mean squared displacement calculated for Li^+ ions in Li_3N at 400 K [Wolf *et al.* 1984]. This material contains layers of Li_2N; mobility of the Li^+ ions is much greater within these planes than perpendicular to them.

The radial distribution function can also be used to monitor the progress of the equilibration. This function is particularly useful for detecting the presence of two phases. Such a situation is characterised by a larger than expected first peak and by the fact that $g(r)$ does not decay towards a value of 1 at long distances. If two-phase behaviour is inappropriate then the simulation should probably be terminated and examined. If, however, a two-phase system is desired, then a long equilibration phase is usually required.

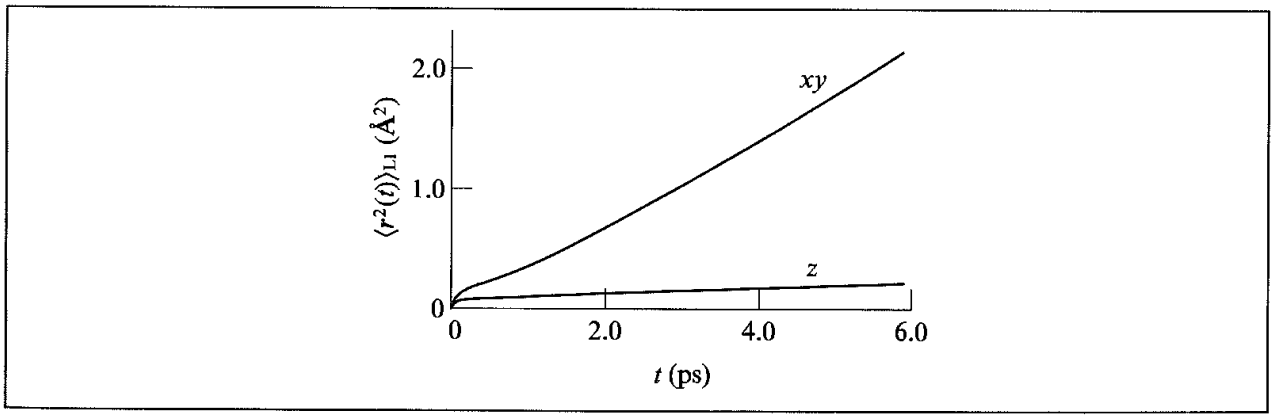

Fig 6 10: Mean squared displacement for Li^+ ions in Li_3N for motion parallel (xy) and perpendicular (z) to the Li_2N layers [Wolf et al 1984].

6.7 Truncating the Potential and the Minimum Image Convention

The most time-consuming part of a Monte Carlo or molecular dynamics simulation (or, indeed, of an energy minimisation) is the calculation of the non-bonded energies and/or forces. The numbers of bond-stretching, angle-bending and torsional terms in a force field model are all proportional to the number of atoms but the number of non-bonded terms that need to be evaluated increases as the square of the number of atoms (for a pairwise model) and is thus of order N^2. In principle, the non-bonded interactions are calculated between every pair of atoms in the system. However, for many interaction models this is not justified. The Lennard-Jones potential falls off very rapidly with distance: at 2.5σ the Lennard-Jones potential has just 1% of its value at σ. This reflects the r^{-6} distance dependence of the dispersion interaction. The most popular way to deal with the non-bonded interactions is to use a *non-bonded cutoff* and to apply the *minimum image convention*. In the minimum image convention, each atom 'sees' at most just one image of every other atom in the system (which is repeated infinitely via the periodic boundary method). The energy and/or force is calculated with the closest atom or image, as illustrated in Figure 6.11. When a cutoff is employed, the interactions between all pairs of atoms that are further apart than the cutoff value are set to zero, taking into account the closest image. When periodic boundary conditions are being used, the cutoff should not be so large that a particle sees its own image, or indeed the same molecule twice. This has the effect of limiting the cutoff to no more than half the length of the cell when simulating atomic fluids in a cubic cell. For rectangular cells the cutoff should be no greater than half the length of the shortest side. For simulations of molecules the upper limit on the cutoff may also be affected by the size of the molecules, as we shall see below in Section 6.7.2 In simulations where the Lennard-Jones potential is the only non-bonded interaction, a cutoff of 2.5σ gives rise to a relatively small error. However, when long-range electrostatic

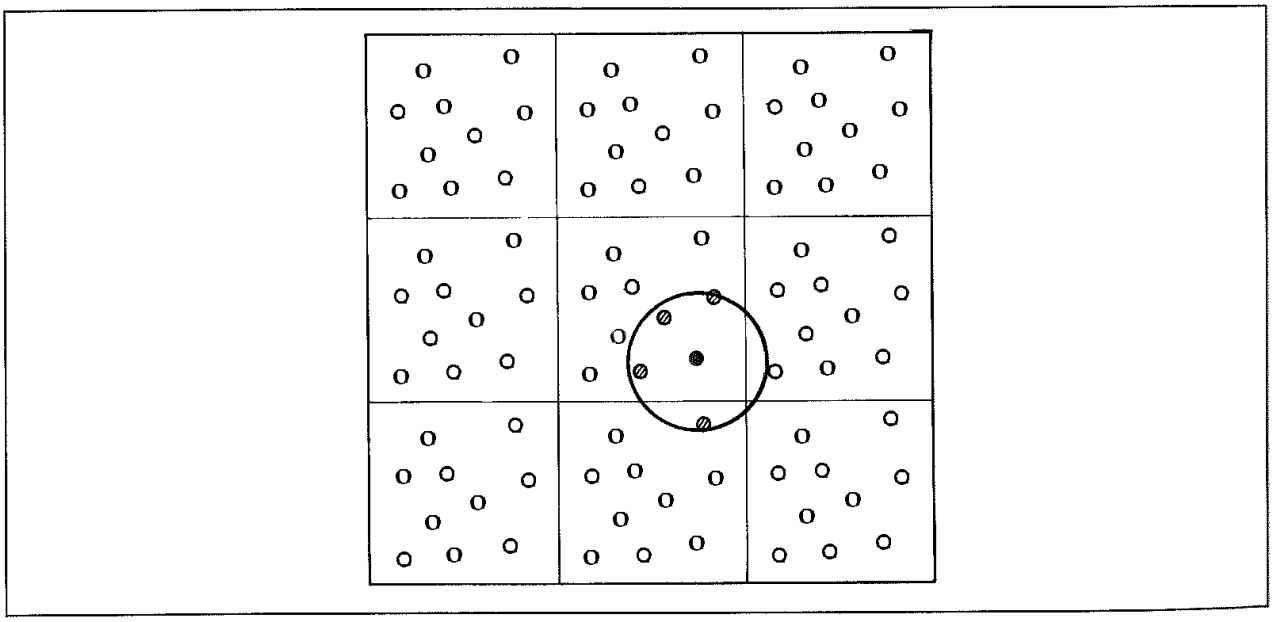

Fig. 6 11: The spherical cutoff and the minimum image convention

interactions are involved, the cutoff should be much greater and indeed there is evidence to suggest that using any cutoff leads to errors. A value of at least 10 Å is generally recommended but even this may be insufficient. More comprehensive methods have been devised for dealing with the electrostatic interactions, which are considered in Section 6.8.

6.7.1 Non-bonded Neighbour Lists

By itself, the use of a cutoff may not dramatically reduce the time taken to compute the number of non-bonded interactions. This is because we would still have to calculate the distance between every pair of atoms in the system simply to decide whether they are close enough to calculate their interaction energy. Calculating all the $N(N-1)$ distances takes almost as much time as calculating the energy itself.

In simulations of fluids, an atom's neighbours (i.e. those atoms that are within the cutoff distance) do not change significantly over 10 or 20 molecular dynamics time steps or Monte Carlo iterations. If we 'knew' which atoms to include in the non-bonded calculation (for example, by storing them in an array), then it would be possible to identify directly each atom's neighbours without having to calculate the distances to all the other atoms in the system. The *non-bonded neighbour* list (first proposed by Verlet) is just such a device. The Verlet neighbour list [Verlet 1967] stores all atoms within the cutoff distance, together with all atoms that are slightly further away than the cutoff distance. This is most efficiently done using a large neighbour list array, L, and a pointer array, P. The pointer array indicates where in the neighbour list the first neighbour for that atom is located. The last neighbour of atom i is stored in element $P[i+1]-1$ of the neighbour list as shown in Figure 6.12. Thus the neighbours of atom i are stored in elements $L[P[i]]$ through $L[P[i+1]-1]$ of the array L. The neighbour list is updated at regular intervals throughout the simulation. Between updates, the neighbour and pointer lists are used to directly identify the nearest neighbours of each atom i. The distance used to calculate each atom's neighbours should be larger than the actual non-bonded cutoff distance so that no atom, initially outside the neighbour cutoff, approaches closer than the non-bonded cutoff distance before the neighbour list is updated again.

It is important to update the neighbour list at the correct frequency. If the update frequency is too high the procedure is inefficient; too low and the energies and forces may be calculated incorrectly due to atoms moving within the non-bonded cutoff region. An update frequency between 10 and 20 steps is common. An algorithm that can automatically update the neighbour list and so circumvent these problems is as follows [Thompson 1983]. An array element is set to zero for each atom whenever the neighbour list is updated. This array is used to store the displacement of each atom or molecule in subsequent steps. When the sum of the maximum displacements of any two atoms exceeds the difference between the non-bonded cutoff distance and the neighbour list distance, then it is time to update the neighbour list again.

There are no fixed rules that determine how much larger the neighbour list cutoff should be than the non-bonded cutoff. Clearly there will be a trade-off between the size of the cutoff and the frequency at which the neighbour list must be updated: the larger the difference, the less frequently will the neighbour list have to be updated. There may also be storage implications if the list is too large.

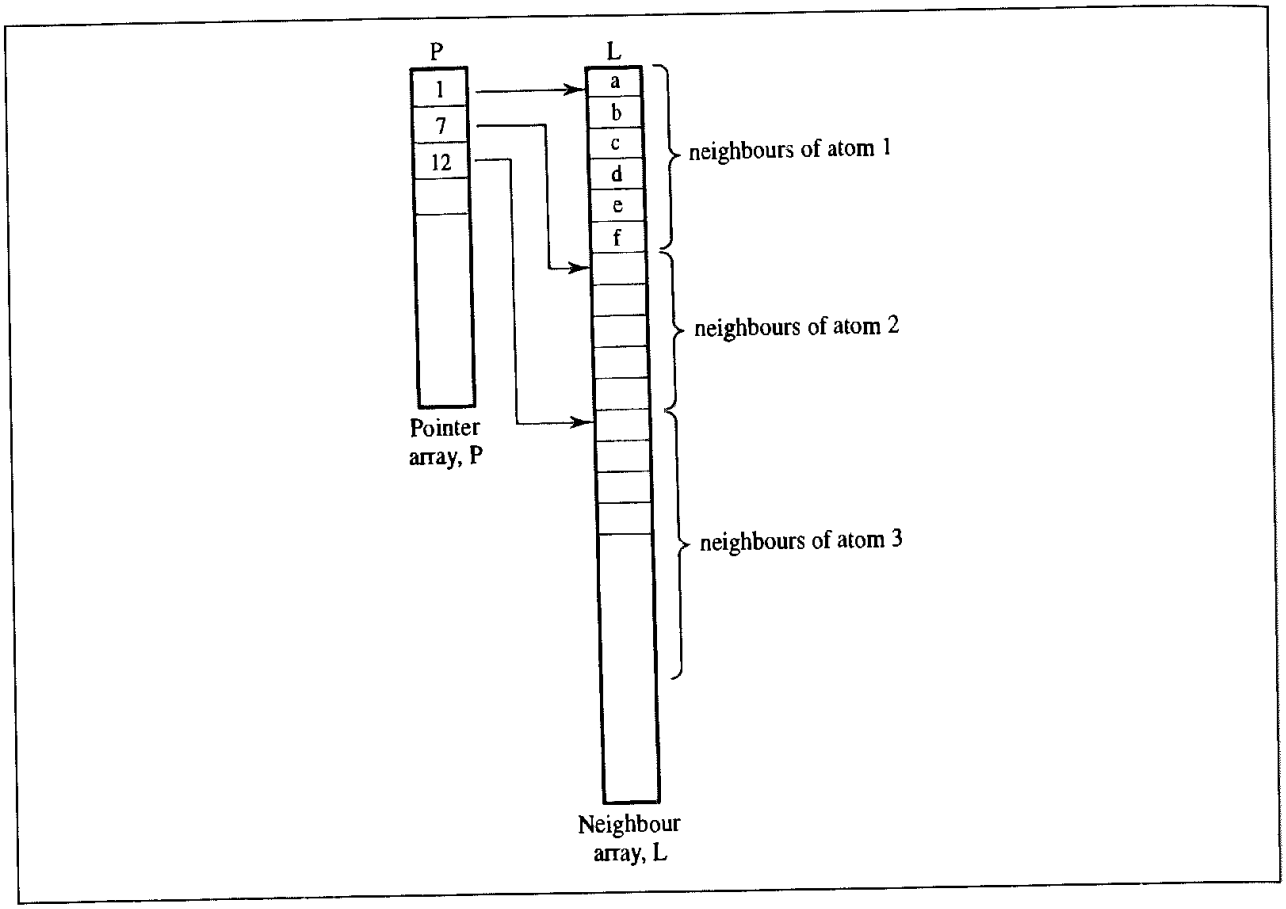

Fig 6.12 Pointer and neighbour arrays can be used to implement the Verlet neighbour list.

When the number of molecules in the simulation is very large, it can require a significant computational effort just to update the neighbour list. This is because the standard way to update the neighbour list requires the distance between all pairs of atoms in the system to be calculated. When the size of the system is much larger than the cutoff distance, a *cell index method* can be used to make the updating procedure more efficient. In the cell index method, the simulation box is divided into a number of cells. The length of each cell is longer than the non-bonded cutoff distance. All of the neighbours of an atom will then be found either in the cell containing the atom or in one of the surrounding cells. If the entire system is divided into M^3 cells, there will be an average of N/M^3 molecules in each cell. To determine the neighbours of a given atom or molecule, it is then necessary to consider just $27N/M^3$ atoms rather than N. The cell index method requires a mechanism for identifying the atoms or molecules in each cell. Two arrays can be used to do this: a linked list array L and a pointer array P. The pointer array indicates the location of one of the atoms or molecules in a given cell. Thus, $P[1]$ would indicate the number of the 'first' atom or molecule in cell 1 and $P[2]$ is the number of the 'first' atom in cell 2. Each element of the linked list array then gives the number of the 'next' atom or molecule in the cell. Thus the value stored in $L[1]$ is the number of the second atom in the first cell. Suppose $P[1]$ is atom 10. Then the value stored in $L[10]$ is the second atom in the cell If this second atom is number 15, then $L[15]$ contains the third atom in the cell. The last molecule in the sequence is identified by the fact that its array element is zero. The cell index method clearly

requires a mechanism for updating the pointer and linked list arrays when atoms or molecules move from one cell to another, which can add to the complexity.

When simulating species with a significant electrostatic contribution, it may be desirable to use different cutoffs for the electrostatic and van der Waals interactions. This is because the electrostatic interaction has a much longer range. Using a longer cutoff for the electrostatic interactions will, of course, significantly increase the number of pairs that must be calculated. A compromise is to use a *twin-range method*, in which two cutoffs are specified. All interactions below the lower cutoff are calculated as normal at each step. Interactions due to atoms between the lower and upper cutoffs are evaluated only when the neighbour list is updated and are kept constant between these updates. The rationale here is that the contribution of the atoms that are further away will not vary much between updates.

The use of a cutoff is amply justified in many cases, if only on the grounds of expediency, but its use will always lead to some fraction of the potential energy being ignored. This lost energy can be easily captured at the end of the simulation if it is assumed that the radial distribution function takes the value of 1 at distances greater than the cutoff. The calculation is analogous to that used to determine the total energy from the radial distribution function, Equation (6.16), but the integration is now performed between the cutoff distance r_c and infinity and $g(r)$ is now taken to be 1 in this range. For N particles, the correction is:

$$E_{\text{correction}} = 2\pi\rho N \int_{r_c}^{\infty} r^2 v(r)\, dr \tag{6.33}$$

For the Lennard-Jones potential the long-range contribution can be determined analytically:

$$E_{\text{correction}} = 8\pi\rho N\varepsilon \left[\frac{\sigma^{12}}{9r^9} - \frac{\sigma^6}{3r^3} \right] \tag{6.34}$$

6.7.2 Group-based Cutoffs

When simulating large molecular systems, it is often advantageous to use a group-based cutoff (sometimes called a residue-based cutoff). Here, the large molecules are divided into 'groups', each of which contains a relatively small number of connected atoms. If the calculation involves small solvent molecules then each solvent molecule is also conveniently regarded as a single unconnected group. Why are groups useful? Let us consider the electrostatic interaction between two water molecules. In the popular TIP3P model there is a charge of $-0\ 834e$ on the oxygen and $0.417e$ on each hydrogen. The electrostatic interaction between two water molecules is calculated as the sum of nine distinct site–site interactions. If we start from the minimum energy arrangement for the water dimer shown in Figure 6.13 and

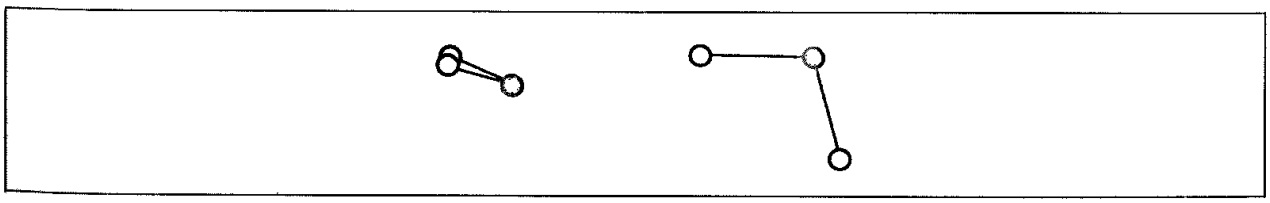

Fig 6.13 Minimum energy structure for water dimer with TIP3P model

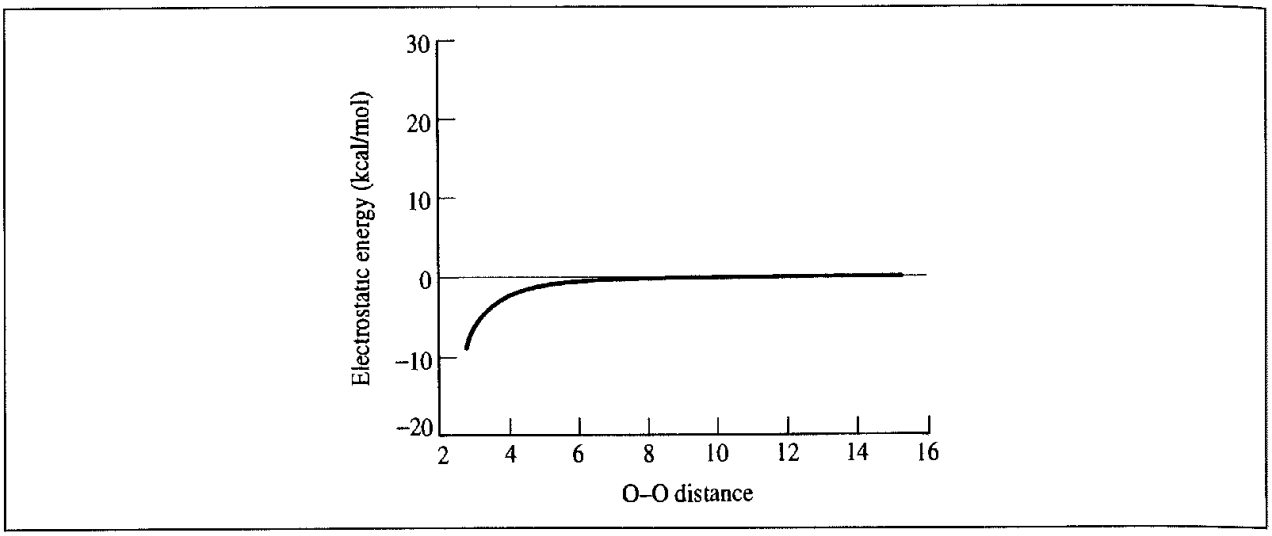

Fig. 6.14: The variation in the electrostatic interaction energy of the water dimer as a function of the O–O distance without a cutoff

gradually move one water molecule relative to the other as indicated then the electrostatic energy varies as shown in the graph in Figure 6.14.

Although the overall interaction energy is relatively small beyond 6 Å or so, each of these energies is the sum of several rather large terms; for example, at an O–O separation of 8 Å, the overall interaction energy is about −0.27 kcal/mol but this comprises an oxygen–oxygen interaction of approximately 29 kcal/mol, oxygen–hydrogen interactions of −59.4 kcal/mol and hydrogen–hydrogen interactions of 29.2 kcal/mol. Suppose that a simple atom-based non-bonded cutoff is applied to the water dimer. The interaction energy then fluctuates violently near the cutoff distance, as shown in Figure 6.15 for a cutoff of 8 Å. This is because only some of the pairwise interactions are included in this case. Clearly such a model would almost certainly lead to serious problems for any

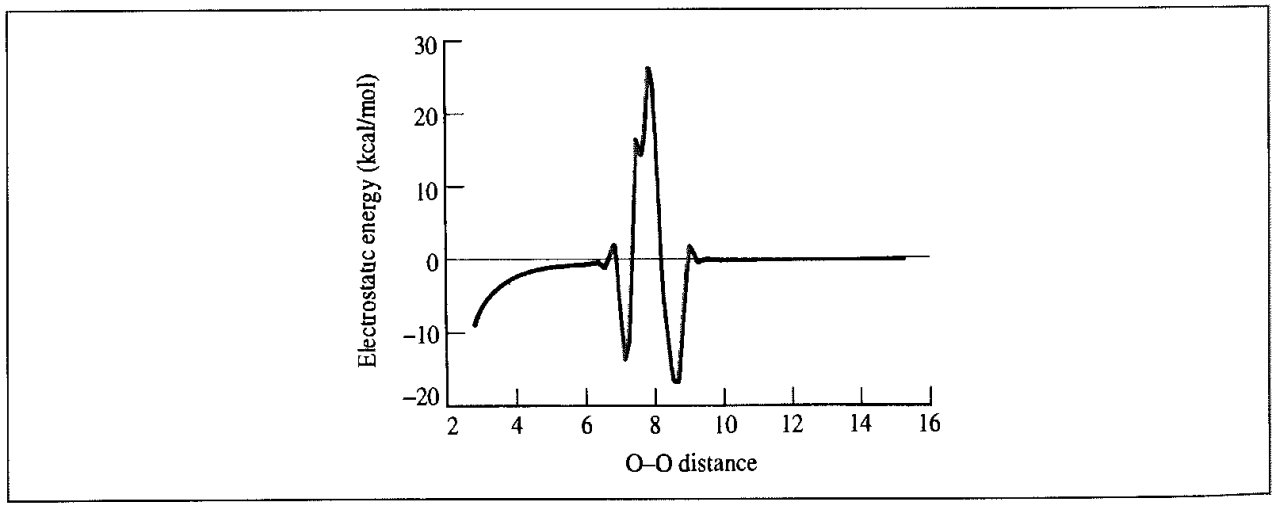

Fig 6.15: The variation in interaction of the water dimer as a function of the O–O distance with an 8 Å atom-based cutoff

simulation. This problem can be avoided by collecting all of the atoms from each water molecule into a single group, and by calculating the interactions on a group-group basis, even though some of the atom pairs may have a separation larger than the cutoff.

How should a molecule be divided into groups? In some cases there may appear to be a chemically obvious way to define the groups, especially when the molecule is a polymer that is constructed from distinct chemical residues. A particularly desirable feature is that each group should, if possible, be of zero charge. The reason for this can be understood if we recall how the different electrostatic interactions vary with distance from Section 4.9.1:

charge-charge $\sim 1/r$
charge-dipole $\sim 1/r^2$
dipole-dipole $\sim 1/r^3$
dipole-quadrupole $\sim 1/r^4$
charge-induced dipole $\sim 1/r^4$
dipole-induced dipole $\sim 1/r^6$

If the groups are electrically neutral, then the leading term in the electrostatic interaction between a pair of groups is the dipole-dipole interaction, which is dependent upon $1/r^3$. By comparison, the charge-charge terms vary as $1/r$. Of course, it is not always possible to arrange atoms in neutral groups as occurs when some of the species are charged.

A further question with the group-based scheme is: how do we decide whether a particular group-group interaction needs to be considered? In other words, how are cutoffs included in the group scheme? One strategy is to include a particular group-group interaction if any pair of atoms in the two groups is closer than the cutoff distance. Alternatively, a 'marker' atom may be nominated within the group; when the marker atoms come closer than the cutoff then the appropriate group-group interaction is included. When using marker atoms, it is important that the groups are not too large; thus the groups used by some simulation programs are much smaller than the 'chemically obvious' groupings. For example, the most obvious choice for proteins and peptides is to define each entire amino acid residue as a single group. However, this is not necessarily the most appropriate strategy. Consider the situation in which two arginine residues are spatially close together (Figure 6.16). Arginine has a long side chain that is comparable in length to the non-bonded cutoff distances often employed. Suppose the alpha-carbon atom (marked C_α in Figure 6.16)

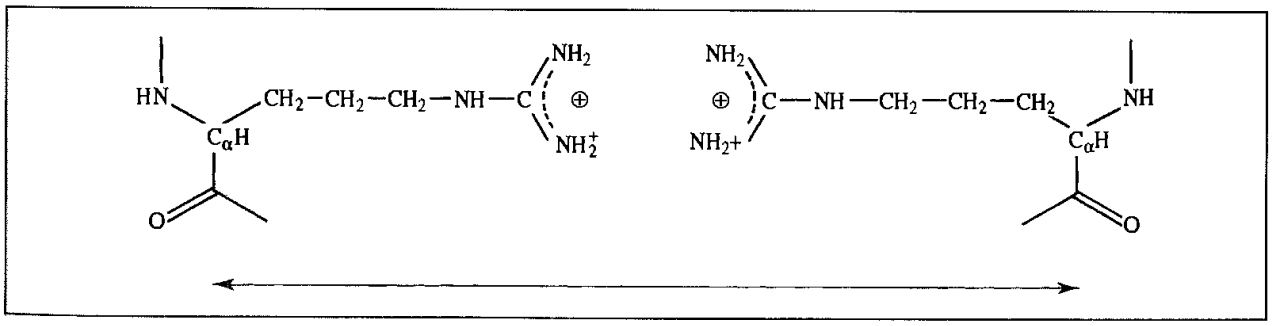

Fig 6.16 The use of a marker atom on the alpha-carbon in an arginine residue may lead to a significant electrostatic interaction being neglected because the distance between the marker atoms exceeds the cutoff

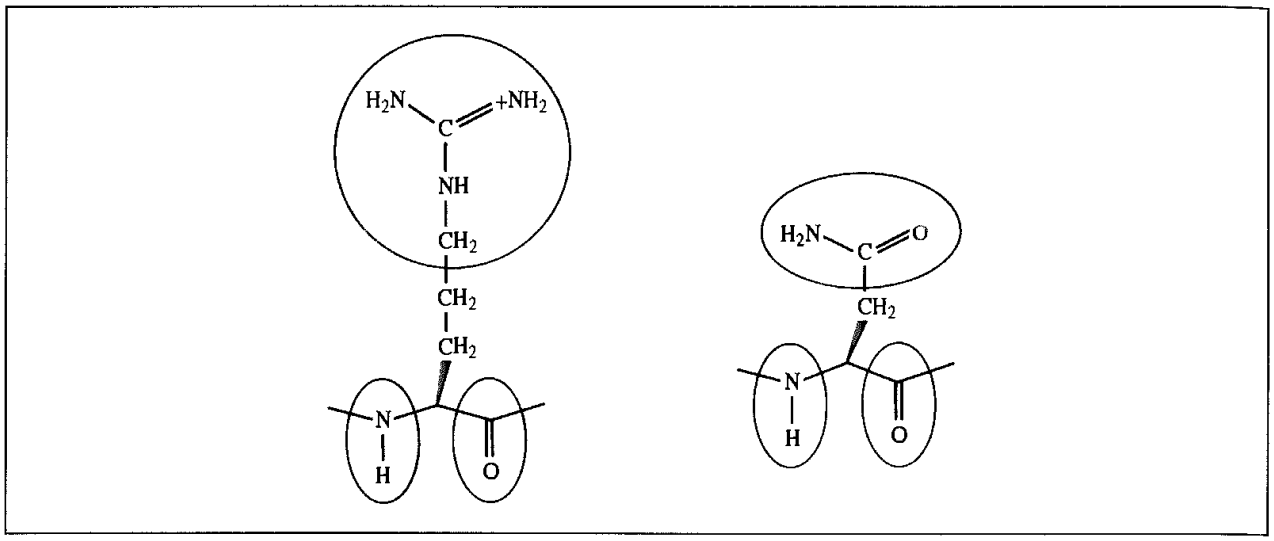

Fig 6 17: The charge groups used in the GROMOS simulation program for simulating proteins [van Gunsteren and Berendsen 1986], illustrated using the amino acids arginine and asparaginine The CH_2 groups have zero charge.

in each arginine residue is chosen as the marker atom. If the distance between the alpha-carbons of two arginine residues is greater than the cutoff, no interactions between any atoms in the arginine residues would be calculated, despite the fact that the positively charged ends of the residues could be very close, as shown in Figure 6.16. Were the alpha-carbons to approach closer than the cutoff, there would then be a dramatic increase in the energy due to the unfavourable interaction between the two side chains, inevitably leading to an unstable simulation. It may therefore be appropriate to define 'charge groups' that contain smaller numbers of atoms than are in the chemically obvious scheme. For example, the groups that are used by the GROMOS simulation program for the amino acids arginine and asparagine are shown in Figure 6.17.

6.7.3 Problems with Cutoffs and How to Avoid Them

A cutoff introduces a discontinuity in both the potential energy and the force near the cutoff value. This creates problems, especially in molecular dynamics simulations where energy conservation is required. There are several ways that the effects of this discontinuity can be counteracted. One approach is to use a shifted potential, in which a constant term is subtracted from the potential at all values (Figure 6.18):

$$v'(r) = v(r) - v_c \quad r \leq r_c \tag{6.35}$$

$$v'(r) = 0 \quad r > r_c \tag{6.36}$$

where r_c is the cutoff distance and v_c is equal to the value of the potential at the cutoff distance. As the additional term is constant, it disappears when the potential is differentiated and so does not affect the force calculation in molecular dynamics. Use of the shifted potential does improve energy conservation, though as the number of atom pairs separated by a distance smaller than the cutoff varies, so the contribution of the shifted potential to the total energy will change. An additional problem is that there is a discontinuity in the force

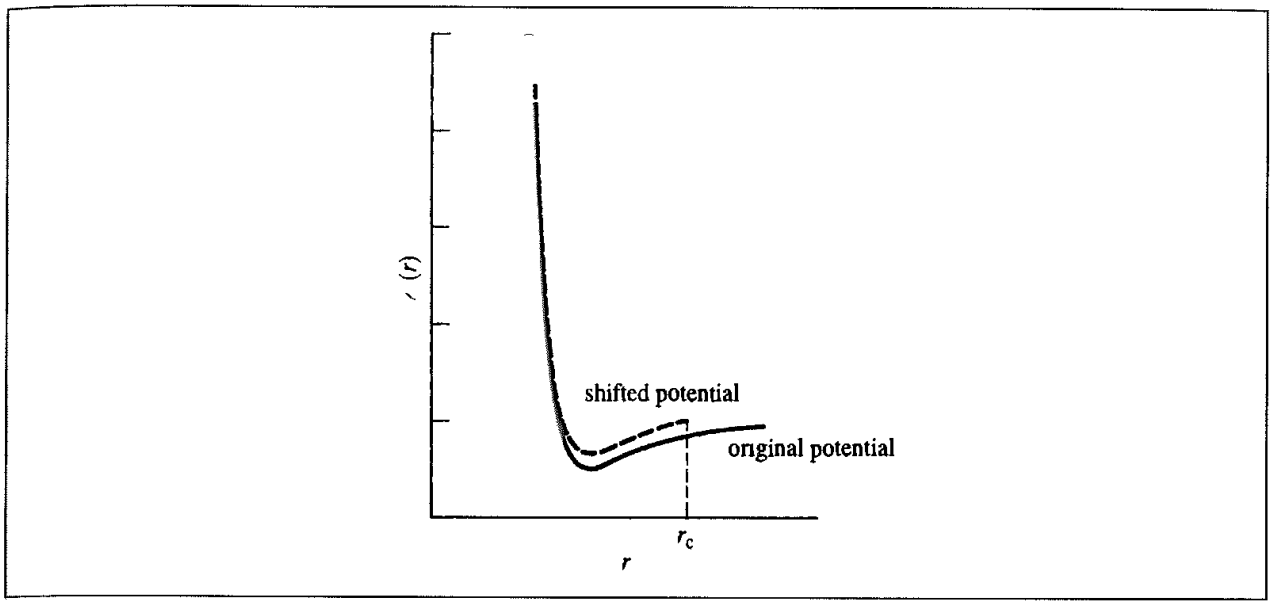

Fig 6 18 A shifted Lennard-Jones potential.

with the shifted potential; at the cutoff distance, the force will have a finite value which drops suddenly to zero just beyond the cutoff. This can also give instabilities in a simulation. To avoid this, a linear term can be added to the potential, making the derivative zero at the cutoff:

$$v'(r) = v(r) - v_c - \left(\frac{dv(r)}{dr}\right)_{r=r_c}(r - r_c) \quad r \leq r_c \tag{6.37}$$

$$v'(r) = 0 \qquad r > r_c \tag{6.38}$$

The shift makes the potential deviate from the 'true' potential, and so any calculated thermodynamic properties will be changed. The 'true' values can be retrieved but it is difficult to do so, and the shifted potential is thus rarely used in 'real' simulations. Moreover, while it is relatively straightforward to implement for a homogeneous system under the influence of a simple potential such as the Lennard-Jones potential, it is not easy for inhomogeneous systems containing many different types of atom.

An alternative way to eliminate discontinuities in the energy and force equations is to use a *switching function*. A switching function is a polynomial function of the distance by which the potential energy function is multiplied. Thus the switched potential $v'(r)$ is related to the true potential $v(r)$ by $v'(r) = v(r)S(r)$. Some switching functions are applied to the entire range of the potential up to the cutoff point. One such function is:

$$v'(r) = v(r)\left[1 - 2\left(\frac{r}{r_c}\right)^2 + \left(\frac{r}{r_c}\right)^4\right] \tag{6.39}$$

The switching function has a value of 1 at $r = 0$ and a value of 0 at $r = r_c$, the cutoff distance. Between these two values it varies as shown in Figure 6.19, which also shows how the potential function is affected.

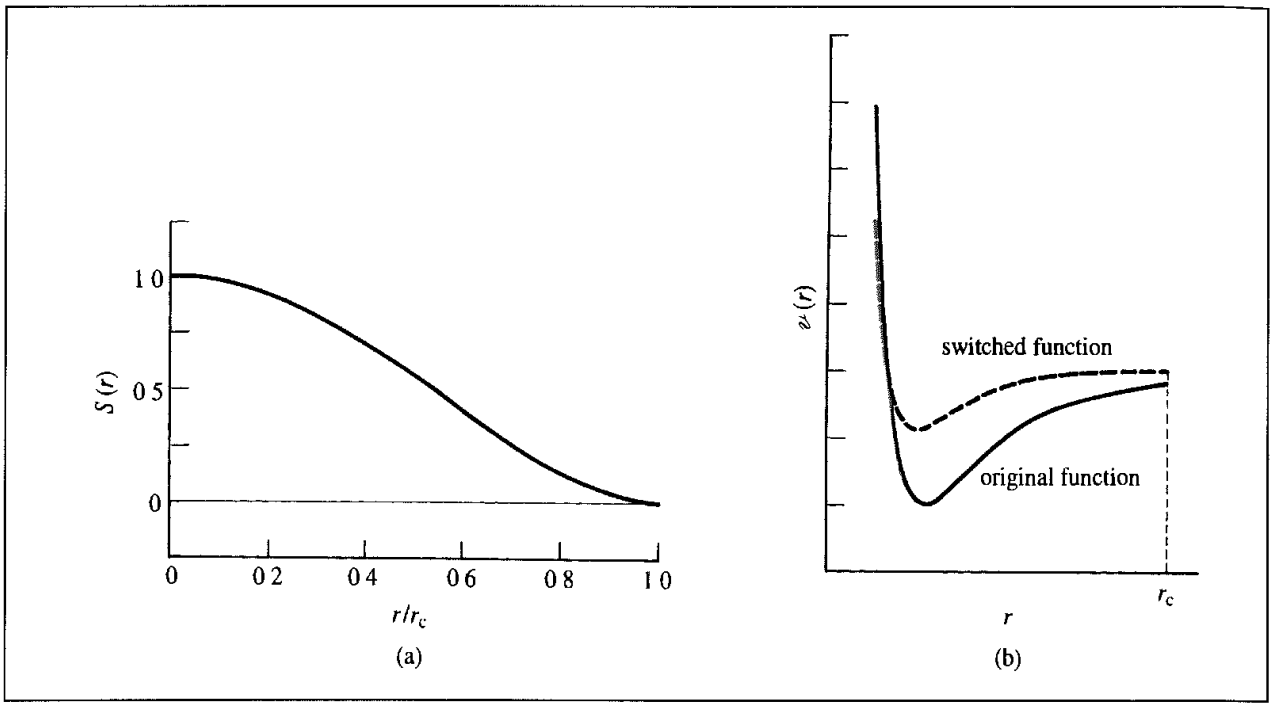

Fig 6.19 (a) The effect of a switching function that applies over the entire range and (b) its effect on the Lennard-Jones potential

A switching function applied to the potential function over the entire range does have drawbacks; for example, equilibrium structures are affected (the minimum energy separation for the argon dimer decreases slightly). A more acceptable alternative is to gradually taper the potential between two cutoff values. The potential takes its usual value until the lower cutoff distance. Between the lower (r_l) and upper cutoff distance (r_u) the potential is multiplied by the switching function, which takes the value 1 at the lower cutoff distance and 0 at the upper cutoff distance. The lower cutoff distance is typically relative close to the upper cutoff distance (for example, r_l might be 9 Å and r_u 10 Å). A simple switching function has the following linear form:

$$S = 1\,0 \qquad r_{ij} < r_l \tag{6.40}$$

$$S = (r_u - r_{ij})/(r_u - r_l) \qquad r_l \leq r_{ij} \leq r_u \tag{6.41}$$

$$S = 0.0 \qquad r_u < r_{ij} \tag{6.42}$$

This suffers from discontinuities in both the energy and the force at the two cutoff values A more acceptable switching function smoothly changes from a value of 1 to a value of 0 (Figure 6.20) between r_l and r_u and satisfies the following requirements:

$$S_{r=r_l} = 1; \qquad \left(\frac{dS}{dr}\right)_{r=r_l} = 0; \qquad \left(\frac{d^2S}{dr^2}\right)_{r=r_l} = 0 \tag{6.43}$$

$$S_{r=r_u} = 0; \qquad \left(\frac{dS}{dr}\right)_{r=r_u} = 0; \qquad \left(\frac{d^2S}{dr^2}\right)_{r=r_u} = 0 \tag{6 44}$$

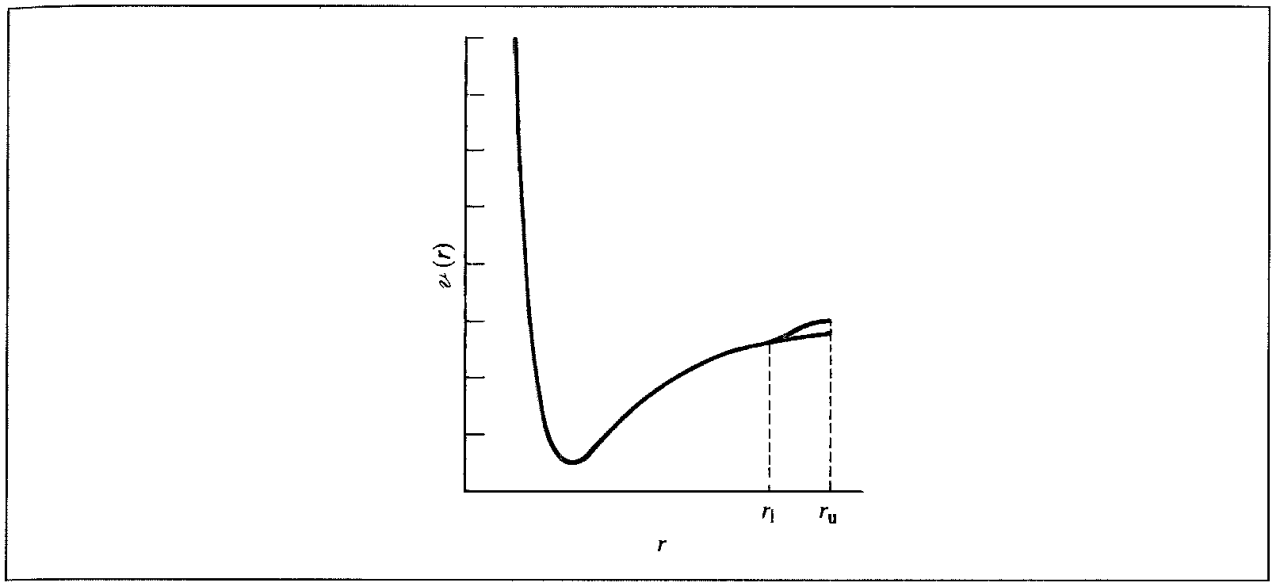

Fig. 6 20 A switching function that applies over a narrow range near the cutoff and its effect on the Lennard-Jones potential.

By ensuring that the first derivative is zero at the endpoints the force also approaches zero smoothly. A continuous second derivative is required to ensure that the integration algorithm works properly. If the switch function is assumed to take the following form:

$$S(r) = c_0 + c_1\left[\frac{r - r_l}{r_u - r_l}\right] + c_2\left[\frac{r - r_l}{r_u - r_l}\right]^2 + c_3\left[\frac{r - r_l}{r_u - r_l}\right]^3 + c_4\left[\frac{r - r_l}{r_u - r_l}\right]^4 + c_5\left[\frac{r - r_l}{r_u - r_l}\right]^5 \quad (6.45)$$

then the following values of the coefficients $c_0 \ldots c_5$ satisfy the six requirements in equations (6.43) and (6.44):

$$c_0 = 1; \quad c_1 = 0; \quad c_2 = 0; \quad c_3 = -10; \quad c_4 = 15; \quad c_5 = -6 \quad (6.46)$$

When using a switching function in a molecular simulation with a residue-based cutoff it is important that the function has the same value for all pairs of atoms in the two interacting groups. Otherwise, severe fluctuations in the energy can arise when the separation is within the cutoff region. These two contrasting situations can be formally expressed as follows:

$$\textit{atom based:} \quad \mathscr{V}_{AB} = \sum_{i=1}^{N_A} \sum_{j=1}^{N_B} S_{ij}(r_{ij}) v_{ij}(r_{ij}) \quad (6.47)$$

$$\textit{residue or molecule based:} \quad \mathscr{V}_{AB} = S_{AB}(|\mathbf{r}_A - \mathbf{r}_B|) \sum_{i=1}^{N_A} \sum_{j=1}^{N_B} v_{ij}(r_{ij}) \quad (6\,48)$$

N_A and N_B are the numbers of atoms in the two groups A and B and S is the switching function. With the group-based switching function, it is necessary to define the 'distance' between the two groups (i.e. the two points $\mathbf{r}_A$ and $\mathbf{r}_B$). There is no definitive way to do this. As with cutoffs, a special marker atom can be nominated within each residue, or the centre of mass, centre of geometry or centre of charge may be used.

Group-based switching functions have several advantages. Better energy conservation can be achieved, and there are advantages when performing energy minimisation, since the potential is defined analytically at all points. However, it is important to beware of possible problems with group-based switching functions when the groups are large. We have already seen how this can arise when an ordinary group-based cutoff is used. Let us re-examine our arginine problem (Figure 6.16) when a switching function is employed. When the two marker atoms have a separation only slightly less than the upper switch cutoff, the switching function will be close to zero, and so there will not be the dramatic increase in energy that is observed with the simple cutoff. Nevertheless, although the switching function does help to prevent the simulation from 'blowing up', the representation of the energy and the forces in the system is still unsatisfactory. The only real alternative is to make the groups smaller or dispense with cutoffs altogether.

6.8 Long-range Forces

Those interactions that decay no faster than r^{-n}, where n is the dimensionality of the system, can be a problem as their range is often greater than half the box length. The charge–charge interaction, which decays as r^{-1}, is particularly problematic in molecular simulations. There is much evidence that it is important to properly model these long-range forces, which are particularly acute when simulating charged species such as molten salts (when it is not possible to construct neutral groups). A proper treatment of long-range forces can also be important when calculating certain properties, such as the dielectric constant. One way to tackle the errors introduced by an inadequate treatment of long-range forces would be to use a much larger simulation cell, but this is usually impractical. Nevertheless, increasing computer power does mean that more rigorous ways of dealing with long-range forces can be considered, even in simulations of large systems. A variety of methods have been developed to handle long-range forces. The methods that we will discuss in detail are the Ewald summation, the reaction field method and the cell multiple method.

6.8.1 The Ewald Summation Method

The Ewald sum was first devised by Ewald [Ewald 1921] to study the energetics of ionic crystals. In this method, a particle interacts with all the other particles in the simulation box and with all of their images in an infinite array of periodic cells. Figure 6.21 illustrates how the array of simulation cells is constructed; in the limit, the cell array is considered to have a spherical shape. The position of each image box (assumed for simplicity to be a cube of side L containing N charges) can be related to the central box by specifying a vector, each of whose components is an integral multiple of the length of the box, $(\pm iL, \pm jL, \pm kL)$; $i,j,k = 0,1,2,3$, etc. The charge–charge contribution to the potential energy due to all pairs of charges in the central simulation box can be written:

$$\mathscr{V} = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\frac{q_i q_j}{4\pi\varepsilon_0 r_{ij}} \tag{6.49}$$

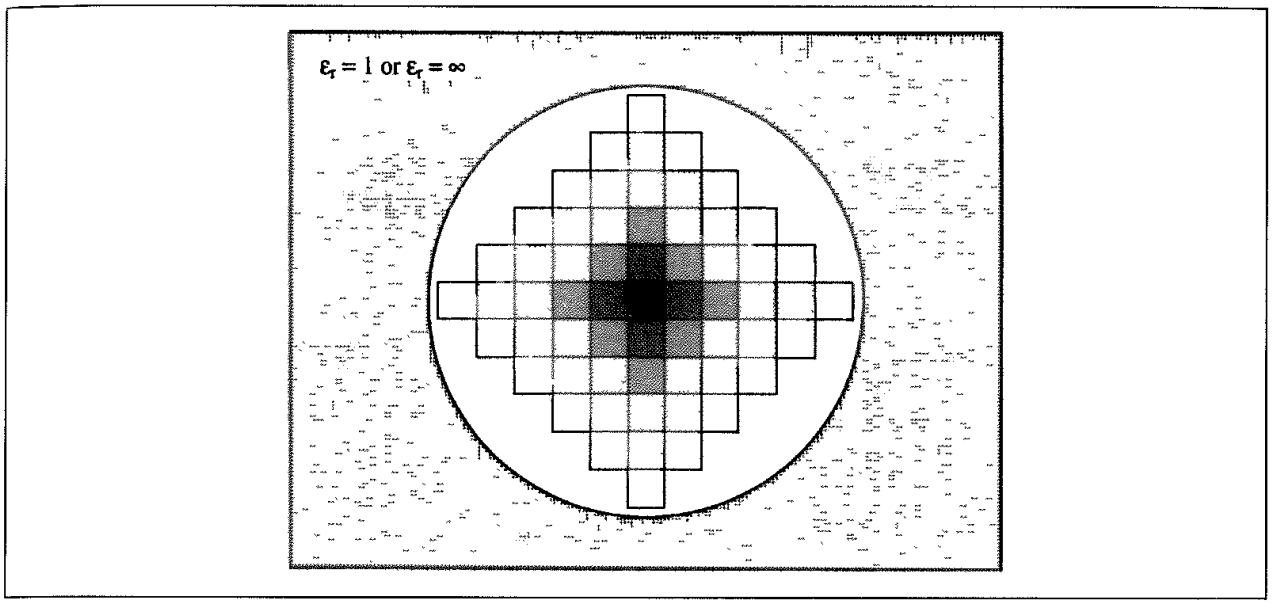

Fig. 6.21: The construction of a system of periodic cells in the Ewald method. (Figure adapted from Allen M P and D J Tildesley 1987 Computer Simulation of Liquids *Oxford, Oxford University Press.)*

where r_{ij} is the minimum distance between the charges i and j. There are six boxes at a distance L from the central box with coordinates ($\mathrm{r_{box}}$) given by $(0,0,L)$, $(0,0,-L)$, $(0,L,0)$, $(0,-L,0)$, $(L,0,0)$ and $(-L,0,0)$ (only four of these are shown in the two-dimensional picture in Figure 6 21). The contribution of the charge–charge interaction between the charges in the central box and all images of all particles in these six surrounding boxes is given by:

$$\mathscr{V} = \frac{1}{2}\sum_{n\,\mathrm{box}=1}^{6}\sum_{i=1}^{N}\sum_{j=1}^{N}\frac{q_i q_j}{4\pi\varepsilon_0|\mathbf{r}_{ij}+\mathbf{r}_{\mathrm{box}}|} \tag{6.50}$$

In general, for a box which is positioned at a cubic lattice point $\mathbf{n}$ ($= (n_xL, n_yL, n_zL)$ with n_x, n_y, n_z being integers):

$$\mathscr{V} = \frac{1}{2}\sum_{\mathbf{n}}\sum_{i=1}^{N}\sum_{j=1}^{N}\frac{q_i q_j}{4\pi\varepsilon_0|\mathbf{r}_{ij}+\mathbf{n}|} \tag{6.51}$$

$|\mathbf{n}|$ thus takes the values $1, \sqrt{2}, \ldots$. This expression is often written in such a way to incorporate the interactions between pairs of charges in the central box (for which $|\mathbf{n}| = 0$):

$$\mathscr{V} = \frac{1}{2}\sum_{|\mathbf{n}|=0}^{\prime}\sum_{i=1}^{N}\sum_{j=1}^{N}\frac{q_i q_j}{4\pi\varepsilon_0|\mathbf{r}_{ij}+\mathbf{n}|} \tag{6.52}$$

The prime on the first summation indicates that the series does not include the interaction $i = j$ for $\mathbf{n} = 0$.

There is thus a contribution to the total energy from the interactions in the central box together with the interactions between the central box and all image boxes. There is also a contribution from the interaction between the spherical array of boxes and the surrounding

medium. The problem is that the summation in Equation (6.52) converges extremely slowly and in fact is *conditionally convergent*. A conditionally convergent series contains a mixture of positive and negative terms such that the positive terms alone form a divergent series (i.e. a series which does not have a finite sum) as do the negative terms when taken alone. The sum of a conditionally convergent series depends on the order in which its terms are considered. An additional problem with the Coulomb interaction is that it can vary rapidly at small distances.

The trick when calculating the Ewald sum is to convert the summation into two series, each of which converges much more rapidly. The mathematical foundation for this is the following identity:

$$\frac{1}{r} = \frac{f(r)}{r} + \frac{1-f(r)}{r} \tag{6.53}$$

The aim is thus to choose an appropriate function $f(r)$ which will deal with the rapid variation of $1/r$ at small r and the slow decay at long r. In the Ewald method each charge is considered to be surrounded by a neutralising charge distribution of equal magnitude but of opposite sign, as shown in Figure 6.22. A Gaussian charge distribution of the following functional form is commonly used:

$$\rho_i(\mathbf{r}) = \frac{q_i\alpha^3}{\pi^{3/2}}\exp(-\alpha^2 r^2) \tag{6.54}$$

The sum over point charges is now converted to a sum of the interactions between the charges *plus* the neutralising distributions This dual summation (the 'real space'

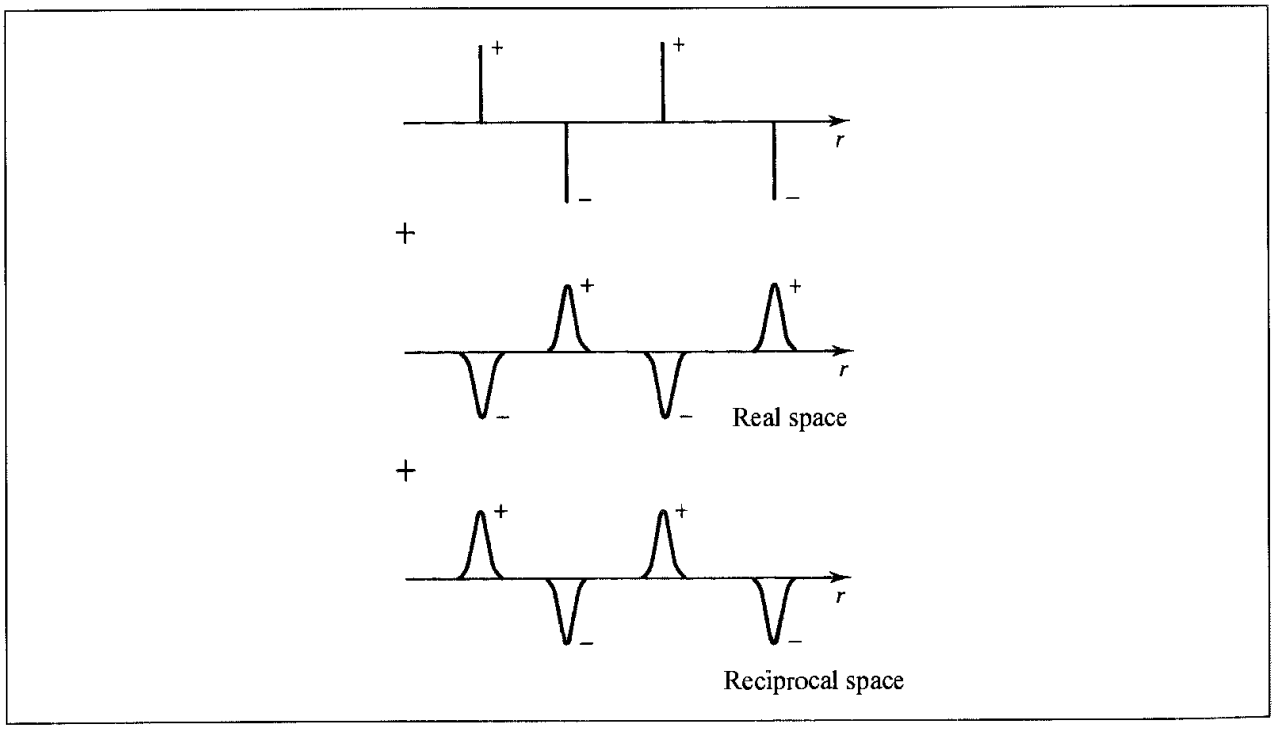

Fig 6 22 In the Ewald summation method the initial set of charges are surrounded by a Gaussian distribution (calculated in real space) to which a cancelling change distribution must be added (calculated in reciprocal space)

summation) is given by:

$$\mathscr{V} = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{|\mathbf{n}|=0}{}' \frac{q_i q_j}{4\pi\varepsilon_0}\frac{\mathrm{erfc}(\alpha|\mathbf{r}_{ij}+\mathbf{n}|)}{|\mathbf{r}_{ij}+\mathbf{n}|} \tag{6.55}$$

erfc is the complementary error function, which is:

$$\mathrm{erfc}(x) = \frac{2}{\sqrt{\pi}}\int_x^{\infty} \exp(-t^2)dt \tag{6.56}$$

The Ewald method thus uses erfc(r) for the function $f(r)$ in Equation (6.53). The crucial point is that this new summation involving the error function converges very rapidly and beyond some cutoff distance its value can be considered negligible. The rate of convergence depends upon the width of the cancelling Gaussian distributions; the wider the Gaussian, the faster the series converges. Specifically, α should be chosen so that the only terms in the series (6.55) are those for which $|\mathbf{n}| = 0$ (i.e. only pairwise interactions involving charges in the central box, or if a cutoff is used α is chosen so that only interactions with other charges within the cutoff are included). A second charge distribution is now added to the system which exactly counteracts the first neutralising distribution (Figure 6 22). The contribution from this second charge distribution is:

$$\mathscr{V} = \frac{1}{2}\sum_{k\neq 0}\sum_{i=1}^{N}\sum_{j=1}^{N}\frac{1}{\pi L^3}\frac{q_1 q_j}{4\pi\varepsilon_0}\frac{4\pi^2}{k^2}\exp\left(-\frac{k^2}{4\alpha^2}\right)\cos(\mathbf{k}\cdot\mathbf{r}_{ij}) \tag{6.57}$$

This summation is performed in *reciprocal space*, the details of which need not concern us here. The vectors $\mathbf{k}$ are reciprocal vectors and are given by $\mathbf{k} = 2\pi\mathbf{n}/L$. This reciprocal sum also converges much more rapidly than the original point-charge sum. However, the number of terms that must be included increases with the width of the Gaussians. There is thus a clear need to balance the real-space and reciprocal-space summations; the former converges more rapidly for large α, whereas the latter converges more rapidly for small α. A value for α of $5/L$ and 100–200 reciprocal vectors $\mathbf{k}$ have been suggested as providing acceptable results. This reciprocal space summation corresponds to the second term $([1-f(r)]/r)$ in Equation (6.53); the requirement for this term is that it is a slowly varying function for all r. As such, its Fourier transform (which is what the summation is) can be represented by a small number of reciprocal vectors. The sum of Gaussian functions in real space includes the interaction of each Gaussian with itself. A third self-term must therefore be subtracted:

$$\mathscr{V} = -\frac{\alpha}{\sqrt{\pi}}\sum_{k=1}^{N}\frac{q_k^2}{4\pi\varepsilon_0} \tag{6.58}$$

A fourth correction term may also be required, depending upon the medium that surrounds the sphere of simulation boxes. If the surrounding medium has an infinite relative permittivity (e.g. if it is a conductor) then no correction term is required. However, if the surrounding medium is a vacuum (with a relative permittivity of 1) then the following energy must be added:

$$\mathscr{V}_{\text{correction}} = \frac{2\pi}{3L^3}\left|\sum_{i=1}^{N}\frac{q_i}{4\pi\varepsilon_0}\mathbf{r}_i\right|^2 \tag{6 59}$$

The final expression is thus:

$$\mathscr{V} = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\left\{\begin{array}{l} \sum_{|\mathbf{n}|=0}^{\infty\prime}\frac{q_iq_j}{4\pi\varepsilon_0}\frac{\mathrm{erfc}(\alpha|\mathbf{r}_{ij}+\mathbf{n}|)}{|\mathbf{r}_{ij}+\mathbf{n}|} \\ +\sum_{k\neq 0}\frac{1}{\pi L^3}\frac{q_iq_j}{4\pi\varepsilon_0}\frac{4\pi^2}{k^2}\exp\left(-\frac{k^2}{4\alpha^2}\right)\cos(\mathbf{k}\cdot\mathbf{r}_{ij}) \\ -\frac{\alpha}{\sqrt{\pi}}\sum_{k=1}^{N}\frac{q_k^2}{4\pi\varepsilon_0}+\frac{2\pi}{3L^3}\left|\sum_{k=1}^{N}\frac{q_k}{4\pi\varepsilon_0}r_k\right|^2 \end{array}\right. \tag{6.60}$$

The Ewald sum is the most 'correct' way yet devised to accurately include all the effects of long-range forces in a computer simulation. It has been extensively used in simulations involving highly charged systems (such as ionic melts and in studies of processes in and on solids) and is increasingly being applied to other systems where electrostatic effects are important, such as lipid bilayers, proteins and DNA. Nevertheless, the Ewald method is not without problems and it tends to reinforce artefacts that arise from imposing periodic boundary conditions. For example, the method artificially results in each charge–charge interaction being minimised at a separation of half the box length. Instantaneous fluctuations in the simulation cell tend to be replicated throughout the infinite system rather than being damped out.

The Ewald summation is computationally quite expensive to implement. Under conditions of constant α (which will give the same density of reciprocal vectors, $\mathbf{k}$) then it scales as the square of the number of particles in the central simulation cell. If α is allowed to vary then the algorithm can be made to scale as $N^{3/2}$ though the consequent value of α might make the range of the Coulomb potential incompatible with the range of the van der Waals interactions. Several methods have been proposed to speed up the computationally demanding reciprocal space part of the calculation, such as the use of polynomial approximations but these do not solve the unfavourable N^2 scaling. The most promising way to tackle this difficulty is to modify the problem so that the fast Fourier transform (FFT) can be used to compute the reciprocal space summation. The fast Fourier algorithm scales as $N \ln N$, which gives considerable advantages over the N^2 alternative. If, in addition, a sufficiently large value of α is chosen such that the interatomic interaction is negligible for r_{ij} greater than a cutoff (e.g. 9 Å) then the real-space summation is reduced to order N and the order of the entire algorithm becomes $N \ln N$.

As outlined in Section 1.10.8, the FFT method requires that the data are not continuous but are discrete values. In order to employ the fast Fourier transform in the Ewald summation the point charges with their continuous coordinates must be replaced by a grid-based charge distribution. Each of the atomic point charges must thus be distributed among the surrounding grid points in some fashion so as to reproduce the potential of the charge at the original location. As usual an element of compromise is required; the more surrounding points that are used the more accurately the potential of the charge at the original location can be approximated but the greater the computational cost per particle. A popular approach is the particle-mesh method of Hockney and Eastwood [Hockney and Eastwood 1988], which uses the nearest 27 points in three dimensions. From this gridded charge density it is possible

to calculate (through use of the FFT algorithm) the potential due to the Gaussian distributions at the grid points, which by interpolation gives rise to the desired potential at (and thus the forces on) each of the particles. A number of variants on this general theme have been described, all of which use the fast Fourier transform algorithm but which differ in other aspects of their implementation. These include the particle–mesh Ewald method [Darden *et al.* 1993] and the particle–particle–particle–mesh approach [Hockney and Eastwood 1988; Luty *et al.* 1994, 1995]. Deserno and Holm presented a unification of these methods and also demonstrated that although very similar in spirit they could have very different accuracies [Deserno and Holm 1998a, b]. The particle–particle–particle–mesh approach was generally preferred as it was believed to be more flexible.

The Ewald method has been widely used to study highly polar or charged systems. Its use is considered routine for many types of solid-state materials. It is increasingly used for calculations on much larger molecular systems, such as proteins and DNA, due both to the increases in computer performance and to the new methodological advances we have just discussed [Darden *et al.* 1999]. For example, an early application of the particle-mesh Ewald method was the molecular dynamics simulation of a crystal of the protein bovine pancreatic trypsin inhibitor [York *et al.* 1994]. The full crystal environment was reproduced, with four protein molecules in the unit cell, together with associated water molecules and chloride counterions. Over the course of the 1 ns simulation the deviation of the simulated structures from the initial crystallographic structure was monitored. Once equilibrium was achieved this deviation (measured as the root-mean-square positional deviation) settled down to a value of 0.63 Å for all non-hydrogen atoms and 0.52 Å for the backbone atoms alone. By contrast, an equivalent simulation run with a 9 Å residue-based cutoff showed a deviation of more than 1.8 Å. In addition, the atomic fluctuations calculated from the Ewald simulation were in close agreement with those derived from the crystallographic temperature factors, unlike the non-Ewald simulation, which was significantly overestimated due to the use of the electrostatic cutoff. The highly charged nature of DNA makes it particularly important to deal properly with the electrostatic interactions and simulations using the particle–mesh Ewald approach are often much more stable, with the trajectories remaining much closer to the experimental structures [Cheatham *et al.* 1995].

6.8.2 The Reaction Field and Image Charge Methods

In the reaction field method, a sphere is constructed around the molecule with a radius equal to the cutoff distance. The interaction with molecules that are within the sphere is calculated explicitly. To this is added the energy of interaction with the medium beyond the sphere, which is modelled as a homogeneous medium of dielectric constant ε_s (Figure 6.23). The electrostatic field due to the surrounding dielectric is given by:

$$\mathrm{E}_i = \frac{2(\varepsilon_s - 1)}{\varepsilon_s + 1}\left(\frac{1}{r_c^3}\right)\sum_{j;r_{ij}\le r_c}\boldsymbol{\mu}_j \tag{6.61}$$

where $\boldsymbol{\mu}_j$ are the dipoles of the neighbouring molecules that are within the cutoff distance (r_c) of the molecule i. The interaction between the molecule i and the reaction field equals $\mathbf{E}_i \cdot \boldsymbol{\mu}_i$,

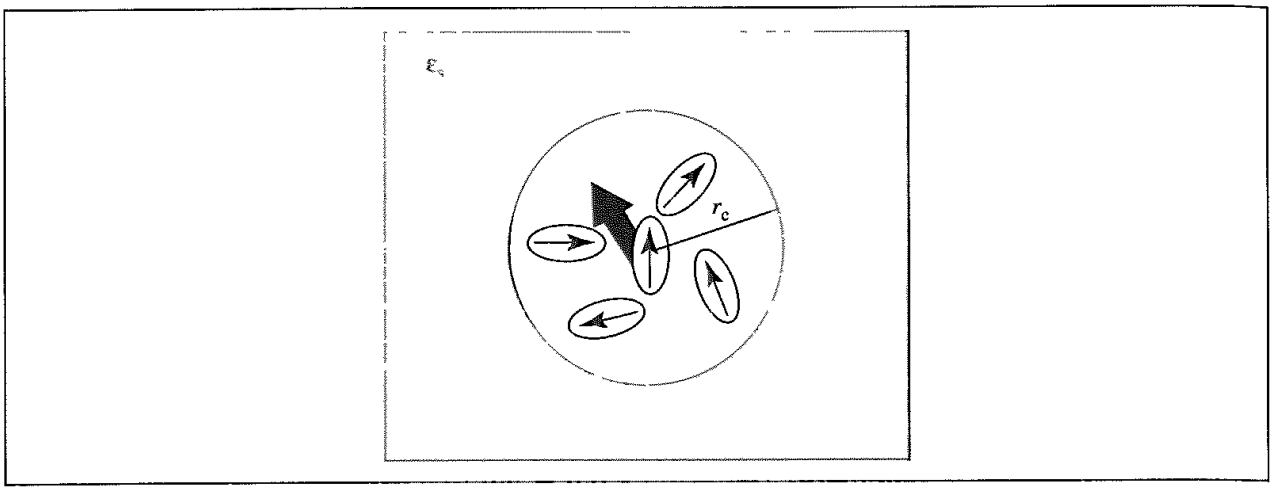

Fig. 6 23 The reaction field method The shaded arrow represents the sum of the dipoles of the other molecules within the cutoff sphere.

which is added to the short-range molecule–molecule interaction. Problems with the reaction field method may arise from discontinuities in the energy and/or force when the number of molecules j within the cavity of the molecule i changes. These problems can be avoided by employing a switching function for molecules that are near the reaction field boundary.

Similar approaches employ a single boundary for the entire system. This boundary may be spherical or may have a more complicated shape that better approximates the true molecular surface of the molecule. In the *image charge method,* a spherical boundary is employed and the reaction field due to a charge inside the boundary is considered to arise from a so-called image charge situated in the continuous dielectric beyond the sphere (Figure 6.24) [Friedman 1975]. If the position of the charge is $\mathbf{r}_i$ then the image charge is located at $(R/r_i)^2\mathbf{r}_i$ (where R is the radius of the bounding sphere) and has magnitude:

$$q_{\text{im}} = -\frac{(\varepsilon_s - \varepsilon_r)}{(\varepsilon_s + \varepsilon_r)} \frac{q_i R}{r_i} \tag{6.62}$$

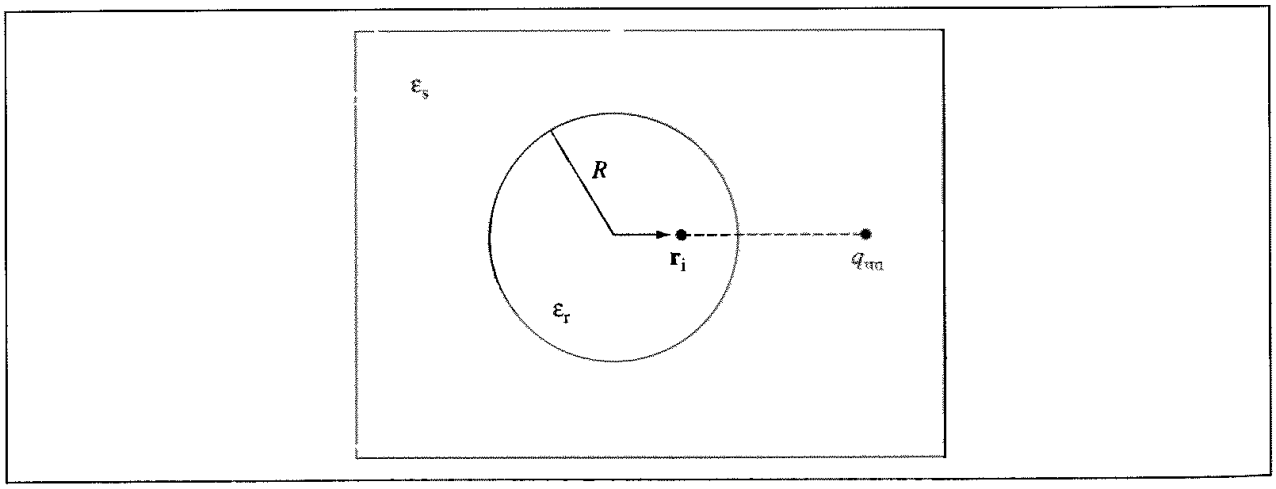

Fig 6 24: The image charge method

where ε_r and ε_s are the dielectric constants inside and outside the boundary, respectively. This expression holds if the dielectric constant beyond the boundary is much greater than that inside ($\varepsilon_s \gg \varepsilon_r$). A drawback with this method is that as a charge approaches the boundary, so too does its image and the method breaks down.

The reaction field and image charge methods have the advantages of being conceptually simple, relatively easy to implement and computationally efficient. However, they do rely upon the assumption that molecules beyond the cutoff can be modelled as a continuous dielectric. This is not necessarily the case but is often a reasonable assumption for homogeneous fluids. A value for the dielectric of the surrounding continuum must also be specified. This can be taken from experimental data, but the dielectric constant may be the property that one is trying to calculate! In practice, it is often necessary only to ensure that $\varepsilon_r \leq \varepsilon_s \leq \infty$. There are several ways in which the dielectric constant can be calculated from a computer simulation. A common approach is via the average of the square of the total dipole moment of the system, $\langle \mathbf{M}^2 \rangle$. With a reaction field boundary the dielectric constant ε_r is given by:

$$\frac{4\pi}{9}\frac{\langle \mathbf{M}^2 \rangle}{VkTB} = \frac{(\varepsilon_r - 1)}{3}\frac{(2\varepsilon_s + 1)}{(2\varepsilon_s + \varepsilon_r)} \tag{6.63}$$

where V is the volume of the simulation system. Even though the value of $\langle \mathbf{M}^2 \rangle$ can vary quite considerably with the reaction field dielectric ε_s, almost identical values of ε_r are obtained. An alternative approach is to determine the polarisation response of the liquid to an electric field $\mathbf{E}_0$. If the average dipole moment per unit volume along the direction of the applied field is $\langle \mathbf{P} \rangle$ then the dielectric constant is given by:

$$\frac{4\pi}{3}\frac{\langle \mathbf{P} \rangle}{\mathbf{E}_0} = \frac{(\varepsilon_r - 1)}{3}\frac{(2\varepsilon_s + 1)}{(2\varepsilon_s + \varepsilon_r)} \tag{6 64}$$

This perturbation method is claimed to be more efficient than the fluctuating dipole method, at least for certain water models [Alper and Levy 1989], but it is important to ensure that the polarisation $\langle \mathbf{P} \rangle$ is linear in the electric field strength to avoid problems with dielectric saturation.

6.8.3 The Cell Multipole Method for Non-bonded Interactions

The cell multipole method (also called the fast multipole method) is an algorithm that enables *all* $N(N-1)$ pairwise non-bonded interactions to be enumerated in a time that scales linearly with N, rather than N^2, as in the standard Ewald approach [Greengard and Roklin 1987; Ding *et al.* 1992a, b; Greengard 1994]. The cell multipole method can be used to evaluate interactions that can be expressed in the following general form:

$$\sum_i \sum_{j>i} \frac{q_i q_j}{|\mathbf{r}_i - \mathbf{r}_j|^p} \tag{6.65}$$

Both the Coulomb and Lennard-Jones potentials can be considered examples of this type. In the cell multipole method the simulation space is divided into uniform cubic

cells. The multipole moments (charge, dipole, quadrupole) of each cell are then calculated by summing over the atoms contained within the cell. The interaction between all of the atoms in the cell and another atom outside the cell (or indeed another cell) can then be calculated using an appropriate multipole expansion (see Section 4.9.1).

This multipole expansion is only valid if the separation between the interacting particles (be they atoms, molecules or cells) is larger than the sum of the radii of convergence of the multipoles. In the cell multipole method, the multipole expansion is used for interactions that are more than one cell distance away. For interactions that are within one cell distance the usual atomic pairwise interaction method is employed.

Consider an atom in a cell, C_0. The interactions with atoms in nearby cells are calculated using the usual pairwise formulae. There are 27 such cells (i e. the cell in which the atom is positioned and the surrounding 26 cells). The interaction between the atom and all of the atoms in each of the faraway cells is then calculated using the multipole expansion. The potential due to a faraway cell will be approximately constant for all atoms in the cell of current interest, C_0 (the cells are usually small, containing on average four atoms). Thus the potential due to each faraway cell can be represented as a Taylor series expansion about the centre of C_0. If there are M cells in total then there are $M-27$ faraway cells; then the calculation of these cell–cell interactions for the entire system will be of order $M(M-27)$. As the number of cells is approximately equal to the number of atoms, this still leaves us with a quadratic dependency upon the number of atoms present (though it does now vary as about $N^2/16$, if there is an average of four atoms per cell).

The algorithm can be converted to one which shows linear dependency by recognising that in the method we have just described, the interactions due to very faraway cells are calculated with the same accuracy as interactions with cells that are much closer. This level of accuracy can be considered unnecessary as any error is largely due to the closer cells. The small cells are thus grouped into larger cells, with the cell size increasing with the distance from the cell of interest, C_0. The accuracy of the calculation remains approximately constant if the ratio of the cell size to the distance remains constant. This grouping scheme is illustrated in Figure 6.25. The multipoles for each of the larger cells are calculated by translating and adding the moments of its constituent smaller cells. The use of multipole expansions and Taylor series approximations does mean that there will be a degree of truncation error, though this can be reduced by simply including more terms in the multipole expansion. The cell multipole algorithm requires an amount of bookkeeping to keep track of the hierarchy of the cells, which means that up to a certain size of problem the exact N^2 algorithm is faster. The algorithm then suddenly switches to a linear dependence. There is some debate over the break-even point at which the cell multipole method is equally as fast as an N^2 method, with estimates ranging from 300 particles to 100 000. Another complication to the debate is the introduction of the fast Fourier transform Ewald methods with their $N \ln N$ scaling. Nevertheless, for calculations on systems with thousands, if not millions, of atoms, the cell multipole methods appear promising, especially as enhanced versions are developed [Petersen *et al.* 1994].

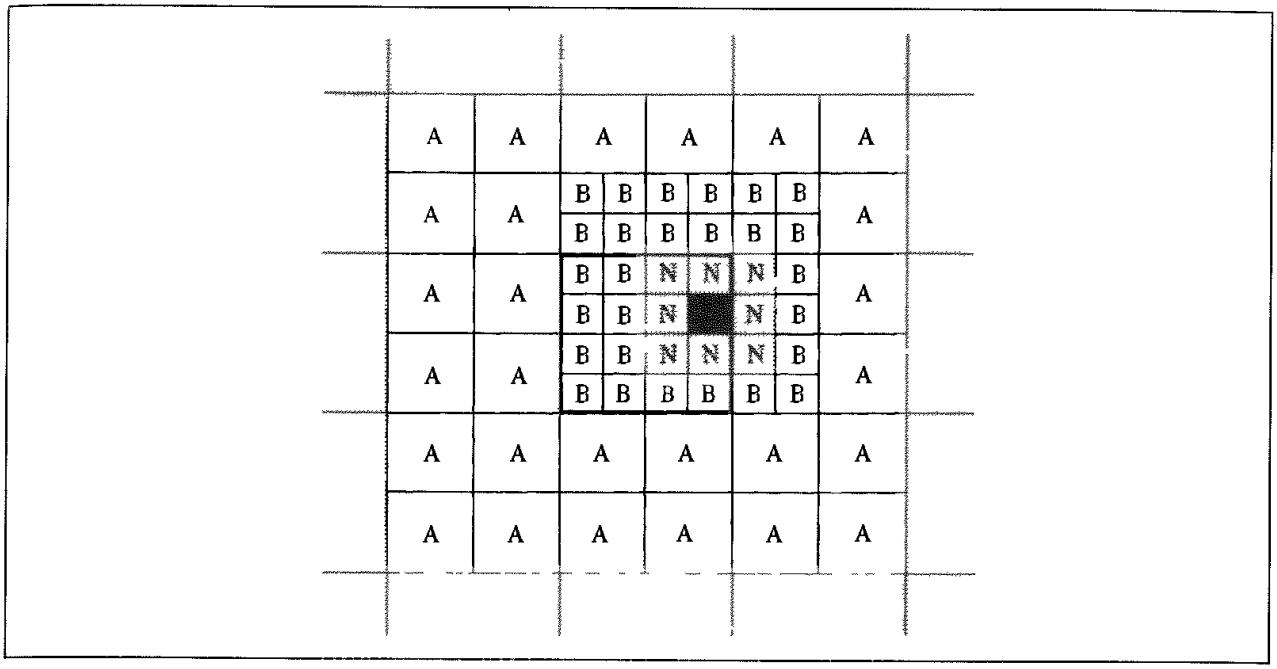

Fig 6.25: The hierarchy of cells used in the cell multipole method For an atom in the black cell, the interactions with atoms in the 26 nearby cells (N) are calculated explicitly Interactions with the atoms in cells labelled A and B are calculated using a Taylor series multipole expansion. (Figure adapted from Ding H-Q, N Karasawa and W A Goddard III, 1992b The Reduced Cell Multipole Method for Coulomb Interactions in Periodic Systems with Million-Atom Unit Cells Chemical Physics Letters *196·6–10)*

6.9 Analysing the Results of a Simulation and Estimating Errors

A simulation can generate an enormous amount of data, which should be properly analysed to extract relevant properties and to check that the calculation has behaved properly. The primary reason for undertaking a particular simulation may be to calculate just a single physical or thermodynamic property or to investigate the conformational properties of a molecule. However, it is also advisable to check that other aspects of the simulation have performed as expected. Some properties can be calculated during the simulation itself (such as the energy and the virial), but it is often sensible not to impose too severe a burden on the simulation program itself. In part this is because many properties do not vary significantly from one step to another but can be calculated at less frequent intervals. The configurations (i.e. positions of each atom or molecule in the system) do not change much from one step to another in either a molecular dynamics or Monte Carlo simulation, and so it is usual to store configurations every 5–25 steps, depending upon the nature of the system and the disk space available. It is good practice to visually examine configurations selected from throughout the simulation to ensure that no strange or unexpected behaviour is present. In many simulations of molecular systems the major objective is to investigate the structural behaviour of the system rather than to calculate thermodynamic properties, and so the focus of the analysis will change accordingly.

A computer simulation is subject to error, and this error should be properly calculated and assessed. Of course, computers only do what they are told to by the programmer, and so a program will always give the same results for the same set of initial conditions (if not, some serious fault should be suspected!). The results of a computer simulation may be subject to two kinds of error, just as any other scientific experiment. These are systematic errors and statistical errors. Systematic error results in a constant bias from the 'proper' behaviour. The most obvious effect of a systematic error is to displace the average property from its proper value. Systematic errors are sometimes due to a fault in the simulation algorithm or in the energy model and may be relatively easy to spot, especially if they have an obvious or even catastrophic effect on the simulation. Systematic errors may also arise from approximations inherent in the algorithm, such as truncation (all finite difference methods used in molecular dynamics generate only an approximation to the true integral of the equations of motion) and round-off errors (due to the limited precision with which numbers can be stored in a computer). Such errors can be more difficult to detect. One way to detect systematic error is to compare the distribution of the values of simple thermodynamic properties about their average values. The distribution of such properties about their average values should be normal (i.e. Gaussian), such that the probability of finding a particular value for the property A is given by:

$$p(A) = \frac{1}{\sigma\sqrt{2\pi}} \exp[-(A - \langle A \rangle)^2 / 2\sigma^2] \tag{6 66}$$

where σ^2 is the variance, given by $\sigma^2 = \langle (A - \langle A \rangle)^2 \rangle$. The standard deviation is the square root of the variance. More information on these statistical terms can be found in Section 1.10.7

The chi-squared test can be used to provide a quantitative estimate of the deviation of a calculated distribution from that expected Suppose that the value of some property (A) has been calculated from the simulation at regular intervals to give a total of M values. The average value of the property A is determined together with the standard deviation. The data, comprising all of the A values from the simulation, are then divided into bins such that the number of values in each bin (M_i) is approximately the same. The number of values that would be expected in the ith bin is:

$$n_i = \frac{M}{\sigma\sqrt{2\pi}} \int_{A_i - \Delta A/2}^{A_i + \Delta A/2} \exp\left[\frac{-(A_i - \langle A \rangle)^2}{2\sigma^2}\right] \tag{6.67}$$

where A_i is the value of the property in the ith bin and ΔA is the width of each bin The number of values that would be expected in each bin, n_i, does not have to be integral, though the actual number as determined from the simulation (M_i) will of course be an integer. The chi-squared function is given by:

$$\chi^2 = \sum_i \frac{(M_i - n_i)^2}{n_i} \tag{6 68}$$

If χ^2 is large (bigger than unity) then it is unlikely that the two distributions are the same. Any significant deviations from the expected behaviour should be investigated further to try to eliminate as much of the systematic error as possible. It is good practice to vary as

many of the parameters as possible: using different computers, different compilers, different algorithms and different ways of implementing a given algorithm, and different simulation methods (Monte Carlo and molecular dynamics) not only to test the component parts of the simulation but also the software used to perform the calculation.

If all sources of systematic error can be eliminated, there will still remain statistical errors. These errors are often reported as standard deviations. What we would particularly like to estimate is the error in the average value, $\langle A \rangle$. The standard deviation of the average value is calculated as follows:

$$\sigma_{\langle A \rangle} = \frac{\sigma_A}{\sqrt{M}} = \frac{\sqrt{\sum_{i=1}^{M} (A(i) - \langle A \rangle)^2}}{\sqrt{M}} \tag{6.69}$$

where $\sigma_{\langle A \rangle}$ is the standard deviation of the average value $\langle A \rangle$ obtained from M data values with respect to the run average, σ_A. Thus the standard deviation of the calculated average is inversely proportional to the square root of the number of data values, and so a longer simulation gives rise to a more accurate value. An important feature of Equation (6.69) is that it applies to *independent* (i.e. random) samples. Thus the number M in the denominator is not simply equal to the number of steps in the simulation. This is because there is a high degree of correlation between successive configurations in either a Monte Carlo or molecular dynamics simulation. What we need to know is the correlation or relaxation 'time' of the simulation; this is the number of steps required for the system to lose its 'memory' of previous configurations. In molecular dynamics, where successive steps are related in a temporal fashion, the correlation 'time' is a true time and will be discussed in more detail in Section 7.6. Usually, the correlation time will be unknown prior to the simulation but it can be estimated as follows. First, the configurations are broken down into a series of blocks. Suppose each block contains t_b successive steps and that there are n_b blocks (so the total simulation contains $t_b n_b$ steps, as shown in Figure 6.26). The average value of the property is calculated for each block:

$$\langle A \rangle_b = \frac{1}{t_b} \sum_{i=1}^{t_b} A_i \tag{6.70}$$

As the number of steps t_b in each block increases, so it would be expected that the block averages become uncorrelated. When this is the case, then the variance of the block averages, $\sigma^2(\langle A \rangle_b)$, will become inversely proportional to t_b. $\sigma^2(\langle A \rangle_b)$ is calculated as follows:

$$\sigma^2(\langle A \rangle_b) = \frac{1}{n_b} \sum_{b=1}^{n_b} (\langle A \rangle_b - \langle A \rangle_{\text{total}})^2 \tag{6.71}$$

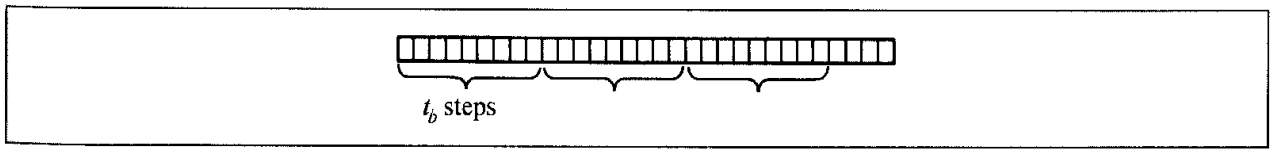

Fig 6 26 Blocking a simulation to calculate the statistical error.

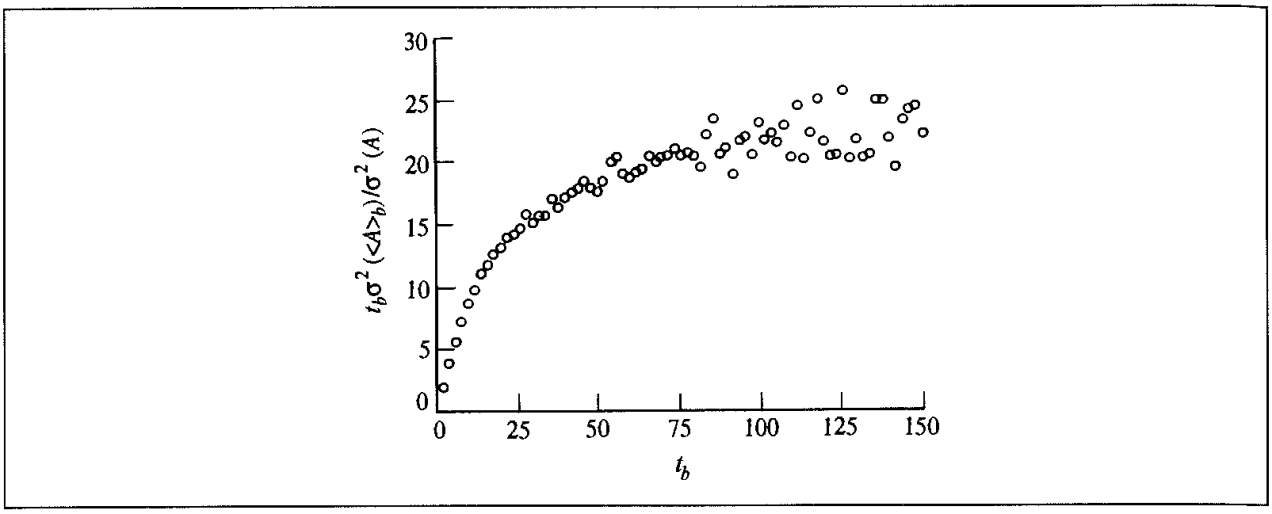

Fig 6 27: Calculating the statistical efficiency, σ A plot of $t_b\sigma^2\langle A\rangle_b/\sigma^2(A)$ against t_b shows a steep rise before levelling off Here the property A corresponds to the pressure calculated from the molecular dynamics simulation of argon

where $\langle A\rangle_{\text{total}}$ is the average over the entire simulation. The limiting number of steps to obtain uncorrelated configurations (the statistical inefficiency, s) can be calculated using:

$$s = \lim_{t_b \to \infty} \frac{t_b\sigma^2(\langle A\rangle_b)}{\sigma^2(A)} \tag{6.72}$$

To determine s, $t_b\sigma^2(\langle A\rangle_b)/\sigma^2(A)$ is plotted against t_b or $\sqrt{t_b}$. The graph should show a steep rise for low t_b and then level off to give a plateau, as shown in Figure 6.27. The plateau value is the limiting value that gives the correlation time ($s \approx 23$ in this case).

Having determined the value of s, the 'true' standard deviation of the average value is related to the 'true' error for an infinite simulation by:

$$\sigma_{\langle A\rangle} \approx \sigma\sqrt{\frac{s}{M}} \tag{6.73}$$

M here is the actual number of steps or iterations in the simulation. If the value of s can be reduced, then a more accurate average value can be calculated for a given length of simulation. This should be an important consideration when deciding what simulation protocol to use. For example, it may be more appropriate to use a complex simulation algorithm than a simpler one if the statistical inefficiency is significantly reduced.

If the relaxation time is known, then sample averages are often best calculated using the block method (Figure 6.26). Each block should contain more steps than the relaxation time. The sample average for the whole run can be obtained in a variety of ways:

1. Stratified systematic sampling, in which a single value of the property is taken from each block;
2. Stratified random sampling, in which a single value is taken at random from each block;
3. Coarse graining, in which the average value for each block is determined and then the average for the run is calculated by averaging the coarse-grain averages.

The coarse-graining approach is commonly used for thermodynamic properties whereas the systematic or random sampling methods are appropriate for static structural properties such as the radial distribution function.

Another way to improve the error in a simulation, at least for properties such as the energy and the heat capacity that depend on the size of the system (the extensive properties), is to increase the number of atoms or molecules in the calculation. The standard deviation of the average of such a property is proportional to $1/\sqrt{N}$. Thus, more accurate values can be obtained by running longer simulations on larger systems. In computer simulation it is unfortunately the case that the more effort that is expended the better the results that are obtained. Such is life!

Appendix 6.1 Basic Statistical Mechanics

The Boltzmann distribution is fundamental to statistical mechanics. The Boltzmann distribution is derived by maximising the entropy of the system (in accordance with the second law of thermodynamics) subject to the constraints on the system. Let us consider a system containing N particles (atoms or molecules) such that the energy levels of the particles are $\varepsilon_1, \varepsilon_2, \ldots$. If there are n_1 particles in the energy level ε_1, n_2 particles in ε_2 and so on, then there are W ways in which this distribution can be achieved:

$$W(n_1, n_2, \ldots) = N!/n_1!\, n_2! \ldots \tag{6.74}$$

The most favourable distribution is the one with the highest weight, and this corresponds to the configuration with just one particle in each energy level ($W = N!$). However, there are two important constraints on the system. First, the total energy is fixed:

$$\sum_i n_i \varepsilon_i = E \tag{6.75}$$

The second constraint arises from the fact that the total number of particles is fixed:

$$\sum_i n_i = N \tag{6.76}$$

The Boltzmann distribution gives the number of particles n_i in each energy level ε_i as:

$$\frac{n_i}{N} = \frac{\exp(-\varepsilon_i/k_B T)}{\sum_i \exp(-\varepsilon_i/k_B T)} \tag{6.77}$$

The denominator in this expression is the molecular partition function:

$$q = \sum_i \exp(-\varepsilon_i/k_B T) \tag{6.78}$$

For translational, rotational and vibrational motion the partition function can be calculated using standard results obtained by solving the Schrödinger equation:

$$\textit{translation:} \qquad q^t = \left(\frac{2\pi m k_B T}{h^2}\right)^{3/2} V \tag{6.79}$$

where V is the volume

$$\textit{rotation:} \qquad q^{r} \approx \left(\frac{\pi^{1/2}}{\sigma}\right)\left(\frac{2I_A k_B T}{\hbar^2}\right)\left(\frac{2I_B k_B T}{\hbar^2}\right)\left(\frac{2I_C k_B T}{\hbar^2}\right) \tag{6 80}$$

where I_A, I_B, I_C are the moments of intertia and σ is the symmetry number (2 for H_2O, 3 for NH_3, 12 for benzene)

$$\textit{vibration:} \qquad r^{v} = \frac{1}{1 - \exp(-\hbar\omega/k_B T)} \tag{6.81}$$

ω is the angular frequency: $\omega = \sqrt{k/\mu}$, where μ is the reduced mass. This form of the vibrational partition function is measured relative to the zero-point energy.

In computer simulations, we are particularly interested in the properties of a system comprising a number of particles. An ensemble is a collection of such systems, as might be generated using a molecular dynamics or a Monte Carlo simulation. Each member of the ensemble has an energy, and the distribution of the system within the ensemble follows the Boltzmann distribution. This leads to the concept of the ensemble partition function, Q.

Various thermodynamic properties can be calculated from the partition function. Here we simply state some of the most common:

$$\textit{internal energy:} \qquad U = \frac{k_B T^2}{Q}\left(\frac{\partial Q}{\partial T}\right)_V = k_B T^2 \left(\frac{\partial \ln Q}{\partial T}\right)_V \tag{6 82}$$

$$\textit{enthalpy:} \qquad H = k_B T^2 \left(\frac{\partial \ln Q}{\partial T}\right)_V + k_B T V \left(\frac{\partial \ln Q}{\partial V}\right)_T \tag{6 83}$$

$$\textit{Helmholtz free energy.} \qquad A = -k_B T \ln Q \tag{6 84}$$

$$\textit{Gibbs free energy:} \qquad G = -k_B T \ln Q + k_B T V \left(\frac{\partial \ln Q}{\partial V}\right)_T \tag{6.85}$$

Appendix 6.2 Heat Capacity and Energy Fluctuations

The heat capacity is related to the internal energy U by

$$C_V = \left(\frac{\partial U}{\partial T}\right)_V \tag{6.86}$$

If we differentiate the expression for the internal energy, Equation (6.20), we can obtain the heat capacity in terms of the partition function:

$$C_V = \frac{\partial}{\partial T}\left(\frac{k_B T^2}{Q}\frac{\partial Q}{\partial T}\right)_V = \frac{k_B T^2}{Q}\frac{\partial^2 Q^2}{\partial T^2} + \frac{2k_B T}{Q}\frac{\partial Q}{\partial T} - \frac{k_B T^2}{Q^2}\left(\frac{\partial Q}{\partial T}\right)^2 \tag{6.87}$$

The desired expression is obtained by writing each of these three terms as a function of the average energy, $\langle E \rangle$. The internal energy is just the expectation value of the energy, $\langle E \rangle$, and so·

$$\langle E \rangle = \frac{k_B T^2}{Q}\frac{\partial Q}{\partial T} \tag{6.88}$$

Thus for the second term in Equation (6.87) we have

$$\frac{2k_BT}{Q}\frac{\partial Q}{\partial T} = \frac{2\langle E\rangle}{T} \tag{6.89}$$

We can also rewrite the third term in Equation (6.87):

$$k_BT\left(\frac{1}{Q}\frac{\partial Q}{\partial T}\right)^2 = \frac{\langle E\rangle^2}{k_BT} \tag{6.90}$$

For the first term, we need to do a little more work The starting point is:

$$\frac{\partial}{\partial T}\left(\frac{\langle E\rangle}{k_BT^2}\right) = \frac{\partial}{\partial T}\left\{\frac{1}{Q}\left(\frac{\partial Q}{\partial T}\right)\right\} \tag{6.91}$$

or

$$-2\frac{\langle E\rangle}{k_BT^3} = \frac{1}{Q}\frac{\partial^2 Q}{\partial T^2} + \frac{\partial Q}{\partial T}\frac{\partial}{\partial T}\left(\frac{1}{Q}\right) \tag{6 92}$$

We can use the chain rule as follows:

$$\frac{\partial Q}{\partial T}\frac{\partial}{\partial T}\left(\frac{1}{Q}\right) = \frac{\partial Q}{\partial T}\frac{\partial Q}{\partial T}\frac{\partial}{\partial T}\left(\frac{1}{Q}\right) = -\left(\frac{\partial Q}{\partial T}\right)^2\left(\frac{1}{Q}\right)^2 \tag{6.93}$$

Thus

$$\frac{k_BT^2}{Q}\frac{\partial^2 Q}{\partial T^2} = -2\frac{\langle E\rangle}{k_BT^3} + \frac{\langle E^2\rangle}{k_B^2T^4} \tag{6 94}$$

So

$$C_V = k_BT^2\left\{-2\frac{\langle E\rangle}{k_BT^3} + \frac{\langle E^2\rangle}{k_B^2T^4}\right\} + 2\frac{\langle E\rangle}{T} - \frac{\langle E\rangle^2}{k_BT^2} \tag{6.95}$$

or

$$C_V = \frac{(\langle E^2\rangle - \langle E\rangle^2)}{k_BT^2} \tag{6 96}$$

Appendix 6.3 The Real Gas Contribution to the Virial

If the gas particles interact through a pairwise potential, then the contribution to the virial from the intermolecular forces can be derived as follows. Consider two atoms i and j separated by a distance r_{ij}.

$$r_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2} \tag{6.97}$$

The contribution to the virial from the interaction $v(r_{ij})$ between atoms i and j is given by:

$$W_{\text{real}} = \left[x_i\frac{\partial}{\partial x_i} + x_j\frac{\partial}{\partial x_j} + y_i\frac{\partial}{\partial y_i} + y_j\frac{\partial}{\partial y_j} + z_i\frac{\partial}{\partial z_i} + z_j\frac{\partial}{\partial z_j}\right]v(r_{ij}) \tag{6.98}$$

Since

$$x_i\frac{\partial r_{ij}}{\partial x_i} = x_i\frac{(x_i - x_j)}{r_{ij}} \quad \text{and} \quad x_j\frac{\partial r_{ij}}{\partial x_i} = -x_j\frac{(x_i - x_j)}{r_{ij}} \tag{6 99}$$

and similarly for the y and z coordinates, we can apply the chain rule, $\partial/\partial x_i = (\partial/\partial r_{ij})(\partial r_{ij}/\partial x_i)$, as follows:

$$W_{\text{real}} = \left[\frac{(x_i - x_j)^2}{r_{ij}} + \frac{(y_i - y_j)^2}{r_{ij}} + \frac{(z_i - z_j)^2}{r_{ij}}\right]\frac{d\upsilon(r_{ij})}{dr_{ij}} = r_{ij}\frac{d\upsilon(r_{ij})}{dr_{ij}} \tag{6 100}$$

When we include the contributions from all pairs of atoms, we obtain:

$$W_{\text{real}} = \sum_{i=1}^{N}\sum_{j=i+1}^{N} r_{ij}\frac{d\upsilon(r_{ij})}{dr_{ij}} \tag{6.101}$$

Appendix 6.4 Translating Particle Back into Central Box for Three Box Shapes

From Smith W 1983. The Periodic Boundary Condition in Non-Cubic MD Cells: Wigner-Seitz Cells with Reflection Symmetry. *CCP5 Quarterly* **10**:37–42. This table is expressed using references to several built-in FORTRAN functions. The AINT function returns the integral part of its argument, e.g. AINT(3.4) = 3.0; AINT(4.7) = 4.0; AINT(−0.5) = 0.0 and AINT(−1.7) = −1.0. ANINT returns the nearest integer, so ANINT(0.49) = 0.0 and ANINT(0.51) = 1.0. SIGN(x, y) returns $|x|$ if $y \geq 0$ and $-|x|$ if $y < 0$. ABS(x) returns the absolute value of x, $|x|$. Equivalent functions also exist in most other programming languages.

Rectangular box, side 2a (x) by 2b (y) by 2c (z)	$x = x - 2 \times \text{a} \times \text{AINT}(x/\text{a})$ $y = y - 2 \times b \times \text{AINT}(y/b)$ $z = z - 2 \times c \times \text{AINT}(z/c)$ A common alternative is $x = x - \text{a} \times \text{ANINT}(x/\text{a})$ $y = y - b \times \text{ANINT}(y/b)$ $z = z - c \times \text{ANINT}(z/c)$
Truncated octahedron derived from cube of side 2a	$x = x - 2 \times \text{a} \times \text{AINT}(x/\text{a})$ $y = y - 2 \times b \times \text{AINT}(y/\text{a})$ $z = z - 2 \times c \times \text{AINT}(z/\text{a})$ if $(ABS(x) + ABS(y) + ABS(z)) \geq 1\,5 \times A$ then $x = x - \text{SIGN}(\text{a}, x)$ $y = y - \text{SIGN}(\text{a}, y)$ $z = z - \text{SIGN}(\text{a}, z)$ endif

Hexagonal prism of length 2a (in z direction) and distance between opposite faces of the hexagon 2b	$z = z - 2 \times \text{a} \times \text{AINT}(z/\text{a})$
	$x = x - 2 \times b \times \text{AINT}(x/b)$
	if $(\text{ABS}(x) + \sqrt{3} \times \text{ABS}(y)) \geq 2 \times B$ then
	$x = x - \text{SIGN}(b, x)$
	$y = y - \text{SIGN}(\sqrt{3} \times b, y)$
	endif

Further Reading

Allen M P and D J Tildesley 1987. *Computer Simulation of Liquids* Oxford, Oxford University Press

Bradbury T C 1968. *Theoretical Mechanics.* Malabar, FL, Krieger.

Chandler D 1987 *Introduction to Modern Statistical Mechanics.* New York, Oxford University Press.

Hansen J P and I R McDonald 1976 *Theory of Simple Liquids.* London, Academic Press.

Smith P E and van Gunsteren W F 1993. Methods for the Evaluation of Long Range Electrostatic Forces. In van Gunsteren W F, P K Weiner and A J Wilkinson (Editors). *Computer Simulation of Biomolecular Systems.* Leiden, ESCOM.

van Gunsteren W F and H J C Berendsen 1990. Computer Simulation of Molecular Dynamics Methodology, Applications and Perspectives in Chemistry. *Angewandte Chemie International Edition in English* **29**·992–1023

References

Adams D J 1983 Alternatives to the Periodic Cube in Computer Simulation. *CCP5 Quarterly* **10**:30–36.

Alper H E and R M Levy 1989 Computer Simulations of the Dielectric Properties of Water – Studies of the Simple Point-Charge and Transferable Intermolecular Potential Models *Journal of Chemical Physics* **91**.1242–1251

Cheatham T E III, J L Miller, T Fox, T A Darden and P A Kollman 1995 Molecular Dynamics Simulations on Solvated Biomolecular Systems· The Particle Mesh Ewald Method Leads to Stable Trajectories of DNA, RNA and Proteins. *Journal of the American Chemical Society* **117** 4193–4194

Darden T A, L Perera, L Li and L Pedersen 1999 New Tricks for Modelers from the Crystallography Toolkit· The Particle Mesh Ewald Algorithm and Its Use in Nucleic Acid Simulations *Structure with Folding and Design* 7 R55–R60.

Darden T A, D York and L Pedersen 1993 Particle-mesh Ewald· An *N*.log(*N*) method for Ewald sums in large systems. *Journal of Chemical Physics* **98** 10089–10092

Deserno M and C Holm 1998a How to Mesh Up Ewald Sums. I A Theoretical and Numerical Comparison of Various Particle Mesh Routines *Journal of Chemical Physics* **109** 7678–7693

Deserno M and C Holm 1998b. How to Mesh Up Ewald Sums. II An Accurate Error Estimate for the Particle–Particle–Particle–Mesh Algorithm. *Journal of Chemical Physics* **109**.7694–7701.

Ding H-Q, N Karasawa and W A Goddard III 1992a. Atomic Level Simulations on a Million Particles: The Cell Multipole Method for Coulomb and London Nonbonding Interactions. *Journal of Chemical Physics* **97**:4309–4315.

Ding H-Q, N Karasawa and W A Goddard III 1992b. The Reduced Cell Multipole Method for Coulomb Interactions in Periodic Systems with Million-Atom Unit Cells *Chemical Physics Letters* **196**.6–10.

Ewald P 1921 Due Berechnung optischer und elektrostatischer Gitterpotentiale. *Annalen der Physik* **64** 253–287.

Friedman H L 1975. Image Approximation to the Reaction Field *Molecular Physics* **29** 1533–1543

Greengard L 1994. Fast Algorithms for Classical Physics. *Science* **265**·909–914

Greengard L and V I Roklin 1987. A Fast Algorithm for Particle Simulations. *Journal of Computational Physics* **73**:325–348

Hockney R W and J W Eastwood 1988 *Computer Simulation using Particles* Bristol, Adam Hilger.

Luty B A, M E David, I G Tironi and W F van Gunsteren 1994 A Comparison of Particle-Particle, Particle-Mesh and Ewald Methods for Calculating Electrostatics Interactions in Periodic Molecular Systems *Molecular Simulation* **14**.11–20

Luty B A, I G Tironi and W F van Gunsteren 1995. Lattice-sum methods for calculating electrostatic interactions in molecular simulations *Journal of Chemical Physics* **103**:3014–3021.

Petersen H G, D Soelvaso, J W Perram and E R Smith 1994 The Very Fast Multipole Method. *Journal of Chemical Physics* **101**.8870–8876

Thompson S M 1983. Use of Neighbour Lists in Molecular Dynamics. *CCP5 Quarterly* **8** 20–28.

van Gunsteren W F and H J C Berendsen 1986. *GROMOS User Guide.*

Verlet L 1967. Computer 'Experiments' on Classical Fluids. II. Equilibrium Correlation Functions *Physical Review* **165**.201–204

Wolf M L, J R Walker and C R A Catlow 1984. A Molecular Dynamics Simulation Study of the Superionic Conductor Lithium Nitride. I. *Journal of Physical Chemistry* **17**.6623–34.

York D M, A Wlodawer, L G Pedersen and T A Darden 1994. Atomic-level Accuracy in Simulations of Large Protein Crystals. *Proceedings of the National Academy of Sciences USA* **91** 8715–8718.

CHAPTER SEVEN

Molecular Dynamics Simulation Methods

7.1 Introduction

In molecular dynamics, successive configurations of the system are generated by integrating Newton's laws of motion. The result is a trajectory that specifies how the positions and velocities of the particles in the system vary with time. Newton's laws of motion can be stated as follows:

1. A body continues to move in a straight line at constant velocity unless a force acts upon it.
2. Force equals the rate of change of momentum.
3. To every action there is an equal and opposite reaction.

The trajectory is obtained by solving the differential equations embodied in Newton's second law ($F = ma$):

$$\frac{d^2x_i}{dt^2} = \frac{F_{x_i}}{m_i} \tag{7 1}$$

This equation describes the motion of a particle of mass m_i along one coordinate (x_i) with F_{x_i} being the force on the particle in that direction.

It is helpful to distinguish three different types of problem to which Newton's laws of motion may be applied. In the simplest case, no force acts on each particle between collisions. From one collision to the next, the position of the particle thus changes by $\mathbf{v}_i\delta t$, where $\mathbf{v}_i$ is the (constant) velocity and δt is the time between collisions. In the second situation, the particle experiences a constant force between collisions. An example of this type of motion would be that of a charged particle moving in a uniform electric field. In the third case, the force on the particle depends on its position relative to the other particles. Here the motion is often very difficult, if not impossible, to describe analytically, due to the coupled nature of the particles' motions

7.2 Molecular Dynamics Using Simple Models

The first molecular dynamics simulation of a condensed phase system was performed by Alder and Wainwright in 1957 using a hard-sphere model [Alder and Wainwright 1957]. In this model, the spheres move at constant velocity in straight lines between collisions. All collisions are perfectly elastic and occur when the separation between the centres of

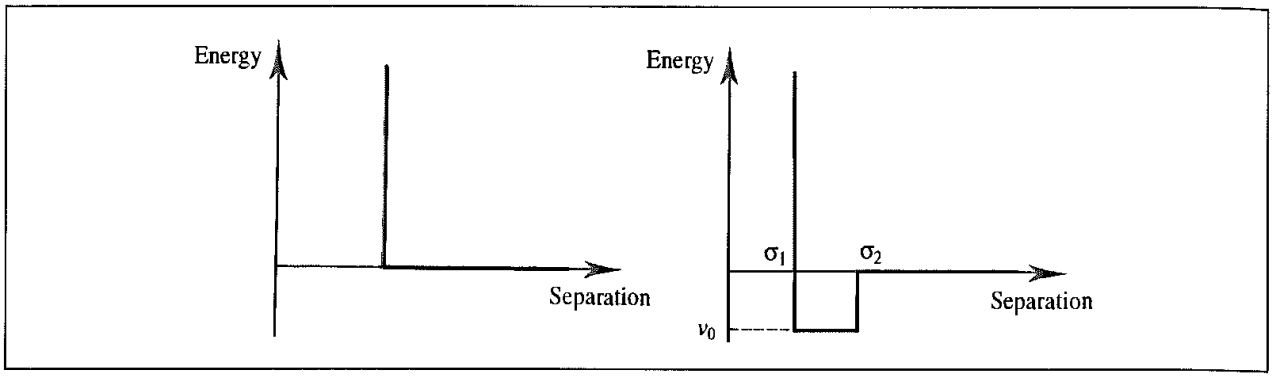

Fig 7 1 The hard-sphere and square-well potentials

the spheres equals the sphere diameter. The pair potential thus has the form shown in Figure 7.1 Some early simulations also used the square-well potential, where the interaction energy between two particles is zero beyond a cutoff distance σ_2; infinite below a smaller cutoff distance σ_1; and equal to v_0 between the two cutoff values (Figure 7.1) The steps involved in the hard-sphere calculation are as follows:

1. Identify the next pair of spheres to collide and calculate when the collision will occur.
2. Calculate the positions of all the spheres at the collision time.
3. Determine the new velocities of the two colliding spheres after the collision.
4. Repeat from 1 until finished.

The new velocities of the colliding spheres are calculated by applying the principle of conservation of linear momentum.

Simple interaction models such as the hard-sphere potential obviously suffer from many deficiencies but have nevertheless provided many useful insights into the microscopic

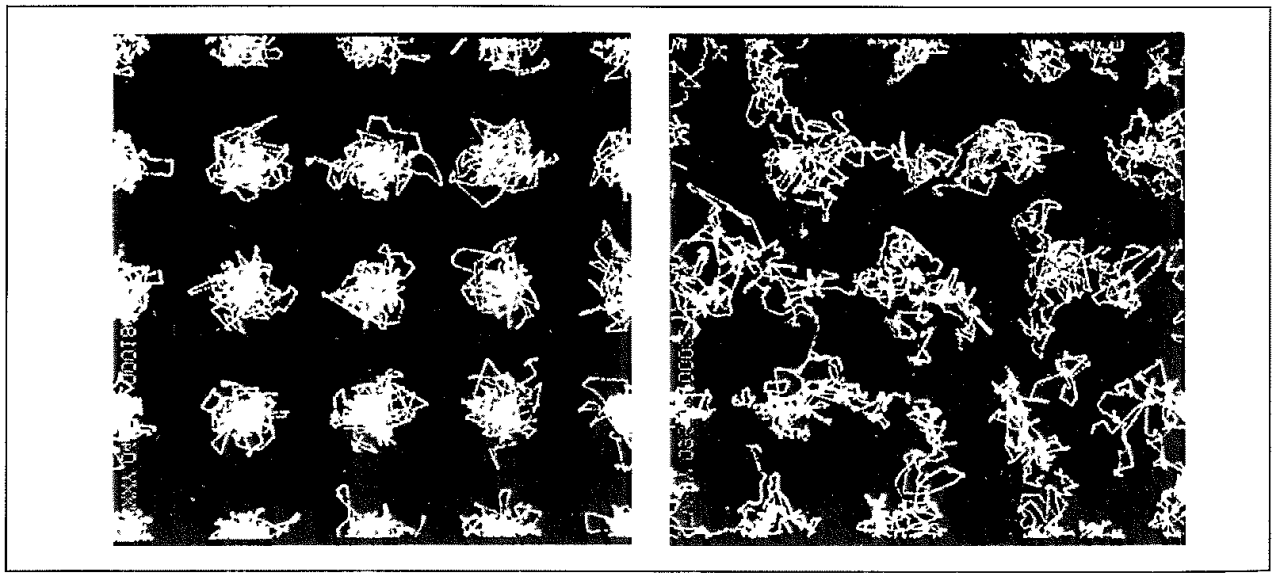

Fig 7 2 Molecular graphics representation of the paths generated by 32 hard spherical particles in the solid (left) and fluid (right) phase (Reproduced from Alder B J and T E Wainwright 1959 Studies in Molecular Dynamics I General Method Journal of Chemical Physics *31* 459–466)

nature of fluids. The early workers were particularly keen to quantify the differences between the solid and fluid phases; it is interesting to note that such investigations were facilitated by early molecular graphics systems, which enabled the trajectories of the particles to be represented simultaneously (Figure 7.2).

7.3 Molecular Dynamics with Continuous Potentials

In more realistic models of intermolecular interactions, the force on each particle will change whenever the particle changes its position, or whenever any of the other particles with which it interacts changes position. The first simulation using continuous potentials was of argon by Rahman [Rahman 1964], who also performed the first simulation of a molecular liquid (water) [Rahman and Stillinger 1971]) and made many other important methodological contributions in molecular dynamics. Under the influence of a continuous potential the motions of all the particles are coupled together, giving rise to a many-body problem that cannot be solved analytically. Under such circumstances the equations of motion are integrated using a *finite difference method.*

7.3.1 Finite Difference Methods

Finite difference techniques are used to generate molecular dynamics trajectories with continuous potential models, which we will assume to be pairwise additive. The essential idea is that the integration is broken down into many small stages, each separated in time by a fixed time δt. The total force on each particle in the configuration at a time t is calculated as the vector sum of its interactions with other particles. From the force we can determine the accelerations of the particles, which are then combined with the positions and velocities at a time t to calculate the positions and velocities at a time $t+\delta t$. The force is assumed to be constant during the time step. The forces on the particles in their new positions are then determined, leading to new positions and velocities at time $t+2\delta t$, and so on.

There are many algorithms for integrating the equations of motion using finite difference methods, several of which are commonly used in molecular dynamics calculations. All algorithms assume that the positions and dynamic properties (velocities, accelerations, etc.) can be approximated as Taylor series expansions:

$$\mathbf{r}(t+\delta t) = \mathbf{r}(t) + \delta t\mathbf{v}(t) + \tfrac{1}{2}\delta t^2\mathbf{a}(t) + \tfrac{1}{6}\delta t^3\mathbf{b}(t) + \tfrac{1}{24}\delta t^4\mathbf{c}(t) + \cdots \tag{7.2}$$

$$\mathbf{v}(t+\delta t) = \mathbf{v}(t) + \delta t\mathbf{a}(t) + \tfrac{1}{2}\delta t^2\mathbf{b}(t) + \tfrac{1}{6}\delta t^3\mathbf{c}(t) + \cdots \tag{7.3}$$

$$\mathbf{a}(t+\delta t) = \mathbf{a}(t) + \delta t\mathbf{b}(t) + \tfrac{1}{2}\delta t^2\mathbf{c}(t)\cdots \tag{7.4}$$

$$\mathbf{b}(t+\delta t) = \mathbf{b}(t) + \delta t\mathbf{c}(t) + \cdots \tag{7.5}$$

where $\mathbf{v}$ is the velocity (the first derivative of the positions with respect to time), $\mathbf{a}$ is the acceleration (the second derivative), $\mathbf{b}$ is the third derivative, and so on. The *Verlet algorithm* [Verlet 1967] is probably the most widely used method for integrating the equations of

motion in a molecular dynamics simulation. The Verlet algorithm uses the positions and accelerations at time t, and the positions from the previous step, $\mathbf{r}(t-\delta t)$, to calculate the new positions at $t+\delta t$, $\mathbf{r}(t+\delta t)$. We can write down the following relationships between these quantities and the velocities at time t:

$$\mathbf{r}(t+\delta t) = \mathbf{r}(t) + \delta t\mathbf{v}(t) + \tfrac{1}{2}\delta t^2\mathbf{a}(t) + \cdots \quad (7.6)$$

$$\mathbf{r}(t-\delta t) = \mathbf{r}(t) - \delta t\mathbf{v}(t) + \tfrac{1}{2}\delta t^2\mathbf{a}(t) - \cdots \quad (7.7)$$

Adding these two equations gives

$$\mathbf{r}(t+\delta t) = 2\mathbf{r}(t) - \mathbf{r}(t-\delta t) + \delta t^2\mathbf{a}(t) \quad (7.8)$$

The velocities do not explicitly appear in the Verlet integration algorithm. The velocities can be calculated in a variety of ways; a simple approach is to divide the difference in positions at times $t+\delta t$ and $t-\delta t$ by $2\delta t$:

$$\mathbf{v}(t) = [\mathbf{r}(t+\delta t) - \mathbf{r}(t-\delta t)]/2\delta t \quad (7.9)$$

Alternatively, the velocities can be estimated at the half-step, $t+\frac{1}{2}\delta t$:

$$\mathbf{v}(t+\tfrac{1}{2}\delta t) = [\mathbf{r}(t+\delta t) - \mathbf{r}(t)]/\delta t \quad (7.10)$$

Implementation of the Verlet algorithm is straightforward and the storage requirements are modest, comprising two sets of positions ($\mathbf{r}(t)$ and $\mathbf{r}(t-\delta t)$) and the accelerations $\mathbf{a}(t)$. One of its drawbacks is that the positions $\mathbf{r}(t+\delta t)$ are obtained by adding a small term ($\delta t^2\mathbf{a}(t)$) to the difference of two much larger terms, $2\mathbf{r}(t)$ and $\mathbf{r}(t-\delta t)$. This may lead to a loss of precision. The Verlet algorithm has some other disadvantages. The lack of an explicit velocity term in the equations makes it difficult to obtain the velocities, and indeed the velocities are not available until the positions have been computed at the next step. In addition, it is not a self-starting algorithm; the new positions are obtained from the current positions $\mathbf{r}(t)$ and the positions from the previous time step, $\mathbf{r}(t-\delta t)$. At $t=0$ there is obviously only one set of positions and so it is necessary to employ some other means to obtain positions at $t-\delta t$. One way to obtain $\mathbf{r}(t-\delta t)$ is to use the Taylor series, Equation (7.2), truncated after the first term. Thus, $\mathbf{r}(-\delta t) = \mathbf{r}(0) - \delta t\mathbf{v}(0) - \delta t\mathbf{v}(0)$.

Several variations on the Verlet algorithm have been developed. The *leap-frog* algorithm [Hockney 1970] uses the following relationships:

$$\mathbf{r}(t+\delta t) = \mathbf{r}(t) + \delta t\mathbf{v}(t+\tfrac{1}{2}\delta t) \quad (7.11)$$

$$\mathbf{v}(t+\tfrac{1}{2}\delta t) = \mathbf{v}(t-\tfrac{1}{2}\delta t) + \delta t\mathbf{a}(t) \quad (7.12)$$

To implement the leap-frog algorithm, the velocities $\mathbf{v}(t+\frac{1}{2}t)$ are first calculated from the velocities at time $t-\frac{1}{2}\delta t$ and the accelerations at time t. The positions $\mathbf{r}(t+\delta t)$ are then deduced from the velocities just calculated together with the positions at time $\mathbf{r}(t)$ using Equation (7.11). The velocities at time t can be calculated from

$$\mathbf{v}(t) = \tfrac{1}{2}[\mathbf{v}(t+\tfrac{1}{2}\delta t) + \mathbf{v}(t-\tfrac{1}{2}\delta t)] \quad (7.13)$$

The velocities thus 'leap-frog' over the positions to give their values at $t+\frac{1}{2}\delta t$ (hence the name). The positions then leap over the velocities to give their new values at $t+\delta t$, ready for the velocities at $t+\frac{3}{2}\delta t$, and so on. The leap-frog method has two advantages over the

standard Verlet algorithm: it explicitly includes the velocity and also does not require the calculation of the differences of large numbers. However, it has the obvious disadvantage that the positions and velocities are not synchronised. This means that it is not possible to calculate the kinetic energy contribution to the total energy at the same time as the positions are defined (from which the potential energy is determined).

The *velocity Verlet* method [Swope *et al.* 1982] gives positions, velocities and accelerations at the same time and does not compromise precision:

$$\mathbf{r}(t+\delta t) = \mathbf{r}(t) + \delta t\mathbf{v}(t) + \tfrac{1}{2}\delta t^2\mathbf{a}(t) \tag{7 14}$$

$$\mathbf{v}(t+\delta t) = \mathbf{v}(t) + \tfrac{1}{2}\delta t[\mathbf{a}(t) + \mathbf{a}(t+\delta t)] \tag{7.15}$$

The velocity Verlet method is actually implemented as a three-stage procedure because, as can be seen from Equation (7.15), to calculate the new velocities requires the accelerations at both t and $t+\delta t$. Thus in the first step the positions at $t+\delta t$ are calculated according to Equation (7.14) using the velocities and the accelerations at time t. The velocities at time $t+\frac{1}{2}\delta t$ are then determined using:

$$\mathbf{v}(t+\tfrac{1}{2}\delta t) = \mathbf{v}(t) + \tfrac{1}{2}\delta t\mathbf{a}(t) \tag{7.16}$$

New forces are next computed from the current positions, thus giving $\mathbf{a}(t+\delta t)$. In the final step, the velocities at time $t+\delta t$ are determined using:

$$\mathbf{v}(t+\delta t) = \mathbf{v}(t+\tfrac{1}{2}\delta t) + \tfrac{1}{2}\delta t\mathbf{a}(t+\delta t) \tag{7 17}$$

Beeman's algorithm [Beeman 1976] is also related to the Verlet method:

$$\mathbf{r}(t+\delta t) = \mathbf{r}(t) + \delta t\mathbf{v}(t) + \tfrac{2}{3}\delta t^2\mathbf{a}(t) - \tfrac{1}{6}\delta t^2\mathbf{a}(t-\delta t) \tag{7.18}$$

$$\mathbf{v}(t+\delta t) = \mathbf{v}(t) + \tfrac{1}{3}\delta t\mathbf{a}(t) + \tfrac{5}{6}\delta t\mathbf{a}(t) - \tfrac{1}{6}\delta t\mathbf{a}(t-\delta t) \tag{7.19}$$

The Beeman integration scheme uses a more accurate expression for the velocity. As a consequence it often gives better energy conservation, because the kinetic energy is calculated directly from the velocities However, the expressions used are more complex than those of the Verlet algorithm and so it is computationally more expensive.

We have already encountered four different integration methods, with more to come! Why should we use one method in preference to another? What features characterise a 'good' integration method? As with any other computer algorithm, an ideal integration scheme should be fast, require minimal memory and be easy to program. However, for most molecular dynamics simulations these issues are of secondary importance; most calculations do not make significant memory demands of even a modest workstation, and the time required for the integration is usually trivial compared to the other parts of the calculation. The most demanding part of a molecular dynamics simulation is invariably the calculation of the force on each particle in the system. More important considerations are that the integration algorithm should conserve energy and momentum, be time-reversible, and should permit a long time step, δt, to be used. The size of the time step is particularly relevant to the computational demands as a simulation using a long time step will require fewer iterations to cover a given amount of phase space. A less important requirement is that the integration algorithm should give the same results as an exact, analytical trajectory

(this can be tested using simple problems for which an analytical solution can be derived). We would, in any case, expect the calculated trajectory to deviate from the exact trajectory because the computer can only store numbers to a given precision.

The *order* of an integration method is the degree to which the Taylor series expansion, Equation (7.2), is truncated: it is the lowest term that is not present in the expansion. The order may not always be apparent from the formulae used. For example, the highest-order derivative that appears in the Verlet formulae is the second, $\mathbf{a}(t)$, yet the Verlet algorithm is, in fact, a fourth-order method. This is because the third-order terms, which cancel when Equation (7.6) is added to Equation (7.7), are still implied in the expansion:

$$\mathbf{r}(t+\delta t) = \mathbf{r}(t) + \delta t\mathbf{v}(t) + \tfrac{1}{2}\delta t^2\mathbf{a}(t) + \tfrac{1}{6}\delta t^3\mathbf{b}(t) + \tfrac{1}{24}\delta t^4\mathbf{c}(t) \quad (7\ 20)$$

$$\mathbf{r}(t-\delta t) = \mathbf{r}(t) - \delta t\mathbf{v}(t) + \tfrac{1}{2}\delta t^2\mathbf{a}(t) - \tfrac{1}{6}\delta t^3\mathbf{b}(t) + \tfrac{1}{24}\delta t^4\mathbf{c}(t) \quad (7.21)$$

7.3.2 Predictor–Corrector Integration Methods

The predictor–corrector methods [Gear 1971] form a general family of integration algorithms from which one can select a scheme that is correct to a given order. These methods have three basic steps. First, new positions, velocities, accelerations and higher-order terms are predicted according to the Taylor expansion, Equations (7.2)–(7.4). In the second stage, the forces are evaluated at the new positions to give accelerations $\mathbf{a}(t+\delta t)$. These accelerations are then compared with the accelerations that are predicted from the Taylor series expansion, $\mathbf{a}^c(t+\delta t)$. The difference between the predicted and calculated accelerations is then used to 'correct' the positions, velocities, etc., in the correction step:

$$\Delta\mathbf{a}(t+\delta t) = \mathbf{a}^c(t+\delta t) - \mathbf{a}(t+\delta t) \quad (7.22)$$

Then

$$\mathbf{r}^c(t+\delta t) = \mathbf{r}(t+\delta t) + c_0\Delta\mathbf{a}(t+\delta t) \quad (7.23)$$

$$\mathbf{v}^c(t+\delta t) = \mathbf{v}(t+\delta t) + c_1\Delta\mathbf{a}(t+\delta t) \quad (7.24)$$

$$\mathbf{a}^c(t+\delta t)/2 = \mathbf{a}(t+\delta t)/2 + c_2\Delta\mathbf{a}(t+\delta t) \quad (7.25)$$

$$\mathbf{b}^c(t+\delta t)/6 = \mathbf{b}(t+\delta t)/6 + c_3\Delta\mathbf{a}(t+\delta t) \quad (7.26)$$

Gear has suggested 'best' values of the coefficients $c_0, c_1, \ldots$ The set of coefficients to use depends upon the order of the Taylor series expansion. In Equations (7.23)–(7.26) the expansion has been truncated after the third derivative of the positions (i.e. $\mathbf{b}(t)$). The appropriate set of coefficients to use in this case is $c_0 = \frac{1}{6}$, $c_1 = \frac{5}{6}$, $c_2 = 1$ and $c_3 = \frac{1}{3}$.

The storage required for the Gear predictor–corrector algorithm is $3 \times (O+1)N$, where O is the highest-order differential used in the Taylor series expansion and N is the number of atoms. Thus the storage required for our example is $15N$, which is rather more than for the Verlet algorithm, which uses $9N$. More importantly, the Gear algorithm requires two time-consuming force evaluations per time step, though this is not necessarily a disadvantage as it may permit a time step more than twice as long as an alternative algorithm

There are many variants of the 'predictor-corrector' theme; of these, we will only mention the algorithm used by Rahman in the first molecular dynamics simulations with continuous potentials [Rahman 1964]. In this method, the first step is to predict new positions as follows:

$$\mathbf{r}(t+\delta t) = \mathbf{r}(t-\delta t) + 2\delta t\mathbf{v}(t) \tag{7.27}$$

New accelerations are calculated at these new positions in the usual way These accelerations are then used to generate a set of new velocities, and then corrected positions:

$$\mathbf{v}(t+\delta t) = \mathbf{v}(t) + \tfrac{1}{2}\delta t(\mathbf{a}(t+\delta t) + \mathbf{a}(t)) \tag{7.28}$$

$$\mathbf{r}^c(t+\delta t) = \mathbf{r}(t) + \tfrac{1}{2}\delta t(\mathbf{v}(t) + \mathbf{v}(t+\delta t)) \tag{7.29}$$

The acceleration can then be recalculated at the new corrected positions to give new velocities The method then iterates over the two Equations (7.28) and (7.29). Two or three passes are usually required to achieve consistency, with a force evaluation at each step. The computational demands of this scheme mean that it is now rarely used, though it does give accurate solutions of the equations of motion.

7.3.3 Which Integration Algorithm is Most Appropriate?

The wide variety of integration schemes available can make it difficult to decide which is the most appropriate one to use Various factors may need to be taken into account when deciding which is most appropriate. Clearly, the computational effort required is a major consideration. As we have already indicated, an algorithm that is nominally more expensive (for example, because it requires more than one force evaluation per iteration) may permit a significantly longer time step to be used and so, in fact, be more cost-effective. One of the most important considerations is energy conservation; this can be calculated as the root-mean-square fluctuation and is often plotted against the time step, as shown in Figure 7.3. In Appendix 7.1 we show why energy conservation would be expected in a molecular dynamics simulation. The kinetic and potential energy components would be expected to fluctuate in equal and opposite directions; this is also shown in Figure 7.3.

As the time step increases, so the RMS energy fluctuation also increases. For the argon simulation reported in Figure 7.3, the RMS fluctuation in the total energy is approximately 0.006 kcal/mol and the RMS fluctuations in the kinetic and potential energies are approximately 2.5 kcal/mol. With a time step of 25 fs the RMS fluctuation rises to 0.04 kcal/mol and with a time step of 5 fs the value is 0.002 kcal/mol. A variation of one part in 10^4 is generally considered acceptable. The different algorithms may vary in the rate at which the error varies with the time step. For example, it has been shown that for short time steps the predictor-corrector methods may be more accurate, but for longer time steps the Verlet algorithm may be better [Fincham and Heyes 1982]. Other factors that may be important when choosing an integration algorithm include the memory required; the synchronisation of positions and velocities; whether they are self-starting (some methods require properties at $t - \delta t$, which obviously do not exist); and

Fig. 12.41: Contour representation of key features from a CoMFA analysis of a series of coumarin substrates and inhibitors of cytochrome $P_{450}2A5$ [Poso et al. 1995]. The red and blue regions indicate positions where it would be favourable and unfavourable, respectively, to place a negative charge and the green and yellow regions would be favourable/unfavourable to locate steric bulk.

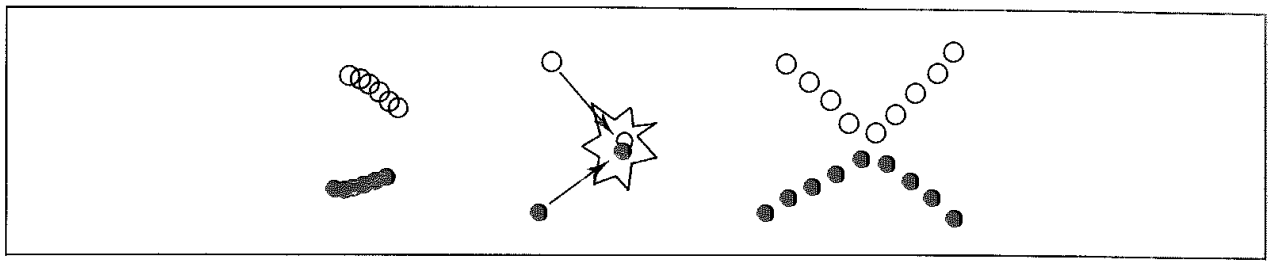

Fig 7 4· With a very small time step (left) phase space is covered very slowly; a large time step (middle) gives instabilities With an appropriate time step (right) phase space is covered efficiently and collisions occur smoothly

consisting of two argon atoms interacting under the Lennard-Jones potential. The behaviour of this system can be determined analytically and so compared with the numerical integration. Suppose the argon atoms are moving towards each other along the x axis with initial velocities of 353 m s^{-1} (this corresponds to the most probable speed of argon at 300 K) We can then plot how the interatomic distance varies with time and compare it to the analytical potential. The result obtained using two time steps (10 fs and 50 fs) are shown in Figure 7 5. In both cases the numerical trajectory initially lags behind the analytical one, but then as the atoms pass through their minimum energy separation and move up the repulsive barrier the atoms 'jump through' the energy barrier. This leads to a gain in energy and the atoms then move apart with velocities that are slightly too high. In both numerical trajectories the total energy rises after the collision. Unfortunately, the atoms move most quickly and take the largest steps in the very region (i.e. near the energy minimum) where it would be best to take the smallest steps. The total error is correlated with the time step, with the largest errors arising for the largest time steps. Of course, with a small time step much more computer time will be required for a given length of calculation; the aim is to find the correct balance between simulating the 'correct' trajectory and covering the phase space. If the time

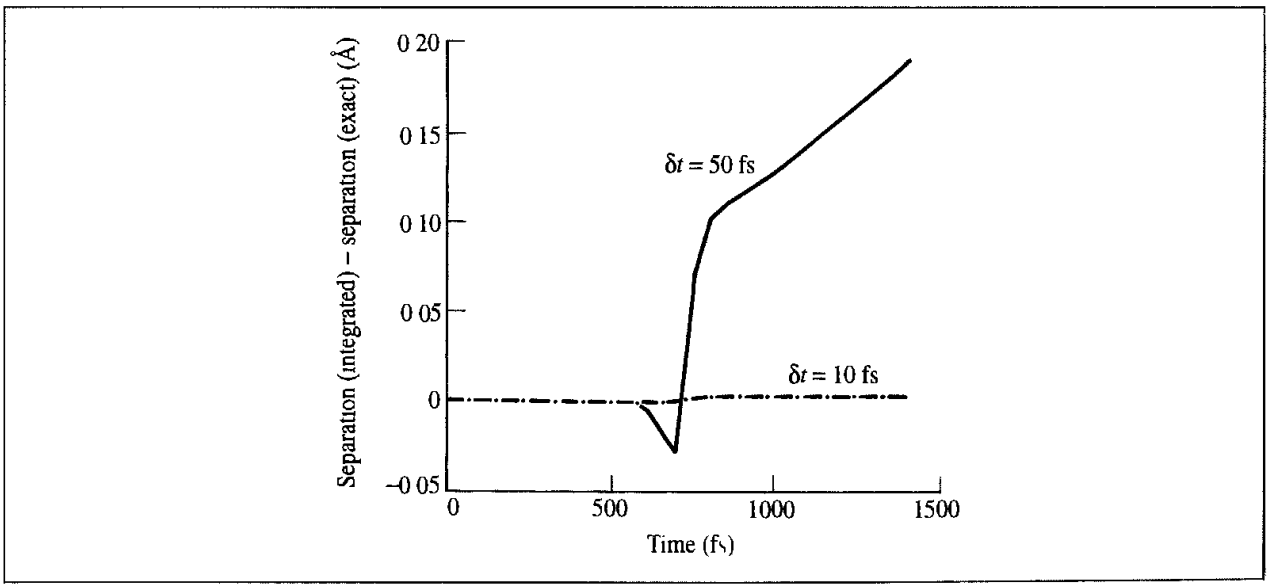

Fig. 7 5. Difference between the exact and numerical trajectories for the approach of two argon atoms with time steps of 10 fs and 50 fs

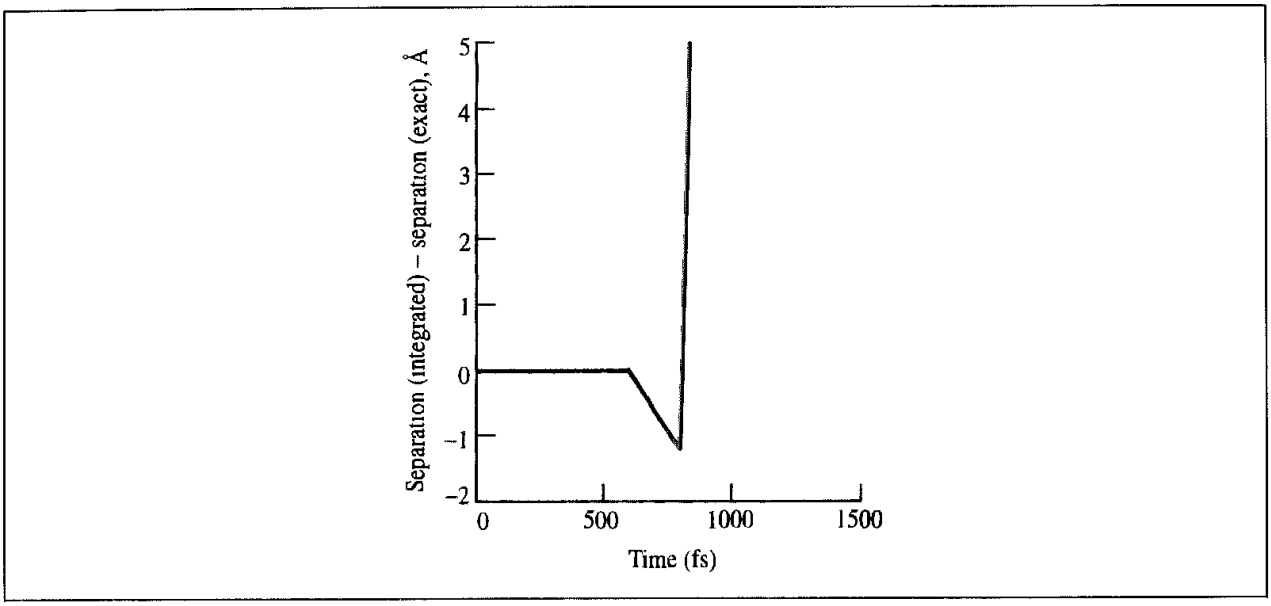

Fig 7 6 Difference between exact and numerical trajectory for the approach of two argon atoms for a time step of 100 fs The simulation 'blows up'

step is too large, then the trajectory will 'blow up', as can be seen for the argon dimer system with a time step of 100 fs (Figure 7 6)

When simulating an atomic fluid the time step should be small compared to the mean time between collisions. When simulating flexible molecules a useful guide is that the time step should be approximately one-tenth the time of the shortest period of motion. In flexible molecules, the highest-frequency vibrations are due to bond stretches, especially those of bonds to hydrogen atoms A C–H bond vibrates with a repeat period of approximately 10 fs. The timescales of some typical motions together with appropriate time steps are shown in Table 7.1, which can be used to choose an appropriate time step.

The requirement that the time step is approximately one order of magnitude smaller than the shortest motion is clearly a severe restriction, particularly as these high-frequency motions are usually of relatively little interest and have a minimal effect on the overall behaviour of the system. One solution to this problem is to 'freeze out' such vibrations by constraining the appropriate bonds to their equilibrium values while still permitting the rest of the degrees of freedom to vary under the intramolecular and intermolecular forces present. This enables a longer time step to be used. We will consider such constraint dynamics methods in Section 7.5.

System	Types of motion present	Suggested time step (s)
Atoms	Translation	10^{-14}
Rigid molecules	Translation and rotation	5×10^{-15}
Flexible molecules, rigid bonds	Translation, rotation, torsion	2×10^{-15}
Flexible molecules, flexible bonds	Translation, rotation, torsion, vibration	10^{-15} or 5×10^{-16}

Table 7 1 The different types of motion present in various systems together with suggested time steps.

7.3.5 Multiple Time Step Dynamics

Table 7.1 presents us with something of a dilemma. We would obviously desire to explore as much of the phase space as possible but this may be compromised by the need for a small time step. One possible approach is to use a multiple time step method. The underlying rationale is that certain interactions evolve more rapidly with time than other interactions The twin-range method (Section 6.7.1) is a crude type of multiple time step approach, in that interactions involving atoms between the lower and upper cutoff distance remain constant and change only when the neighbour list is updated. However, this approach can lead to an accumulation of numerical errors in calculated properties. A more sophisticated approach is to approximate the forces due to these atoms using a Taylor series expansion [Streett *et al.* 1978]:

$$\mathbf{f}(t+\tau\delta t) = \mathbf{f}(t) + (\tau\delta t)\mathbf{df}(t)/\mathbf{dt} + \tfrac{1}{2}(\tau\delta t)\mathbf{d}^2\mathbf{f}(t)/\mathbf{dt}^2 + \cdots \tag{7.30}$$

This series expansion is truncated at a specified order and is probably most easily implemented within a predictor–corrector type of algorithm, where the higher-order terms are already computed. This method has been applied to relatively simple systems such as molecular fluids [Streett *et al.* 1978] and alkane chain liquids [Swindoll and Haile 1984].

An alternative formulation of a multiple time step method is the 'reversible reference system propagation algorithm' (r-RESPA) method [Tuckerman *et al.* 1992]. In this method, the forces within a system are classified into a number of groups according to how rapidly the force varies over time. Each group then has its own time step while maintaining accuracy and numerical stability. The starting point for this algorithm is the Liouville equation, which defines how the state of the system, $\Gamma(t)$, evolves over time:

$$\Gamma(t) = e^{iLt}\Gamma \quad (t=0) \tag{7.31}$$

The exponential $\exp(iLt)$ in Equation (7.31) involves the so-called *Liouville operator*, L, which in the case of a molecular system containing N atoms (and so $3N$ coordinates) can be expressed:

$$iL = \sum_{i=1}^{3N}\left[\frac{\partial x_i}{\partial t}\frac{\partial}{\partial x_i} + F_i(x)\frac{\partial}{\partial p_i}\right] \tag{7.32}$$

In the r-RESPA method this operator is decomposed into two or more parts, for example:

$$L = L_1 + L_2 + L_3 + L_4 \tag{7 33}$$

Each of these parts is then associated with specific terms in the force equation. For example, L_1 may correspond to the bond-stretching terms, L_2 to the angle-bending and torsional terms, L_3 to the short-range non-bonded interactions and L_4 to the long-range non-bonded interactions. Suppose the time step with which we evaluate the bond-stretching terms is δt_1. Integers n_1, n_2 and n_3 then define the time steps for the three other forces as follows:

$$\delta t_2 = n_1\delta t_1; \quad \delta t_3 = n_1 n_2 \delta t_1; \quad \delta t_4 = n_1 n_2 n_3 \delta t_1 \tag{7.34}$$

The underlying theory of r-RESPA is somewhat involved, but the final result and consequent implementation is actually rather straightforward, being very closely related to the velocity Verlet integration scheme. For our four-way decomposition the algorithm would

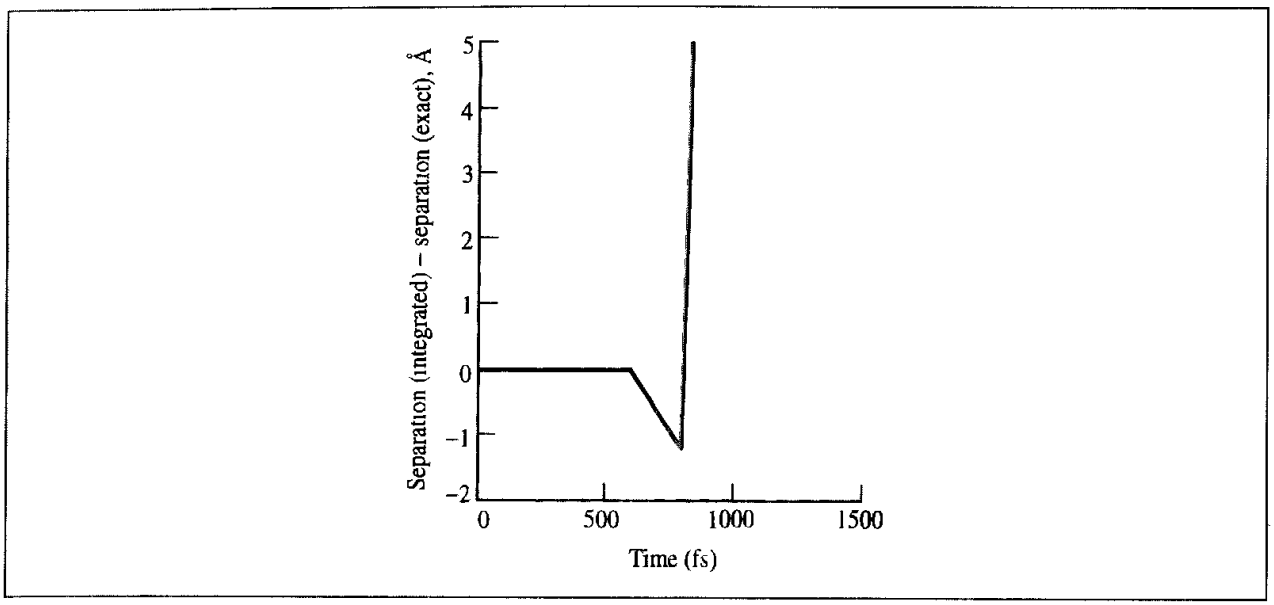

Fig 7 6. Difference between exact and numerical trajectory for the approach of two argon atoms for a time step of 100 fs The simulation 'blows up'

step is too large, then the trajectory will 'blow up', as can be seen for the argon dimer system with a time step of 100 fs (Figure 7.6)

When simulating an atomic fluid the time step should be small compared to the mean time between collisions. When simulating flexible molecules a useful guide is that the time step should be approximately one-tenth the time of the shortest period of motion In flexible molecules, the highest-frequency vibrations are due to bond stretches, especially those of bonds to hydrogen atoms A C–H bond vibrates with a repeat period of approximately 10 fs. The timescales of some typical motions together with appropriate time steps are shown in Table 7.1, which can be used to choose an appropriate time step.

The requirement that the time step is approximately one order of magnitude smaller than the shortest motion is clearly a severe restriction, particularly as these high-frequency motions are usually of relatively little interest and have a minimal effect on the overall behaviour of the system One solution to this problem is to 'freeze out' such vibrations by constraining the appropriate bonds to their equilibrium values while still permitting the rest of the degrees of freedom to vary under the intramolecular and intermolecular forces present. This enables a longer time step to be used. We will consider such constraint dynamics methods in Section 7.5.

System	Types of motion present	Suggested time step (s)
Atoms	Translation	10^{-14}
Rigid molecules	Translation and rotation	5×10^{-15}
Flexible molecules, rigid bonds	Translation, rotation, torsion	2×10^{-15}
Flexible molecules, flexible bonds	Translation, rotation, torsion, vibration	10^{-15} or 5×10^{-16}

Table 7 1 The different types of motion present in various systems together with suggested time steps

7.3.5 Multiple Time Step Dynamics

Table 7.1 presents us with something of a dilemma. We would obviously desire to explore as much of the phase space as possible but this may be compromised by the need for a small time step. One possible approach is to use a multiple time step method The underlying rationale is that certain interactions evolve more rapidly with time than other interactions. The twin-range method (Section 6.7.1) is a crude type of multiple time step approach, in that interactions involving atoms between the lower and upper cutoff distance remain constant and change only when the neighbour list is updated. However, this approach can lead to an accumulation of numerical errors in calculated properties. A more sophisticated approach is to approximate the forces due to these atoms using a Taylor series expansion [Streett *et al.* 1978]:

$$\mathbf{f}(t+\tau\delta t) = \mathbf{f}(t) + (\tau\delta t)\mathbf{df}(t)/\mathbf{dt} + \tfrac{1}{2}(\tau\delta t)\mathbf{d}^2\mathbf{f}(t)/\mathbf{dt}^2 + \cdots \tag{7.30}$$

This series expansion is truncated at a specified order and is probably most easily implemented within a predictor–corrector type of algorithm, where the higher-order terms are already computed. This method has been applied to relatively simple systems such as molecular fluids [Streett *et al.* 1978] and alkane chain liquids [Swindoll and Haile 1984].

An alternative formulation of a multiple time step method is the 'reversible reference system propagation algorithm' (r-RESPA) method [Tuckerman *et al.* 1992]. In this method, the forces within a system are classified into a number of groups according to how rapidly the force varies over time. Each group then has its own time step while maintaining accuracy and numerical stability. The starting point for this algorithm is the Liouville equation, which defines how the state of the system, $\Gamma(t)$, evolves over time:

$$\Gamma(t) = e^{iLt}\Gamma \quad (t=0) \tag{7.31}$$

The exponential $\exp(iLt)$ in Equation (7.31) involves the so-called *Liouville operator*, L, which in the case of a molecular system containing N atoms (and so $3N$ coordinates) can be expressed:

$$iL = \sum_{i=1}^{3N}\left[\frac{\partial x_i}{\partial t}\frac{\partial}{\partial x_i} + F_i(x)\frac{\partial}{\partial p_i}\right] \tag{7.32}$$

In the r-RESPA method this operator is decomposed into two or more parts, for example:

$$L = L_1 + L_2 + L_3 + L_4 \tag{7.33}$$

Each of these parts is then associated with specific terms in the force equation. For example, L_1 may correspond to the bond-stretching terms, L_2 to the angle-bending and torsional terms, L_3 to the short-range non-bonded interactions and L_4 to the long-range non-bonded interactions. Suppose the time step with which we evaluate the bond-stretching terms is δt_1. Integers n_1, n_2 and n_3 then define the time steps for the three other forces as follows.

$$\delta t_2 = n_1\delta t_1; \quad \delta t_3 = n_1 n_2 \delta t_1; \quad \delta t_4 = n_1 n_2 n_3 \delta t_1 \tag{7.34}$$

The underlying theory of r-RESPA is somewhat involved, but the final result and consequent implementation is actually rather straightforward, being very closely related to the velocity Verlet integration scheme. For our four-way decomposition the algorithm would

be implemented as follows:

```
Calculate forces-1 (i.e. a_1(t))
Calculate forces-2 (i.e. a_2(t))
Calculate forces-3 (i.e. a_3(t))
Calculate forces-4 (i.e. a_4(t))
do step = 1, N_steps
    v = v + (1/2) n_1 n_2 n_3 δt_1 a_4
    do i_3 = 1, n_3
        v = v + (1/2) n_1 n_2 δt_1 a_3
        do i_2 = 1, n_2
            v = v + (1/2) n_1 δt_1 a_2
            do i_1 = 1, n_1
                v = v + (1/2) δt_1 a_1
                r = r + δt_1 v
                calculate forces-1 (i.e. a_1)
                v = v + (1/2) δt_1 a_1
            enddo
            calculate forces-2 (i.e. a_2)
            v = v + (1/2) n_1 δt_1 a_2
        enddo
        calculate forces-3 (i e. a_3)
        v = v + (1/2) n_1 n_2 δt_1 a_3
    enddo
    calculate forces-4 (i.e. a_4)
    v = v + (1/2) n_1 n_2 n_3 δt_1 a_4
enddo
```

In this scheme, **v** and r refer to one of the $3N$ velocities or positions, respectively. Note that the different types of force are calculated throughout the algorithm. It can be readily seen that the method reduces to the standard velocity Verlet method if n_1, n_2 and n_3 are set equal to 1.

The r-RESPA method has been applied to a variety of systems, including simple model systems [Tuckerman *et al.* 1992] but also organic molecules [Watanabe and Karplus 1993], fullerene crystals [Procacci and Berne 1994] and also proteins [Humphreys *et al.* 1994, 1996]. In these studies the reduction in computational time compared with the standard velocity Verlet method varied between 4–5 and 20–40, depending upon the size of the system, without any noticeable loss in accuracy. Other developments of the r-RESPA algorithm include its coupling to the fast multipole method (see Section 6.8.3) [Zhou and Berne 1995].

7.4 Setting Up and Running a Molecular Dynamics Simulation

In this section we will examine some of the steps involved in performing a molecular dynamics simulation in the microcanonical ensemble First, it is necessary to establish an

initial configuration of the system. As discussed in Section 6.4.2, the initial configuration may be obtained from experimental data, from a theoretical model or from a combination of the two. It is also necessary to assign initial velocities to the atoms. This can be done by randomly selecting from a Maxwell–Boltzmann distribution at the temperature of interest:

$$p(v_{ix}) = \left(\frac{m_i}{2\pi k_B T}\right)^{1/2} \exp\left[-\frac{1}{2}\frac{m_i v_{ix}^2}{k_B T}\right] \tag{7.35}$$

The Maxwell–Boltzmann equation provides the probability that an atom i of mass m_i has a velocity v_{ix} in the x direction at a temperature T. A Maxwell–Boltzmann distribution is a Gaussian distribution, which can be obtained using a random number generator. Most random number generators are designed to produce random numbers that are uniform in the range 0 to 1. However, it is relatively straightforward to convert such a random number generator to sample from a Gaussian distribution (or indeed from one of several other distributions [Rubinstein 1981]). The probability of generating a value from a Gaussian (normal) distribution with mean $\langle x \rangle$ and variance σ^2 ($\sigma^2 = \langle (x - \langle x \rangle)^2 \rangle$) is:

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(x - \langle x \rangle)^2}{2\sigma^2}\right] \tag{7.36}$$

One option is to first generate two random numbers ξ_1 and ξ_2 between 0 and 1. The corresponding two numbers from the normal distribution are then calculated using

$$x_1 = \sqrt{-2\ln\xi_1}\cos(2\pi\xi_2) \quad \text{and} \quad x_2 = \sqrt{-2\ln\xi_1}\sin(\pi\xi_2) \tag{7.37}$$

An alternative approach is to generate twelve random numbers $\xi_1, \ldots, \xi_2$ and then calculate:

$$x = \sum_{i=1}^{12} \xi_i - 6 \tag{7.38}$$

These two methods generate random numbers in the normal distribution with zero mean and unit variance. A number (x) generated from this distribution can be related to its counterpart (x') from another Gaussian distribution with mean $\langle x' \rangle$ and variance σ using

$$x' = \langle x' \rangle + \sigma x \tag{7.39}$$

The initial velocities may also be chosen from a uniform distribution or from a simple Gaussian distribution. In either case the Maxwell–Boltzmann distribution of velocities is usually rapidly achieved.

The initial velocities are often adjusted so that the total momentum of the system is zero. Such a system then samples from the constant *NVEP* ensemble. To set the total linear momentum of the system to zero, the sum of the components of the atomic momenta along the x, y and z axes is calculated. This gives the total momentum of the system in each direction, which, when divided by the total mass, is subtracted from the atomic velocities to give an overall momentum of zero.

Having set up the system and assigned the initial velocities, the simulation proper can commence. At each step the force on each atom must be calculated by differentiating the potential function. The force on an atom may include contributions from the various

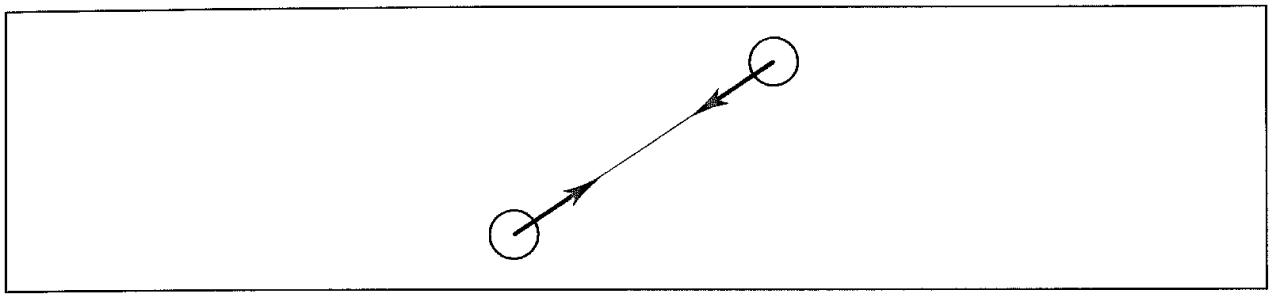

Fig. 7 7· The force between two particles acts along the line joining their centres of mass, in accordance with Newton's third law

terms in the force field such as bonds, angles, torsional terms and non-bonded interactions. The force is straightforward to calculate for two atoms interacting under the Lennard-Jones potential:

$$\mathbf{f}_{ij} = \frac{\mathbf{r}_{ij}}{|\mathbf{r}_{ij}|} \frac{24\varepsilon}{\sigma} \left[2\left(\frac{\sigma}{r_{ij}}\right)^{13} - \left(\frac{\sigma}{r_{ij}}\right)^{7} \right] \tag{7.40}$$

The force between the two atoms is equal in magnitude and opposite in direction and applies along the line connecting the two nuclear centres, in accordance with Newton's third law (Figure 7.7). It is necessary to calculate the force between each atom pair just once. This is most easily achieved by arranging to compute the force between an atom and those atoms with a higher index (i.e. for an atom i the forces are calculated with atoms $i+1, i+2, \ldots, N$). Having calculated the force between an atom i and an atom with a higher index j, minus the force is added to the accumulating sum of the forces on j. The force calculation is most easily implemented using two loops, as outlined in the following pseudocode:

```
set elements on force array to zero
while atom1 = 1 to N − 1
    while atom2 = atom1 + 1 to N
        calculate force on atom1 due to interaction with atom2
        add the force to the array element, atom1
        subtract the force from the appropriate force array element atom2
    enddo
enddo
```

At the end of these two loops, the total force on each atom is known. A consequence of the fact that the force between the two atoms is equal and opposite is that the neighbour list for each atom need only contain those atoms with a higher number as the force on an atom due to interactions with lower numbered atoms will be calculated earlier in the loop. This organisation of the neighbour list contrasts with the structure used for Monte Carlo simulations, where all the neighbours of each atom (with both lower and higher indices) must be stored.

Analytical expressions for the forces due to other terms in the molecular mechanics potential function have been published for most of the functional forms encountered in common force

fields. These expressions can seem rather complicated because the intramolecular terms (e.g. bonds, angles, torsions) are calculated in terms of the internal coordinates, whereas molecular dynamics is typically performed using Cartesian coordinates. The chain rule must therefore be used to obtain the desired functional forms. However, the resulting expressions are relatively easy to implement in a computer program.

The first stage of a molecular dynamics simulation is the equilibration phase, the purpose of which is to bring the system to equilibrium from the starting configuration. During equilibration, various parameters are monitored together with the actual configurations When these parameters achieve stable values then the production phase can commence. It is during the production phase that thermodynamic properties and other data are calculated. The parameters that are used to characterise whether equilibrium has been reached depend to some extent on the system being simulated but invariably include the kinetic, potential and total energies, the velocities, the temperature and the pressure. As we have indicated, the kinetic and potential energies would be expected to fluctuate in a simulation in the microcanonical ensemble but the total energy should remain constant. The components of the velocities should describe a Maxwell–Boltzmann distribution (in all three directions x, y and z) and the kinetic energy should be equally distributed among the three directions x, y and z. It is usually desired to perform a simulation at a specified temperature and so it is common practice to adjust the temperature of the system by scaling the velocities (see Section 7.7.1) during the equilibration phase. During the production phase the temperature is a variable of the system. Order parameters may be calculated to monitor changes in structure, which can supplement visual examination of the evolving trajectory.

When simulating an inhomogeneous system a more detailed equilibration procedure is usually desirable. A typical procedure suitable for a molecular dynamics simulation of a macromolecular solute, such as a protein in solution, would be as follows. First, the solvent alone together with any mobile counterions is subject to energy minimisation with the solute kept fixed in its initial conformation. The solvent and any counterions are then allowed to evolve using either a molecular dynamics (or indeed Monte Carlo) simulation, again keeping the structure of the solute molecule fixed. This solvent equilibration phase should be sufficiently extensive to allow the solvent to completely readjust to the potential field of the solute. For molecular dynamics this implies that the length of this solvent equilibration phase should be longer than the relaxation time of the solvent (the time taken for a molecule to lose any 'memory' of its original orientation, which for water is about 10 ps). Next, the entire system (solute and solvent) is minimised. Only then does the molecular dynamics simulation of the whole system commence.

At the start of the production phase all counters are set to zero and the system is permitted to evolve. In a microcanonical ensemble no velocity scaling is performed during the production phase and so the temperature becomes a calculated property of the system Various properties are routinely calculated and stored during the production phase for subsequent analysis and processing. Careful monitoring of these properties during the simulation can show whether the simulation is 'well behaved' or not; it may be necessary to restart a simulation if problems are encountered. It is also usual to store the positions, energies

and velocities of configurations at regular intervals (e.g. every five to twenty time steps), from which other properties can be determined once the simulation has finished.

7.4.1 Calculating the Temperature

Many thermodynamic properties can be calculated from a molecular dynamics simulation. Most of these were dealt with in Section 6.2; here we just discuss the calculation of temperature. The instantaneous value of the temperature is related to the kinetic energy via the particles' momenta as follows:

$$\mathscr{K} = \sum_{i=1}^{N} \frac{|\mathbf{p}_i|^2}{2m_i} = \frac{k_B T}{2}(3N - N_c) \tag{7.41}$$

where N_c is the number of constraints and so $3N - N_c$ is the total number of degrees of freedom. For an isolated system (i.e. for a simulation of a system *in vacuo*) the total translational momentum of the system and the total angular momentum are conserved and can be made equal to zero by an appropriate choice of initial velocities. For a simulation performed using periodic boundary conditions, the total linear momentum is conserved but the total angular momentum is not. It is common practice to choose a set of initial velocities that ensures that the total linear momentum and the total angular momentum are zero. As the system evolves, the linear momentum should remain zero but the angular momentum will not. Molecular dynamics with periodic boundary conditions thus strictly samples from the constant $NVE\mathbf{P}$ ensemble where $\mathbf{P}$ is the total linear momentum. This differs trivially from the standard microcanonical ensemble but it should be remembered that the appropriate number of degrees of freedom must be subtracted from the total when calculating the kinetic energy per degree of freedom. Specifically, for a system *in vacuo* where the total linear and angular momenta have been set to zero, six degrees of freedom need to be subtracted. For a simulation using periodic boundary conditions three degrees of freedom need to be subtracted if the centre-of-mass motion of the system is removed. In constraint dynamics, discussed in the next section, rather more degrees of freedom may be fixed and N_c must be calculated accordingly.

7.5 Constraint Dynamics

The earliest molecular dynamics simulations using 'realistic' potentials were of atoms interacting under the Lennard-Jones potential. In such calculations the only forces on the atoms are those due to non-bonded interactions. It is rather more difficult to simulate molecules because the interaction between two non-spherical molecules depends upon their relative orientation as well as the distance between them If the molecules are flexible then there will also be intramolecular interactions, which give rise to changes in conformation. Clearly, the simplest model is to treat the species present as rigid bodies with no intramolecular conformational freedom. In such cases the dynamics of each molecule can often be considered in terms of translations of its centre of mass and rotations about its centre of mass. The force on the molecule equals the vector sum of all the forces acting at the

centre of mass, and the rotational motion is determined by the torque about the centre of mass. To deal with these rotational motions is considerably more complicated than for the translational motions, but in favourable cases they can be programmed quite efficiently

When the simulation involves conformationally flexible molecules then the motion is inevitably described in terms of the atomic Cartesian coordinates The conformational behaviour of a flexible molecule is usually a complex superposition of different motions. The high frequency motions (e.g. bond vibrations) are usually of less interest than the lower frequency modes, which often correspond to major conformational changes. Unfortunately, the time step of a molecular dynamics simulation is dictated by the highest frequency motion present in the system. It would therefore be of considerable benefit to be able to increase the time step without prejudicing the accuracy of the simulation. Constraint dynamics enables individual internal coordinates or combinations of specified coordinates to be constrained, or 'fixed' during the simulation without affecting the other internal degrees of freedom.

Before we consider in detail the use of constraint dynamics, it is helpful to establish the difference between *constraints* and *restraints*; we shall discuss the method of restrained molecular dynamics in a later chapter (see Section 9.10). A constraint is a requirement that the system is forced to satisfy. As we shall see, in constraint dynamics bonds or angles are forced to adopt specific values throughout a simulation. When a bond or angle is restrained then it *is* able to deviate from the desired value; the restraint only acts to 'encourage' a particular value. Restraints are most easily incorporated using additional terms in the force field which impose a penalty for deviations from the reference value. An additional difference is that restrained degrees of freedom still have an energy $k_B T/2$ associated with them, whereas constrained degrees of freedom do not.

The most commonly used method for applying constraints, particularly in molecular dynamics, is the SHAKE procedure of Ryckaert, Ciccotti and Berendsen [Ryckaert *et al.* 1977]. In constraint dynamics the equations of motion are solved while simultaneously satisfying the imposed constraints. Constrained systems have been much studied in classical mechanics; we shall illustrate the general principles using a simple system comprising a box sliding down a frictionless slope in two dimensions (Figure 7.8). The box is constrained to remain on the slope and so the box's x and y coordinates must always satisfy the equation of the slope (which we shall write as $y = mx + c$). If the slope were not present then the box

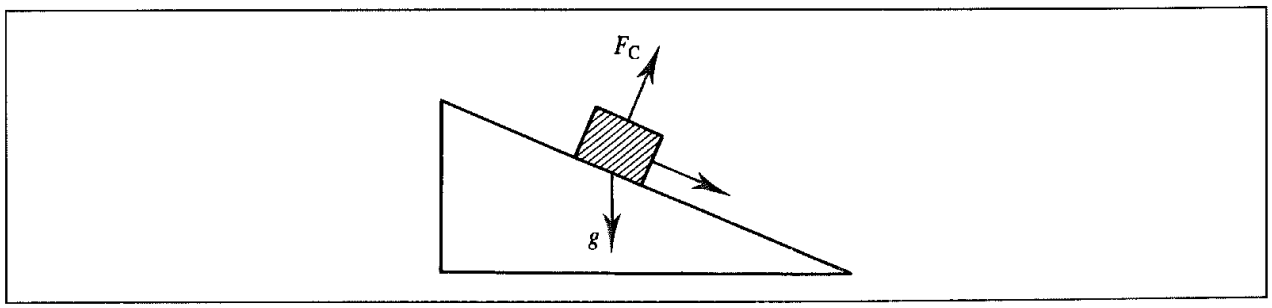

Fig. 7 8 A box sliding down a slope under the influence of gravity is subject to the constraint that it must remain on the slope. The constraint force F_c acts perpendicular to the direction of motion

would fall vertically downwards. Constraints are often categorised as *holonomic* or *non-holonomic*. Holonomic constraints can be expressed in the form

$$f(q_1, q_2, q_3, \ldots, t) = 0 \tag{7.42}$$

q_1, q_2, etc., are the coordinates of the particles. Non-holonomic constraints cannot be expressed in this way. For example, the motion of a particle constrained to lie on the surface of a sphere is subject to a holonomic constraint, but if the particle is able to fall off the sphere under the influence of gravity then the constraint becomes non-holonomic. A holonomic constraint that keeps a particle on the surface of a sphere can be written:

$$r^2 - a^2 = 0 \tag{7.43}$$

r is the distance of the particle from the origin where the sphere of radius a is centred. The equivalent non-holonomic constraint is written as an inequality:

$$r^2 - a^2 \geq 0 \tag{7.44}$$

SHAKE uses holonomic constraints. In a constrained system the coordinates of the particles are not independent and the equations of motion in each of the coordinate directions are connected. A second difficulty is that the magnitude of the constraint forces is unknown. Thus in the case of the box on the slope, the gravitational force acting on the box is in the y direction whereas the motion is down the slope. The motion is thus not in the same direction as the gravitational force As such, the total force on the box can be considered to arise from two sources one due to gravity and the other a constraint force that is perpendicular to the motion of the box (Figure 7 8) As there is no motion perpendicular to the surface of the slope, the constraint force does no work

As we know, the motion of a system of N particles can be described in terms of $3N$ independent coordinates or degrees of freedom. If there are k holonomic constraints then the number of degrees of freedom is reduced to $3N - k$. It is possible, in principle at least, to find $3N - k$ independent coordinates (the *generalised coordinates*), which can then be used to solve the problem directly. For example, the motion of the box can be described using the single coordinate, q, along the direction of the slope. The component of the gravitational force that acts along the slope is $Mg \sin\theta$ and so the acceleration down the slope is $g \sin\theta$. The position at any time t can thus be obtained by integrating the following equation of motion:

$$\frac{d^2q}{dt^2} = g \sin\theta \tag{7.45}$$

The solution to this equation is:

$$q(t) = q(0) + t\dot{q}(0) + \frac{t^2}{2} g \sin\theta \tag{7.46}$$

where $q(0)$ is the value of ξ at time $t = 0$ and $\dot{q}(0)$ is the initial velocity of the box along the slope. In this simple example it is quite easy to identify the single generalised coordinate that can be used to describe the motion in the constrained system. When there are many constraints, it can be difficult to determine the generalised coordinates. In any case, it is usually desirable to work with the atomic Cartesian coordinates. The motion of the box can be more generally described in terms of the Cartesian (x, y) coordinates of the box as follows

Newton's equations in the x and y directions are:

$$M\frac{d^2x}{dt^2} = F_{cx} \tag{7 47}$$

$$M\frac{d^2y}{dt^2} = -Mg + F_{cy} \tag{7 48}$$

where F_{cx} and F_{cy} are the components of the as yet unknown constraint force in the x and the y directions, respectively. We know that the constraint force acts perpendicular to the slope, and so the ratio of its x and y components must be:

$$\frac{F_{cx}}{F_{cy}} = -m \tag{7.49}$$

The constraint force can be introduced into Newton's equations as a Lagrange multiplier (see Section 1.10.5). To achieve consistency with the usual Lagrangian notation, we write F_{cy} as $-\lambda$ and so F_{cx} equals λm. Thus:

$$M\frac{d^2x}{dt^2} = \lambda m \tag{7.50}$$

$$M\frac{d^2y}{dt^2} = -Mg - \lambda \tag{7 51}$$

Equations (7.50) and (7.51) contain three unknowns (d^2x/dt^2, d^2y/dt^2 and λ). A third equation that links x and y is the equation of the slope, which can be written in the following form:

$$\sigma = mx - y + c = 0 \tag{7.52}$$

This constraint equation is expressed in terms of x and y rather than their second derivatives. However, as $\sigma(x, y) = 0$ holds for all x, y, it follows that $d\sigma = 0$ and $d^2\sigma = 0$ also. Consequently, the constraint equation can be written:

$$m\frac{d^2x}{dt^2} - \frac{d^2y}{dt^2} = 0 \tag{7 53}$$

Solving the three equations gives:

$$\frac{d^2x}{dt^2} = -g\frac{m}{1+m^2} \tag{7.54}$$

$$\frac{d^2y}{dt^2} = -g\frac{m^2}{1+m^2} \tag{7 55}$$

The x and y coordinates at time t are thus given by:

$$x(t) = x(0) + t\frac{dx(0)}{dt} - g\frac{t^2}{2}\frac{m}{(1+m^2)} \tag{7.56}$$

$$y(t) = y(0) + t\frac{dy(0)}{dt} - g\frac{t^2}{2}\frac{m^2}{(1+m^2)} \tag{7 57}$$

In the general case, the equations of motion for a constrained system involve two types of force: the 'normal' forces arising from the intra- and intermolecular interactions, and the forces due to the constraints. We are particularly interested in the case where the constraint σ_k requires the bond between atoms i and j to remain fixed. The constraint influences the Cartesian coordinates of atoms i and j. The force due to this constraint can be written as follows:

$$F_{ckx} = \lambda_k \frac{\partial \sigma_k}{\partial x} \tag{7.58}$$

where λ_k is the Lagrange multiplier and x represents one of the Cartesian coordinates of the two atoms. Applying Equation (7.58) to the above example, we would write $F_{cx} = \lambda \partial\sigma/\partial x = \lambda m$ and $F_{cy} = \lambda \partial\sigma/\partial y = -\lambda$. If an atom is involved in a number of constraints (because it is involved in more than one constrained bond) then the total constraint force equals the sum of all such terms. The nature of the constraint for a bond between atoms i and j is:

$$\sigma_{ij} = (\mathbf{r}_i - \mathbf{r}_j)^2 - d_{ij}^2 = 0 \tag{7.59}$$

The constraint force lies along the bond at all times. For each constrained bond, there is an equal and opposite force on the two atoms that comprise the bond. The overall effect is that the constraint forces do no work. Suppose the constraint k corresponds to the bond length between atoms i and j. The constraint forces are obtained by differentiating the constraint with respect to the coordinates of atoms i and j and multiplying by an (as yet) undetermined multiplier:

$$\partial\sigma_k/\partial\mathbf{r}_i = 2(\mathbf{r}_i - \mathbf{r}_j) \quad \text{so} \quad F_{ci} = \lambda(\mathbf{r}_i - \mathbf{r}_j) \tag{7.60}$$

$$\partial\sigma_k/\partial\mathbf{r}_j = -2(\mathbf{r}_i - \mathbf{r}_j) \quad \text{and} \quad F_{cj} = -\lambda(\mathbf{r}_i - \mathbf{r}_j) \tag{7.61}$$

The factor of 2 that arises when we differentiate the square term has been incorporated into the Lagrange multiplier λ. The above expression for the forces can be incorporated into the Verlet algorithm as follows:

$$\mathbf{r}_i(t+\delta t) = 2\mathbf{r}_i(t) - \mathbf{r}_i(t-\delta t) + \frac{\delta t^2}{m_i}\mathbf{F}_i(t) + \sum_k \frac{\lambda_k \delta t^2}{m_i}\mathbf{r}_{ij}(t) \tag{7.62}$$

Recall that the positions that would be obtained from the Verlet algorithm without constraints are $\mathbf{r}_i'(t+\delta t) = 2\mathbf{r}_i(t) - \mathbf{r}_i(t-\delta t) + \delta t^2 \mathbf{F}_i(t)/m_i$. The summation in Equation (7.62) is over all constraints k that affect atom i. These constraints perturb the positions that would otherwise have been obtained from the integration algorithm, and so the above expression can be written:

$$\mathbf{r}_i(t+\delta t) = \mathbf{r}_i'(t+\delta t) + \sum_k \frac{\lambda_k \delta t^2}{m_i}\mathbf{r}_{ij}(t) \tag{7.63}$$

The next problem is to determine the multipliers λ_k that enable all the constraints to be satisfied simultaneously. This can be done algebraically in simple cases. Suppose we wish to fix the bond in a diatomic molecule. There is just one constraint in this case, and if the

atoms are labelled 1 and 2 we can write:

$$\mathbf{r}_1(t+\delta t) = \mathbf{r}'_1(t+\delta t) + \lambda_{12}(\delta t^2/m_1)(\mathbf{r}_1(t) - \mathbf{r}_2(t)) \quad (7\,64)$$

$$\mathbf{r}_2(t+\delta t) = \mathbf{r}'_2(t+\delta t) - \lambda_{12}(\delta t^2/m_2)(\mathbf{r}_1(t) - \mathbf{r}_2(t)) \quad (7\,65)$$

A third equation is derived from the requirement that the new positions keep the bond at the required distance:

$$|\mathbf{r}_1(t+\delta t) - \mathbf{r}_2(t+\delta t)|^2 = |\mathbf{r}_1(t) - \mathbf{r}_2(t)|^2 = d_{12}^2 \quad (7\,66)$$

We now have three equations and three unknowns ($\mathbf{r}_1(t+\delta t)$, $\mathbf{r}_2(t+\delta t)$ and λ_{12}). Subtracting, and putting $\mathbf{r}_{12}(t) = (\mathbf{r}_1(t) - \mathbf{r}_2(t))$ and $\mathbf{r}'_{12}(t+\delta t) = \mathbf{r}'_1(t+\delta t) - \mathbf{r}'_2(t+\delta t)$ gives:

$$\mathbf{r}_1(t+\delta t) - \mathbf{r}_2(t+\delta t) = \mathbf{r}'_{12}(t+\delta t) + \lambda_{12}\delta t^2(1/m_1 + 1m_2)\mathbf{r}_{12}(t) \quad (7\,67)$$

Squaring both sides and imposing the constraint gives:

$$\mathbf{r}'_{12}(t+\delta t)^2 + 2\lambda_{12}\delta t^2(1/m_1 + 1/m_2)\mathbf{r}_{12}(t) + \lambda_{12}^2\delta t^4(1/m_1 + 1/m_2)^2\mathbf{r}_{12}(t)^2 = d_{12}^2 \quad (7.68)$$

Solving this quadratic equation for λ_{12} enables the new positions $\mathbf{r}_1(t+\delta t)$ and $\mathbf{r}_2(t+\delta t)$ to be determined.

In the case of a triatomic molecule with two bonds (between atoms 1,2 and 2,3), two constraint equations are obtained:

$$\mathbf{r}_{12}(t+\delta t) = \mathbf{r}'_{12}(t+\delta t) + \delta t^2(1/m_1 + 1/m_2)\lambda_{12}\mathbf{r}_{12}(t) - (\delta t^2/m_2)\lambda_{23}\mathbf{r}_{23}(t) \quad (7.69)$$

$$\mathbf{r}_{23}(t+\delta t) = \mathbf{r}'_{23}(t+\delta t) + \delta t^2(1/m_2 + 1/m_3)\lambda_{23}\mathbf{r}_{23}(t) - (\delta t^2/m_2)\lambda_{12}\mathbf{r}_{12}(t) \quad (7.70)$$

These expressions could be solved algebraically but even in this simple case the algebra becomes rather complicated. A solution can be obtained by ignoring the terms that are quadratic in λ as this produces equations which are linear in the Lagrange multipliers λ. When there are many constraints, the problem is equivalent to inverting a $k \times k$ matrix, even when the quadratic terms are ignored. The SHAKE method uses an alternative approach in which each constraint is considered in turn and solved. Satisfying one constraint may cause another constraint to be violated, and so it is necessary to iterate around the constraints until they are all satisfied to within some tolerance. The tolerance should be tight enough to ensure that the fluctuations in the simulation due to the SHAKE algorithm are much smaller than the fluctuations due to other sources, such as the use of cutoffs. Another important requirement is that the constrained degrees of freedom should be only weakly coupled to the remaining degrees of freedom, so that the motion of the molecule is not affected by the application of the constraints. The sampling of unconstrained degrees of freedom should also be unaffected. For example, if the bond lengths and angles are constrained in butane then the only degree of freedom remaining is the torsion angle. It is important that this torsion is able to explore its entire range of values in a way that is not biased because of the SHAKE procedure.

Our discussion so far has considered the use of SHAKE with the Verlet algorithm. Versions have also been derived for other integration schemes, such as the leap-frog algorithm, the predictor–corrector methods and the velocity Verlet algorithm. In the case of the velocity Verlet algorithm, the method has been named RATTLE [Anderson 1983].

When velocities appear in the integration algorithm these must be corrected as well as the positions.

Angle constraints can be easily accommodated in the SHAKE scheme by recognising that an angle constraint simply corresponds to an additional distance constraint. The angle in a triatomic molecule could thus be maintained at the desired value by requiring the distance between the two end atoms to adopt the appropriate value. This is how some small molecules (e.g. water) are maintained in a rigid geometry. For example, the simple point-charge (SPC) model of water uses three distance constraints. However, it is generally accepted that to constrain the bond angles in simulations of conformationally flexible molecules can have a deleterious effect on the efficiency with which the system explores configurational space. This is because many conformational transitions involve some opening or closing of angles as well as rotations about bonds. The most common use of SHAKE is for constraining bonds involving hydrogen atoms due to their much higher vibrational frequencies. This can enable the time step in a molecular dynamics simulation to be increased (e.g. from 1 fs to 2 fs).

The SHAKE method has been extended by Tobias and Brooks [Tobias and Brooks 1988] to enable constraints to be applied to arbitrary internal coordinates. This enables the torsion angle of a rotatable bond to be constrained to a particular value during a molecular dynamics simulation, which is particularly useful when used in conjunction with methods for calculating free energies (see Section 11.7)

7.6 Time-dependent Properties

Molecular dynamics generates configurations of the system that are connected in time and so an MD simulation can be used to calculate time-dependent properties. This is a major advantage of molecular dynamics over the Monte Carlo method. Time-dependent properties are often calculated as *time correlation coefficients.*

7.6.1 Correlation Functions

Suppose we have two sets of data values, x and y, and we wish to determine what correlation (if any) exists between them. For example, imagine that we are performing a simulation of a fluid in a capillary, and that we wish to determine the correlation between the absolute velocity of an atom and its distance from the wall of the tube. One way to do this would be to plot the sets of values as a graph. A correlation function (also known as a correlation coefficient) provides a numerical value that encapsulates the data and quantifies the strength of the correlation. A series of simulations with different capillary diameters could then be compared by examining the correlation coefficients. A variety of correlation functions can be defined, a commonly used one being:

$$C_{xy} = \frac{1}{M}\sum_{i=1}^{M} x_i y_i \equiv \langle x_i y_i \rangle \tag{7.71}$$

We have assumed that there are M values of x_i and y_i in the data sets This correlation function can be normalised to a value between -1 and $+1$ by dividing by the root-mean-square values of x and y:

$$c_{xy} = \frac{\frac{1}{M}\sum_{i=1}^{M} x_i y_i}{\sqrt{\left(\frac{1}{M}\sum_{i=1}^{M} x_i^2\right)\left(\frac{1}{M}\sum_{i=1}^{M} y_i^2\right)}} = \frac{\langle x_i y_i \rangle}{\sqrt{\langle x_i^2 \rangle \langle y_i^2 \rangle}} \tag{7 72}$$

A value of 0 indicates no correlation and an absolute value of 1 indicates a high degree of correlation (which may be positive or negative). We will use a lowercase c to indicate a normalised correlation coefficient.

Sometimes the quantities x and y will fluctuate about non-zero mean values $\langle x \rangle$ and $\langle y \rangle$. Under such circumstances it is typical to consider just the fluctuating part and to define the correlation function as:

$$c_{xy} = \frac{\frac{1}{M}\sum_{i=1}^{M} (x_i - \langle x \rangle)(y_i - \langle y \rangle)}{\sqrt{\left(\frac{1}{M}\sum_{i=1}^{M} (x_i - \langle x \rangle)^2\right)\left(\frac{1}{M}\sum_{i=1}^{M} (y_i - \langle y \rangle)^2\right)}} = \frac{\langle (x_i - \langle x \rangle)(y_i - \langle y \rangle) \rangle}{\sqrt{\langle (x_i - \langle x \rangle)^2 \rangle \langle (y_i - \langle y \rangle)^2 \rangle}} \tag{7.73}$$

c_{xy} can also be written in the following useful way:

$$c_{xy} = \frac{\sum_{i=1}^{M} x_i y_i - \frac{1}{M}\left(\sum_{i=1}^{M} x_i\right)\left(\sum_{i=1}^{M} y_i\right)}{\sqrt{\left[\sum_{i=1}^{M} x_i^2 - \frac{1}{M}\left(\sum_{i=1}^{M} x_i\right)^2\right]\left[\sum_{i=1}^{M} y_i^2 - \frac{1}{M}\left(\sum_{i=1}^{M} y_i\right)^2\right]}} \tag{7.74}$$

Equation (7.74) does not require the mean values $\langle x \rangle$ and $\langle y \rangle$ to be determined before the correlation coefficient can be calculated and so values can be accumulated as the simulation proceeds.

A molecular dynamics simulation provides data values at specific times. This enables the value of some property at some instant to be correlated with the value of the same or another property at a later time t. The resulting values are known as *time correlation coefficients*. The correlation function is then written:

$$C_{xy}(t) = \langle x(t) y(0) \rangle \tag{7 75}$$

The following two results are useful:

$$\lim t \to 0 \quad C_{xy}(0) = \langle xy \rangle \tag{7 76}$$

$$\lim t \to \infty \quad C_{xy}(t) = \langle x \rangle \langle y \rangle \tag{7 77}$$

If the quantities x and y are different, then the correlation function is sometimes referred to as a *cross-correlation function* When x and y are the same then the function is usually called an *autocorrelation function*. An autocorrelation function indicates the extent to which the system retains a 'memory' of its previous values (or, conversely, how long it takes the system to 'lose' its memory). A simple example is the velocity autocorrelation coefficient whose value indicates how closely the velocity at a time t is correlated with the velocity at time 0. Some correlation functions can be averaged over all the particles in the system (as can the velocity autocorrelation function) whereas other functions are a property of the entire system (e.g. the dipole moment of the sample). The value of the velocity autocorrelation coefficient can be calculated by averaging over the N atoms in the simulation:

$$C_{vv}(t) = \frac{1}{N}\sum_{i=1}^{N} \mathbf{v}_i(t) \cdot \mathbf{v}_i(0) \tag{7.78}$$

To normalise the function, we divide by $\langle \mathbf{v}_i(0) \cdot \mathbf{v}_i(0) \rangle$:

$$c_{vv}(t) = \frac{1}{N}\sum_{i=1}^{N} \frac{\langle \mathbf{v}_i(t) \cdot \mathbf{v}_i(0) \rangle}{\langle \mathbf{v}_i(0) \cdot \mathbf{v}_i(0) \rangle} \tag{7.79}$$

In general, an autocorrelation function such as the velocity autocorrelation coefficient has an initial value of 1 and at long times has the value 0. The time taken to lose the correlation is often called the *correlation time*, or the *relaxation time*. If the duration of the simulation is significantly longer than the relaxation time (as it should be), then many sets of data can be extracted from the simulation to calculate the correlation function and to reduce the uncertainty in the calculation. If P steps of molecular dynamics are required for complete relaxation, and the simulation has been run for a total of Q steps, then $(Q - P)$ different sets of values could be used to calculate a value for the correlation function. The first set would run from step 1 to step N; the second from step 2 to step $N + 1$, and so on (Figure 7.9). In fact, as we saw in Section 6.9 the high degree of correlation between successive time steps means that it is common to use time origins that are separated by several time steps, as shown in Figure 7.9. If we use M time origins (t_j) then the velocity autocorrelation function is given by:

$$C_{vv}(t) = \frac{1}{MN}\sum_{j=1}^{M}\sum_{i=1}^{N} \mathbf{v}_i(t_j) \cdot \mathbf{v}_i(t_j + t) \tag{7 80}$$

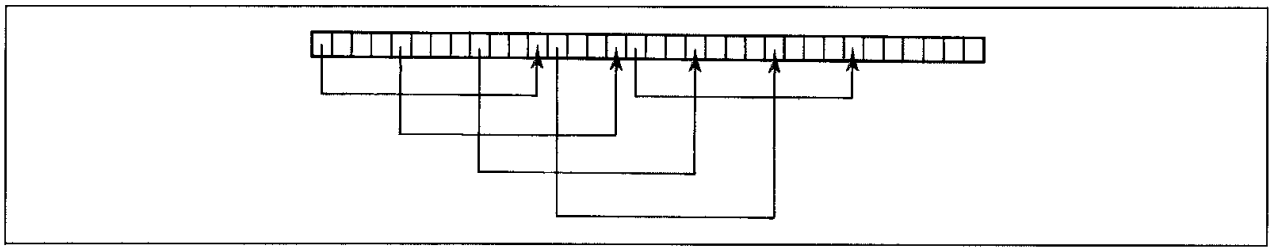

Fig 7 9· The use of different time origins improves the accuracy with which time correlation functions can be calculated.

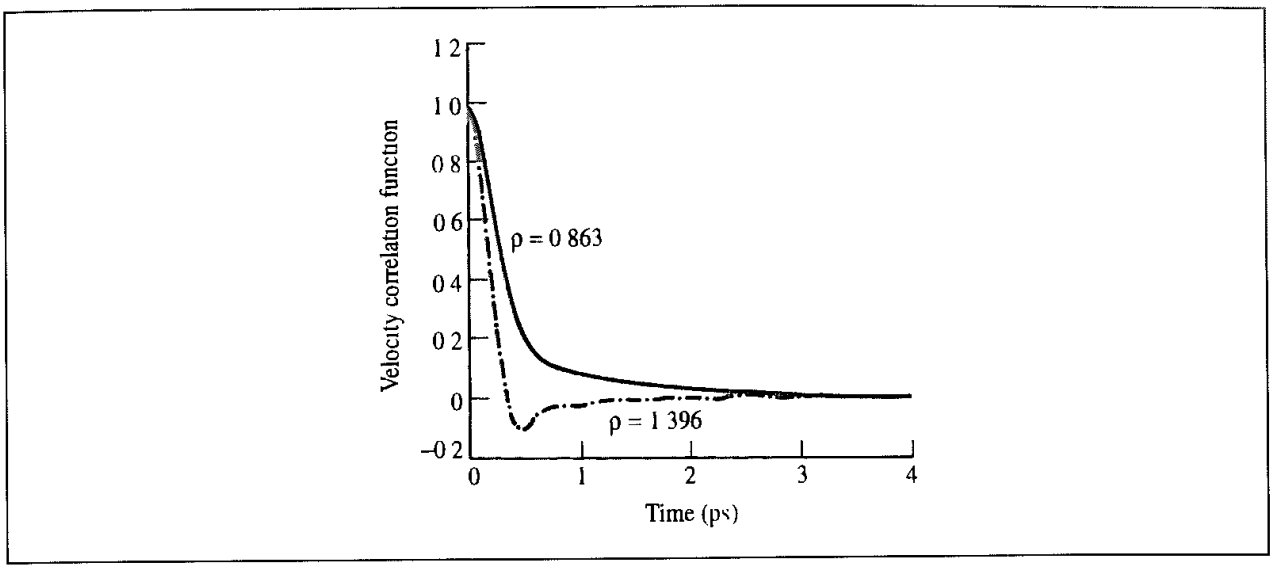

Fig 7.10· Velocity autocorrelation functions for liquid argon at densities of 1.396 g cm^{-3} and 0 863 g cm^{-3}

Quantities with small relaxation times can thus be determined with greater statistical precision, as it will be possible to include a greater number of data sets from a given simulation. Moreover, no quantity with a relaxation time greater than the length of the simulation can be determined accurately.

The velocity autocorrelation functions obtained from molecular dynamics simulations of argon at two different densities are shown in Figure 7.10. The correlation coefficient has an initial value of 1 and is then quadratic at short times, a result that is predicted theoretically. The subsequent behaviour depends upon the density of the fluid. For a low-density fluid, the velocity autocorrelation coefficient gradually decreases to zero. At high densities $c_{vv}(t)$ crosses the axis and then becomes negative. A negative correlation coefficient simply means that the particle is now moving in the direction opposite to that at $t = 0$. This result has been interpreted in terms of a 'cage' structure of the liquid; the atom hits the side of the cage formed by its nearest neighbours and rebounds, reversing the direction of its motion. At both low density and high density the decay towards zero is significantly slower than the exponential decay predicted by kinetic theory. In fact, the decay varies as $t^{-3/2}$. This was one of the most interesting results obtained from the early molecular dynamics simulations and can be observed even with a hard-sphere model [Alder and Wainwright 1970]. The phenomenon is ascribed to the formation of a 'hydrodynamic vortex'. As the atom moves through the fluid it pushes other atoms out of the way. These atoms circle around and eventually give it a final 'push' so resulting in a less rapid decrease to zero (Figure 7.11).

The slow decay of the velocity autocorrelation function can present practical problems when deriving other properties, such as the transport coefficients, that require the correlation function to be integrated between $t = 0$ and $t = \infty$. The so-called 'long time-tail' of the autocorrelation function makes a significant contribution to the integral, but unfortunately the statistical uncertainty with which this part of the function can be calculated is greater as fewer segments of the appropriate length can be extracted from the simulation.

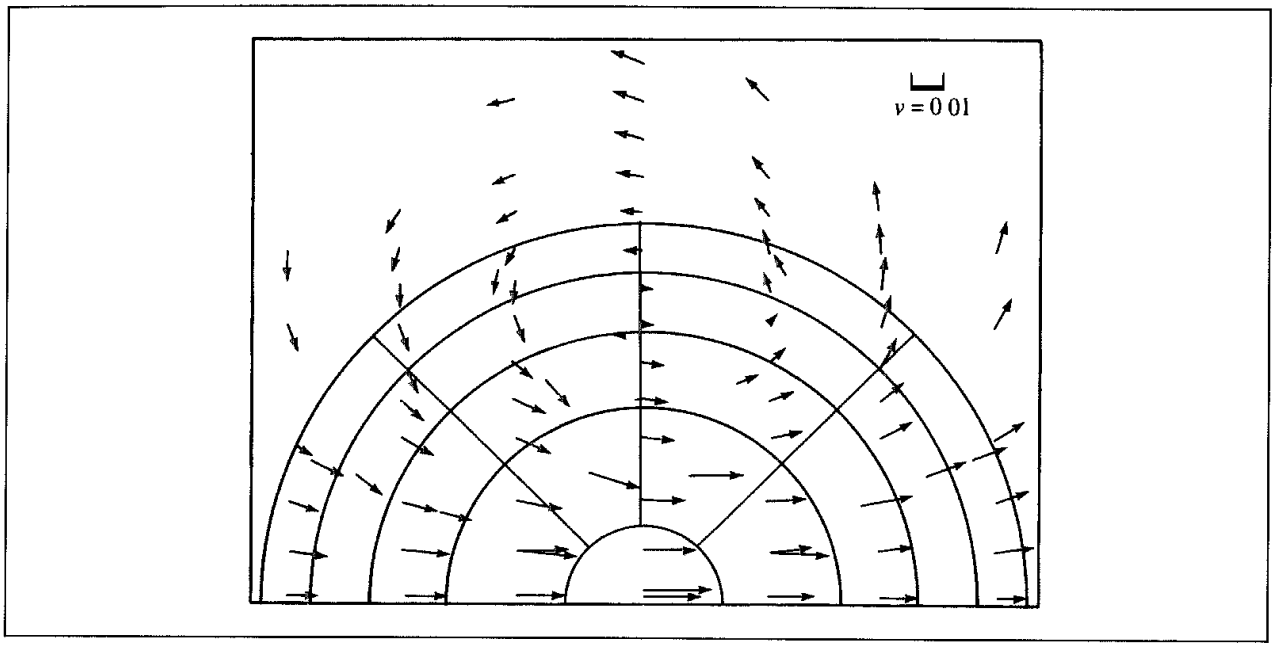

Fig 7 11 The slow decay of the velocity autocorrelation function towards zero can be explained in terms of the formation of a hydrodynamic vortex (Figure adapted from Alder B J and T E Wainwright 1970. Decay of the Velocity Autocorrelation Function Physical Review *A 1:18–21)*

The velocity autocorrelation function is an example of a single-particle correlation function, in which the average is calculated not only over time origins but also over all the atoms. Some properties are calculated for the entire system. One such property is the net dipole moment of the system, which is the vector sum of all the individual dipoles of the molecules in the system (clearly the dipole moment of the system can be non-zero only if each individual molecule has a dipole). The magnitude and orientation of the net dipole will change with time and is given by:

$$\boldsymbol{\mu}_{\text{tot}}(t) = \sum_{i=1}^{N} \boldsymbol{\mu}_i(t) \tag{7.81}$$

$\boldsymbol{\mu}_i(t)$ is the dipole moment of molecule i at time t. The total dipolar correlation function is given by:

$$c_{\text{dipole}}(t) = \frac{\langle \boldsymbol{\mu}_{\text{tot}}(t) \cdot \boldsymbol{\mu}_{\text{tot}}(0) \rangle}{\langle \boldsymbol{\mu}_{\text{tot}}(0) \cdot \boldsymbol{\mu}_{\text{tot}}(0) \rangle} \tag{7.82}$$

The dipole correlation time of the system is a valuable quantity to calculate as it is related to the sample's absorption spectrum. Liquids usually absorb in the infrared region of the electromagnetic spectrum, a typical spectrum being shown in Figure 7.12. As can be seen, the spectrum is very broad with none of the sharp peaks characteristic of a well-resolved spectrum for a species in the gas phase. This is because the overall dipole of a liquid does not change at a constant rate but, rather, there is a distribution of frequencies. The intensity of absorption at any frequency depends upon the relative contribution of that frequency to the overall distribution. If, on average, the overall dipole changes very rapidly (i.e. the relaxation time is short) then the maximum in the absorption spectrum will occur at a

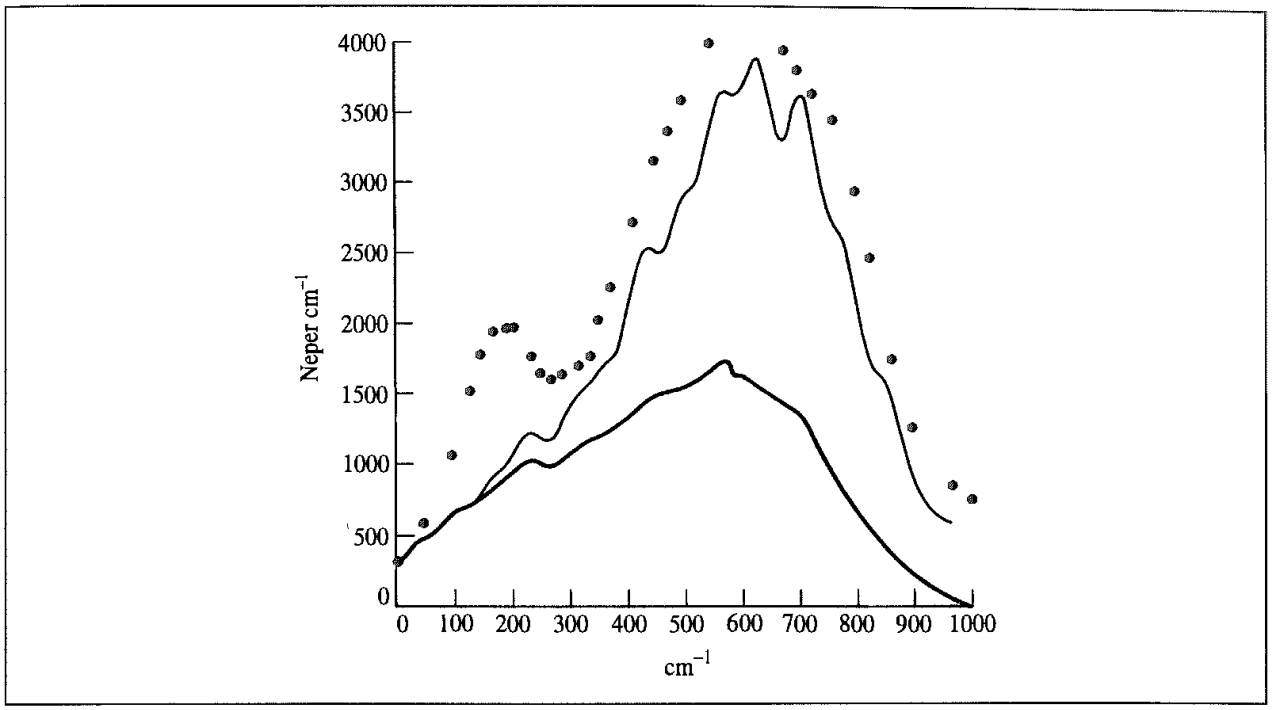

Fig 7 12. Experimental and calculated infrared spectra for liquid water The black dots are the experimental values The thick curve is the classical profile produced by the molecular dynamics simulation. The thin curve is obtained by applying quantum corrections. (Figure redrawn from Guillot B 1991. A Molecular Dynamics Study of the Infrared Spectrum of Water. Journal of Chemical Physics *95 1543–1551.)*

higher frequency than if the relaxation time is long. To predict the spectrum from the correlation function it is therefore necessary to extract the relative contribution of each dipole fluctuation. This is done using Fourier analysis techniques, in which the correlation function is transformed from the time domain into the frequency domain (an introduction to Fourier analysis is provided in Section 1.10.8). The Fourier analysis picks out the intensity of dipole fluctuation at each frequency using the following relationship:

$$\hat{c}_{\text{dipole}}(\nu) = \int_{-\infty}^{\infty} c_{\text{dipole}}(t) \exp(-i2\pi\nu t)\, dt \tag{7 83}$$

Having calculated the Fourier transform it is then possible to plot the simulated spectrum and compare it to that obtained by experiment, as in Figure 7.12.

7.6.2 Orientational Correlation Functions

Other orientational correlation coefficients can be calculated in the same way as the correlation coefficients that we have discussed already. Thus, the reorientational correlation coefficient of a single rigid molecule indicates the degree to which the orientation of a molecule at a time t is related to its orientation at time 0. The angular velocity autocorrelation function is the rotational equivalent of the velocity correlation function:

$$c_{\omega\omega}(t) = \frac{\langle \omega_i(t) \cdot \omega_i(0) \rangle}{\langle \omega_i(0) \cdot \omega_i(0) \rangle} \tag{7 84}$$

In a liquid, the rotation of a molecule is influenced by neighbouring molecules and over time the correlation will decay to zero. The information embodied in the orientational correlation functions can be compared to a variety of spectroscopic experiments, including infrared, Raman and NMR spectra. For non-spherical molecules it can be useful to derive separate auto-correlation functions for the angular velocity along each of the principal axes of rotation. For example, for a spherical molecule such as CBr_4 neighbouring molecules have a relatively small influence on the loss in correlation in the angular velocity. By contrast, a linear molecule such as CS_2 experiences significant torques as it rotates. This has the effect of damping the rotational motion more severely than for the spherical case, and indeed the correlation function can change sign, indicating that the molecule is now rotating in the opposite direction. For some molecules such as water the presence of specific interactions between molecules (for example, due to hydrogen bonding) can give rise to very rapid damping and several minima in $c_{\omega\omega}(t)$.

7.6.3 Transport Properties

Transport refers to a phenomenon that gives rise to a flow of material from one region to another. For example, if a solution is prepared with a non-equilibrium solute distribution, then the solute diffuses until the concentration is equal throughout. If a thermal gradient is created, energy flows until the temperature is equalised. A momentum gradient gives rise to viscosity. The very existence of transport implies that the system is not in equilibrium. Techniques have been developed to perform non-equilibrium molecular dynamics simulations from which transport properties can be calculated, but we will not consider these here. We are thus faced with the problem of calculating non-equilibrium properties from equilibrium simulations This may seem an impossible task but can be achieved by considering the microscopic local fluctuations that occur even in systems at equilibrium. We should be aware, however, that non-equilibrium molecular dynamics simulations can be a more efficient way to calculate transport properties and other quantities [Allen and Tildesley 1987].

To a first approximation the rate at which transport of the relevant quantity occurs (called the *flux*) is proportional to the gradient of the property with the constant of proportionality being the relevant transport property coefficient. For example, the flux of matter J_z (i.e. the amount passing through unit area in unit time) equals the diffusion coefficient (D) multiplied by the concentration gradient; this is Fick's first law of diffusion:

$$J_z = -D(d\mathcal{N}/dz) \tag{7.85}$$

$\mathcal{N}$ is the number density (the number of atoms per unit volume). Equation (7.85) refers to diffusion in the z direction. The minus sign indicates that flux increases in the direction of negative concentration gradient. The time dependence of diffusive behaviour (which applies if a distribution is established at some time and is then allowed to evolve) is governed by Fick's second law, which gives the rate of change of concentration with time:

$$\frac{\partial \mathcal{N}(z,t)}{\partial t} = D\frac{\partial^2 \mathcal{N}(z,t)}{\partial z^2} \tag{7.86}$$

To solve Fick's second law equation it is necessary to impose two boundary conditions for the spatial dependence and one boundary condition for the temporal dependence (the

equation is second order in space and first order in time). For example, we might require that at time zero all N_0 particles have $z = 0$. The solution to the equation is then.

$$\mathcal{N}(z,t) = \frac{N_0}{A\sqrt{\pi Dt}} \exp\left[-\frac{z^2}{4Dt}\right] \tag{7.87}$$

where A is the cross-sectional area of the sample. Equation (7.87) is a Gaussian function which initially has a sharp peak at $z = 0$ but which gradually becomes more spread out as time progresses. When the material being simulated is a pure liquid then the coefficient D is often referred to as a *self-diffusion coefficient*. The diffusion coefficient is related to the mean square distance, $\langle|[\mathbf{r}(t) - \mathbf{r}(0)]^2|\rangle$, which Einstein showed was equal to $2Dt$. In three dimensions, the mean square displacement is given by:

$$3D = \lim_{t\to\infty} \frac{\langle|\mathbf{r}(t) - \mathbf{r}(0)|^2\rangle}{2t} \tag{7.88}$$

As indicated, the relationship strictly holds only in the limit as $t \to \infty$.

The Einstein relationship can thus be used to calculate the diffusion coefficient from an equilibrium simulation, by plotting the mean square displacement as a function of time and then attempting to obtain the limiting behaviour as $t \to \infty$ (Fick's law is inapplicable at short times). The quantity $|\mathbf{r}(t) - \mathbf{r}(0)|$ can be averaged over the particles in the system to reduce the statistical error. It is also usual practice to average over time origins where possible. When using this method for calculating the diffusion coefficient the mean-squared distances should not be limited by the edges of the periodic box. In other words, we require a set of positions that have not been translated back into the central simulation cell. This can be achieved either by storing a set of 'uncorrected' positions or indeed by not correcting any of the positions during the simulation and simply generating the appropriate minimum image positions as required for the calculation of the energy or forces.

Einstein relationships hold for other transport properties, e.g. the shear viscosity, the bulk viscosity and the thermal conductivity. For example, the shear viscosity η is given by:

$$\eta_{xy} = \frac{1}{Vk_BT} \lim_{t\to\infty} \frac{\langle(\sum_{i=1}^{N} m\dot{x}_i(t)y_i(t) - \sum_{i=1}^{N} m\dot{y}_i(t)x_i(t))^2\rangle}{2t} \tag{7.89}$$

The shear viscosity is a tensor quantity, with components η_{xy}, η_{xz}, η_{yx}, η_{yz}, η_{zx}, η_{zy}. It is a property of the whole sample rather than of individual atoms and so cannot be calculated with the same accuracy as the self-diffusion coefficient. For a homogeneous fluid the components of the shear viscosity should all be equal and so the statistical error can be reduced by averaging over the six components. An estimate of the precision of the calculation can then be determined by evaluating the standard deviation of these components from the average. Unfortunately, Equation (7.89) cannot be directly used in periodic systems, even if the positions have been unfolded, because the 'unfolded' distance between two particles may not correspond to the distance of the minimum image that is used to calculate the force. For this reason alternative approaches are required.

One alternative approach to the calculation of the diffusion and other transport coefficients is via an appropriate autocorrelation function. For example, the diffusion coefficient

depends upon the way in which the atomic position $\mathbf{r}(t)$ changes with time. At a time t the difference between $\mathbf{r}(t)$ and $\mathbf{r}(0)$ is given by:

$$|\mathbf{r}(t) - \mathbf{r}(0)| = \int_0^t \mathbf{v}(t')\, dt' \tag{7.90}$$

If both sides of Equation (7.90) are now squared then we obtain the following for the mean-square value:

$$\langle |\mathbf{r}(t) - \mathbf{r}(0)| \rangle^2 = \int_0^t dt' \int_0^t dt'' \langle \mathbf{v}(t') \cdot \mathbf{v}(t'') \rangle \tag{7.91}$$

The crucial feature to recognise is that the relevant correlation functions are unaffected by changing the origin, which means that the following holds:

$$\langle \mathbf{v}(t') \cdot \mathbf{v}(t'') \rangle = \langle \mathbf{v}(t'' - t') \cdot \mathbf{v}(0) \rangle \tag{7.92}$$

Integration of the double integral, Equation (7.91) leads to the *Green–Kubo* formula:

$$\frac{\langle |\mathbf{r}(t) - \mathbf{r}(0)|^2 \rangle}{2t} = \int_0^t \langle \mathbf{v}(\tau) \cdot \mathbf{v}(0) \rangle \left(1 - \frac{\tau}{t}\right) d\tau \tag{7.93}$$

In the limit:

$$\int_0^\infty \langle \mathbf{v}(\tau) \cdot \mathbf{v}(0) \rangle\, d\tau = \lim_{t \to \infty} \frac{\langle |\mathbf{r}(t) - \mathbf{r}(0)|^2 \rangle}{2t} = 3D \tag{7.94}$$

We can now see why long time-tails in the autocorrelation functions are so important The area under the curve during the slow decay towards zero may be a significant part of the integral in the Green–Kubo formula. In practice, these integrals are determined numerically. The long time-tail may be dealt with by fitting a function to the curve and then attempting to integrate to infinity.

7.7 Molecular Dynamics at Constant Temperature and Pressure

Molecular dynamics is traditionally performed in the constant *NVE* (or *NVEP*) ensemble. Although thermodynamic results can be transformed between ensembles, this is strictly only possible in the limit of infinite system size ('the thermodynamic limit'). It may therefore be desired to perform the simulation in a different ensemble. The two most common alternative ensembles are the constant *NVT* and the constant *NPT* ensembles. In this section we will therefore consider how molecular dynamics simulations can be performed under conditions of constant temperature and/or constant pressure.

7.7.1 Constant Temperature Dynamics

There are several reasons why we might want to maintain or otherwise control the temperature during a molecular dynamics simulation. Even in a constant *NVE* simulation it is common practice to adjust the temperature to the desired value during the equilibration phase. A constant temperature simulation may be required if we wish to determine how

the behaviour of the system changes with temperature, such as the unfolding of a protein or glass formation. Simulated annealing algorithms require the temperature of the system to be reduced gradually while the system explores its degrees of freedom. Simulated annealing is used in searching conformational space and in the elucidation of macromolecular structure from NMR and X-ray data and is discussed in Section 9.9.2.

The temperature of the system is related to the time average of the kinetic energy, which for an unconstrained system is given by:

$$\langle \mathcal{K} \rangle_{NVT} = \tfrac{3}{2} N k_B T \tag{7.95}$$

An obvious way to alter the temperature of the system is thus to scale the velocities [Woodcock 1971]. If the temperature at time t is $T(t)$ and the velocities are multiplied by a factor λ, then the associated temperature change can be calculated as follows:

$$\Delta T = \frac{1}{2}\sum_{i=1}^{N} \frac{2}{3} \frac{m_i(\lambda v_i)^2}{N k_B} - \frac{1}{2}\sum_{i=1}^{N} \frac{2}{3} \frac{m_i v_i^2}{N k_B} \tag{7.96}$$

$$\Delta T = (\lambda^2 - 1)T(t) \tag{7.97}$$

$$\lambda = \sqrt{T_{new}/T(t)} \tag{7.98}$$

The simplest way to control the temperature is thus to multiply the velocities at each time step by the factor $\lambda = \sqrt{T_{req}/T_{curr}}$, where T_{curr} is the current temperature as calculated from the kinetic energy and T_{req} is the desired temperature.

An alternative way to maintain the temperature is to couple the system to an external heat bath that is fixed at the desired temperature [Berendsen *et al.* 1984]. The bath acts as a source of thermal energy, supplying or removing heat from the system as appropriate. The velocities are scaled at each step, such that the rate of change of temperature is proportional to the difference in temperature between the bath and the system:

$$\frac{dT(t)}{dt} = \frac{1}{\tau}(T_{bath} - T(t)) \tag{7.99}$$

τ is a coupling parameter whose magnitude determines how tightly the bath and the system are coupled together. This method gives an exponential decay of the system towards the desired temperature. The change in temperature between successive time steps is:

$$\Delta T = \frac{\delta t}{\tau}(T_{bath} - T(t)) \tag{7.100}$$

The scaling factor for the velocities is thus:

$$\lambda^2 = 1 + \frac{\delta t}{\tau}\left(\frac{T_{bath}}{T(t)} - 1\right) \tag{7.101}$$

If τ is large, then the coupling will be weak. If τ is small, the coupling will be strong and when the coupling parameter equals the time step ($\tau = \delta t$) then the algorithm is equivalent to the simple velocity scaling method. A coupling constant of approximately 0.4 ps has been suggested as an appropriate value to use when the time step is 1 fs, giving $\delta t/\tau \approx 0\ 0025$. The advantage of this approach is that it does permit the system to fluctuate about the desired temperature.

These two relatively simple temperature scaling methods do not generate rigorous canonical averages. Velocity scaling artificially prolongs any temperature differences among the components of the system, which can lead to the phenomenon of 'hot solvent, cold solute', in which the 'temperature' of the solute is lower than that of the solvent, even though the overall temperature of the system is at the desired value. One 'solution' to this problem is to apply temperature coupling separately to the solute and to the solvent but the problem of unequal distribution of energy between the various components (and between the various modes of motion) may still remain. Two methods that do generate rigorous canonical ensembles if properly implemented are the *stochastic collisions* method and the *extended system* method.

In the stochastic collisions method a particle is randomly chosen at intervals and its velocity is reassigned by random selection from the Maxwell-Boltzmann distribution [Anderson 1980]. This is equivalent to the system being in contact with a heat bath that randomly emits 'thermal particles' which collide with the atoms in the system. Between each collision the system is simulated at constant energy and so the overall effect is equivalent to a series of microcanonical simulations, each performed at a slightly different energy. The distribution of the energies of these 'mini microcanonical' simulations will be a Gaussian function. The stochastic collisions method does not, of course, generate a smooth trajectory, which may be a drawback. By calculating the energy change due to a collision, Anderson showed that the mean rate (ν) at which each particle should suffer a stochastic collision is given by:

$$\nu = \frac{2a\kappa}{3k_B \mathcal{N}^{1/3} N^{2/3}} \tag{7.102}$$

a is a dimensionless constant, κ is the thermal conductivity and $\mathcal{N}$ is the number density of the particles. If the thermal conductivity is not known then a suitable value of ν can be obtained from the intermolecular collision frequency ν_c:

$$\nu = \nu_c / N^{2/3} \tag{7.103}$$

If the collision rate is too low then the system does not sample from a canonical distribution of energies. If it is too high then the temperature control algorithm dominates and the system does not show the expected fluctuations in kinetic energy. The velocity of more than one particle can be changed in the stochastic collision method; in the limit the velocities of all the particles are changed simultaneously, though it is preferable to do this at quite long intervals. A distinction can thus be made between 'minor' collisions, in which only one (or a few) particles are affected, and 'major' (or 'massive') collisions, where the velocities of all particles are changed. It is also possible to use a combined approach, with minor collisions occurring relatively frequently and major collisions at longer intervals.

Extended system methods, originally introduced for performing constant temperature molecular dynamics by Nosé [Nosé 1984] and subsequently developed by Hoover [Hoover 1985], consider the thermal reservoir to be an integral part of the system. The reservoir is represented by an additional degree of freedom, labelled s. The reservoir has potential energy $(f+1)k_B T \ln s$, where f is the number of degrees of freedom in the physical system and T is the desired temperature. The reservoir also has kinetic energy $(Q/2)(ds/dt)^2$. Q is a parameter with the dimensions of energy $\times$ (time)^2 and can be considered the

(fictitious) mass of the extra degree of freedom. The magnitude of Q determines the coupling between the reservoir and the real system and so influences the temperature fluctuations.

Each state of the extended system that is generated by the molecular dynamics simulation corresponds to a unique state of the real system. There is not, however, a direct correspondence between the velocities and the time in the real and the extended systems The velocities of the atoms in the real system are given by:

$$\mathbf{v}_i = s\frac{d\mathbf{r}_i}{dt} \tag{7 104}$$

$\mathbf{r}_i$ is the position of particle i in the simulation and $\mathbf{v}_i$ is considered to be the real velocity of the particle. The time step $\delta t'$ is related to the time step in 'real time' δt by

$$\delta t = s\delta t' \tag{7.105}$$

The value of the additional degree of freedom s can change and so the time step in real time can fluctuate. Thus regular time intervals in the extended system correspond to a trajectory of the real system which is unevenly space in time.

The parameter Q controls the energy flow between the system and the reservoir. If Q is large then the energy flow is slow; in the limit of infinite Q, conventional molecular dynamics is regained and there is no energy exchange between the reservoir and the real system. However, if Q is too small then the energy oscillates, resulting in equilibration problems. Nosé has suggested that Q should be proportional to fk_BT; the constant of proportionality can then be obtained by performing a series of trial simulations for a test system and observing how well the system maintains the desired temperature.

7.7.2 Constant Pressure Dynamics

Just as one may wish to specify the temperature in a molecular dynamics simulation, so it may be desired to maintain the system at a constant pressure. This enables the behaviour of the system to be explored as a function of the pressure, enabling one to study phenomena such as the onset of pressure-induced phase transitions. Many experimental measurements are made under conditions of constant temperature and pressure, and so simulations in the isothermal–isobaric ensemble are most directly relevant to experimental data. Certain structural rearrangements may be achieved more easily in an isobaric simulation than in a simulation at constant volume. Constant pressure conditions may also be important when the number of particles in the system changes (as in some of the 'test particle' methods for calculating free energies and chemical potentials; see Section 8.9).

The pressure often fluctuates much more than quantities such as the total energy in a constant *NVE* molecular dynamics simulation. This is as expected because the pressure is related to the virial, which is obtained as the product of the positions and the derivative of the potential energy function. This product, $r_{ij}d\nu(r_{ij})/dr_{ij}$, changes more quickly with r than does the internal energy, hence the greater fluctuation in the pressure.

A macroscopic system maintains constant pressure by changing its volume. A simulation in the isothermal–isobaric ensemble also maintains constant pressure by changing the volume

of the simulation cell. The amount of volume fluctuation is related to the isothermal compressibility, κ.

$$\kappa = -\frac{1}{V}\left(\frac{\partial V}{\partial P}\right)_T \tag{7.106}$$

An easily compressible substance has a larger value of κ, so larger volume fluctuations occur at a given pressure than in a more incompressible substance. Conversely, in a constant volume simulation a less compressible substance shows larger fluctuations in the pressure. The isothermal compressibility is the pressure analogue of the heat capacity, which is related to the energy fluctuations.

A volume change in an isobaric simulation can be achieved by changing the volume in all directions, or in just one direction. It is instructive to consider the range of volume changes that one might expect to observe in a constant pressure simulation of a 'typical' system. The isothermal compressibility is related to the mean square volume displacement by:

$$\kappa = \frac{1}{k_B T}\frac{\langle V^2\rangle - \langle V\rangle^2}{\langle V^2\rangle} \tag{7.107}$$

The isothermal compressibility of an ideal gas is approximately $1\,\text{atm}^{-1}$. So for a simulation in a box of side 20 Å (volume 8000 Å^3) at 300 K, the root mean square change in the volume is approximately 18 100 Å^3. This is larger than the initial size of the box! For a relatively incompressible substance such as water ($\kappa = 44\,75 \times 10^{-6}\,\text{atm}^{-1}$) the fluctuation is 121 Å^3, which corresponds to the box only changing by about 0 1 Å in each direction. These values have clear implications for the appropriate size of the simulation system.

Many of the methods used for pressure control are analogous to those used for temperature control Thus, the pressure can be maintained at a constant value by simply scaling the volume. An alternative is to couple the system to a 'pressure bath', analogous to a temperature bath [Berendsen *et al.* 1984]. The rate of change of pressure is given by:

$$\frac{dP(t)}{dt} = \frac{1}{\tau_p}(P_{\text{bath}} - P(t)) \tag{7.108}$$

τ_p is the coupling constant, P_{bath} is the pressure of the 'bath', and $P(t)$ is the actual pressure at time t. The volume of the simulation box is scaled by a factor λ, which is equivalent to scaling the atomic coordinates by a factor $\lambda^{1/3}$. Thus:

$$\lambda = 1 - \kappa\frac{\delta t}{\tau_p}(P - P_{\text{bath}}) \tag{7.109}$$

The new positions are given by:

$$\mathbf{r}_i' = \lambda^{1/3}\mathbf{r}_i \tag{7 110}$$

The constant κ can be combined with the relaxation constant τ_p as a single constant. This expression can be applied isotropically (i.e. such that the scaling factor is equal for all three directions) or anisotropically (where the scaling factor is calculated independently for each of the three axes). In general, it is best to use the anisotropic approach as this enables the box dimensions to change independently. Unfortunately, it has not been possible to determine from which ensemble this method samples.

In the extended pressure-coupling system methods, first introduced by Anderson [Anderson 1980], an extra degree of freedom, corresponding to the volume of the box, is added to the system. The kinetic energy associated with this degree of freedom (which can be considered to be equivalent to a piston acting on the system) is $\frac{1}{2}Q(dV/dt)^2$, where Q is the 'mass' of the piston. The piston also has potential energy PV, where P is the desired pressure and V is the volume of the system. A piston of small mass gives rise to rapid oscillations in the box, whereas a large mass has the opposite effect. An infinite mass returns normal molecular dynamics behaviour. The volume can vary during the simulation, with the average volume being determined by the balance between the internal pressure of the system and the desired external pressure. The extended-system temperature-scaling method of Nosé uses a scaled time; in the extended pressure method the coordinates of the extended system are related to the 'real' coordinates by:

$$\mathbf{r}_i' = V^{-1/3}\mathbf{r}_i \tag{7.111}$$

7.8 Incorporating Solvent Effects into Molecular Dynamics: Potentials of Mean Force and Stochastic Dynamics

In many simulations of solute–solvent systems the primary focus is the behaviour of the solute; the solvent is of relatively little interest, particularly in regions far from the solute molecule. The use of non-rectangular periodic boundary conditions, stochastic boundaries and 'solvent shells' can all help to reduce the number of solvent molecules required and enable a larger proportion of the computing time to be spent simulating the solute In this section we consider a group of techniques that incorporate the effects of solvent without requiring any explicit specific solvent molecules to be present.

One approach to this problem is to use a *potential of mean force* (PMF), which describes how the free energy changes as a particular coordinate (such as the separation of two atoms or the torsion angle of a bond) is varied. The free energy change described by the potential of mean force includes the averaged effects of the solvent.

Potentials of mean force may be determined using a molecular dynamics or Monte Carlo simulation using the techniques of umbrella sampling or free energy perturbation, which will be discussed in Chapter 11. Here we illustrate the concept using an example The energy difference between the *trans* and *gauche* conformations for an isolated molecule of 1,2-dichloroethane (i.e. in the gas phase) is approximately 1.14 kcal mol^{-1} with a population containing 77% *trans* and 23% *gauche* conformers. In liquid 1,2-dichloroethane, however, the relative population of the *gauche* conformer is significantly increased relative to the *trans* conformer by comparison with the isolated molecule, with 44% *trans* and 56% *gauche*. These experimental results were reproduced by Jorgensen (see Figure 7.13) using Monte Carlo simulations [Jorgensen *et al.* 1981]. The potential of mean force would be designed to reproduce this new population and so enable a single 1,2-dichloroethane molecule to be simulated as if it were present in the liquid.

A simulation performed using a potential of mean force enables the modulating effects of the solvent to be taken into account. The solvent also influences the dynamic behaviour of the

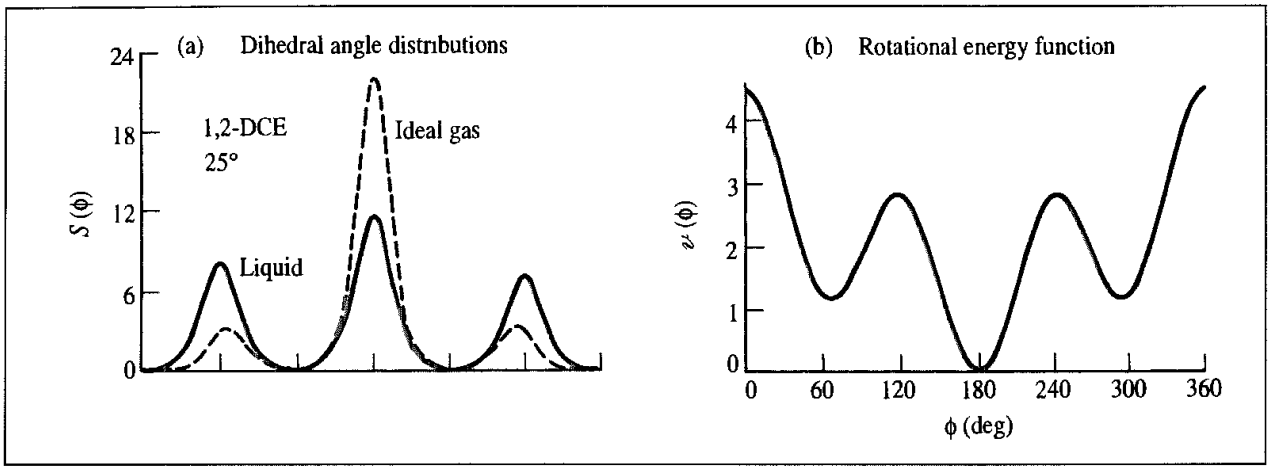

Fig 7 13· Population distribution for 1,2-dichloroethane in the gas and liquid phases (Figure redrawn from Jorgensen W L, R C Binning Jr and B Bigot 1981. Structures and Properties of Organic Liquids. n-*Butane and 1,2-Dichloroethane and Their Conformational Equilibria.* Journal of the American Chemical Society ***103*** *4393–4399)*

solute via random collisions, and by imposing a frictional drag on the motion of the solute through the solvent. The Langevin equation of motion is the starting point for the *stochastic dynamics* models, which also incorporate these two effects. In stochastic dynamics the force on a particle is considered to arise from three sources. The first component is due to interactions between the particle and other particles This force ($\mathbf{F}_i$) depends upon the position of the particle relative to the other particles and is modelled using a potential of mean force. The second force arises from the motion of the particle through the solvent and is equivalent to the frictional drag on the particle due to the solvent. This frictional force is proportional to the speed of the particle with the constant of proportionality being the friction coefficient:

$$\mathbf{F}_{\text{frictional}} = -\xi\mathbf{v} \tag{7.112}$$

where $\mathbf{v}$ is the velocity and ξ is the friction coefficient. The friction coefficient is related to the collision frequency (γ) by $\gamma = \xi/m$ (m is the mass of the particle). γ^{-1} can be considered as the time taken for the particle to lose memory of its initial velocity (the velocity relaxation time). For a spherical particle the friction coefficient is related to the diffusion constant D by:

$$\xi = k_B T/D \tag{7.113}$$

If the radius of the spherical particle is a then the frictional force is given by Stokes' law:

$$\mathbf{F}_{\text{frictional}} = 6\pi a\eta\mathbf{v} \tag{7.114}$$

where η is the viscosity of the fluid.

The third contribution to the force on the particle is due to random fluctuations caused by interactions with solvent molecules. We will write this force as $\mathbf{R}(t)$. The Langevin equation of motion for a particle i can therefore be written*:

$$m_i \frac{d^2x_i(t)}{dt^2} = \mathbf{F}_i\{x_i(t)\} - \gamma_i \frac{dx_i(t)}{dt} m_i + \mathbf{R}_i(t) \tag{7.115}$$

* γ_i in Equation (7 115) is often referred to as the friction coefficient in the literature.

A number of simulation methods based on Equation (7.115) have been described. These differ in the assumptions that are made about the nature of frictional and random forces. A common simplifying assumption is that the collision frequency γ is independent of time and position. The random force $\mathbf{R}(t)$ is often assumed to be uncorrelated with the particle velocities, positions and the forces acting on them, and to obey a Gaussian distribution with zero mean. The force $\mathbf{F}_i$ is assumed to be constant over the time step of the integration.

Three different situations can be considered, depending upon the relative magnitudes of the integration time step and the velocity relaxation time. The first category corresponds to timescales that are short relative to the velocity relaxation time ($\gamma\delta t \ll 1$). Under such circumstances the solvent does not activate or deactivate the particle to any significant extent. In the limit of zero γ (when there are no effects due to solvent) then the Langevin Equation (7.115) reduces to that obtained from Newton's laws of motion. At the other extreme the velocity relaxation time is much smaller than the time step. This corresponds to the diffusive regime, where the motion is rapidly damped by the solvent. The third situation is intermediate between these two extremes Various methods have been proposed for integrating the Langevin equation of motion in these three regions.

In the region where $\gamma\delta t \ll 1$ the following is a simple integration algorithm [van Gunsteren *et al.* 1981]:

$$x_{i+1} = x_i + v_i\delta t + \tfrac{1}{2}(\delta t)^2\{-\gamma v_i + m^{-1}(F_i + R_i)\} \tag{7.116}$$

$$v_{i+1} = v_i + (\delta t)\{-\gamma v_i + m^{-1}(F_i + R_i)\} \tag{7 117}$$

The average random force over the time step is taken from a Gaussian with a variance $2mk_BT\gamma(\delta t)^{-1}$. x_i is one of the $3N$ coordinates at time step i; F_i and R_i are the relevant components of the frictional and random forces at that time; v_i is the velocity component.

An alternative expression is based on the following finite difference approximations [Brunger *et al.* 1984]:

$$d^2x/dt^2 \approx (x_{i+1} - 2x_i + x_{i-1})/\delta t^2 \tag{7 118}$$

$$dx/dt \approx (x_{i+1} - x_{i-1})/2\delta t \tag{7 119}$$

This leads to the following expressions for the coordinates x_{i+1}:

$$x_{i+1} = x_i + (x_i - x_{i-1})\frac{1 - \frac{1}{2}\gamma\delta t}{1 + \frac{1}{2}\gamma\delta t} + \left(\frac{\delta t^2}{m}\right)\frac{F_i + R_i}{1 + \frac{1}{2}\gamma\delta t} \tag{7.120}$$

In the region where $\gamma\delta t \gg 1$ then if the interparticle force is assumed to be constant over the integration time step the following result is obtained [van Gunsteren *et al.* 1981]:

$$x_{i+1} = x_i + F_i(m\gamma)^{-1}\delta t + X_i(\delta t) \tag{7.121}$$

where X_i is a Gaussian distribution with zero mean and a variance of $2k_BT(m\gamma)^{-1} = 2D\delta t$ An extension of this treatment is to permit force F_i to vary linearly over the time step, giving:

$$x_{i+1} = x_i + \frac{\delta t}{m\gamma}(F_i + \tfrac{1}{2}\dot{F}_i\delta t) + X_i \tag{7 122}$$

$\dot{F}_i$ is the derivative of the force at the time step i and is obtained numerically·

$$\dot{F}_i = (F_i - F_{i-1})/\delta t \tag{7.123}$$

In the intermediate region, where there are no restrictions on $\gamma\delta t$, then integration of the equations of motion gives the following rather complicated result [van Gunsteren and Berendsen 1982]:

$$x_{i+1} = x_i + v_i\gamma^{-1}(1 - \exp(-\gamma\delta t)) + F_i(m\gamma)^{-1}[\delta t - \gamma^{-1}(1 - \exp(-\gamma\delta t))] + (mg)^{-1}\int_{t_i}^{t_{i+1}}[1 - \exp(-\gamma(t_{i+1} - t'))]R(t')\,dt' \tag{7.124}$$

$$v_{i+1} = v_i \exp(-\gamma\delta t) + F_i(m\gamma)^{-1}(1 - \exp(-\gamma\delta t)) + (m)^{-1}\int_{t_i}^{t_{i+1}} \exp(-\gamma(t_{i+1} - t'))R(t')\,dt' \tag{7.125}$$

The important feature of these two equations is that the new positions and the new velocities both depend upon an integral over the random force, $R(t)$ (the final terms in Equations (7.124) and (7.125). As both of these integrals depend upon $R_i(t)$ they are correlated. Specifically, they obey a *bivariate* Gaussian distribution. Such a distribution provides the probability that a particle located at x_i at time t with velocity v_i and experiencing a force F_i will be at x_{i+1} at time $t + \delta t$ with velocity v_{i+1}. In practice, this means that the distribution for the second variable depends upon the value selected for the first variable. It can be difficult to properly sample from such distributions, but van Gunsteren and Berendsen showed that the equations can be reformulated in terms of sampling from two independent Gaussian functions.

More complex stochastic dynamics treatments are possible; our treatment has only provided a rather simple treatment of solvent effects. For example, we have assumed that the frictional force at a given instant is proportional only to its velocity at the same time. A more realistic model assumes that the frictional forces are correlated; they have a 'memory' of previous values. The friction coefficient can also be made to depend on the coordinates of the other particles.

7.8.1 Practical Aspects of Stochastic Dynamics Simulations

A stochastic dynamics simulation requires a value to be assigned to the collision frequency friction coefficient γ. For simple particles such as spheres this can be related to the diffusion constant in the fluid. For the simulation of a rigid molecule it may be possible to derive γ via the diffusion coefficient from a standard molecular dynamics situation. In the more general case we require the friction coefficient of each atom. For simple molecules such as butane the friction coefficient can be considered to be the same for all atoms. The optimal value for γ can be determined by trial and error, performing a stochastic dynamics simulation for different values of γ and comparing the results with those from experiment (where available) or from standard molecular dynamics simulations. For large molecules the atomic friction coefficient is considered to depend upon the degree to which each atom is in contact with the solvent and is usually taken to be proportional to the accessible surface area of the atom (as defined in Section 1.5).

One of the main advantages of the stochastic dynamics methods is that dramatic time savings can be achieved, which enables much longer stimulations to be performed For example, Widmalm and Pastor performed 1 ns molecular dynamics and stochastic dynamics simulations of an ethylene glycol molecule in aqueous solution of the solute and 259 water molecules [Widmalm and Pastor 1992]. The molecular dynamics simulation required 300 hours whereas the stochastic dynamics simulation of the solute alone required just 24 minutes. The dramatic reduction in time for the stochastic dynamics calculation is due not only to the very much smaller number of molecules present but also to the fact that longer time steps can often be used in stochastic dynamics simulations.

Stochastic dynamics has been widely used to study the behaviour of long-chain molecules and polymers. The advantages of stochastic dynamics are especially important for polymers [Helfand 1984], where many interesting phenomena occur over relatively long time periods, so putting them beyond the scope of conventional molecular dynamics. However, one must take care with the Langevin method when simulating systems in which specific solute–solvent interactions are present. For example, Yun-Yu, Lu and van Gunsteren used both stochastic dynamics and molecular dynamics to study the immunosuppressant drug cyclosporin (Figure 7.14) in two solvents: carbon tetrachloride and water [Yun-Yu *et al.* 1988]. The time-averaged structures obtained from each method were compared to determine the similarity between the average structure obtained for each simulation. Fluctuations in torsion angles were also compared. The analysis showed that the structures obtained from the molecular dynamics and stochastic dynamics simulations of cyclosporin in carbon tetrachloride were very similar, but that the results were very different for the Langevin and molecular dynamics simulations performed in water. This was due to an excessive degree of internal hydrogen bonding in the stochastic dynamics simulation; the equivalent

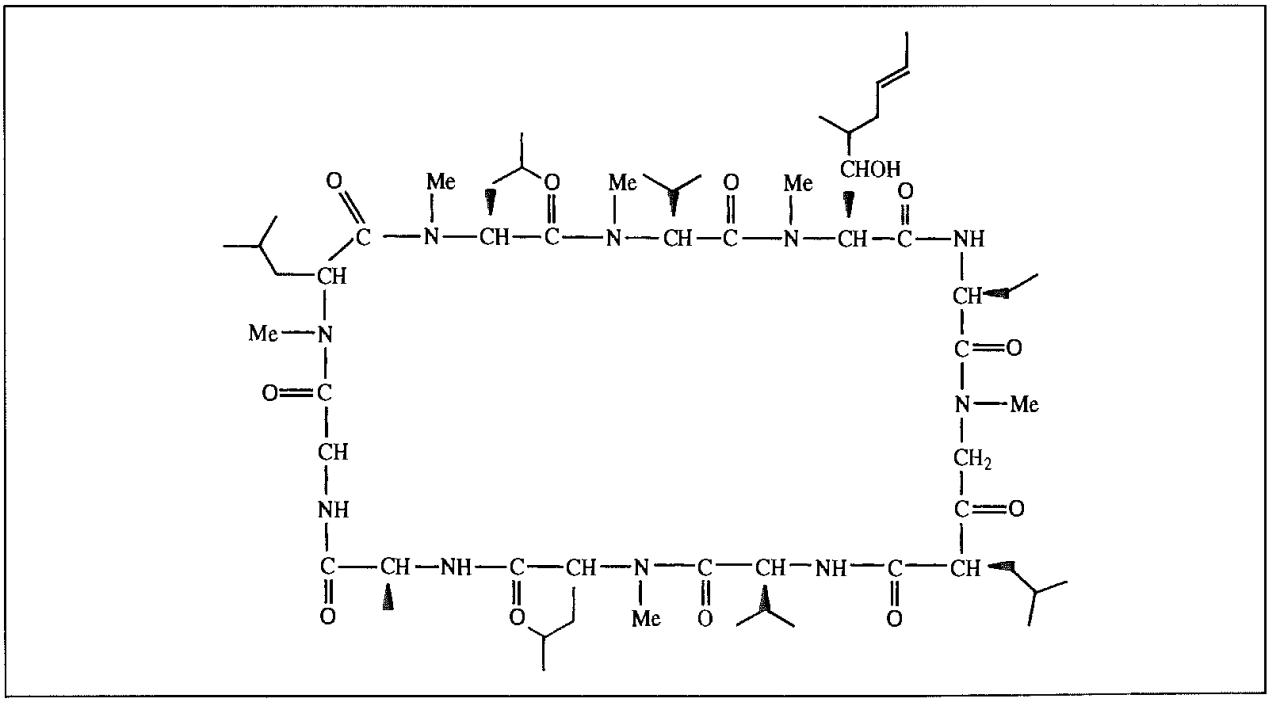

Fig 7 14· Cyclosporin

molecular dynamics simulation contained much more hydrogen bonding between cyclosporin and the solvent.

7.9 Conformational Changes from Molecular Dynamics Simulations

Molecular dynamics can provide information about the conformational properties of molecular systems and the way in which the conformation changes with time. Molecular graphics programs can facilitate the analysis of such simulations by displaying the structural parameters of interest in a manner that enables the time dimension to be taken into account. Perhaps the most direct way to demonstrate the conformational behaviour of the system is as a movie, where coordinate sets saved at regular intervals are displayed in sequence. For publication purposes, time-dependent data can be displayed graphically, with one of the axes corresponding to the time, such as the plots of energy or autocorrelation function versus time (Figures 7.3 and 7.10). The representation of bond rotations is difficult using x/y plots due to the 2π periodicity of a torsion angle. Lavery and Sklenar have developed a method to represent torsion data as a polar plot [Lavery and Sklenar 1988], where the distance from the origin corresponds to the time (Figure 7.15). Such 'dials' are very useful for detecting the presence of correlated conformational changes.

When viewing a movie of a molecular dynamics simulation of a complex molecule one is often struck by the chaotic nature of the motion. This should be expected; the motion of complex molecules *is* chaotic, but there are often underlying low-frequency motions which correspond to more significant and more interesting conformational changes. Fourier analysis techniques can be used to filter out the unwanted high-frequency motions, enabling the important low-frequency changes to be observed unhindered. Here we describe the filtering method of Dauber-Osguthorpe and Osguthorpe [Dauber-Osguthorpe and Osguthorpe 1990, 1993].

A Fourier transform enables one to convert the variation of some quantity as a function of time into a function of frequency, and vice versa. Thus, if we represent the quantity that varies in time as $x(t)$, then Fourier analysis enables us to also represent that quantity as a function $X(\nu)$, where ν is the frequency $(-\infty < \nu < \infty)$. Fourier analysis is usually introduced by considering functions that vary in a periodic manner with time which can be written as a superposition of sine and cosine functions (a Fourier series; see Section 1 10.8). If the period of the function $x(t)$ is τ then the cosine and sine terms in the Fourier series are functions of frequencies $2\pi n/\tau$, where n can take integer values 1, 2, 3,....

A Fourier series is rarely relevant to the interpretation of a molecular dynamics simulation as the movement of the atoms is not periodic but chaotic. The Fourier transform enables a non-periodic function to be converted into the equivalent frequency function (and vice versa). The Fourier transform can be developed from the Fourier series simply by considering the effect of increasing the period of a periodic function to infinity. The frequency function obtained from a Fourier transform is a continuous function rather than one written as a series of discrete frequencies. Further details are provided in Section 1.10.8.

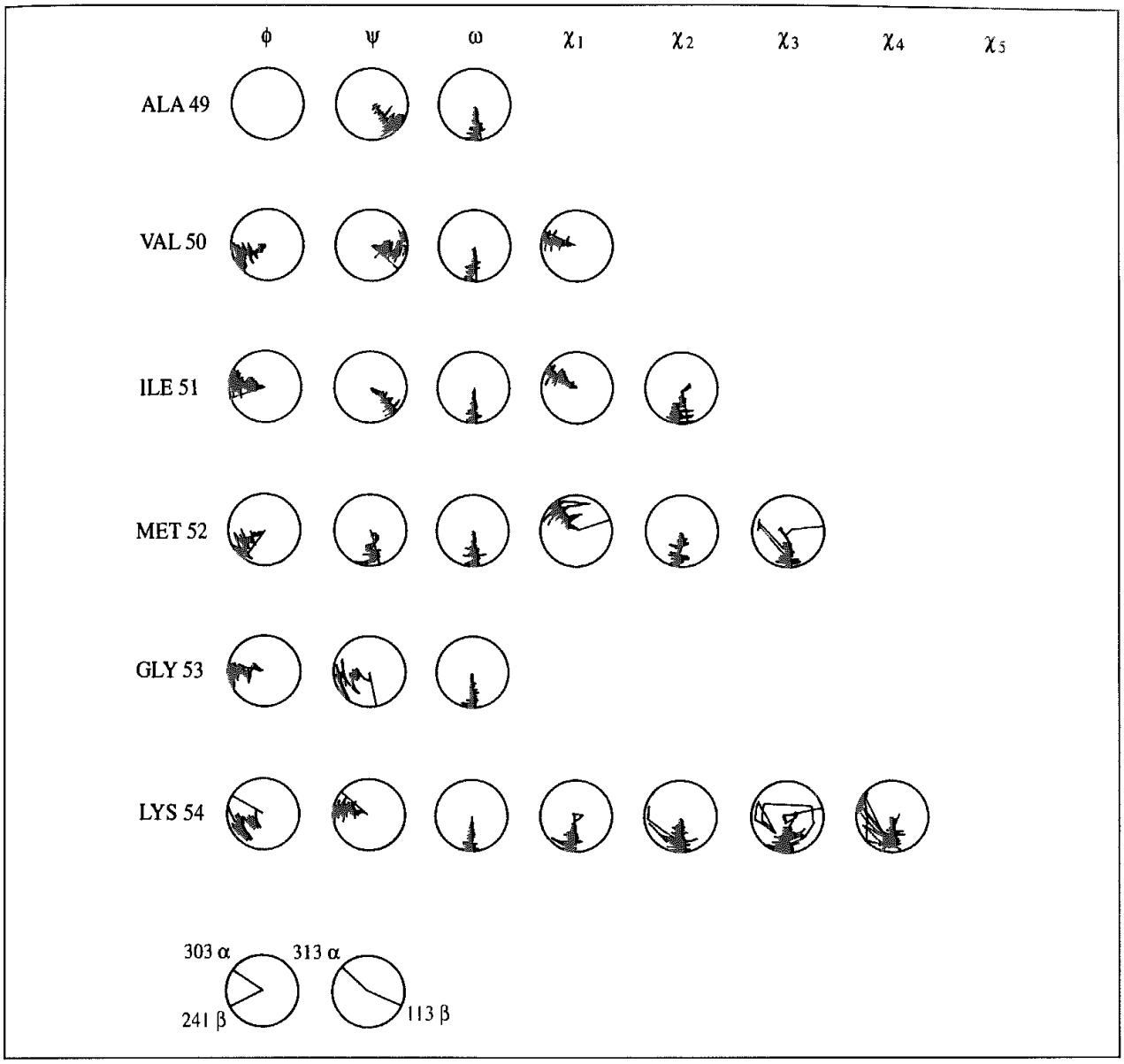

Fig 7 15 The variation in torsion angles can be effectively represented as a series of 'dials', where the time corresponds to the distance from the centre of the dial Data from a molecular dynamics simulation of an intermolecular complex between the enzyme dihydrofolate reductase and a triazine inhibitor [Leach and Klein 1995]

At each step of the Fourier analysis of a molecular dynamics simulation the variation with time of one of the Cartesian coordinates of one of the atoms in the system is converted into the correponding frequency function. Fast Fourier techniques are usually employed for this step. The frequency spectrum can then be filtered to remove high frequencies. This is achieved simply by setting the coefficients of the unwanted frequencies in the frequency function to zero. The resulting spectrum is then converted back to the time domain to give a new set of coordinate values at each of the time steps in the trajectory. This new coordinate set includes only the selected frequencies. This process can be repeated for the three coordinates of each atom to give a filtered trajectory for the entire system. It is also possible to select just a single frequency (i.e. a single normal mode) from the frequency spectrum and view this in isolation.

7.10 Molecular Dynamics Simulations of Chain Amphiphiles

The molecular dynamics technique is widely used for simulating large molecular systems, some of which have many degrees of conformational freedom. In this section we will examine the application of molecular dynamics to chain amphiphiles, a class of molecules of interest to both the 'biological' and 'materials science' communities These molecules have a polar head group attached to one or more hydrocarbon chains. Some examples are shown in Figure 7.16. The head group has a high affinity for water, whereas the hydrocarbon tail prefers to exist in a hydrophobic environment. The molecules therefore exist in both phases at a water/oil interface. A characteristic feature of these molecules is their ability

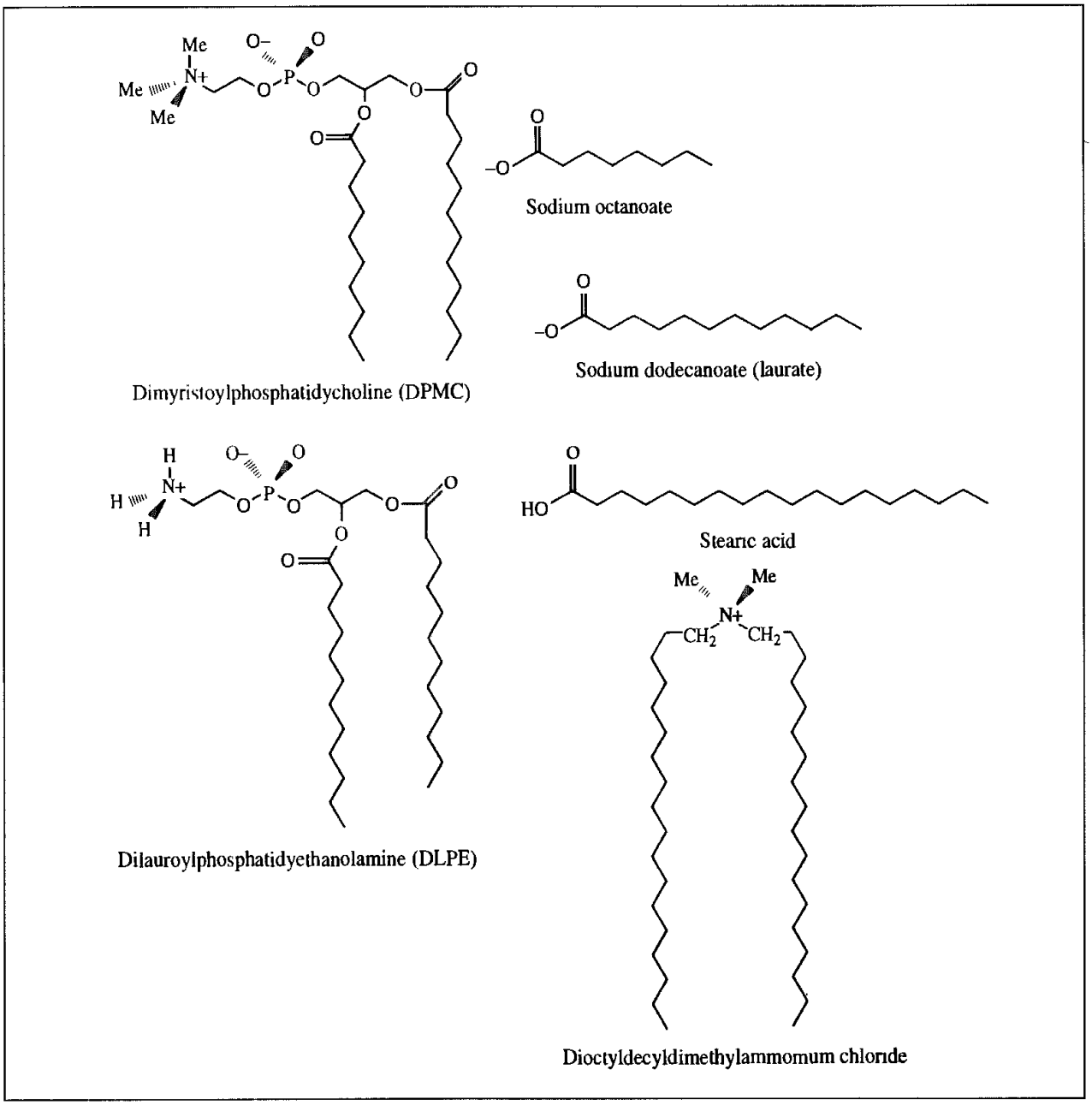

Fig. 7.16· Some typical amphiphiles

to form extended layer structures. Monolayers, bilayers and multiple layers are all possible. A monolayer at the water/air interface is known as a Langmuir film; when this is transferred to a solid substrate it is known as a Langmuir–Blodgett film. Langmuir–Blodgett films with many layers can be constructed in the laboratory but most simulation studies of these systems have been restricted to monolayers or bilayers. The ability to control the thickness of a Langmuir–Blodgett film and their high degree of order means that they are intensively investigated as insulators in semiconductors, filtration devices and as anti-reflective coatings. Amphiphiles are important in biology as cell membranes are formed from lipid bilayers. At a high enough concentration some amphiphiles can form micelles, which are globular structures that have the head groups all pointing into solution and the tails inside (Figure 7.17).

Amphiphiles often have a complex phase behaviour with several liquid crystalline phases. These liquid crystalline phases are often characterised by long-range order in one direction together with the formation of a layer structure. The molecules may nevertheless be able to move laterally within the layer and perpendicular to the surface of the layer. Structural information can be obtained using spectroscopic techniques including X-ray and neutron diffraction and NMR. The quadrupolar splitting in the deuterium NMR spectrum can be

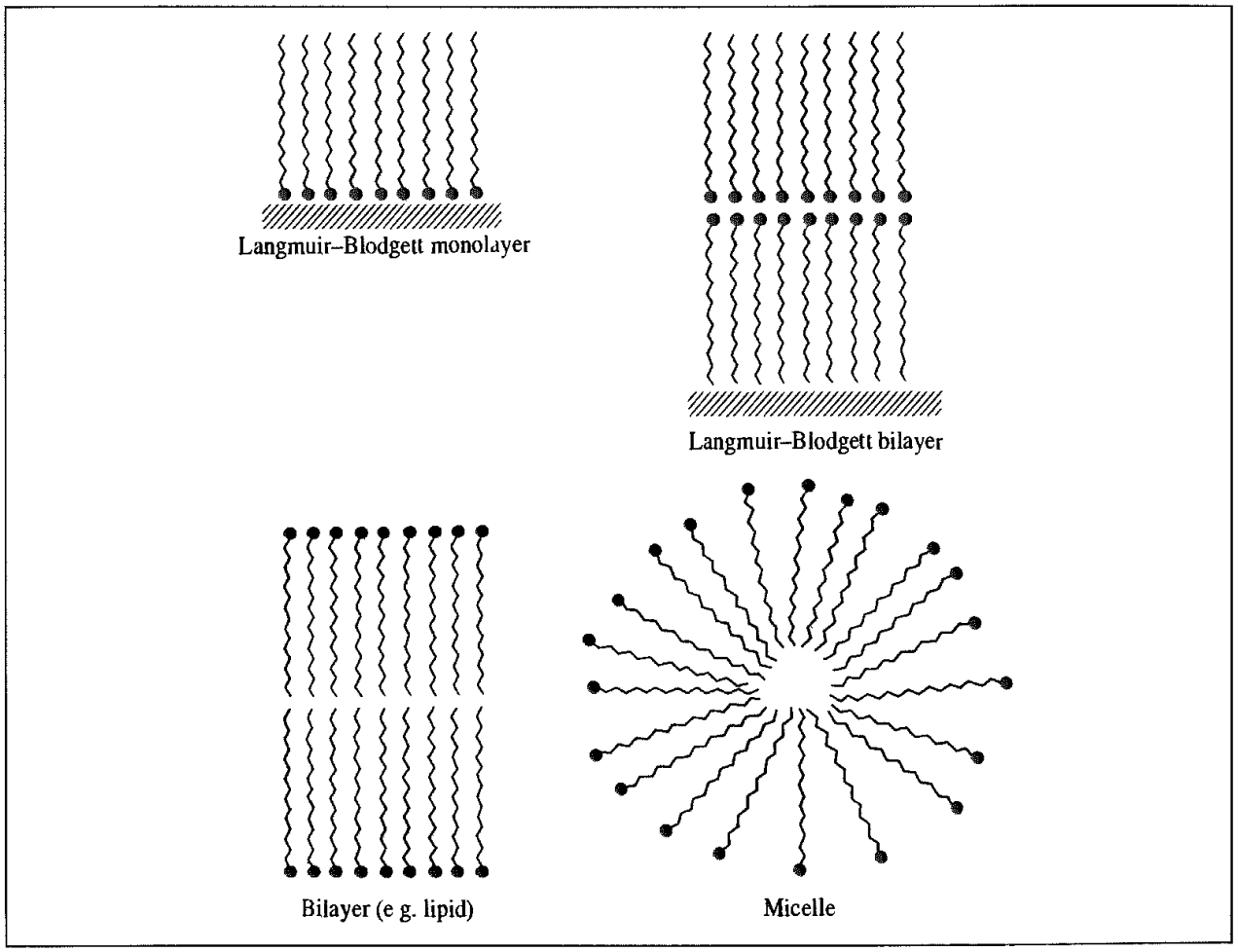

Fig 7 17 Some of the various phases that amphiphiles may form.

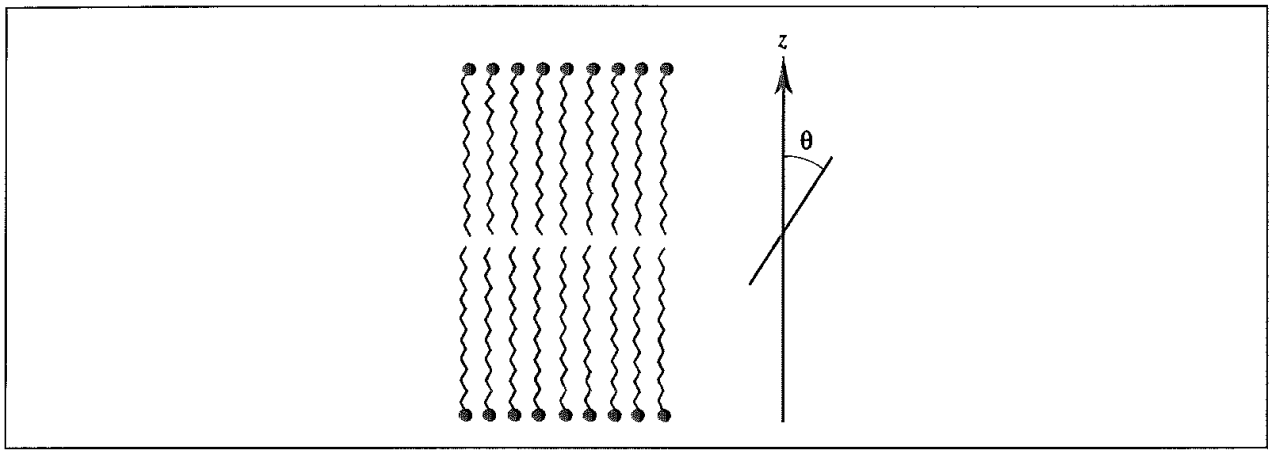

Fig 7.18 Definition of the order parameter

used to determine an *order parameter* for the carbon atoms on the hydrocarbon tail. The order parameter is defined as.

$$S = 0.5\langle 3\cos\theta_i \cos\theta_j - \delta_{ij}\rangle \tag{7.126}$$

θ_i is the angle between the ith molecular axis and the *director,* which is the average of the molecular axes over the sample. For a bilayer in the $L\alpha$ phase (the one present in cell membranes) the director is the same as the bilayer normal and is conventionally taken to be the z axis; see Figure 7.18. δ_{ij} is the Kronecker delta function ($\delta_{ij} = 1$ if $i = j$; $\delta_{ij} = 0$ if $i \neq j$) The expression for S is averaged over time and over molecules. The deuterium NMR experiment provides the order parameter S_{CD}, which indicates the average orientation of the C–D bond vector with respect to the bilayer normal. The experimental order parameters S_{CD} can range from 1 0 (indicating full order along the bilayer normal) to −0.5 (full order perpendicular to the bilayer normal) [Seelig and Seelig 1974]. A value of zero is considered to indicate full isotropic motion of the group. Experimental values are determined using molecules with deuterium-substituted methylene groups at positions along the hydrocarbon chain. Many simulations of amphiphiles are performed using united atom models for the hydrocarbon chains and it is therefore necessary to be able to relate the experimental order parameters to values that can be calculated from a simulation. This is done as follows [Essex *et al.* 1994]. Molecular axes are defined for each CH_2 unit in the chain as shown in Figure 7 19. These molecular axes are defined for the nth CH_2 unit as follows:

z: vector from C_{n-1} to C_{n+1}
y: vector perpendicular to z and in the plane through C_{n-1}, C_n and C_{n+1}
x: perpendicular to y and z

Using these definitions, components of the molecular order parameter tensor can be determined (for example, S_{zz} is determined by measuring the angle between the molecular z axis and the bilayer normal). The experimental order parameter can be related to the molecular order parameter using the equation:

$$S_{CD} = 2S_{xx}/3 + S_{yy}/3 \tag{7.127}$$

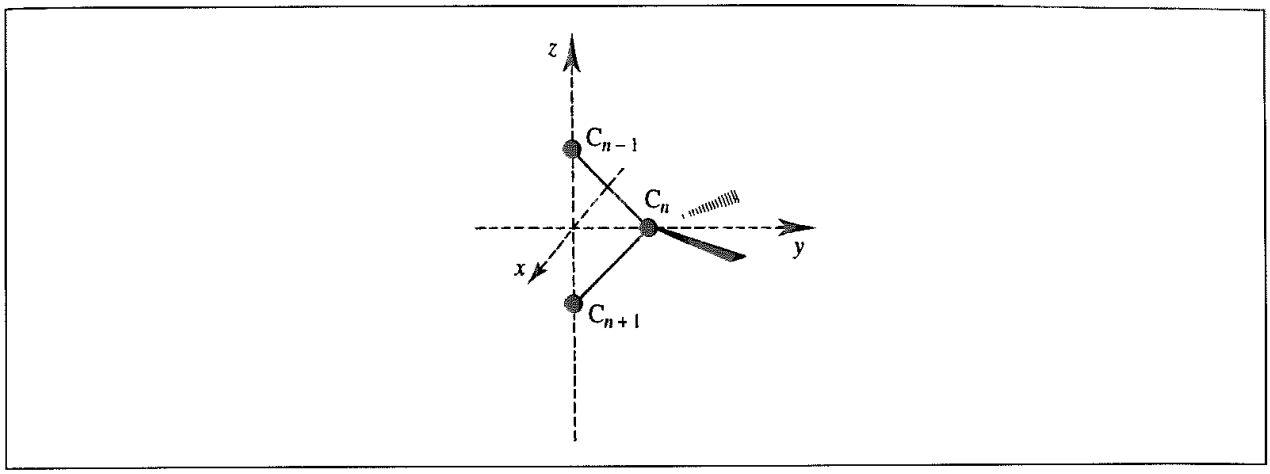

Fig 7 19. Calculation of the order parameter for united atom simulations.

With all-atom simulations the locations of the hydrogen atoms are known and so the order parameters can be calculated directly. Another structural property of interest is the ratio of *trans* conformations to *gauche* conformations for the CH_2–CH_2 bonds in the hydrocarbon tail. The *trans* : *gauche* ratio can be estimated using a variety of experimental techniques such as Raman, infrared and NMR spectroscopy.

7.10.1 Simulation of Lipids

There has been considerable interest in the simulation of lipid bilayers due to their biological importance. Early calculations on amphiphilic assemblies were limited by the computing power available, and so relatively simple models were employed. One of the most important of these is the mean field approach of Marcelja [Marcelja 1973, 1974], in which the interaction of a single hydrocarbon chain with its neighbours is represented by two additional contributions to the energy function. The energy of a chain in the mean field is given by:

$$\mathscr{V}_{\text{tot}} = \mathscr{V}_{\text{int}} + \mathscr{V}_{\text{disp}} + \mathscr{V}_{\text{rep}} \tag{7.128}$$

where $\mathscr{V}_{\text{int}}$ is the internal energy of a chain, which can be calculated using standard force field methods. $\mathscr{V}_{\text{disp}}$ simulates the van der Waals interactions with the neighbouring molecules. It is often modelled using a Maier-Saupe potential:

$$\mathscr{V}_{\text{disp}} = -\Phi \sum_{i=1}^{\text{carbons}} \tfrac{1}{2}(3\cos^2\theta_i - 1) \tag{7 129}$$

The summation runs over all carbon atoms in the chain. θ_i is the angle between the bilayer normal and the molecular axis, as discussed above. Φ is the field strength; this may be parametrised to reproduce appropriate experimental data such as the deuterium NMR order parameters or it may be obtained by a self-consistent protocol, as described below. In his work on lipid bilayers Marcelja used a slightly different expression for $\mathscr{V}_{\text{disp}}$ which

involved the fraction of *trans* bonds in the system:

$$\mathscr{V}_{\text{disp}} = -\Phi \frac{n_{\text{trans}}}{n} \sum_{i=1}^{\text{carbons}} \tfrac{1}{2}(3\cos^2\theta_i - 1) \tag{7.130}$$

This additional factor was introduced to ensure the proper behaviour over both liquid crystalline and solid phases. In simulations of the liquid crystalline phase alone this term may be omitted for computational efficiency.

The repulsive contribution, $\mathscr{V}_{\text{rep}}$, is due to lateral pressure on each chain. In Marcelja's original treatment, this was set equal to the product of the lateral pressure, γ, and the cross-sectional area of the chain. The cross-sectional area was approximated by:

$$A = A_0 l_0 / l \tag{7.131}$$

where l_0 and A_0 are the length and cross-sectional area, respectively, of the hydrocarbon chain in a fully extended conformation. l is the length of the chain in the current conformation, projected onto the bilayer normal. If the bilayer normal is along the z axis then l is taken to be the z coordinate of the last carbon atom in the hydrocarbon chain. In other mean field models [Pastor *et al.* 1988] the product $\gamma A_0/l_0$ is replaced with a single adjustable parameter Γ and so $\mathscr{V}_{\text{rep}}$ is given by:

$$\mathscr{V}_{\text{rep}} = \sum_{\text{chains}} \frac{\Gamma}{(z_n - z_0)} \tag{7.132}$$

where z_n is the z coordinate of the last carbon in the chain and z_0 is the coordinate of the surface of the monolayer or bilayer. This force acts to keep the last carbon away from the surface; the closer it gets the larger the force pulling it away.

In his calculations, Marcelja generated all possible conformations of the hydrocarbon chain, restricting each carbon–carbon bond to the *trans* and *gauche* conformations. The energy of each conformation was evaluated. From the ensemble of conformations a partition function can be computed:

$$Z = \sum_{\text{all conformations}} \exp[-\mathscr{V}_{\text{tot}}/k_{\text{B}}T] \tag{7 133}$$

The molecular field is related to the partition function:

$$\Phi = \sum_{\text{all conformations}} \left\{ \frac{\dfrac{n_{\text{trans}}}{n} \displaystyle\sum_{i=1}^{\text{carbons}} \tfrac{1}{2}(3\cos^2\theta_i - 1)\exp[-\mathscr{V}_{\text{tot}}/k_{\text{B}}T]}{Z} \right\} \tag{7.134}$$

The molecular field is thus related to the partition function and so it is possible to generate a self-consistent value of the molecular field, Φ. Thermodynamic properties can then be calculated from the partition function. For example, Marcelja calculated the pressure as a function of the area per polar head group for surface monolayers at a variety of temperatures. His results showed good qualitative agreement with experimental results for such systems.

The mean field approach can be incorporated into a molecular dynamics simulation. It is particularly useful when used in conjunction with Langevin dynamics, as very long simulations can be performed. For example, Pearce and Harvey were able to perform simulations of three unsaturated phospholipids for 100 ns (i.e. 0.1 μs) in single-molecule Langevin dynamics calculations [Pearce and Harvey 1993]. An extension of this strategy is to use a central 'core' containing one or more molecules that are simulated using molecular dynamics This core is surrounded by a shell of molecules that are simulated using Langevin dynamics with the mean field. In this way one attempts to simulate a more 'realistic' system without incurring the computational penalty of a full molecular dynamics simulation of the entire system [De Loof *et al.* 1991].

The first molecular dynamics simulations of a lipid bilayer which used an explicit representation of all the molecules was performed by van der Ploeg and Berendsen in 1982 [van der Ploeg and Berendsen 1982]. Their simulation contained 32 decanoate molecules arranged in two layers of sixteen molecules each. Periodic boundary conditions were employed and a united atom force potential was used to model the interactions. The head groups were restrained using a harmonic potential of the form:

$$v(z) = \frac{k_h}{2}(z - \langle z \rangle)^2 \qquad (7\,135)$$

By writing the restraint in terms of the average z coordinates of the head groups ($\langle z \rangle$) van der Ploeg and Berendsen ensured that the bilayer was able to change its thickness to reach its equilibrium value. This restraining potential was designed to reproduce the interactions between the head groups and the water layer, neither of which was explicitly included in the calculation. A key feature of the simulation was the long equilibration time required. By explicitly representing all the molecules in the system it was possible to determine the collective motion of the system as a whole. One distinct feature was a slowly fluctuating collective tilt of the molecules away from the normal to the bilayer surface (Figure 7.20). The degree to which the molecules were aligned with each other was also correlated with the tilt angle. When the average tilt angle reached a maximum the chains were much more likely to be well aligned, but when the average tilt angle was close to zero (i.e. such that the average orientation of the chains was almost normal to the bilayer surface) much less order was observed. In their original simulations this collective tilt phenomenon was observed to extend over the entire simulation cell, suggesting that the cell dimensions were too small and that the use of periodic boundary conditions was enhancing the long-range correlations. Simulations using a larger system subsequently showed that this collective tilt could be observed for subsets of the molecules.

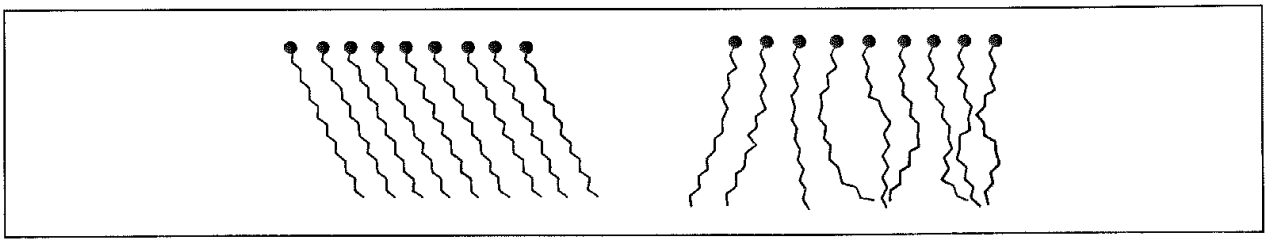

Fig. 7 20 Variation in alignment of chains in lipid simulation with tilt angle [van der Ploeg and Berendsen 1982]

Faster computers have enabled more realistic simulations of lipid bilayers to be performed, of larger systems, with more accurate models and for longer times [Stouch 1993; Tobias *et al.* 1997]. The trend is very much towards simulations that use full representations of all species present (i e 'all-atom' models with explicit solvent and counterions). The charged and highly polar nature of lipid head groups means that a proper representation of the long-range electrostatic forces can be critical, using a method such as the Ewald summation. Equilibration of such systems often requires hundreds of picoseconds and certain phenomena are only observed on a nanosecond timescale. In addition, molecules such as cholesterol and proteins may be included within the membrane These simulations have revealed many hitherto unknown features of the behaviour of such systems. For example, considerable conformational mobility of the hydrocarbon chains is often observed in the liquid crystalline phases. This is illustrated in Figure 7.21 (colour plate section), which shows a snapshot of a lipid bilayer after a molecular dynamics simulation of several hundred picoseconds. The considerable degree of disorder in the hydrocarbon chains near the middle of the bilayer is clear from this figure and is very different to the idealised, 'textbook' pictures in which the chains are perfectly aligned in completely extended conformations. The distribution of *gauche* conformations tends to be higher towards the end of the chain, though in some systems a *gauche* link is required near the head group to enable the chain to lie perpendicular to the interface. 'Kinks' are often observed in the chains; these are arrangements of three successive bonds with *gauche*(+)-*trans*-*gauche*(−) torsion angle, which enable the chain to remain perpendicular to the surface.

7.10.2 Simulations of Langmuir–Blodgett films

The simulations of Langmuir-Blodgett systems can be difficult due to the need to correctly model the solid support To illustrate the procedure we will describe the calculations of Kim, Moller, Tildesley and Quirke [Kim *et al.* 1994a] who simulated stearic acid ($CH_3(CH_2)_{16}COOH$) adsorbed onto graphite. The surface was modelled using a Lennard-Jones 9-3 potential that depends upon the height of the atom (α) above the surface (z_α):

$$v_{\alpha s}(z_\alpha) = \frac{2\pi\rho}{3}\varepsilon_{ss}\left[\frac{2}{15}\left(\frac{\sigma_{\alpha s}}{z_\alpha}\right)^9 - \left(\frac{\sigma_{\alpha s}}{z_\alpha}\right)^3\right] \tag{7.136}$$

where ρ is the density of the solid and ε_{xx} and δ_{ss} are its Lennard-Jones parameters. An image-change method was also applied to the acid head group with the interaction between a charge and its image being:

$$v_{ic}(z) = \frac{1}{2}\frac{(\varepsilon - \varepsilon')}{(\varepsilon + \varepsilon')}\left[\frac{q_\alpha^2}{8\pi\varepsilon_0(z - z_{ip})}\right] \tag{7.137}$$

where ε' is the relative permittivity of the solid (taken to be 4.0) and ε is the permittivity above the surface ($\varepsilon = 1.0$). The image plane is located at $z_{ip} = \sigma_{ss}/2$. Each charge interacts with its own image and with the images of other charges, but there are no interactions between the image charges themselves. The hydrocarbon chain of the stearic acid was modelled using an all-atom model, with explicit hydrogen atoms.

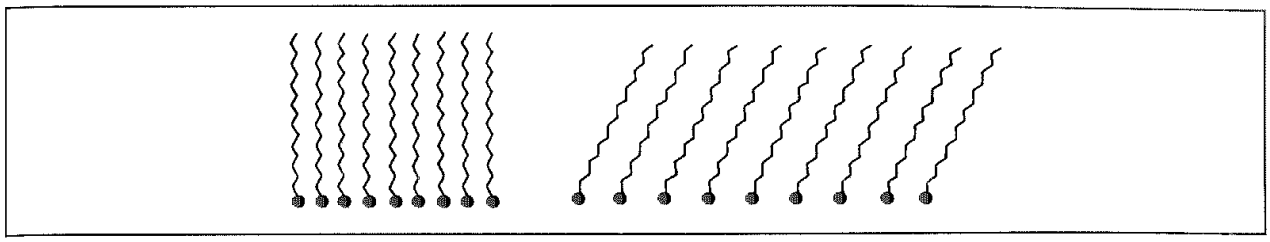

Fig 7.22 Simulations of a Langmuir-Blodgett film [Kim et al *1994a]· as the area per head group increases the chains tilt away from the normal*

A molecular dynamics simulation of 64 molecules with periodic boundary conditions confirmed the presence of a transition in which the collective tilt of the chains changed from being upright (i.e. perpendicular to the surface) to having an angle of around 20° (Figure 7.22). This transition was induced by increasing the area per head group. The proportion of molecules in the all-*trans* conformation decreased significantly as the head group area was increased (97.7% of molecules were fully extended for a head group area of 20.6 $Å^2$ but only 66.9% for an area of 21.2 $Å^2$). The bond linking the chain to the acid head growth showed a considerable degree of rotational disorder.

Bilayers of stearic acid were also simulated on a hydrophobic surface [Kim *et al.* 1994b]. In the bilayer the molecules are arranged head to head, with the hydrocarbon tail on the surface. In this arrangement hydrogen bonds form between the head groups (Figure 7.23). The bilayer also showed the tilt angle transition that was observed for the monolayer, though the degree of tilt was considerably less for the bilayer, suggesting that hydrogen bonding between the head groups was important in controlling the orientation of the molecules.

An extension of these calculations to cationic dialkylamide salts required an even more complex model [Adolf *et al.* 1995]. These molecules have the general formula $(CH_3)_2N^+[(CH_2)_{n-1}CH_3][(CH_2)_{m-1}CH_3]Cl^-$ and the isomer with $m = n = 18$ is one of the main active ingredients in commercial fabric softeners. The presence of two long alkyl

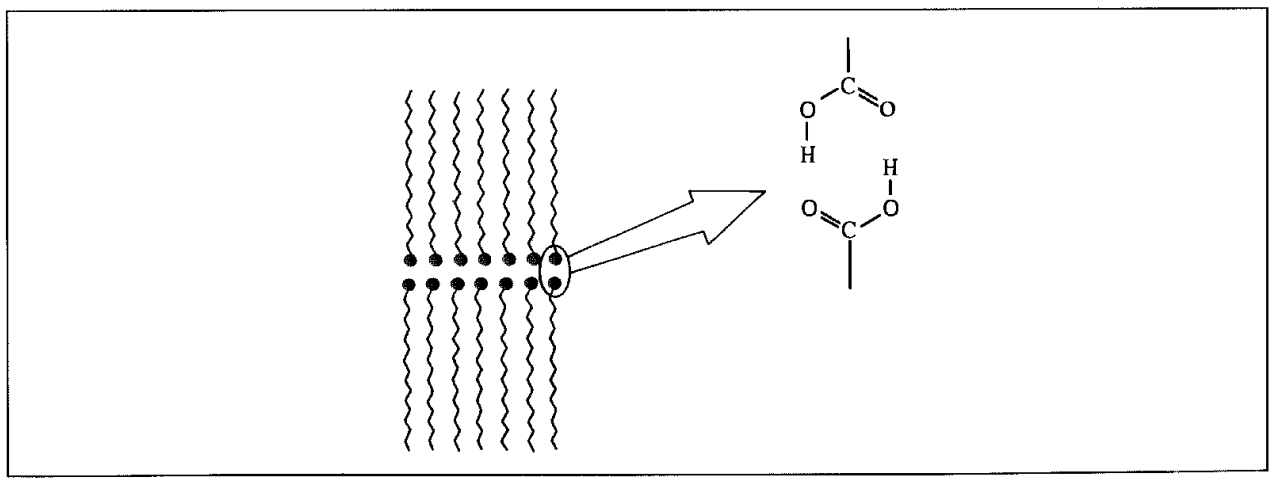

Fig 7.23 In simulations of stearic acid on a hydrophobic surface hydrogen bonding between the head groups is important in controlling the orientation of the molecules [Kim et al *1994b]*

chains and an ionic head group means that these molecules are also structurally similar to phospholipids. A modified Ewald method was used to calculate electrostatic interactions in the two dimensions parallel to the surface, and the anisotropic potential model of Toxvaerd (see Section 4.15) was employed to retain the computational savings of a united-atom model. This system also showed a variation in the tilt with head group area, though the results at the highest head group densities were not as 'solid-like' as was suggested by the experimental data. Nevertheless, there were some areas where the model could be improved, including the need to incorporate water molecules and use a more appropriate representation of the chloride anion.

7.10.3 Mesoscale Modelling: Dissipative Particle Dynamics

The molecular dynamics methods that we have discussed in this chapter, and the examples that have been used to illustrate them, fall into the category of 'atomistic simulations', in that all of the actual atoms (or at least the non-hydrogen atoms) in the core system are represented explicitly. Atomistic simulations can provide very detailed information about the behaviour of the system, but as we have discussed this typically limits a simulation to the nanosecond timescale. Many processes of interest occur over a longer timescale. In the case of processes which occur on a 'macroscopic' timescale (i.e. of the order of seconds) then rather simple models may often be applicable. Between these two extremes are phenomena that occur on an intermediate scale (of the order of microseconds). This is the realm of the *mesoscale* Dissipative particle dynamics (DPD) is particularly useful in this region, examples include complex fluids such as surfactants and polymer melts.

These three general regions (atomistic, mesoscopic and macroscopic) are not only characterised by different timescales but also varying length scales Indeed, there is a general inverse relationship between the time and the length In the case of the dissipative particle dynamics method the fast motion of the atoms is integrated out, leaving as the fundamental 'unit' a set of beads that interact with other beads via an appropriate potential [Koelman and Hoogerbrugge 1993]. Each bead represents a small 'droplet' of the fluid. The total force on each bead is due to a combination of direct interactions with other beads together with random and dissipative forces. The trajectory of the system is calculated by integrating Newton's laws of motion in the usual way, from which properties can be derived.

The underlying model in dissipative particle dynamics is usually developed in such a way that the mass, length and timescales are all unity. This is similar to the use of reduced units for the Lennard-Jones potential (Section 4.10.5). A particular advantage of such an approach is that a single simulation may often be able to explain the behaviour of many different systems. With a mass of 1 the force acting on a particle is equal to its acceleration. In DPD there are three forces on each bead [Groot and Warren 1997]:

$$\mathbf{f}_i = \sum_{j=1;j\neq i}^{N} (\mathbf{F}_{ij}^{\mathrm{C}} + \mathbf{F}_{ij}^{\mathrm{D}} + \mathbf{F}_{ij}^{\mathrm{R}}) \tag{7.138}$$

The summation is over all other particles j which are within a certain cutoff radius r_c of i. This cutoff radius becomes the unit of length in the subsequent treatment (i.e $r_c = 1$). The

first of these forces, the conservative force, $\mathbf{F}_{ij}^{C}$, is a soft repulsion that acts along the line joining i to j.

$$F_{ij}^{C} = \begin{cases} a_{ij}(1 - r_{ij})\hat{\mathbf{r}}_{ij} & r_{ij} < 1 \\ 0 & r_{ij} > 1 \end{cases} \tag{7.139}$$

r_{ij} is the distance between beads i and j and $\hat{\mathbf{r}}_{ij}$ is the corresponding unit vector. The second force is a dissipative (or drag) force, which is given by:

$$\mathbf{F}_{ij}^{D} = \begin{cases} -\gamma w^{D}(r_{ij})(\hat{\mathbf{r}}_{ij} \cdot \mathbf{v}_{ij})\hat{\mathbf{r}}_{ij} & r_{ij} < 1 \\ 0 & r_{ij} > 1 \end{cases} \tag{7.140}$$

This dissipative force is proportional to the relative velocity of the two beads and acts so as to reduce their relative momentum. $\mathbf{v}_{ij}$ is the difference between the two velocities ($\mathbf{v}_{ij} = \mathbf{v}_i - \mathbf{v}_j$) and $w^{D}(r_{ij})$ is a weight function that depends upon the distance r_{ij} and disappears for inter-bead distances greater than unity (i.e. r_c).

The third and final force acting between any pair of beads is a random force:

$$\mathbf{F}_{ij}^{R} = \begin{cases} \sigma w^{R}(r_{ij})\theta_{ij}\hat{\mathbf{r}}_{ij} & r_{ij} < 1 \\ 0 & r_{ij} > 1 \end{cases} \tag{7.141}$$

$w^{R}(r_{ij})$ is a distance-dependent weight function similar to that for the dissipative force. θ_{ij} is a function which ensures that the random force between each pair of particles averages to zero over time and is independent of the force between every other pair of particles. The random force can be more usefully expressed in terms of the timestep in the integration scheme:

$$\mathbf{F}_{ij}^{R} = \frac{\sigma w^{R}(r_{ij})\zeta_{ij}\hat{\mathbf{r}}_{ij}}{\sqrt{\delta t}} \tag{7 142}$$

ζ_{ij} is a random number with zero mean and unit variance, chosen independently for each pair of particles and at each time step in the integration.

Both the dissipative force and the random force act along the line joining the pair of beads and also conserve linear and angular momentum. The model thus has two unknown functions $w^{D}(r_{ij})$ and $w^{R}(r_{ij})$ and two unknown constants γ and σ. In fact, only one of the two weight functions can be chosen arbitrarily as they are related [Espanol and Warren 1995]. Moreover, the temperature of the system relates the two constants:

$$w^{D}(r) = [w^{R}(r)]^2 \tag{7 143}$$

$$\sigma^2 = 2\gamma k_B T \tag{7.144}$$

The usual choice for the weight functions is to make the random force the same as the conservative force:

$$w^{D}(r) = [w^{R}(r)]^2 = \begin{cases} (1 - r)^2 & r < 1 \\ 0 & r > 1 \end{cases} \tag{7 145}$$

The equations of motion are integrated using a modified velocity Verlet algorithm. The modification is required because the force depends upon the velocity, the extra step involves

a prediction followed by a correction. If, in addition to the use of units of mass and length, we assume that k_BT is equal to 1, then the unit of time is:

$$\tau = r_c\sqrt{m/k_BT} \tag{7.146}$$

It remains to assign values to the noise amplitude, σ, the time step for the integration, δt, and the repulsion parameter, a_{ij}. The effects of the first two of these upon the stability of the simulation are also related to the integration method. Groot and Warren determined that when the noise amplitude was larger than 8 the integration scheme became unstable and that a value of 3 gave good results over a range of temperatures [Groot and Warren 1997]. The integration time step with the modified Verlet algorithm should have a value between 0.04 and 0.06; any larger and the temperature would artificially increase by an unacceptable amount. The repulsion parameter is the key determinant of the interactions between the beads. This can be achieved by relating the DPD model to bulk properties For example, to model the compressibility of water at room temperature the repulsion parameter is related to the density, ρ, by:

$$a_{ii}\rho = 75k_BT \tag{7.147}$$

These interaction parameters between particles of the same type can be used to derive the values of a_{ij} between unlike beads. For polymers which involve beads of different types the repulsion between unlike beads is made larger than between like beads.

A good example of the use of DPD is the study by Groot and Madden of the microphase separation of diblock copolymer melts [Groot and Madden 1998; Groot *et al.* 1999]. Block copolymers are surfactants which are present in many consumer products such as foods (e.g. ice cream and margarine), detergents and personal care products (e g. shampoo). The properties of these materials are strongly dependent upon their bulk organisation (or *morphology*), which in turn depends upon the relative sizes of the head and the tail groups and how they interact. The diblock copolymers of interest can be represented by the general formula A_mB_n where A and B represent amalgamations of the smaller building blocks from which the polymer is constructed. Of particular interest was the way in which the behaviour of the system varied as the ratio of A to B was changed, for a fixed polymer length. In this particular case the length was fixed at 10 beads and the entire simulation contained a total of 40 000 particles. A variety of systems were investigated, such as A_2B_8, A_3B_7 and A_5B_5. The beads in each polymer chain were kept together by adding an extra term to the force (Equation (7.139)) of the form Cr_{ij} if i is connected to j.

Due to the greater degree of repulsion between unlike beads the final configuration of the system contains domains which are rich in either the A or the B type of bead. Some regions are rich in A; others are rich in B. The organisation of the A-rich and B-rich domains can be visualised by plotting a three-dimensional contour that connects regions where the density is intermediate between purely A and purely B.

For the 1:1 polymer (A_5B_5) a lamellar phase was obtained, in which the A- and B-rich domains form parallel planes of alternating A and B beads (Figure 7.24(a), colour plate section). For other configurations, however, different structures were observed. The A_3B_7 system evolved to a hexagonal phase (Figure 7.24(b)) and the A_2B_8 structure produces a set of peanut-shaped micelles (Figure 7.24(c))

Appendix 7.1 Energy Conservation in Molecular Dynamics

The total energy is the sum of the kinetic $\mathscr{K}(t)$ and potential energies $\mathscr{V}(t)$:

$$E(t) = \mathscr{K}(t) + \mathscr{V}(t) \tag{7 148}$$

We want to derive an expression for the rate of change of the energy with time, dE/dt First, we differentiate the kinetic energy term with respect to time:

$$\frac{d\mathscr{K}}{dt} = \sum_{i=1}^{N} \frac{d}{dt}(\tfrac{1}{2} m_i v_i^2) = \sum_{i=1}^{N} m_i v_i \frac{dv_i}{dt} \tag{7.149}$$

As $m_i\, dv_i/dt$ is equal to the force on the atom i, the result can be written:

$$\frac{d\mathscr{K}}{dt} = \sum_{i=1}^{N} v_i f_i \tag{7.150}$$

f_i is the force on atom i.

The potential energy is written as a series of pairwise interaction terms:

$$\mathscr{V}(t) = \sum_{i=1}^{N} \sum_{j=i+1}^{N} v(r_{ij}(t)) \tag{7 151}$$

The derivative of the potential energy with respect to time can be written:

$$\frac{d\mathscr{V}}{dt} = \sum_{i=1}^{N} \sum_{j=i+1}^{N} \frac{\partial \mathscr{V}}{\partial v(r_{ij})} \frac{dv(r_{ij})}{dt} \tag{7.152}$$

$\partial \mathscr{V}/\partial v(r_{ij})$ equals 1 for each pairwise combination i and j. Each term $v(r_{ij})$ is a function of the positions of atom i and j ($\mathbf{r}_i$ and $\mathbf{r}_j$) and we can then write:

$$\frac{dv(r_{ij})}{dt} = \frac{dv(r_{ij})}{d\mathbf{r}_i} \frac{d\mathbf{r}_i}{dt} + \frac{dv(r_{ij})}{d\mathbf{r}_j} \frac{d\mathbf{r}_j}{dt} \tag{7.153}$$

For a given atom i, there will be a total of $N-1$ terms of the form $v(r_{ij})$ in the expression for the potential energy due to the interactions between i and all other atoms j. Hence we can write $d\mathscr{V}/dt$ as follows:

$$\frac{d\mathscr{V}}{dt} = \sum_{i=1}^{N} \sum_{j=1,\, j\neq i}^{N} \frac{\partial v(r_{ij})}{\partial \mathbf{r}_i} \frac{d\mathbf{r}_i}{dt} = \sum_{i=1}^{N} \frac{d\mathbf{r}_i}{dt} \sum_{j=1;\, j\neq i}^{N} \frac{\partial v(r_{ij})}{\partial \mathbf{r}_i} \tag{7 154}$$

The force on atom i due to its interaction with atom j equals minus the gradient with respect to $\mathbf{r}_i$, or $-dv(r_{ij})/d\mathbf{r}_i$. Thus the total force on the atom is equal to

$$-\sum_{j=1;\, j\neq i}^{N} \frac{\partial v(r_{ij})}{\partial \mathbf{r}_i} \tag{7.155}$$

and so we have:

$$\frac{d\mathscr{V}}{dt} = -\sum_{i=1}^{N} \frac{d\mathbf{r}_i}{dt} f_i = -\sum_{i=1}^{N} v_i f_i \tag{7.156}$$

Thus $(d\mathscr{V}/dt) + (d\mathscr{K}/dt) = dE/dt = 0$, which implies that the energy is constant. In practice, the total energy fluctuates about a constant value.

Further Reading

Allen M P and D J Tildesley 1987 *Computer Simulation of Liquids*. Oxford, Oxford University Press.

Berendsen H C and W F van Gunsteren 1984. Molecular Dynamics Simulations Techniques and Approaches. In Barnes A J, W J Orville-Thomas and J Yarwood (Editors) *Molecular Liquids, Dynamics and Interactions*. NATO ASI Series C135, New York, Reidel, pp. 475–600

Berendsen H C and W F van Gunsteren 1986. Practical Algorithms for Dynamic Simulations. Molecular Dynamics Simulation of Statistical Mechanical Systems. *Proceedings of the Enrico Fermi Summer School Varenna Soc Italian di Fiscia* Bologna, pp 43–65

Brooks C L III, M Karplus and B M Pettitt 1988 Proteins. A Theoretical Perspective of Dynamics, Structure and Thermodynamics. *Advances in Chemical Physics* Volume LXXI. New York, John Wiley & Sons

Goldstein H 1980. *Classical Mechanics* (2nd Edition). Reading, MA, Addison-Wesley.

Haile J M 1992. *Molecular Dynamics Simulation Elementary Methods* New York, John Wiley & Sons

McCammon J A and S C Harvey 1987. *Dynamics of Proteins and Nucleic Acids*. Cambridge, Cambridge University Press

van Gunsteren W F 1994 Molecular Dynamics and Stochastic Dynamics Simulations: A Primer. In van Gunsteren W F, P K Weiner and A J Wilkinson (Editors) *Computer Simulations of Biomolecular Systems* Volume 2 Leiden, ESCOM

van Gunsteren W F and H J C Berendsen 1990 Computer Simulation of Molecular Dynamics: Methodology, Applications and Perspectives in Chemistry *Angewandte Chemie International Edition in English* **29**.992–1023

References

Adolf D B, D J Tildesley, M R S Pinches, J B Kingdon, T Madden and A Clark 1995. Molecular Dynamics Simulations of Dioctadecyldimethylammonium Chloride Monolayers. *Langmuir* **11**:237–246.

Alder B J and T E Wainwright 1957 Phase Transition for a Hard-sphere System. *Journal of Chemical Physics* **27**:1208–1209

Alder B J and T E Wainwright 1970. Decay of the Velocity Autocorrelation Function *Physical Review* **A1**:18–21.

Allen M P and D J Tildesley 1987 *Computer Simulation of Liquids*. Oxford, Oxford University Press

Anderson H C 1980. Molecular Dynamics Simulations at Constant Pressure and/or Temperature. *Journal of Chemical Physics* **72**:2384–2393.

Anderson H C 1983. Rattle A 'Velocity' Version of the Shake Algorithm for Molecular Dynamics Calculations. *Journal of Computational Physics* **54**:24–34.

Beeman D 1976 Some Multistep Methods for Use in Molecular Dynamics Calculations. *Journal of Computational Physics* **20** 130–139.

Berendsen H J C, J P M Postma, W F van Gunsteren, A Di Nola and J R Haak 1984 Molecular Dynamics with Coupling to an External Bath. *Journal of Chemical Physics* **81**:3684–3690

Brunger A, C B Brooks and M Karplus 1984 Stochastic Boundary Conditions for Molecular Dynamics Simulations of ST2 Water. *Chemical Physics Letters* **105**:495–500.

Dauber-Osguthorpe P and D J Osguthorpe 1990 Analysis of Intramolecular Motions by Filtering Molecular Dynamics Trajectories. *Journal of the American Chemical Society* **112**:7921–7935

Dauber-Osguthorpe P and D J Osguthorpe 1993 Partitioning the Motion in Molecular Dynamics Simulations into Characteristic Modes of Motion *Journal of Computational Chemistry* **14** 1259–1271.

De Loof H, S C Harvey, J P Segrest and R W Pastor 1991. Mean Field Stochastic Boundary Molecular Dynamics Simulation of a Phospholipid in a Membrane *Biochemistry* **30** 2099–2113

Espanol P and P B Warren 1995 Statistical Mechanics of Dissipative Particle Dynamics. *Europhysics Letters* **30**.191–196.

Essex J W, M M Hann and W G Richards 1994. Molecular Dynamics of a Hydrated Phospholipid Bilayer *Philosophical Transactions of the Royal Society of London* **B344**.239–260

Fincham D and Heyes D M 1982 Integration Algorithms in Molecular Dynamics. *CCP5 Quarterly* **6**.4–10

Gear C W 1971 *Numerical Initial Value Problems in Ordinary Differential Equations* Englewood Cliffs, NJ, Prentice Hall

Groot R D and T J Madden 1998. Dynamic Simulation of Diblock Copolymer Microphase Separation. *Journal of Chemical Physics* **108**:8713–8724

Groot R D, T J Madden and D J Tildesley 1999 On the Role of Hydrodynamic Interactions in Block Copolymer Microphase Separation. *Journal of Chemical Physics* **110** 9739–9749.

Groot R D and P B Warren 1997. Dissipative Particle Dynamics: Bridging the Gap Between Atomistic and Mesoscopic Simulation *Journal of Chemical Physics* **107** 4423–4435.

Helfand E 1984 Dynamics of Conformational Transitions in Polymers *Science* **226**:647–650.

Hockney R W 1970. The Potential Calculation and Some Applications *Methods in Computational Physics* **9** 136–211

Hoover W G 1985. Canonical Dynamics. Equilibrium Phase-space Distributions. *Physical Review* **A31**:1695–1697.

Humphreys D D, R A Friesner and B J Berne 1994. A Multiple Time-step Molecular Dynamics Algorithm for Macromolecules *Journal of Physical Chemistry* **98**:6885–6892.

Humphreys D D, R A Friesner and B J Berne 1995. Simulated Annealing of a Protein in a Continuum Solvent by Multiple Time-step Molecular Dynamics. *Journal of Physical Chemistry* **99**:10674–10685.

Humphreys, D D, R A Friesner and B J Berne 1996 A Multiple Time-step Molecular Dynamics Algorithm for Macromolecules *Journal of Physical Chemistry* **98**:6885–6892.

Jorgensen W L, R C Binning Jr and B Bigot 1981 Structures and Properties of Organic Liquids: *n*-Butane and 1,2-Dichloroethane and Their Conformational Equilibria *Journal of the American Chemical Society* **103**.4393–4399.

Kim K S, M A Moller, D J Tildesley and N Quirke 1994a. Molecular Dynamics Simulations of Langmuir-Blodgett Monolayers with Explicit Head-group Interactions *Molecular Simulation* **13**:77–99.

Kim K S, D J Tildesley and N Quirke 1994b. Molecular Dynamics of Langmuir-Blodgett Films: II. Bilayers. *Molecular Simulation* **13**.101–114

Koelman J M V A and P J Hoogerbrugge 1993 Dynamic Simulations of Hard-sphere Suspensions Under Steady Shear *Europhysics Letters* **21** 363–368.

Lavery R and H Sklenar 1988 The Definition of Generalized Helicoidal Parameters and of Axis Curvature for Irregular Nucleic Acids *Journal of Biomolecular Structure and Dynamics* **6**.63–91

Leach A R and T E Klein 1995 A Molecular Dynamics Study of the Inhibitors of Dihydrofolate Reductase by a Phenyl Triazine. *Journal of Computational Chemistry* **16**:1378–1393

Marcelja S 1973. Molecular Model for Phase Transition in Biological Membranes *Nature* **241**:451–453

Marcelja S 1974. Chain Ordering in Liquid Crystals. II. Structure of Bilayer Membranes. *Biochimica et Biophysica Acta* **367**·165–176

Nosé S 1984 A Molecular Dynamics Method for Simulations in the Canonical Ensemble. *Molecular Physics* **53**·255–268.

Pastor R W, R M Venable and M Karplus 1988. Brownian Dynamics Simulation of a Lipid Chain in a Membrane Bilayer *Journal of Chemical Physics* **89**·1112–1127.

Pearce L L and S C Harvey 1993. Langevin Dynamics Studies of Unsaturated Phospholipids in a Membrane Environment. *Biophysical Journal* **65**:1084–1092

Procacci P and B Berne 1994 Computer Simulation of Solid C_{60} Using Multiple Time-step Algorithms *Journal of Chemical Physics* **101**·2421–2431.

Rahman A 1964. Correlations in the Motion of Atoms in Liquid Argon. *Physical Review* **A136**:405–411.

Rahman A and F H Stillinger 1971. Molecular Dynamics Study of Liquid Water. *Journal of Chemical Physics* **55** 3336–3359

Robinson A J, W G Richards, P J Thomas and M M Hann 1994. Head Group and Chain Behaviour in Biological Membranes—A Molecular Dynamics Simulation *Biophysical Journal* **67**.2345–2354.

Rubinstein R Y 1981. *Simulation and Monte Carlo Methods* New York, John Wiley & Sons.

Ryckaert J P, G Cicotti and H J C Berensden 1977 Numerical Integration of the Cartesian Equations of Motion of a System with Constraints Molecular Dynamics of *n*-Alkanes *Journal of Computational Physics* **23**:327–341

Seelig A and J Seelig 1974 The Dynamics Structure of Fatty Acyl Chains in a Phospholipid Bilayer Measured by Deuterium Magnetic Resonance *Biochemistry* **13**·4839–4845

Stouch T R 1993 Lipid Membrane Structure and Dynamics Studied by All-atom Molecular Dynamics Simulations of Hydrated Phospholipid Bilayers. *Molecular Simulation* **10**·335–362.

Streett W B, D Tildesley and G Saville 1978 Multiple Time-step Methods in Molecular Dynamics *Molecular Physics* **35**:639–648.

Swindoll R D and J M Haile 1984. A Multiple Time-step Method for Molecular Dynamics Simulations of Fluids of Chain Molecules. *Journal of Compututational Physics* **53**.289–298

Swope W C, H C Anderson, P H Berens and K R Wilson 1982. A Computer Simulation Method for the Calculation of Equilibrium Constants for the Formation of Physical Clusters of Molecules: Application to Small Water Clusters *Journal of Chemical Physics* **76**.637–649

Tobias D J and C L Brooks III 1988 Molecular Dynamics with Internal Coordinate Constraints. *Journal of Chemical Physics* **89**·5115–5126

Tobias D J, K Tu and M L Klein 1997. Atomic-scale Molecular Dynamics Simulations of Lipid Membranes *Current Opinion in Colloid and Interface Science* **2**·15–26

Tuckerman M, B J Berne and G J Martyna 1992. Reversible Multiple Time Scale Molecular Dynamics. *Journal of Chemical Physics* **97**.1990–2001

van der Ploeg P and H J C Berendsen 1982. Molecular Dynamics Simulation of a Bilayer Membrane. *Journal of Chemical Physics* **76**:3271–3276

van Gunsteren W F and H J C Berendsen 1982 Algorithms for Brownian Dynamics. *Molecular Physics* **45**.637–547.

van Gunsteren W F, H J C Berendsen and J A C Rullmann 1981. Stochastic Dynamics for Molecules with Constraints. Brownian Dynamics of *n*-Alkanes. *Molecular Physics* 44:69–95.

Verlet L 1967. Computer 'Experiments' on Classical Fluids. I Thermodynamical Properties of Lennard-Jones Molecules *Physical Review* **159**:98–103

Watanabe M and M Karplus 1993 Dynamics of Molecules with Internal Degrees of Freedom by Multiple Time-step Methods. *Journal of Chemical Physics* **99**·8063–8074

Widmalm G and R W Pastor 1992. Comparison of Langevin and Molecular Dynamics Simulations. *Journal of the Chemical Society Faraday Transactions* **88**:1747–1754

Woodcock L V 1971 Isothermal Molecular Dynamics Calculations for Liquid Salts *Chemical Physics Letters* **10**:257–261.

Yun-Yu S, W Lu and W F van Gunsteren 1988 On the Approximation of Solvent Effects on the Conformation and Dynamics of Cyclosporin A by Stochastic Dynamics Simulation Techniques *Molecular Simulation* **1** 369–383

Zhou R and B J Berne 1995 A New Molecular Dynamics Method Combining the Reference System Propagator Algorithm with a Fast Multipole Method for Simulating Proteins and Other Complex Systems. *Journal of Chemical Physics* **103** 9444–9459.