

Monte Carlo Simulation Methods

8.1 Introduction

The Monte Carlo simulation method occupies a special place in the history of molecular modelling, as it was the technique used to perform the first computer simulation of a molecular system. A Monte Carlo simulation generates configurations of a system by making random changes to the positions of the species present, together with their orientations and conformations where appropriate. Many computer algorithms are said to use a 'Monte Carlo' method, meaning that some kind of random sampling is employed. In molecular simulations 'Monte Carlo' is almost always used to refer to methods that use a technique called *importance sampling*. Importance sampling methods are able to generate states of low energy, as this enables properties to be calculated accurately. We can calculate the potential energy of each configuration of the system, together with the values of other properties, from the positions of the atoms. The Monte Carlo method thus samples from a $3N$ -dimensional space of the positions of the particles. There is no momentum contribution in a Monte Carlo simulation, in contrast to a molecular dynamics simulation. How then can a Monte Carlo simulation be used to calculate thermodynamic quantities, given that phase space is $6N$ -dimensional?

To resolve this difficulty, let us return to the canonical ensemble partition, Q , which for a system of N identical particles of mass m can be written:

$$Q_{NVT} = \frac{1}{N!} \frac{1}{h^{3N}} \iint d\mathbf{p}^N d\mathbf{r}^N \exp \left[-\frac{\mathcal{H}(\mathbf{p}^N, \mathbf{r}^N)}{k_B T} \right] \quad (8.1)$$

The factor $N!$ disappears when the particles are no longer indistinguishable. $\mathcal{H}(\mathbf{p}^N, \mathbf{r}^N)$ is the Hamiltonian that corresponds to the total energy of the system. The value of the Hamiltonian depends upon the $3N$ positions and $3N$ momenta of the particles in the system (one position and one momentum for each of the three coordinates of each particle). The Hamiltonian can be written as the sum of the kinetic and potential energies of the system:

$$\mathcal{H}(\mathbf{p}^N, \mathbf{r}^N) = \sum_{i=1}^N \frac{|\mathbf{p}_i|^2}{2m} + \mathcal{V}(\mathbf{r}^N) \quad (8.2)$$

The crucial point to recognise is that the double integral in Equation (8.1) can be separated into two separate integrals, one over positions and the other over the momenta:

$$Q_{NVT} = \frac{1}{N!} \frac{1}{h^{3N}} \int d\mathbf{p}^N \exp \left[-\frac{|\mathbf{p}|^2}{2mk_B T} \right] \int d\mathbf{r}^N \exp \left[-\frac{\mathcal{V}(\mathbf{r}^N)}{k_B T} \right] \quad (8.3)$$

This separation is possible only if the potential energy function, $\psi(\mathbf{r}^N)$, is not dependent upon the velocities (this is a safe assumption for almost all potential functions in common use). The integral over the momenta can now be performed analytically, the result being:

$$\int d\mathbf{p}^N \exp\left[-\frac{|\mathbf{p}|^2}{2mk_B T}\right] = (2\pi mk_B T)^{3N/2} \quad (8.4)$$

The partition function can thus be written:

$$Q_{NVT} = \frac{1}{N!} \left(\frac{2\pi mk_B T}{h^2}\right)^{3N/2} \int d\mathbf{r}^N \exp\left(-\frac{\psi(\mathbf{r}^N)}{k_B T}\right) \quad (8.5)$$

The integral over the positions is often referred to as the *configurational integral*, Z_{NVT} :

$$Z_{NVT} = \int d\mathbf{r}^N \exp\left(-\frac{\psi(\mathbf{r}^N)}{k_B T}\right) \quad (8.6)$$

In an ideal gas there are no interactions between the particles and so the potential energy function, $\psi(\mathbf{r}^N)$, equals zero. $\exp(-\psi(\mathbf{r}^N)/k_B T)$ is therefore equal to 1 for every gas particle in the system. The integral of 1 over the coordinates of each atom is equal to the volume, and so for N ideal gas particles the configurational integral is given by V^N ($V \equiv$ volume). This leads to the following result for the canonical partition function of an ideal gas:

$$Q_{NVT} = \frac{V^N}{N!} \left(\frac{2\pi k_B T m}{h^2}\right)^{3N/2} \quad (8.7)$$

This is often written in terms of the *de Broglie thermal wavelength*, Λ :

$$Q_{NVT} = \frac{V^N}{N! \Lambda^{3N}} \quad (8.8)$$

where $\Lambda = \sqrt{h^2/2\pi k_B T m}$.

By combining Equations (8.4) and (8.6) we can see that the partition function for a 'real' system has a contribution due to ideal gas behaviour (the momenta) and a contribution due to the interactions between the particles. Any deviations from ideal gas behaviour are due to interactions within the system as a consequence of these interactions. This enables us to write the partition function as:

$$Q_{NVT} = Q_{NVT}^{\text{ideal}} Q_{NVT}^{\text{excess}} \quad (8.9)$$

The excess part of the partition function is given by:

$$Q_{NVT}^{\text{excess}} = \frac{1}{V^N} \int d\mathbf{r}^N \exp\left[-\frac{\psi(\mathbf{r}^N)}{k_B T}\right] \quad (8.10)$$

A consequence of writing the partition function as a product of a real gas and an ideal gas part is that thermodynamic properties can be written in terms of an ideal gas value and an excess value. The ideal gas contributions can be determined analytically by integrating over the momenta. For example, the Helmholtz free energy is related to the canonical partition function by:

$$A = -k_B T \ln Q_{NVT} \quad (8.11)$$

Writing the partition function as the product, Equation (8.9), leads to:

$$A = A^{\text{ideal}} + A^{\text{excess}} \quad (8.12)$$

The important conclusion is that all of the deviations from ideal gas behaviour are due to the presence of interactions between the atoms in the system, as calculated using the potential energy function. This energy function is dependent only upon the positions of the atoms and not their momenta, and so a Monte Carlo simulation is able to calculate the excess contributions that give rise to deviations from ideal gas behaviour.

8.2 Calculating Properties by Integration

Having established that we can indeed explore configurational phase space and derive useful thermodynamic properties, let us consider how we might achieve this in practice. For example, the average potential energy can, in principle at least, be determined by evaluating the integral:

$$\langle \psi(\mathbf{r}^N) \rangle = \int d\mathbf{r}^N \psi(\mathbf{r}^N) \rho(\mathbf{r}^N) \quad (8.13)$$

This is a multidimensional integral over the $3N$ degrees of freedom of the N particles in the system. $\rho(\mathbf{r}^N)$ is the probability of obtaining the configuration \mathbf{r}^N and is given by

$$\rho(\mathbf{r}^N) = \frac{\exp[-\psi(\mathbf{r}^N)/k_B T]}{Z} \quad (8.14)$$

The denominator, Z , is the configurational integral (Equation (8.6)). For the potential functions commonly used in molecular modelling, it is not possible to evaluate these integrals analytically. However, we could attempt to obtain values for the integrals using numerical methods. One simple numerical integration method is the trapezium rule. This approximates the integral as a series of trapeziums between the two limits, as illustrated for a one-dimensional problem in Figure 8.1. In this case we have divided the integral into ten trapeziums, which requires eleven function evaluations. Simpson's rule involves a similar procedure and may provide a more accurate value of the integral [Stephenson 1973]. For a function of two variables ($f(x, y)$), it is necessary to square the number of function

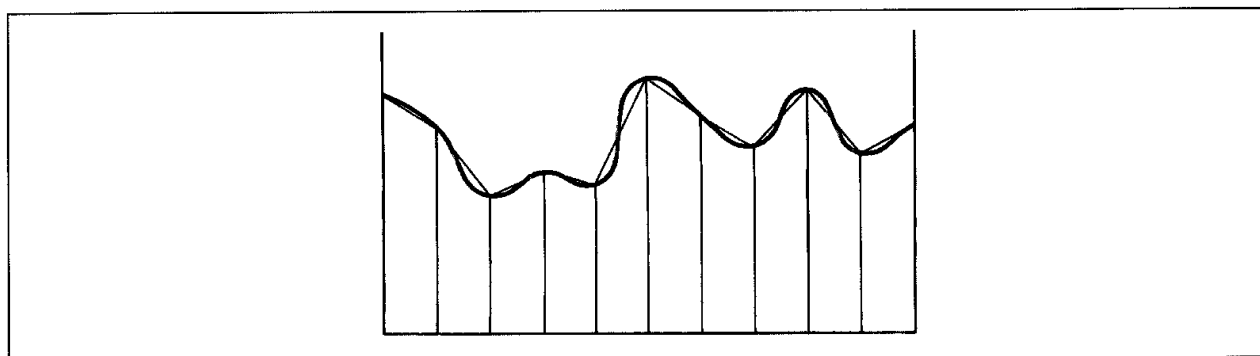


Fig 8.1 Evaluation of a one-dimensional integral using the trapezium rule. The area under the curve is approximated as the sum of the areas of the trapeziums

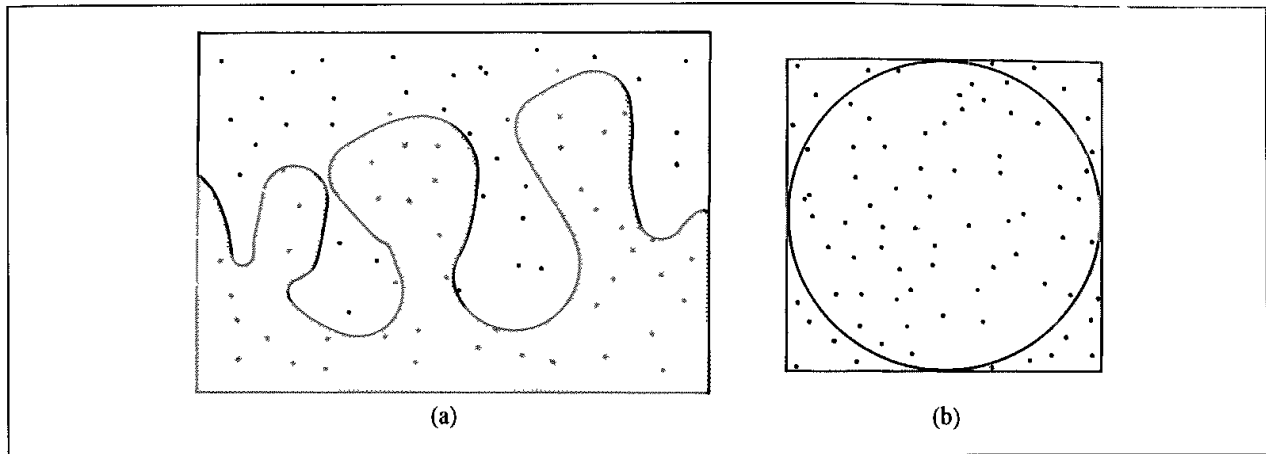


Fig. 8.2 Simple Monte Carlo integration (a) The shaded area under the irregular curve equals the ratio of the number of random points under the curve to the total number of points, multiplied by the area of the bounding area (b) An estimate of π can be obtained by generating random numbers within the square. π then equals the number of points within the circle divided by the total number of points within the square, multiplied by 4

evaluations required. For a $3N$ -dimensional integral the total number of function evaluations required to determine the integral would be m^{3N} , where m is the number of points needed to determine the integral in each dimension. This number is enormous even for very small numbers of particles. For example, with just 50 particles and three points per dimension, a total of 3^{150} ($\sim 10^{71}$) evaluations would be required. Integration using the trapezium rule or Simpson's rule is clearly not a feasible approach.

We could consider a random method as a possible alternative. The general principle can be illustrated using the function shown in Figure 8.2. To determine the area under the curve in Figure 8.2 a series of random points would be generated within the bounding area. The area under the curve is then calculated by multiplying the bounding area A by the ratio of the number of trial points that lie under the curve to the total number of points generated. An estimate of π can be determined in this way, as illustrated in Figure 8.2.

To calculate the partition function for a system of N atoms using this simple Monte Carlo integration method would involve the following steps:

1. Obtain a configuration of the system by randomly generating $3N$ Cartesian coordinates, which are assigned to the particles.
2. Calculate the potential energy of the configuration, $\mathcal{V}(\mathbf{r}^N)$.
3. From the potential energy, calculate the Boltzmann factor, $\exp(-\mathcal{V}(\mathbf{r}^N)/k_B T)$.
4. Add the Boltzmann factor to the accumulated sum of Boltzmann factors and the potential energy contribution to its accumulated sum and return to step 1.
5. After a number, N_{trial} , of iterations, the mean value of the potential energy would be calculated using:

$$\langle \mathcal{V}(\mathbf{r}^N) \rangle = \frac{\sum_{i=1}^{N_{\text{trial}}} \mathcal{V}_i(\mathbf{r}^N) \exp[-\mathcal{V}_i(\mathbf{r}^N)/k_B T]}{\sum_{i=1}^{N_{\text{trial}}} \exp[-\mathcal{V}_i(\mathbf{r}^N)/k_B T]} \quad (8.15)$$

Unfortunately, this is not a feasible approach for calculating thermodynamic properties due to the large number of configurations that have extremely small (effectively zero) Boltzmann

factors caused by high-energy overlaps between the particles. This reflects the nature of the phase space, most of which corresponds to non-physical configurations with very high energies. Only a very small proportion of the phase space corresponds to low-energy configurations where there are no overlapping particles and where the Boltzmann factor has an appreciable value. These low-energy regions coincide with the physically observed phases such as solid, liquid, etc

One way around this impasse is to generate configurations that make a large contribution to the integral (8.15), which is the strategy adopted in importance sampling and which is the essence of the method described by Metropolis, Rosenbluth, Rosenbluth, Teller and Teller in 1953 [Metropolis *et al.* 1953]. For many thermodynamic properties of a molecular system, those states with a high probability ρ are also the ones that make a significant contribution to the integral (there are some notable exceptions to this, such as the free energy). The Metropolis method has become so widely adopted that in the simulation and molecular modelling communities it is usually referred to as ‘the Monte Carlo method’. Fortunately, there is rarely any confusion with the simple Monte Carlo methods. The crucial feature of the Metropolis approach is that it biases the generation of configurations towards those that make the most significant contribution to the integral. Specifically, it generates states with a probability $\exp(-\mathcal{V}(\mathbf{r}^N)/k_B T)$ and then counts each of them equally. By contrast, the simple Monte Carlo integration method generates states with equal probability (both high- and low-energy) and then assigns them a weight $\exp(-\mathcal{V}(\mathbf{r}^N)/k_B T)$.

8.3 Some Theoretical Background to the Metropolis Method

The Metropolis algorithm generates a *Markov chain* of states. A Markov chain satisfies the following two conditions:

1. The outcome of each trial depends only upon the preceding trial and not upon any previous trials.
2. Each trial belongs to a finite set of possible outcomes.

Condition (1) provides a clear distinction between the molecular dynamics and Monte Carlo methods, for in a molecular dynamics simulation all of the states are connected in time. Suppose the system is in a state m . We denote the probability of moving to state n as π_{mn} . The various π_{mn} can be considered to constitute an $N \times N$ matrix π (the transition matrix), where N is the number of possible states. Each row of the transition matrix sums to 1 (i.e. the sum of the probabilities π_{mn} for a given m equals 1). The probability that the system is in a particular state is represented by a probability vector $\boldsymbol{\rho}$:

$$\boldsymbol{\rho} = (\rho_1, \rho_2, \dots, \rho_m, \rho_n, \dots, \rho_N) \quad (8.16)$$

Thus ρ_1 is the probability that the system is in state 1 and ρ_m the probability that the system is in state m . If $\boldsymbol{\rho}(1)$ represents the initial (randomly chosen) configuration, then the probability of the second state is given by:

$$\boldsymbol{\rho}(2) = \boldsymbol{\rho}(1)\boldsymbol{\pi} \quad (8.17)$$

The probability of the third state is:

$$\rho(3) = \rho(2)\pi = \rho(1)\pi\pi \quad (8.18)$$

The equilibrium distribution of the system can be determined by considering the result of applying the transition matrix an infinite number of times. This limiting distribution of the Markov chain is given by $\rho_{\text{limit}} = \lim_{N \rightarrow \infty} \rho(1)\pi^N$.

One feature of the limiting distribution is that it is independent of the initial guess $\rho(1)$. The limiting or equilibrium distribution for a molecular or atomic system is one in which the probabilities of each state are proportional to the Boltzmann factor. We can illustrate the use of the probability distribution and the transition matrix by considering a two-level system in which the energy levels are such that the ratio of the Boltzmann factors is 2:1. The expected limiting distribution thus corresponds to a configuration vector $(\frac{2}{3}, \frac{1}{3})$. The following transition matrix enables the limiting distribution to be achieved.

$$\pi = \begin{pmatrix} 0.5 & 0.5 \\ 1 & 0 \end{pmatrix} \quad (8.19)$$

We can illustrate the use of this transition matrix as follows. Suppose the initial probability vector is $(1, 0)$ and so the system starts with a 100% probability of being in state 1 and no probability of being in state 2. Then the second state is given by:

$$\rho(2) = (1 \ 0) \begin{pmatrix} 0.5 & 0.5 \\ 1 & 0 \end{pmatrix} = (0.5 \ 0.5) \quad (8.20)$$

The third state is $\rho(3) = (0.75, 0.25)$. Successive applications of the transition matrix give the limiting distribution $(2/3, 1/3)$.

When the limiting distribution is reached then application of the transition matrix must return the same distribution back:

$$\rho_{\text{limit}} = \rho_{\text{limit}}\pi \quad (8.21)$$

Thus, if an ensemble can be prepared that is at equilibrium, then one Metropolis Monte Carlo step should return an ensemble that is still at equilibrium. A consequence of this is that the elements of the probability vector for the limiting distribution must satisfy:

$$\sum_m \rho_m \pi_{mn} = \rho_n \quad (8.22)$$

This can be seen to hold for our simple two-level example:

$$(2/3 \ 1/3) \begin{pmatrix} 1/2 & 1/2 \\ 1 & 0 \end{pmatrix} = (2/3 \ 1/3) \quad (8.23)$$

We will henceforth use the symbol ρ to refer to the limiting distribution

Closely related to the transition matrix is the *stochastic matrix*, whose elements are labelled α_{mn} . This matrix gives the probability of choosing the two states m and n between which the move is to be made. It is often known as the *underlying matrix* of the Markov chain. If the probability of accepting a trial move from m to n is p_{mn} then the probability of making a transition from m to n (π_{mn}) is given by multiplying the probability of choosing states m

and n (α_{mn}) by the probability of accepting the trial move (p_{mn}):

$$\pi_{mn} = \alpha_{mn} p_{mn} \quad (8.24)$$

It is often assumed that the stochastic matrix \mathbf{a} is symmetrical (i.e. the probability of choosing the states m and n is the same whether the move is made from m to n or from n to m). If the probability of state n is greater than that of state m in the limiting distribution (i.e. if the Boltzmann factor of n is greater than that of m because the energy of n is lower than the energy of m) then in the Metropolis recipe, the transition matrix element π_{mn} for progressing from m to n equals the probability of selecting the two states in the first place (i.e. $\pi_{mn} = \alpha_{mn}$ if $\rho_n \geq \rho_m$). If the Boltzmann weight of the state n is less than that of state m , then the probability of permitting the transition is given by multiplying the stochastic matrix element α_{mn} by the ratio of the probabilities of the state n to the previous state m . This can be written:

$$\pi_{mn} = \alpha_{mn} \quad (\rho_n \geq \rho_m) \quad (8.25)$$

$$\pi_{mn} = \alpha_{mn} (\rho_n / \rho_m) \quad (\rho_n < \rho_m) \quad (8.26)$$

These two conditions apply if the initial and final states m and n are different. If m and n are the same state, then the transition matrix element is calculated from the fact that the rows of the stochastic matrix sum to 1:

$$\pi_{mm} = 1 - \sum_{m \neq n} \pi_{mn} \quad (8.27)$$

Let us now try to reconcile the Metropolis algorithm as outlined in Section 6.1.3 with the more formal approach that we have just developed. We recall that in the Metropolis method a new configuration n is accepted if its energy is lower than the original state m . If the energy is higher, however, then we would like to choose the move with a probability according to Equation (8.24). This is achieved by comparing the Boltzmann factor $\exp(-\Delta\mathcal{V}(\mathbf{r}^N)/k_B T)$ ($\Delta\mathcal{V}(\mathbf{r}^N) = [\mathcal{V}(\mathbf{r}^N)_n - \mathcal{V}(\mathbf{r}^N)_m]$) to a random number between 0 and 1. If the Boltzmann factor is greater than the random number then the new state is accepted. If it is smaller then the new state is rejected. Thus if the energy of the new state (n) is very close to that of the old state (m) then the Boltzmann factor of the energy difference will be very close to 1, and so the move is likely to be accepted. If the energy difference is very large, however, then the Boltzmann factor will be close to zero and the move is unlikely to be accepted.

The Metropolis method is derived by imposing the condition of microscopic reversibility: at equilibrium the transition between two states occurs at the same rate. The rate of transition from a state m to a state n equals the product of the population (ρ_m) and the appropriate element of the transition matrix (π_{mn}). Thus, at equilibrium we can write:

$$\pi_{mn} \rho_m = \pi_{nm} \rho_n \quad (8.28)$$

The ratio of the transition matrix elements thus equals the ratio of the Boltzmann factors of the two states:

$$\frac{\pi_{mn}}{\pi_{nm}} = \exp[-(\mathcal{V}(\mathbf{r}^N)_n - \mathcal{V}(\mathbf{r}^N)_m)/k_B T] \quad (8.29)$$

8.4 Implementation of the Metropolis Monte Carlo Method

A Monte Carlo program to simulate an atomic fluid is quite simple to construct. At each iteration of the simulation a new configuration is generated. This is usually done by making a random change to the Cartesian coordinates of a single randomly chosen particle using a random number generator. If the random number generator produces numbers (ξ) in the range 0 to 1, moves in both positive and negative directions are possible if the coordinates are changed as follows:

$$x_{\text{new}} = x_{\text{old}} + (2\xi - 1)\delta r_{\text{max}} \quad (8.30)$$

$$y_{\text{new}} = y_{\text{old}} + (2\xi - 1)\delta r_{\text{max}} \quad (8.31)$$

$$z_{\text{new}} = z_{\text{old}} + (2\xi - 1)\delta r_{\text{max}} \quad (8.32)$$

A unique random number is generated for each of the three directions x , y and z . δr_{max} is the maximum possible displacement in any direction. The energy of the new configuration is then calculated; this need not require a complete recalculation of the energy of the entire system but only those contributions involving the particle that has just been moved. As a consequence, the neighbour list used by a Monte Carlo simulation must contain *all* the neighbours of each atom, because it is necessary to identify all the atoms which interact with the moving atom (recall that in molecular dynamics the neighbour list for each atom contains only neighbours with a higher index). Proper account should be taken of periodic boundary conditions and the minimum image convention when generating new configurations and calculating their energies. If the new configuration is lower in energy than its predecessor then the new configuration is retained as the starting point for the next iteration. If the new configuration is higher in energy than its predecessor then the Boltzmann factor, $\exp(-\Delta\mathcal{V}/k_{\text{B}}T)$, is compared to a random number between 0 and 1. If the Boltzmann factor is greater than the random number then the new configuration is accepted; if not then it is rejected and the initial configuration is retained for the next move. This acceptance condition can be written in the following concise fashion:

$$\text{rand}(0, 1) \leq \exp(-\Delta\mathcal{V}(\mathbf{r}^N)/k_{\text{B}}T) \quad (8.33)$$

The size of the move at each iteration is governed by the maximum displacement, δr_{max} . This is an adjustable parameter whose value is usually chosen so that approximately 50% of the trial moves are accepted. If the maximum displacement is too small then many moves will be accepted but the states will be very similar and the phase space will only be explored very slowly. Too large a value of δr_{max} and many trial moves will be rejected because they lead to unfavourable overlaps. The maximum displacement can be adjusted automatically while the program is running to achieve the desired acceptance ratio by keeping a running score of the proportion of moves that are accepted. Every so often the maximum displacement is then scaled by a few percent: if too many moves have been accepted then the maximum displacement is increased; too few and δr_{max} is reduced.

As an alternative to the random selection of particles it is possible to move the atoms sequentially (this requires one fewer call to the random number generator per iteration). Alternatively, several atoms can be moved at once; if an appropriate value for the maximum displacement is chosen then this may enable phase space to be covered more efficiently.

As with a molecular dynamics simulation, a Monte Carlo simulation comprises an equilibration phase followed by a production phase. During equilibration, appropriate thermodynamic and structural quantities such as the total energy (and the partitioning of the energy among the various components), mean square displacement and order parameters (as appropriate) are monitored until they achieve stable values, whereupon the production phase can commence. In a Monte Carlo simulation from the canonical ensemble, the temperature and volume are, of course, fixed. In a constant pressure simulation the volume will change and should therefore also be monitored to ensure that a stable system density is achieved.

8.4.1 Random Number Generators

The random number generator at the heart of every Monte Carlo simulation program is accessed a very large number of times, not only to generate new configurations but also to decide whether a given move should be accepted or not. Random number generators are also used in other modelling applications; for example, in a molecular dynamics simulation the initial velocities are normally assigned using a random number generator. The numbers produced by a random number generator are not, in fact, truly random; the same sequence of numbers should always be generated when the program is run with the same initial conditions (if not, then a serious error in the hardware or software must be suspected!). The sequences of numbers are thus often referred to as 'pseudo-random' numbers as they possess the statistical properties of 'true' sequences of random numbers. Most random number generators are designed to generate different sequences of numbers if a different 'seed' is provided. In this way, several independent runs can be performed using different seeds. One simple strategy is to use the time and/or date as the seed; this is information that can often be obtained automatically by the program from the computer's operating system.

The numbers produced by a random number generator should satisfy certain statistical properties. This requirement usually supersedes the need for a computationally very fast algorithm as other parts of a Monte Carlo simulation take much more time (such as calculating the change in energy). One useful and simple test of a random number generator is to break a sequence of random numbers into blocks of k numbers, which are taken to be coordinates in a k -dimensional space. A good random number should give a random distribution of points. Many of the common generators do not satisfy this test because the points lie on a plane or because they show clear correlations [Sharp and Bays 1992].

The *linear congruential* method is widely used for generating random numbers. Each number in the sequence is generated by taking the previous number, multiplying by a constant (the multiplier, a), adding a second constant (the increment, b), and taking the remainder when dividing by a third constant (the modulus, m). The first value is the seed, supplied by the user. Thus:

$$\xi[1] = \text{seed} \quad (8.34)$$

$$\xi[i] = \text{MOD}\{(\xi[i-1] \times a + b), m\} \quad (8.35)$$

The MOD function returns the remainder when the first argument is divided by the second (for example, MOD(14,5) equals 4). If the constants are chosen carefully, the linear congruential method generates all possible integers between 0 and $m - 1$, and the period (i.e. the number of iterations before the sequence starts to repeat itself) will be equal to

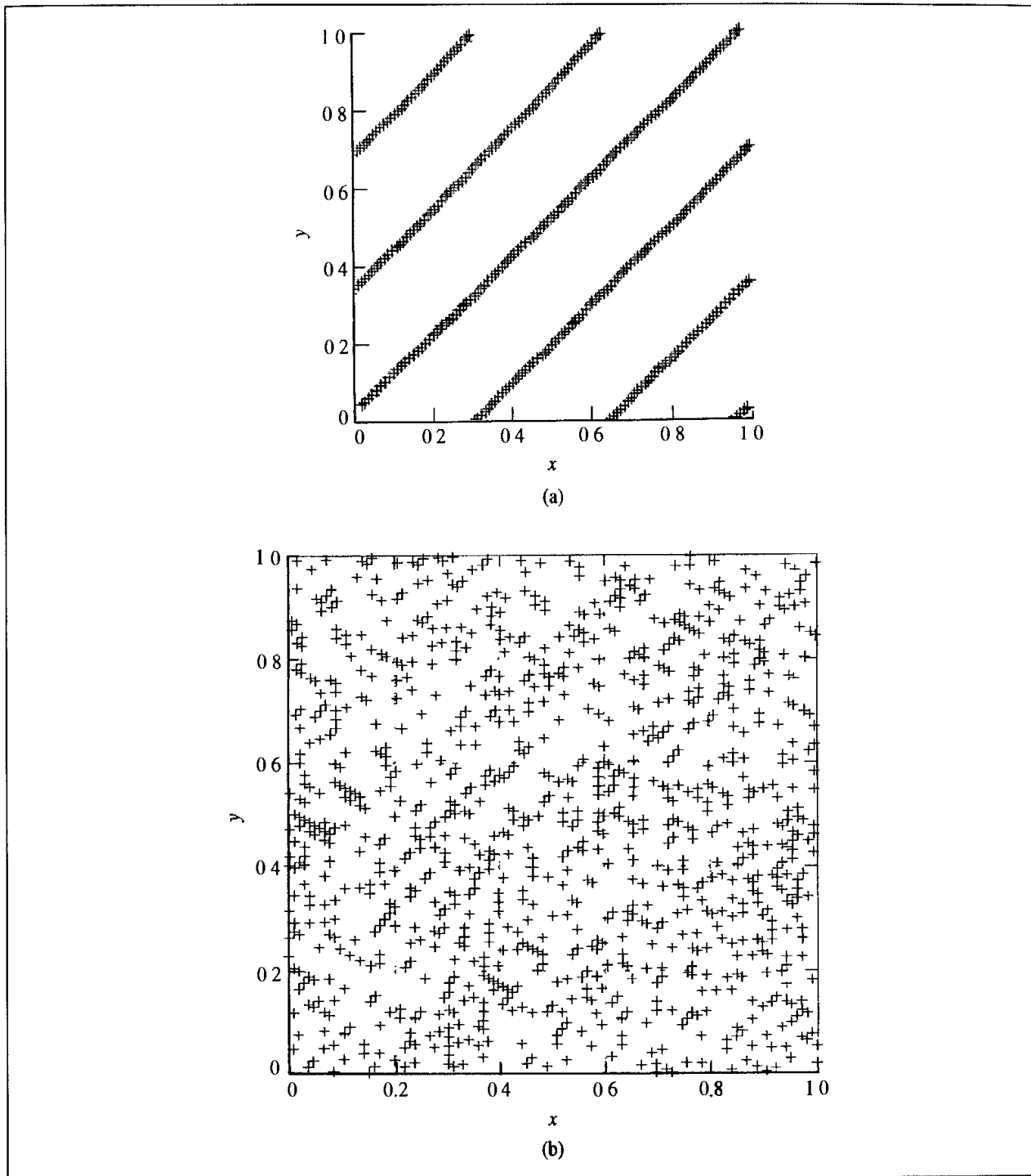


Fig. 8.3 Two 'random' distributions obtained by plotting pairs of values from a linear congruential random generator. The distribution (a) was obtained using $m = 32\,769$, $a = 10\,924$, $b = 11\,830$. The distribution (b) was obtained using $m = 6075$, $a = 106$, $b = 1283$. Data from [Sharp and Bays 1992]

the modulus. The period cannot of course be greater than m . The linear congruential method generates integral values, which can be converted to real numbers between 0 and 1 by dividing by m . The modulus is often chosen to be the largest prime number that can be represented in a given number of bits (usually chosen to be the number of bits per word; $2^{31} - 1$ is thus a common choice on a 32-bit machine).

Although popular, by virtue of the ease with which it can be programmed, the linear congruential method does not satisfy all of the requirements that are now regarded as important in a random number generator. For example, the points obtained from a linear congruential generator lie on $(k - 1)$ -dimensional planes rather than uniformly filling up the space. Indeed, if the constants a , b and m are chosen inappropriately then the linear congruential method can give truly terrible results, as shown in Figure 8.3. One random number generator that is claimed to perform well in all of the standard tests is that of G Marsaglia, which is described in Appendix 8.1.

8.5 Monte Carlo Simulation of Molecules

The Monte Carlo method is most easily implemented for atomic systems because it is only necessary to consider the translational degrees of freedom. The algorithm is easy to implement and accurate results can be obtained from relatively short simulations of a few tens of thousands of steps. There can be practical problems in applying the method to molecular systems, and especially to molecules which have a significant degree of conformational flexibility. This is because, in such systems, it is necessary to permit the internal degrees of freedom to vary. Unfortunately, such changes often lead to high-energy overlaps either within the molecule or between the molecule and its neighbours and thus a high rejection rate.

8.5.1 Rigid Molecules

For rigid, non-spherical molecules, the orientations of the molecules must be varied as well as their positions in space. It is usual to translate and rotate one molecule during each Monte Carlo step. Translations are usually described in terms of the position of the centre of mass. There are various ways to generate a new orientation of a molecule. The simplest approach is to choose one of the three Cartesian axes (x , y or z) and to rotate about the chosen axis by a randomly chosen angle $\delta\omega$, chosen to lie within the maximum angle variation, $\delta\omega_{\max}$ [Barker and Watts 1969]. The rotation is achieved by applying routine trigonometric relationships. For example, if the vector $(x\mathbf{i}, y\mathbf{j}, z\mathbf{k})$ describes the orientation of a molecule then the new vector $(x'\mathbf{i}, y'\mathbf{j}, z'\mathbf{k})$ that corresponds to rotation by $\delta\omega$ about the x axis is calculated as follows:

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \delta\omega & \sin \delta\omega \\ 0 & -\sin \delta\omega & \cos \delta\omega \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} \quad (8.36)$$

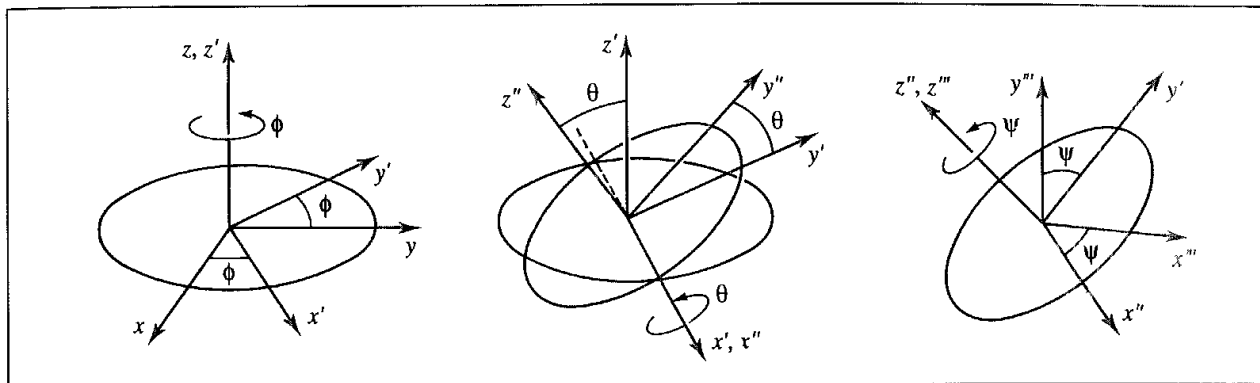


Fig. 8.4: The Euler angles ϕ , θ and ψ

The Euler angles are often used to describe the orientations of a molecule. There are three Euler angles; ϕ , θ and ψ . ϕ is a rotation about the Cartesian z axis; this has the effect of moving the x and y axes. θ is a rotation about the new x axis. Finally, ψ is a rotation about the new z axis (Figure 8.4). If the Euler angles are randomly changed by small amounts $\delta\phi$, $\delta\theta$ and $\delta\psi$ then a vector \mathbf{v}_{old} is moved according to the following matrix equation:

$$\mathbf{v}_{new} = \mathbf{A}\mathbf{v}_{old} \tag{8.37}$$

where the matrix \mathbf{A} is

$$\begin{pmatrix} \cos \delta\phi \cos \delta\psi - \sin \delta\phi \cos \delta\theta \sin \delta\psi & \sin \delta\phi \cos \delta\psi + \cos \delta\phi \cos \delta\theta \sin \delta\psi & \sin \delta\theta \sin \delta\psi \\ -\cos \delta\phi \sin \delta\psi - \sin \delta\phi \cos \delta\theta \cos \delta\psi & -\sin \delta\phi \sin \delta\psi + \cos \delta\phi \cos \delta\theta \cos \delta\psi & \sin \delta\theta \cos \delta\psi \\ \sin \delta\phi \sin \delta\theta & -\cos \delta\phi \sin \delta\theta & \cos \delta\theta \end{pmatrix} \tag{8.38}$$

It is important to note that simply sampling displacements of the three Euler angles does not lead to a uniform distribution; it is necessary to sample from $\cos \theta$ rather than θ (Figure 8.5).

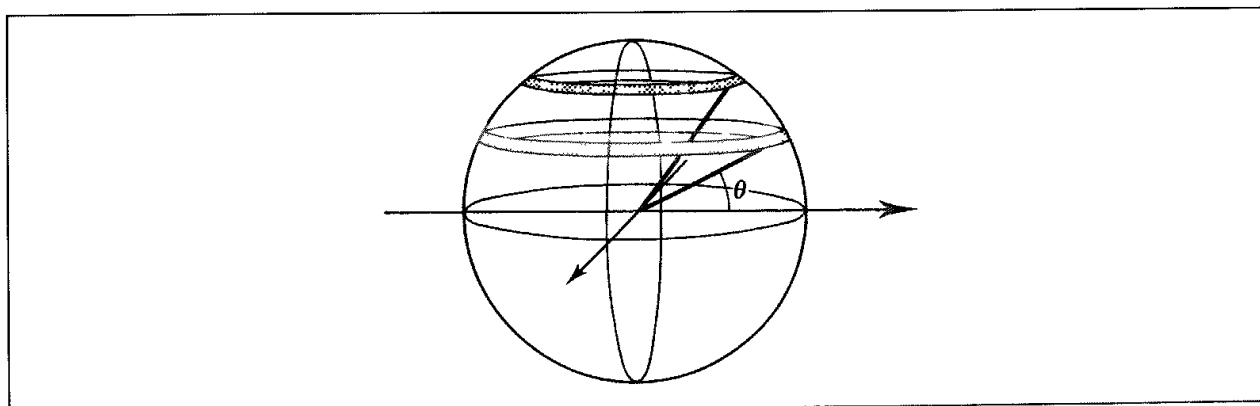


Fig. 8.5: To achieve a uniform distribution of points over the surface of a sphere it is necessary to sample from $\cos \theta$ rather than θ . If the sampling is uniform in θ then the number of points per unit area increases with θ , leading to an uneven distribution over the sphere

The preferred approach is to sample directly in $\cos \theta$ as follows:

$$\phi_{\text{new}} = \phi_{\text{old}} + 2(\xi - 1)\delta\phi_{\text{max}} \quad (8.39)$$

$$\cos \theta_{\text{new}} = \cos \theta_{\text{old}} + (2\xi - 1)\delta(\cos \theta)_{\text{max}} \quad (8.40)$$

$$\psi_{\text{new}} = \psi_{\text{old}} + 2(\xi - 1)\delta\psi_{\text{max}} \quad (8.41)$$

The alternative is to sample in θ and to modify the acceptance or rejection criteria as follows:

$$\theta_{\text{new}} = \theta_{\text{old}} + (2\xi - 1)\delta\theta_{\text{max}} \quad (8.42)$$

$$\frac{\rho_{\text{new}}}{\rho_{\text{old}}} = \exp(-\Delta\mathcal{V}/k_{\text{B}}T) \frac{\sin \theta_{\text{new}}}{\sin \theta_{\text{old}}} \quad (8.43)$$

This second approach may give problems if θ_{old} equals zero.

A disadvantage of the Euler angle approach is that the rotation matrix contains a total of six trigonometric functions (sine and cosine for each of the three Euler angles). These trigonometric functions are computationally expensive to calculate. An alternative is to use *quaternions*. A quaternion is a four-dimensional vector such that its components sum to 1: $q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1$. The quaternion components are related to the Euler angles as follows:

$$q_0 = \cos \frac{1}{2}\theta \cos \frac{1}{2}(\phi + \psi) \quad (8.44)$$

$$q_1 = \sin \frac{1}{2}\theta \cos \frac{1}{2}(\phi + \psi) \quad (8.45)$$

$$q_2 = \sin \frac{1}{2}\theta \sin \frac{1}{2}(\phi + \psi) \quad (8.46)$$

$$q_3 = \cos \frac{1}{2}\theta \sin \frac{1}{2}(\phi + \psi) \quad (8.47)$$

The Euler angle rotation matrix can then be written

$$\mathbf{A} = \begin{pmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 + q_0q_3) & 2(q_1q_3 - q_0q_2) \\ 2(q_1q_2 - q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 + q_0q_1) \\ 2(q_1q_3 + q_0q_2) & 2(q_2q_3 - q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{pmatrix} \quad (8.48)$$

To generate a new orientation, it is necessary to rotate the quaternion vector to a new (random) orientation. As it is a four-dimensional vector, the orientation must be performed in four-dimensional space. This can be achieved as follows [Vesely 1982]:

1. Generate pairs of random numbers (ξ_1, ξ_2) between -1 and 1 until $S_1 = \xi_1^2 + \xi_2^2 < 1$.
2. Do the same for pairs ξ_3 and ξ_4 until $S_2 = \xi_3^2 + \xi_4^2 < 1$.
3. Form the random unit four-dimensional vector $(\xi_1, \xi_2, \xi_3\sqrt{(1-S_1)/S_2}, \xi_4\sqrt{(1-S_1)/S_2})$.

To achieve an appropriate acceptance rate the angle between the two vectors that describe the new and old orientations should be less than some value; this corresponds to sampling randomly and uniformly from a region on the surface of a sphere.

The introduction of an orientational component as well as a translational component increases the number of maximum displacement parameters that determine the acceptance ratio. It is important to check that the desired acceptance ratio is achieved, and also that an appropriate proportion of orientational and translational moves are made. Trial and error is often the most effective way to find the best combination of parameters.

8.5.2 Monte Carlo Simulations of Flexible Molecules

Monte Carlo simulations of flexible molecules are often difficult to perform successfully unless the system is small, or some of the internal degrees of freedom are frozen out, or special models or methods are employed. The simplest way to generate a new configuration of a flexible molecule is to perform random changes to the Cartesian coordinates of individual atoms, in addition to translations and rotations of the entire molecule. Unfortunately, it is often found that very small atomic displacements are required to achieve an acceptable acceptance ratio, which means that the phase space is covered very slowly. For example, even small movements away from an equilibrium bond length will cause a large increase in the energy. One obvious tactic is to freeze out some of the internal degrees of freedom, usually the 'hard' degrees of freedom such as the bond lengths and the bond angles. Such algorithms have been extensively used to investigate small molecules such as butane. However, for large molecules, even relatively small bond rotations may cause large movements of atoms down the chain. This invariably leads to high-energy configurations as illustrated in Figure 8.6. The rigid bond and rigid angle approximation must be used with care, for freezing out some of the internal degrees of freedom can affect the distributions of other internal degrees of freedom.

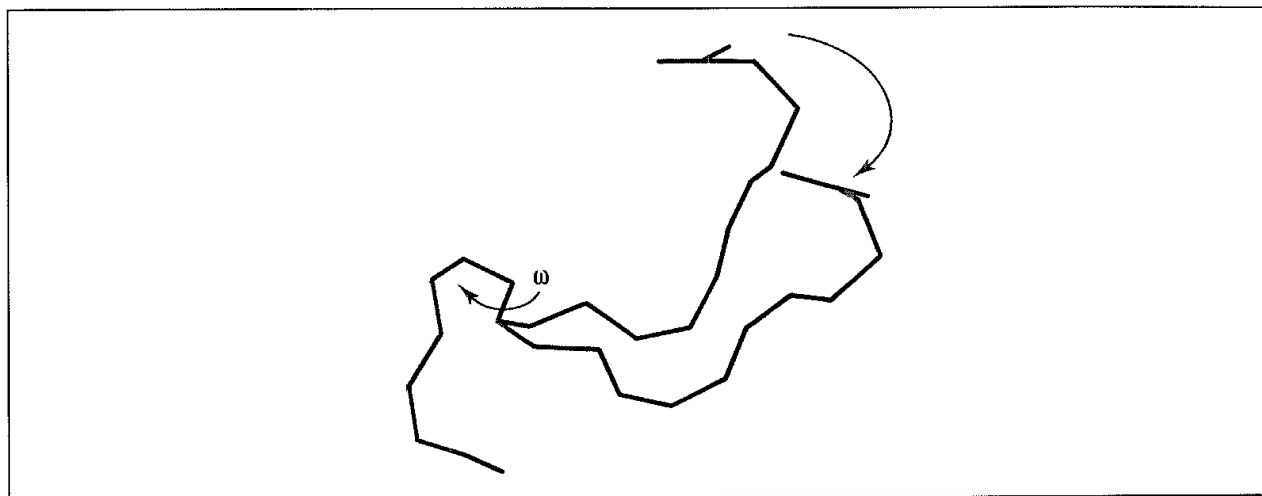


Fig 8.6 A bond rotation in the middle of a molecule may lead to a large movement at the end

8.6 Models Used in Monte Carlo Simulations of Polymers

A polymer is a macromolecule that is constructed by chemically linking together a sequence of molecular fragments. In simple synthetic polymers such as polyethylene or polystyrene all of the molecular fragments comprise the same basic unit (or monomer). Other polymers contain mixtures of monomers. Proteins, for example, are polypeptide chains in which each unit is one of the twenty amino acids. Cross-linking between different chains gives rise to yet further variations in the constitution and structure of a polymer. All of these features may affect the overall properties of the molecule, sometimes in a dramatic way. Moreover, one

may be interested in the properties of the polymer under different conditions, such as in solution, in a polymer melt or in the crystalline state. Molecular modelling can help to develop theories for understanding the properties of polymers and can also be used to predict their properties.

A wide range of time and length scales are needed to completely describe a polymer's behaviour. The timescale ranges from approximately 10^{-14} s (i.e. the period of a bond vibration) through to seconds, hours or even longer for collective phenomena. The size scale ranges from the 1–2 Å of chemical bonds to the diameter of a coiled polymer, which can be several hundreds of ångstroms. Many kinds of model have been used to represent and simulate polymeric systems and predict their properties. Some of these models are based upon very simple ideas about the nature of the intra- and intermolecular interactions within the system but have nevertheless proved to be extremely useful. One famous example is Flory's rotational isomeric state model [Flory 1969]. Increasing computer performance now makes it possible to use techniques such as molecular dynamics and Monte Carlo simulations to study polymer systems.

Most simulations on polymers are performed using empirical energy models (though with faster computers and new methods it is becoming possible to apply quantum mechanics to larger and larger systems). Moreover, there are various ways in which the configurational and conformational degrees of freedom may be restricted so as to produce a computationally more efficient model. The simplest models use a lattice representation in which the polymer is constructed from connected interaction centres, which are required to occupy the vertices of a lattice. At the next level of complexity are the bead models, where the polymer is composed of a sequence of connected 'beads'. Each bead represents an 'effective monomer' and interacts with the other beads to which it is bonded and also with other nearby beads. The ultimate level of detail is achieved with the atomistic models, in which each non-hydrogen atom is explicitly represented (and sometimes all of the hydrogens as well). Our aim here is to give a flavour of the way in which Monte Carlo methods can be used to investigate polymeric systems. We divide the discussion into lattice and continuum models but recognise that there is a spectrum of models from the simplest to the most complex.

8.6.1 Lattice Models of Polymers

Lattice models have provided many insights into the behaviour of polymers despite the obvious approximations involved. The simplicity of a lattice model means that many states can be generated and examined very rapidly. Both two-dimensional and three-dimensional lattices are used. The simplest models use cubic or tetrahedral lattices in which successive monomers occupy adjacent lattice points (Figure 8.7). The energy models are usually very simple, in part to reflect the simplicity of the representation but also to permit the rapid calculation of the energy.

More complex models have been developed in which the lattice representation is closer to the 'true' geometry of the molecule. For example, in Figure 8.8 we show the bond fluctuation model of polyethylene, in which the 'bond' between successive monomers on the lattice

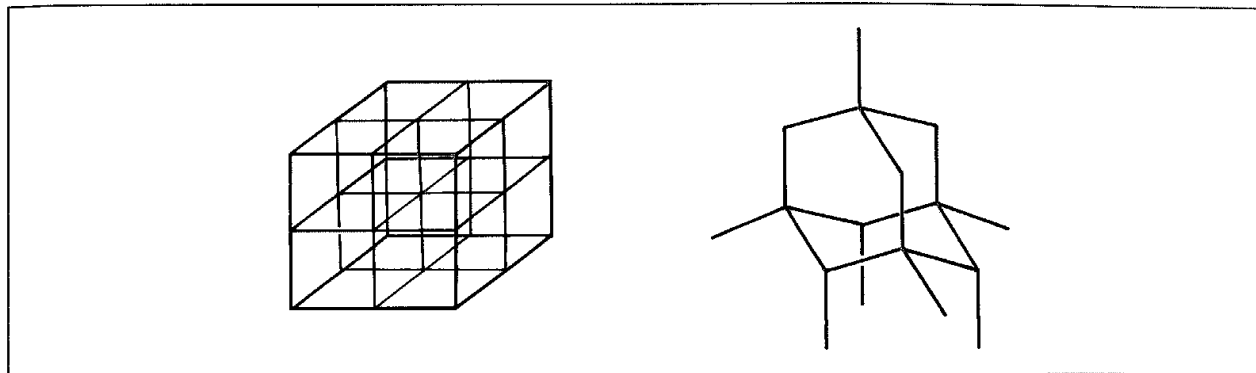


Fig 8 7 Cubic and tetrahedral (diamond) lattices, which are commonly used for lattice simulations of polymers

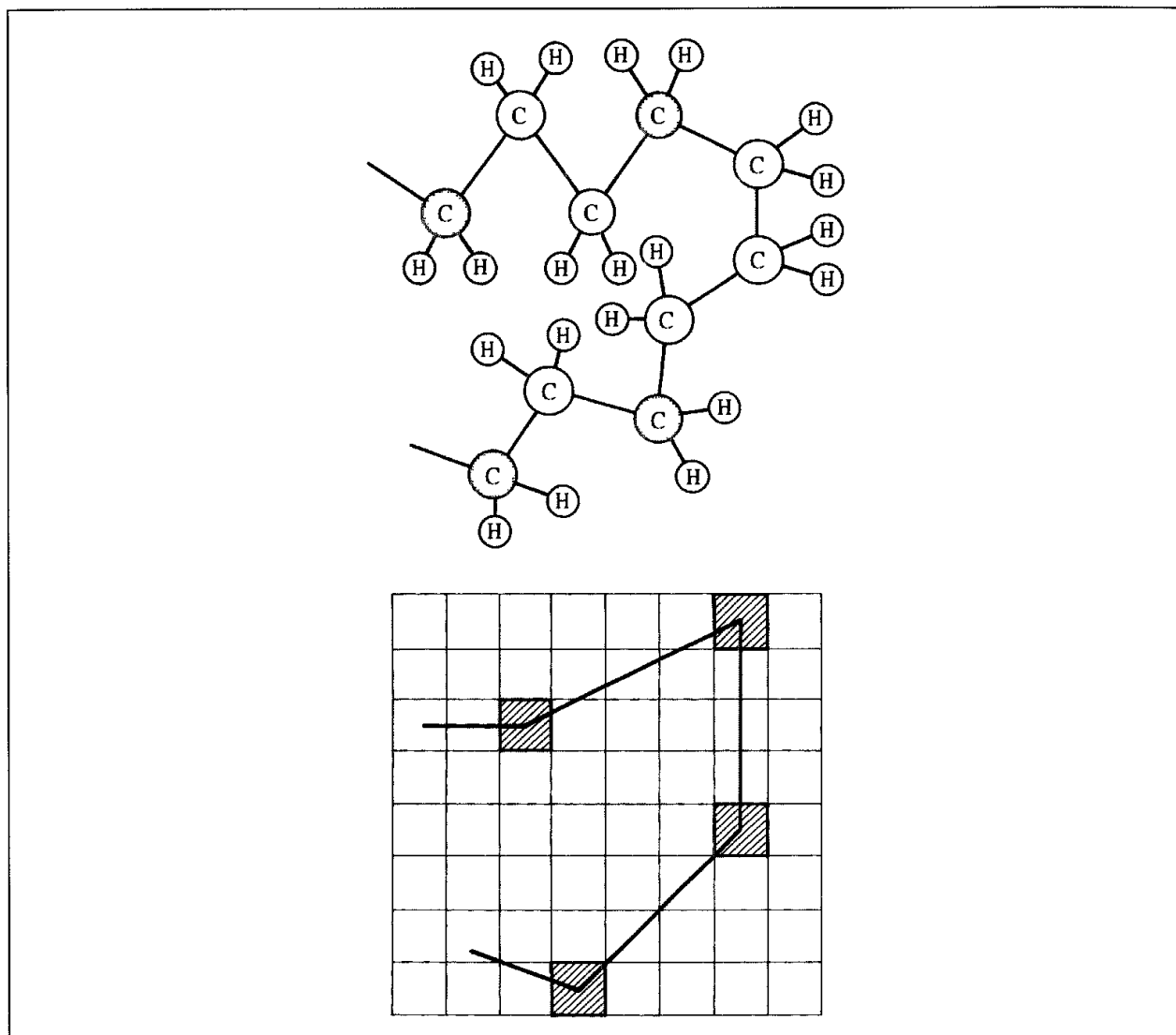


Fig 8 8 The bond fluctuation model In this example three bonds in the polymer are incorporated into a single 'effective bond' between 'effective monomers' (Figure adapted from Baschnagel J, K Binder, W Paul, M Laso, U Suter, I Batoulis, W Jilge and T Bürger 1991 On the Construction of Coarse-Grained Models for Linear Flexible Polymer-Chains - Distribution-Functions for Groups of Consecutive Monomers Journal of Chemical Physics 95 6014-6025)

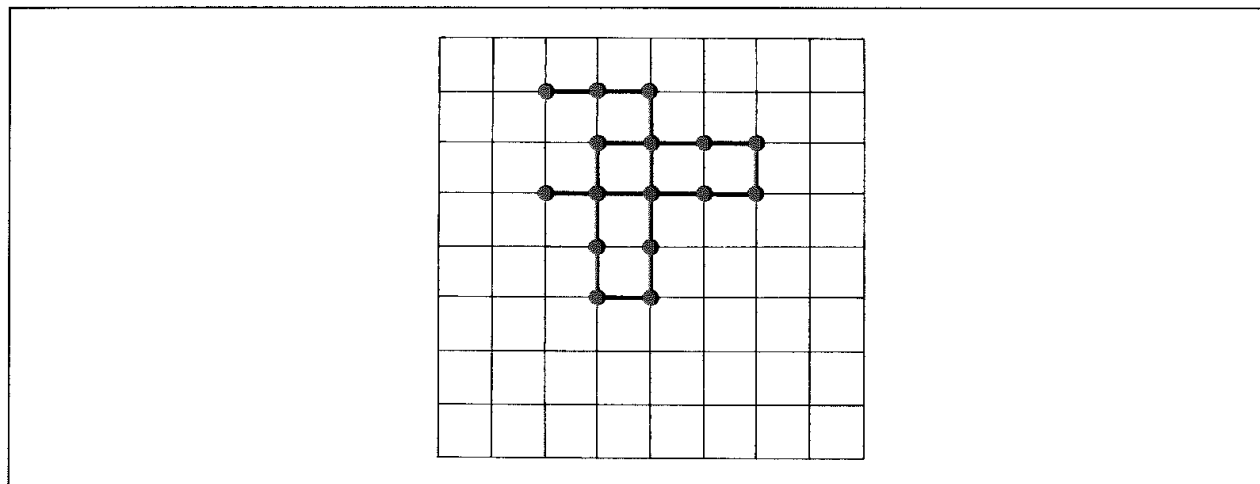


Fig 8.9: In a random walk on a square lattice the chain can cross itself.

represent three bonds in the actual molecule [Baschnagel *et al.* 1991]. In this model each monomer is positioned at the centre of a cube within the lattice and five different distances are possible for the monomer-monomer bond lengths.

Lattices can be used to study a wide variety of polymeric systems, from single polymer chains to dense mixtures. The simplest type of simulation is a 'random walk', in which the chain is randomly grown in the lattice until it contains the desired number of bonds (Figure 8.9). In this model the chain is free to cross itself (i.e. excluded volume effects are ignored). Various properties can be calculated from such simulations, by averaging the results over a large number of trials. For example, a simple measure of the size of a polymer is the mean square end-to-end distance, $\langle R_n^2 \rangle$. For the random walk model $\langle R_n^2 \rangle$ is related to the number of bonds (n) and the length of each bond (l) by:

$$\langle R_n^2 \rangle = nl^2 \quad (8.49)$$

The radius of gyration is another commonly calculated property; this is the root mean square distance of each atom (or monomer) from the centre of mass. For the random walk model the radius of gyration $\langle s^2 \rangle$ is given in the asymptotic limit by:

$$\langle s^2 \rangle = \langle R_n^2 \rangle / 6 \quad (8.50)$$

The ability of the chain to cross itself in the random walk may seem to be a serious limitation, but it is found to be valid under some circumstances. When excluded volume effects are not important (also known as 'theta' conditions) then a subscript '0' is often added to properties such as the mean square end-to-end distance, $\langle R_n^2 \rangle_0$. Excluded volume effects can be taken into account by generating a 'self-avoiding walk' of the chain in the lattice (Figure 8.10). In this model only one monomer can occupy each lattice site. Self-avoiding walks have been used to exhaustively enumerate all possible conformations for a chain of a given length on the lattice. If all states are known then the partition function can be determined and thermodynamic quantities calculated. The 'energy' of each state may be calculated using an appropriate interaction model. For example, the energy may be proportional to the number of adjacent pairs of occupied lattice sites. A variation on this is to use polymers

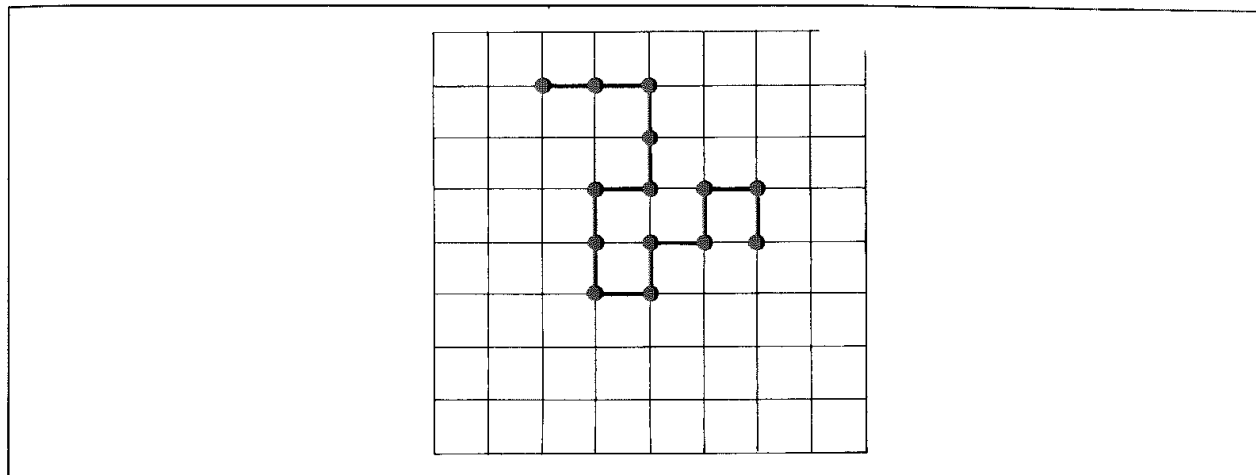


Fig 8 10· Self-avoiding walk only one monomer can occupy each lattice site

consisting of two types of monomer (A and B), which have up to three different energy values: A-A, B-B and A-B. Again, the energy is determined by counting the number of occupied adjacent lattice sites. The relationship between the mean square end-to-end distance and the length of the chain (n) has been investigated intensively; with the self-avoiding walk the result obtained is different from the random walk, with $\langle R_n^2 \rangle$ being proportional to $n^{1.18}$ in the asymptotic limit.

Having grown a polymer onto the lattice, we now have to consider the generation of alternative configurations. Motion of the entire polymer chain or large-scale conformational changes are often difficult, especially for densely packed polymers. In variants of the Verdier-Stockmayer algorithm [Verdier and Stockmayer 1962] new configurations are generated using combinations of 'crankshaft', 'kink jump' and 'end rotation' moves (Figure 8.11). Another widely used algorithm in Monte Carlo simulations of polymers (not just in lattice models) is the 'slithering snake' model. Motion of the entire polymer chain is very difficult, especially for densely packed polymers, and one way in which the polymer can move is by wriggling around obstacles, a process known as *reptation*. To implement a slithering snake algorithm, one end of the polymer chain is randomly chosen as the 'head' and an attempt is made to grow a new bead at one of the available adjacent lattice positions. Each of the remaining beads is then advanced to that of its predecessor in the chain as illustrated in Figure 8.12. The procedure is then repeated. Even if it is impossible to move the chosen 'head', the configuration must still be included when ensemble averages are calculated.

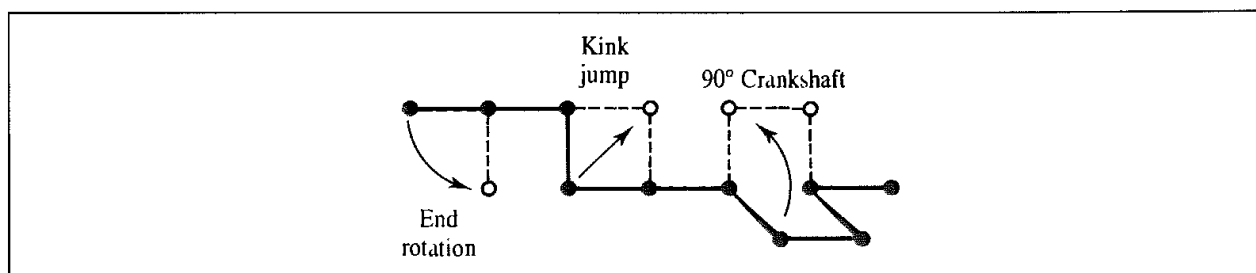


Fig 8 11· The 'crankshaft', 'kink jump' and 'end rotation' moves used in Monte Carlo simulations of polymers

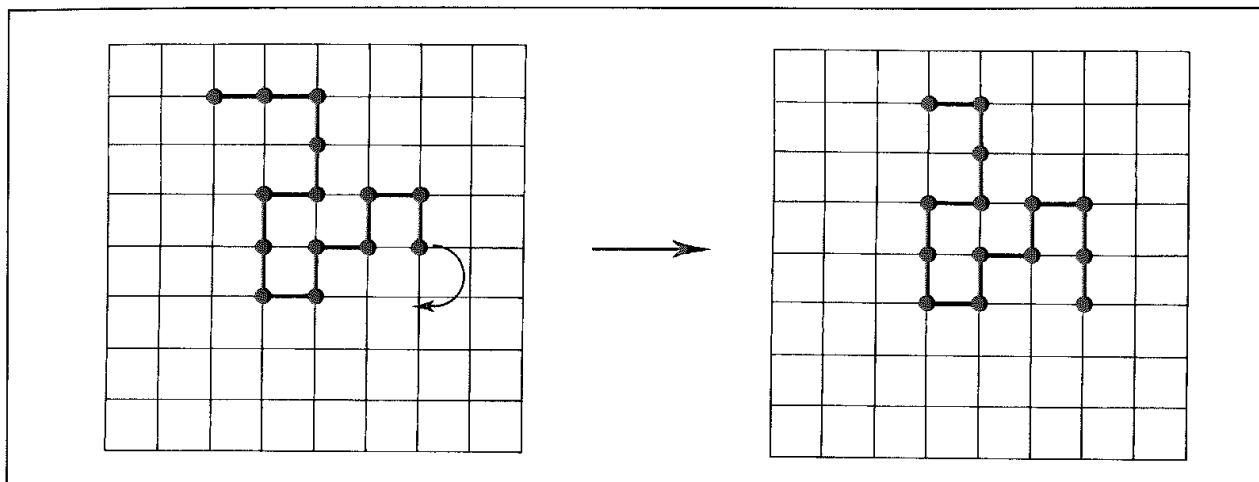


Fig 8 12. The 'slithering snake' algorithm

8.6.2 'Continuous' Polymer Models

The simplest of the continuous polymer models consists of a string of connected beads (Figure 8.13). The beads are freely jointed and interact with the other beads via a spherically symmetric potential such as the Lennard-Jones potential. The beads should not be thought of as being identical to the monomers in the polymer, though they are often referred to as such ('effective monomers' is a more appropriate term). Similarly, the links between the beads should not be thought of as bonds. The links may be modelled as rods of a fixed and invariant length or may be permitted to vary using a harmonic potential function.

In Monte Carlo studies with this freely jointed chain model the beads can sample from a continuum of positions. The *pivot algorithm* is one way that new configurations can be generated. Here, a segment of the polymer is randomly selected and rotated by a random amount, as illustrated in Figure 8.13. For isolated polymer chains the pivot algorithm can give a good sampling of the configurational/conformational space. However, for polymers in solution or in the melt, the proportion of accepted moves is often very small due to high-energy steric interactions

The most unrealistic feature of the freely jointed chain model is the assumption that the bond angles can vary continuously. In the *freely rotating chain model* the bond angles are held fixed but free rotation is possible about the bonds, such that any torsion angle value between 0°

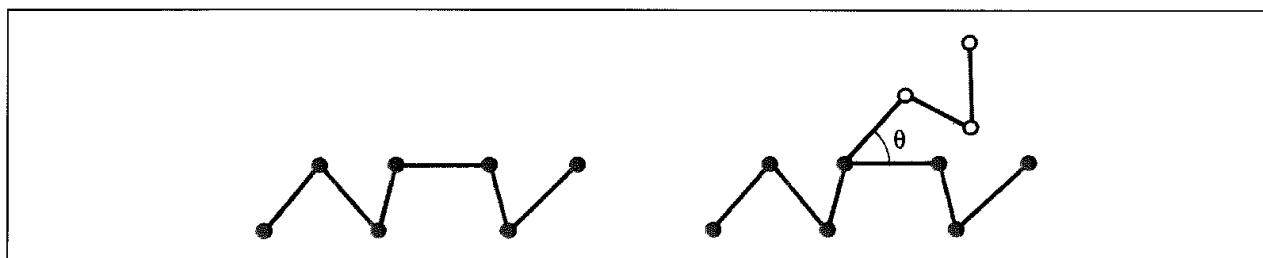


Fig 8 13 The bead model for polymer simulations The beads may be connected by stiff rods or by harmonic springs

and 360° is equally likely. Fixing the bond angles in this way obviously affects the properties of the chain when compared to the freely jointed chain; one way to quantify this is via the characteristic ratio C_n , which is defined as:

$$C_n = \frac{\langle R_n^2 \rangle_0}{nl^2} \quad (8.51)$$

The characteristic ratio approximately indicates how extended the chain is. For the freely rotating chain the characteristic ratio is given by:

$$C_n = \frac{1 + \cos \theta'}{1 - \cos \theta'} - \frac{2 \cos \theta'}{n} \frac{1 - \cos^n \theta'}{(1 - \cos \theta')^2} \quad (8.52)$$

where θ' is the supplement of the normal bond angle (i.e. $\theta' = 180^\circ - \theta$). For an infinitely long chain the characteristic ratio becomes:

$$C_\infty = \frac{1 + \cos \theta'}{1 - \cos \theta'} \quad (8.53)$$

To move up the scale of complexity one now needs to consider the energetics of rotation about each bond. The simplest approach is to assume that each bond can be treated independently and that the total energy of the chain is the sum of the individual torsional energies for each bond. However, this particular model has some serious shortcomings arising from the assumption of independence.

The *rotational isomeric state model* (RIS) developed by Flory [Flory 1969] is probably the best known of the 'approximate' approaches to modelling polymer chains. Each bond is assumed to adopt one of a small number of discrete rotational states, which usually correspond to minima in the potential energy. For example, one might use three rotational states for a typical polyalkane, corresponding to the *trans*, *gauche(+)* and *gauche(-)* conformations. A key part of the RIS approach is its elegant use of various matrices to simplify the calculation. *Generator matrices* are used to establish certain conformation-dependent properties. Thus for a property A one would write:

$$A(\tau_1 \dots \tau_n) = \prod_{i=1}^n \mathbf{F}_i \quad (8.54)$$

where \mathbf{F}_i is the generator matrix for the particular property for bond i (with torsion angle τ_i). An example is the generator matrix for the square end-to-end distance R^2 , which takes the following form:

$$\mathbf{G}_i = \begin{bmatrix} 1 & 2\mathbf{l}^T \mathbf{T} & l^2 \\ 0 & \mathbf{T} & \mathbf{l} \\ 0 & 0 & 1 \end{bmatrix} \quad (8.55)$$

The vector \mathbf{l} is the bond vector for bond i and \mathbf{T} is the 3×3 matrix that transforms coordinates in the reference frame for bond $(i+1)$ to those in the frame of bond i . In this case the square end-to-end distance can be calculated from:

$$R^2 = \mathbf{G}_{[1} \mathbf{G}_2^{n-2} \mathbf{G}_n \quad (8.56)$$

The nomenclature is such that $\mathbf{G}_{[1}$ represents the first row of the matrix \mathbf{G}_1 and \mathbf{G}_n represents the last column of \mathbf{G}_n .

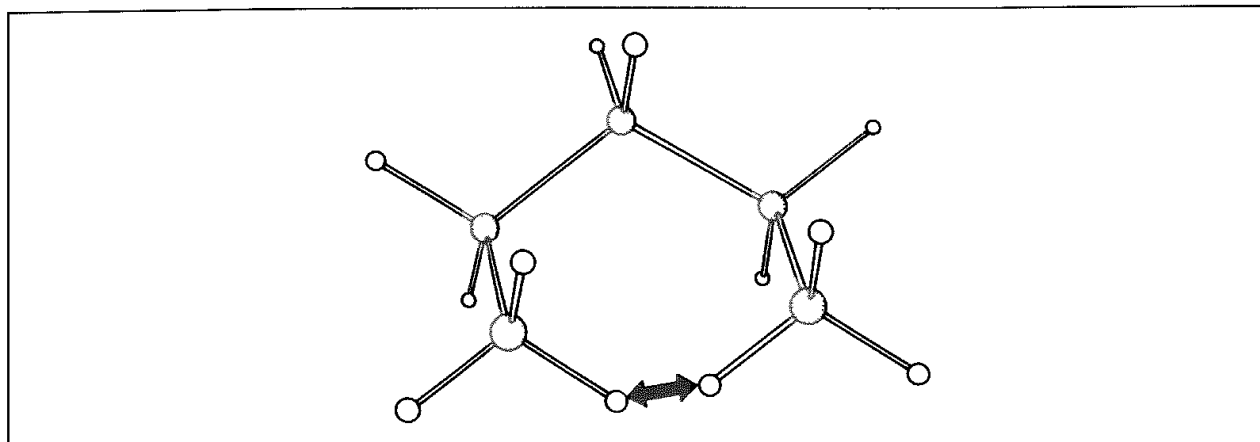


Fig 8.14. Pentane violation

In order to calculate average properties of the polymer chain one uses a standard statistical mechanical approach involving a summation over all possible conformations, with each term multiplied by the appropriate Boltzmann factor. This involves the use of a *statistical weights matrix*, which Flory introduced to deal with the influence a bond has on its neighbours. The pentane violation is the most important of these, wherein a sequence of gauche(+) and gauche(-) bonds gives rise to an unfavourable high-energy interaction (Figure 8.14). The statistical weights matrix associated with bond i has v_{i-1} rows and v_i columns, which correspond to the v_{i-1} rotational states of bond $(i-1)$ and the v_i rotational states of bond i . For example, for a typical polymer with the trans, gauche(+) and gauche(-) rotational states the statistical weights matrix is:

$$\mathbf{U}_i = \begin{bmatrix} u_{tt} & u_{tg^+} & u_{tg^-} \\ u_{g^+t} & u_{g^+g^+} & u_{g^+g^-} \\ u_{g^-t} & u_{g^-g^+} & u_{g^-g^-} \end{bmatrix} \quad (8.57)$$

Some typical values for the elements of this matrix would be:

$$\mathbf{U}_i = \begin{bmatrix} 1.0 & 0.54 & 0.54 \\ 1.0 & 0.54 & 0.05 \\ 1.0 & 0.05 & 0.54 \end{bmatrix} \quad (8.58)$$

The key point to note is the small weight (0.05) for adjacent gauche(+)-gauche(-) bonds. By combining the generator and statistical weights matrices it is possible to reduce the problem of calculating the average value of a property from a set of complex integrals to a series of straightforward matrix multiplications. Some of the properties that can be determined from the RIS model include the mean square end-to-end distance, the mean square radius of gyration and the mean square dipole moment.

The RIS model can be combined with the Monte Carlo simulation approach to calculate a wider range of properties than is available from the simple matrix multiplication method. In the RIS Monte Carlo method the statistical weight matrices are used to generate chain conformations with a probability distribution that is implied in their statistical weights.

Each conformation is generated by starting at one end of the chain and setting the backbone torsion angles one at a time until the entire chain has been constructed. The probability that a particular torsional state is selected for a given bond depends upon the *a priori* probabilities of each state and also upon the torsional state selected for the previous bond in the chain. These probabilities are used at each step by the Monte Carlo procedure to generate the whole chain. A large number of such chains is grown, calculating for each the properties of interest, which are then averaged. Properties which can be determined by the RIS-MC approach include the pair correlation function (which gives the relative probability of finding two atoms within the same chain separated by a distance r), the scattering function (which indicates how the polymer may scatter neutrons or X-rays) and the force-elongation relation (which gives the mean end-to-end distance of a chain subjected to an external force).

The ultimate level of detail in polymer modelling is achieved with the atomistic models, which as the name implies explicitly represent the atoms in the system. An atomistic model is clearly the closest to 'reality' and is necessary if one wishes to calculate accurately certain properties. One of the major problems with simulations of polymers that is particularly pertinent to the atomistic models is how to generate an initial configuration of the system. Amorphous polymers by definition do not adopt a characteristic and reproducible three-dimensional structure. It is important that the properties of the initial configuration are similar to the state one wishes to simulate else the computer time needed to move to the required state can be prohibitive. For short chains containing approximately 20–30 backbone bonds it is feasible to start from a regular crystalline structure, which can then be melted, but to 'melt' a long chain may require a prohibitive amount of computer time. For longer chains an initial configuration may be generated using a random walk and periodic boundary conditions (Figure 8.15). Such an arrangement will inevitably contain high-energy overlaps. These unfavourable interactions may be removed by relaxing the system using minimisation and/or computer simulation, during which the force field potentials are gradually turned on.

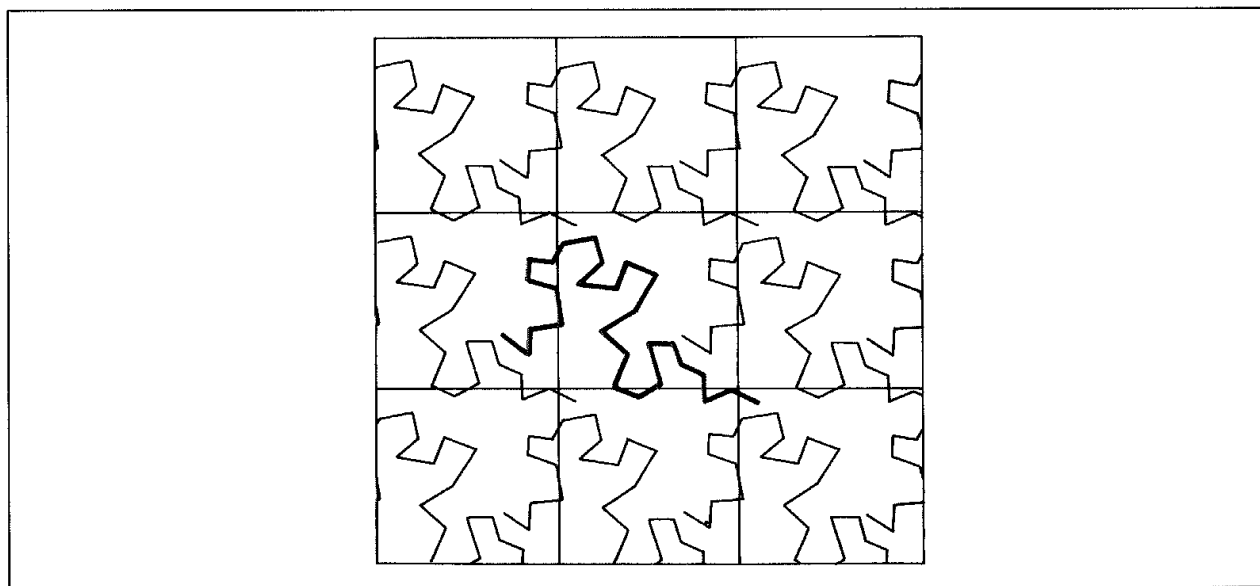


Fig 8 15 Generation of an initial configuration of a polymer using periodic boundary conditions

8.7 'Biased' Monte Carlo Methods

In some situations one is particularly interested in the behaviour of just a part of the system. For example, if we were simulating a solute-solvent system that contained a single solute molecule surrounded by a large number of solvent molecules then the behaviour of the solute and its interactions with the solvent would be of most interest. Solvent molecules far from the solute would be expected to behave almost like bulk solvent. A variety of techniques have been developed which can enhance the ability of the Monte Carlo method to explore the most important regions of phase space in such cases. One relatively simple procedure is *preferential sampling*, where the molecules in the vicinity of the solute are moved more frequently than those further away. This can be implemented by defining a cutoff region around the solute; molecules outside the cutoff region are moved less frequently than those inside the region as determined by a probability parameter p . At each Monte Carlo iteration a molecule is randomly chosen. If the molecule is inside the cutoff region then it is moved; if it is outside the region then a random number is generated between 0 and 1 and compared to the probability p . If p is greater than the random number a trial move is attempted; otherwise no move is made, no averages are accumulated, and a new molecule is randomly selected. The closer p is to zero, the more often 'closer' molecules are moved than 'further' molecules.

An alternative to the use of a fixed cutoff region is to relate the probability of choosing a solvent molecule to its distance from the solute, usually by some inverse power of the distance

$$p \propto r^{-n} \quad (8.59)$$

In preferential sampling it is necessary to ensure that the correct procedures are followed when deciding whether to accept or reject a move in a manner that is consistent with the principle of microscopic reversibility. Suppose a molecule inside the cutoff region is moved outside the cutoff region. In the preferential sampling scheme the chances of the molecule now being selected for an out \rightarrow in move are less than for the original in \rightarrow out move and this must be taken into account when determining the acceptance criteria

The *force-bias* Monte Carlo method [Pangali *et al.* 1978; Rao and Berne 1979] biases the movement according to the direction of the forces on it. Having chosen an atom or a molecule to move, the force on it is calculated. The force corresponds to the direction in which a 'real' atom or molecule would move. In the force-bias Monte Carlo method the random displacement is chosen from a probability distribution function that peaks in the direction of this force. The *smart Monte Carlo method* [Rosky *et al.* 1978] also requires the forces on the moving atom to be calculated. The displacement of an atom or molecule in this method has two components; one component is the force, and the other is a random vector $\delta\mathbf{r}^G$:

$$\delta\mathbf{r}_i = \frac{A\mathbf{f}_i}{k_B T} + \delta\mathbf{r}_i^G \quad (8.60)$$

where \mathbf{f}_i is the force on the atom and A is a parameter. The random displacement $\delta\mathbf{r}_i^G$ is chosen from a normal distribution with zero mean and variance equal to $2A$.

The main difference between the force-bias and the smart Monte Carlo methods is that the latter does not impose any limit on the displacement that an atom may undergo. The displacement in the force-bias method is limited to a cube of the appropriate size centred on the atom. However, in practice the two methods are very similar and there is often little to choose between them. In suitable cases they can be much more efficient at covering phase space and are better able to avoid bottlenecks in phase space than the conventional Metropolis Monte Carlo algorithm. The methods significantly enhance the acceptance rate of trial moves, thereby enabling larger moves to be made as well as simultaneous moves of more than one particle. However, the need to calculate the forces makes the methods much more elaborate, and comparable in complexity to molecular dynamics.

8.8 Tackling the Problem of Quasi-ergodicity: J-walking and Multicanonical Monte Carlo

If there are high-energy barriers between the potential energy minima in a system then a normal Metropolis Monte Carlo simulation may become trapped in just a few of the low-energy regions and fail to properly sample large regions of the thermally accessible space. Such a simulation may appear to possess all the qualities of a good simulation, in terms of its convergence, yet may give results that are completely incorrect. Such a simulation is often referred to as *quasi-ergodic*. This problem arises when studying systems as diverse as rare gas clusters near their melting temperature or protein folding, but it can also be demonstrated in even the simplest of model systems, such as the one-dimensional double-well potential (Figure 8.16). At low temperatures the simulation is unable to cross the high-energy barriers because of the favourable Boltzmann factor. A variety of methods have been suggested for tackling this problem, of which we shall consider two: J-walking and the multicanonical Monte Carlo method.

8.8.1 J-walking

In the J-walking (or Jump-walking) method [Frantz *et al.* 1990] a low-temperature Monte Carlo simulation is permitted occasionally to attempt jumps to regions of space that are accessible to a simulation run at a high temperature. The simplest way to implement this method is to perform the two simulations (at the high and low temperatures) in tandem. The low-temperature simulation is periodically permitted to attempt a jump to the configuration of the high-temperature simulation (the J-walker). The same Metropolis criteria are applied when deciding whether or not to accept the move. The high-temperature simulation will still in principle tend to be biased towards the low-energy regions so there will still be a reasonable chance that one of these special attempted jumps will be accepted.

In practice, it is found that this simple implementation is not the most effective approach. There are two particular problems. First, when the two simulations are run in tandem then significant correlations can arise, which results in large systematic errors. There are a number of ways to avoid these correlations, such as moving the J-walker an extra number

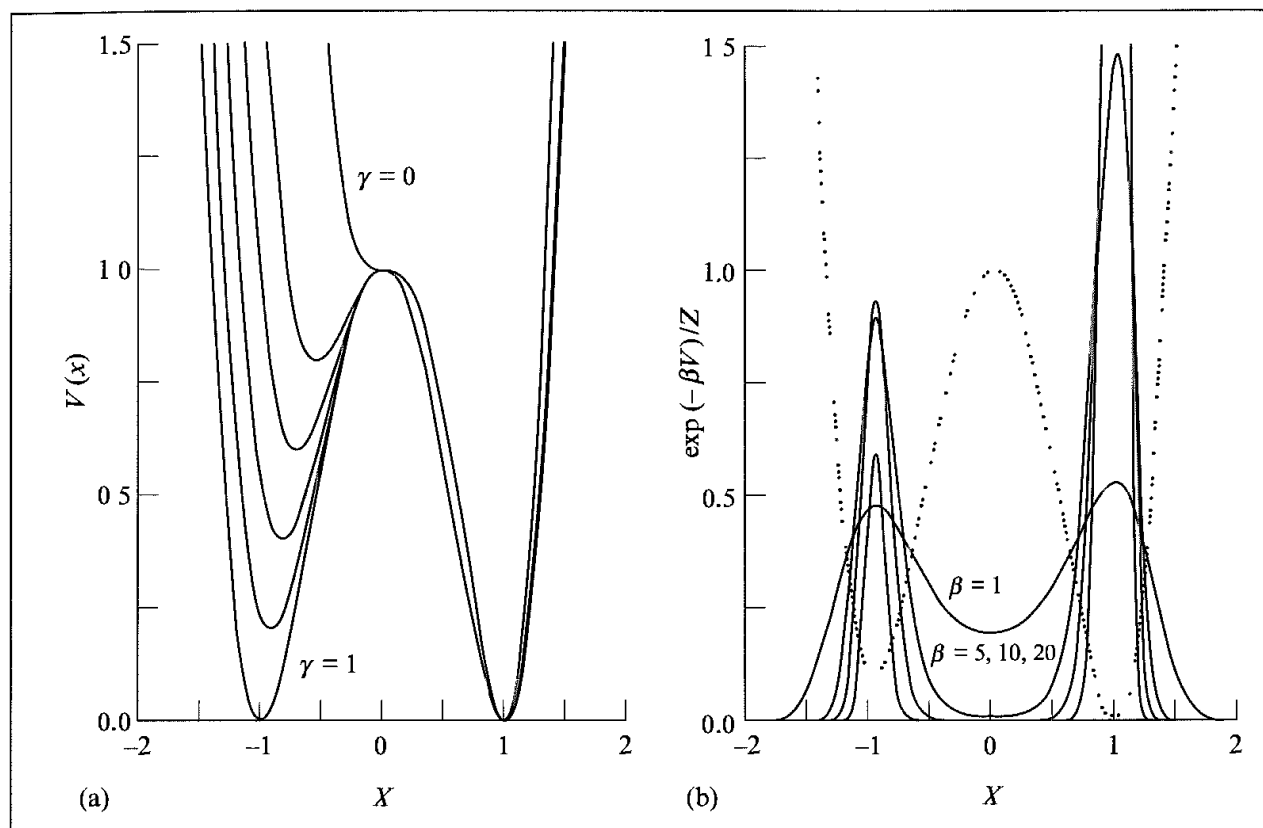


Fig 8.16 Double-well potential which shows quasi-ergodicity. The potential is described by the quartic $\mathcal{V}(x) = 3\delta x^4 + 4\delta(\alpha - 1)x^3 - 6\delta\alpha x^2 + 1$, where $\delta = 1/(2\alpha + 1)$, and the potential is characterised by a parameter $\gamma (= (\alpha^4 + 2\alpha^3)/(2\alpha + 1))$. When $\gamma = 0$ there is just a single well and when $\gamma = 1$ the potential is symmetrical. On the right are Boltzmann distribution functions for the $\gamma = 0.9$ potential for various temperatures, expressed in terms of $\beta (= 1/k_B T)$ (Figure redrawn from Frantz, D D, D L Freeman and J D Doll 1990 Reducing quasi-ergodic behavior in Monte Carlo simulations by J-walking applications to atomic clusters Journal of Chemical Physics 93 2769–2784)

of steps whenever a jump is attempted or running several high-temperature J-walkers and selecting jumps from them at random. However, it is found that a more effective approach is to perform the high-temperature simulation first. The low-temperature simulation first reads in and stores the configurations from the high-temperature simulation, which are then selected at random for each special jump. This approach obviously requires that sufficient storage is available to save the high-temperature configurations; however, it is not necessary to store every configuration (as there will be a high degree of correlation between successive configurations) but rather a representative sample.

One of the first applications of J-walking was to argon clusters [Frantz *et al.* 1990]. These clusters (containing up to 30 atoms) are experimentally known to exhibit strange melting behaviour, wherein the melting temperature is very heavily dependent upon the number of atoms in the cluster. Thus argon clusters Ar_n with seven, thirteen or nineteen atoms have particularly high melting temperatures, whereas those clusters with eight, fourteen or twenty atoms have particularly low melting temperatures. Moreover, certain clusters have different melting and freezing temperatures, implying that there is a range of

temperatures where both solid-like and liquid-like forms coexist. Simulation of these clusters in this transition region is difficult because of the problems of quasi-ergodicity and in obtaining satisfactory convergence. When applying the J-walking method to these clusters it was found necessary to generate the J-walker distribution in stages. The objective was to study an Ar_{13} cluster over a temperature range of 24–41 K using a J-walker at 50 K. However, the distribution of potential energies for the 50 K J-walker did not overlap with the distribution for a 20 K walker, which would mean that at 20 K hardly any of the attempted jumps to the J-walker distribution would be accepted. A series of simulations was thus performed. First, a 50 K J-walker was used to generate a distribution at 40 K, which in turn was used to produce a distribution at 30 K. Finally, the 30 K distribution acted as the J-walker for the 20 K simulation. Significantly better convergence for the cluster configurational energy and the heat capacity when compared with an equivalent standard Metropolis Monte Carlo procedure were noted, even when the simulations were started from a random configuration rather than the icosahedral low-energy geometry.

A related approach designed to be used for conformationally flexible molecules is the ‘jumping between wells’ (JBW) approach [Senderowitz *et al.* 1995]. Here a conformational analysis (see Chapter 9) is first performed on the molecule in order to identify its thermally accessible minimum-energy conformations (e.g. all within 5 kcal/mol of the global minimum-energy conformation). These minimum-energy conformations are stored in a list. The changes in internal coordinates that would be required to interconvert each pair of minimum-energy conformations are then determined. At each stage of the iterative cycle the minimum-energy conformation that is closest to the current structure is identified. A minimum-energy conformation is then selected at random from the conformation list and the appropriate transformation applied to the current structure. Small random changes in this new structure are then made to give a new trial structure, which is then accepted or rejected using the Metropolis criterion with reference to the initial starting structure. The process is then repeated. It is important with this method to avoid oversampling some of the potential energy wells which can occur if the combination of a conformational jump and the subsequent randomisation enters the space of a different minimum-energy conformation. This problem can be overcome by only including minimum-energy conformations that are significantly different from each other and by making the randomisation step small relative to the distance between the conformations.

8.8.2 The Multicanonical Monte Carlo Method

In a canonical ensemble the probability $P_{\text{canon}}(T, E)$ of visiting a point in phase space with an energy E is proportional to the Boltzmann factor, $w_B = \exp(-E/k_B T)$, multiplied by the density of states, $n(E)$, where the number of states between E and $E + dE$ is given by $n(E)\delta E$. Thus:

$$P_{\text{canon}}(T, E) \propto n(E)w_B(E) \quad (8.61)$$

The density of states increases rapidly with energy but the Boltzmann factor decreases exponentially, meaning that $P_{\text{canon}}(T, E)$ is bell-shaped, with values that can vary by many orders of magnitude as the energy changes. In the multicanonical method the simulation

is performed in an artificial multicanonical ensemble in which the probability of visiting a state is independent of its energy over a certain energy range. This is equivalent to replacing the Boltzmann factor by a multicanonical weight factor, $w_{\text{mu}}(E)$:

$$P_{\text{mu}}(E) \propto n(E)w_{\text{mu}}(E) = \text{constant} \quad (8.62)$$

This implies that the multicanonical weight factor is proportional to $n(E)^{-1}$. A simulation performed in the multicanonical ensemble is able to overcome any energy barrier, in contrast to the situation in a normal, canonical simulation. The main task in performing a multicanonical simulation is to determine the weight factor, which is not known *a priori* (unlike the case for the canonical ensemble, where it is equal to the Boltzmann factor). The multicanonical weight factor is usually determined in an iterative fashion from a series of short simulations. One approach is as follows [Okamoto and Hansmann 1995]. First, a simulation is carried out at a temperature (T_0) which is high enough (e.g. 1000 K) to ensure that all energy barriers can be overcome. An array $S(E)$ is established with all its elements initially set to zero. Each element of this array refers to a particular energy range E to $E + \delta E$, where δE might be (say) 1 kcal/mol. From this initial simulation a histogram is constructed which gives the number of times a state with an energy in the range E to $E + \delta E$ is determined. These histogram values are stored in an array $H(E)$. Each of the values in this array ($H(E)$) should thus initially approximate the energy distribution at the temperature T_0 :

$$H(E) \propto n(E) \exp(-E/k_B T_0) \quad (8.63)$$

During this simulation the minimum and maximum energies visited during the simulation are recorded (E_{min} and E_{max} , respectively). The array $S(E)$ is now updated according to the following formula:

$$S(E) = S(E) + \ln H(E) \quad (8.64)$$

for all energy values E where the energy is between E_{min} and E_{max} and where the value of the appropriate array element $H(E)$ is greater than some minimum value (e.g. 20). In other words, that particular energy level must have been visited at least twenty times during the simulation. The following two parameters are now calculated:

$$\beta(E) = \begin{cases} 1/k_B T_0 & E \geq E_{\text{max}} \\ 1/k_B T_0 + \frac{S(E') - S(E)}{E' - E} & E_{\text{min}} \leq E \leq E_{\text{max}} \\ \beta(E_{\text{min}}) & E < E_{\text{min}} \end{cases} \quad (8.65)$$

$$\alpha(E) = \begin{cases} 0 & E \geq E_{\text{max}} \\ \alpha(E') + (\beta(E') - \beta(E))E' & E < E_{\text{max}} \end{cases} \quad (8.66)$$

E' refers to that element in the array $S(E)$ which succeeds E (i.e. E and E' are adjacent elements). Having determined the parameters $\alpha(E)$ and $\beta(E)$, a multicanonical weight factor is calculated as:

$$w_{\text{mu}}(E) = \exp[-\beta(E)E - \alpha(E)] \quad (8.67)$$

A new simulation is now initiated using this multicanonical weight factor (rather than the Boltzmann factor), from which new values of $S(E)$ and thus $H(E)$ can be determined. This cycle is continued until the distribution in $H(E)$ is reasonably flat in the energy range being considered.

Once the final multicanonical weight factor has been derived it provides the distribution for the production simulation in which high-energy configurations will be sampled adequately and high-energy barriers can be crossed with ease. Moreover, from this single simulation it is possible to derive the canonical distribution $P_{\text{canon}}(T, E)$ at any temperature (hence the name 'multicanonical'):

$$P_{\text{canon}}(T, E) \propto P_{\text{mu}}(E) \omega_{\text{mu}}^{-1} e^{-E/k_{\text{B}}T} \quad (8.68)$$

The average value of any property A at a temperature T can be determined from the multicanonical simulation using the following formula:

$$\langle A \rangle_T = \frac{\int A(E) P_{\text{canon}}(T, E) dE}{\int P_{\text{canon}}(T, E) dE} \quad (8.69)$$

In practice, one is restricted to energies between E_{min} and E_{max} , and a range of temperatures $T_{\text{min}} \leq T \leq T_{\text{max}}$ where the range of permitted temperatures is determined by calculating the expectation value of the energy at temperature T :

$$E_{\text{min}} \leq \langle E \rangle_T \leq E_{\text{max}} \quad (8.70)$$

The array $S(E)$ is intimately related to the entropy of the system as can be demonstrated by expanding the logarithm of $H(E)$ as:

$$\ln H(E) = \ln n(E) - E/k_{\text{B}}T_0 + \text{constant} \quad (8.71)$$

and recalling that the entropy is related to the density of states by $S(E) = \ln n(E)$.

The multicanonical Monte Carlo method can be used to study a wide range of systems and is particularly useful where traditional Metropolis Monte Carlo methods encounter difficulties. In addition to the simulation of systems such as clusters of rare gas atoms, the multicanonical method has been used to study the properties of macromolecular systems. There has been particular interest in using the approach to study the properties of amino acid polymers, and in particular their ability to form certain types of regular structure such as alpha helices. These structures will be discussed in more detail in Chapter 10. It suffices for now to note that certain types of amino acid confer a greater propensity to adopt a helical structure. Traditional simulation techniques can be used to study the equilibrium between the helical and 'random' (or coil) structures but typically have to start from the regular (i.e. helical) structure, which is then 'unfolded' using molecular dynamics or Monte Carlo simulations. The large number of minima on the potential energy surface means that it is not practical to start from the unfolded structure and observe helix formation. However, the multicanonical Monte Carlo method does provide a mechanism for more completely exploring the energy surface (including the helical conformations), starting from a random structure. In one such study, Okamoto and Hansmann were able to compare the thermodynamics of the equilibrium between the helix and coil structures for three amino acids (alanine, valine and glycine) which have different observed propensities to form helix structures [Okamoto and Hansmann 1995].

One of the drawbacks of the multicanonical method is that, during the simulations to derive the weight factor, the energy distribution in $H(E)$ can oscillate rather than steadily approaching a limiting distribution. Another drawback is that it can fail to properly

sample the low-energy regions adequately. The multicanonical method samples energies within a certain range with approximately equal probability, but at the ends of this range the probability drops dramatically. Thus little sampling is done from the low-energy regions. These low-energy regions are proportionally more important at low temperatures, leading to poor statistics. To some extent the problems with J-walking are complementary to the limitations of the multicanonical method, and so attempts have been made to combine the two [Xu and Berne 1999]. In the combined method (termed 'multicanonical jump walking') the multicanonical weight factor is first derived and then used to perform a long multicanonical simulation. The configurations generated by this multicanonical simulation are saved and used as the 'high-temperature' component in the subsequent J-walking simulation. The key modification is that the standard acceptance criterion for the jump steps during this last phase must be multiplied by the ratio of the weight factors for the two energies (i.e. rather than comparing the usual Boltzmann factor, $\exp(-\Delta\mathcal{V}/k_B T)$, to the random number a modified factor $\exp(-\Delta\mathcal{V}/k_B T)[w(\mathcal{V}_{\text{old}})/w(\mathcal{V}_{\text{new}})]$ is used). This combined approach appears to provide more efficient sampling of phase space for a given number of Monte Carlo steps when compared to the regular J-walking or multicanonical sampling method, on both low-dimensional trial potentials and for systems such as rare gas clusters.

8.9 Monte Carlo Sampling from Different Ensembles

A Monte Carlo simulation traditionally samples from the constant NVT (canonical) ensemble, but the technique can also be used to sample from different ensembles. A common alternative is the isothermal-isobaric, or constant NPT , ensemble. To simulate from this ensemble, it is necessary to have a scheme for changing the volume of the simulation cell in order to keep the pressure constant. This is done by combining random displacements of the particles with random changes in the volume of the simulation cell. The size of each volume change is governed by the maximum volume change, δV_{max} . Thus a new volume is generated from the old volume as follows:

$$V_{\text{new}} = V_{\text{old}} + \delta V_{\text{max}}(2\xi - 1) \quad (8.72)$$

As usual, ξ is a random number between 0 and 1. When the volume is changed, it is in principle necessary to recalculate the interaction energy of the entire system, not just the interactions involving the one atom or molecule that has been displaced. However, for simple interatomic potentials the change in energy associated with a volume change can be calculated very rapidly by using *scaled coordinates*. For a set of particles that are modelled by a Lennard-Jones potential in a cubic box of length L_{old} , the potential energy can be written:

$$\mathcal{V}_{\text{old}}(\mathbf{r}^N) = 4\epsilon \sum_{i=1}^N \sum_{j=i+1}^N \left(\frac{\sigma}{L_{\text{old}} s_{ij}} \right)^{12} - 4\epsilon \sum_{i=1}^N \sum_{j=i+1}^N \left(\frac{\sigma}{L_{\text{old}} s_{ij}} \right)^6 \quad (8.73)$$

where s_{ij} is a scaled coordinate which is related to the actual interatomic distance by $s_{ij} = L_{\text{old}}^{-1} r_{ij}$. It is necessary to write the energy as the sum of two components, one from

the repulsive part of the Lennard-Jones potential and the other from the attractive part:

$$\mathcal{V}_{\text{old}}(\mathbf{r}^N) = \mathcal{V}_{\text{old}}(12) + \mathcal{V}_{\text{old}}(6) \quad (8.74)$$

The advantage of using scaled coordinates is that they are independent of the size of the simulation box. Thus the energy of the configuration in a different-sized box (with side L_{new}) is:

$$\mathcal{V}_{\text{new}}(\mathbf{r}^N) = 4\epsilon \sum_{i=1}^N \sum_{j=i+1}^N \left(\frac{\sigma}{L_{\text{new}} s_{ij}} \right)^{12} - 4\epsilon \sum_{i=1}^N \sum_{j=i+1}^N \left(\frac{\sigma}{L_{\text{new}} s_{ij}} \right)^6 \quad (8.75)$$

The energy $\mathcal{V}_{\text{new}}(\mathbf{r}^N)$ is related to the energy $\mathcal{V}_{\text{old}}(\mathbf{r}^N)$ as follows:

$$\mathcal{V}_{\text{new}}(\mathbf{r}^N) = \mathcal{V}_{\text{old}}(12) \left\{ \frac{L_{\text{old}}}{L_{\text{new}}} \right\}^{12} + \mathcal{V}_{\text{old}}(6) \left\{ \frac{L_{\text{old}}}{L_{\text{new}}} \right\}^6 \quad (8.76)$$

The change in energy from the old to the new system is thus:

$$\Delta\mathcal{V}(\mathbf{r}^N) = \mathcal{V}_{\text{old}}(12) \left\{ \frac{L_{\text{old}}}{L_{\text{new}}} - 1 \right\}^{12} + \mathcal{V}_{\text{old}}(6) \left\{ \frac{L_{\text{old}}}{L_{\text{new}}} - 1 \right\}^6 \quad (8.77)$$

Any long-range corrections to the potential must also be taken into account when the volume changes. One way to deal with these is to assume that the non-bonded cutoff scales with the box length. Under such circumstances, the long-range corrections to both the repulsive and attractive parts of the potential scale in exactly the same manner as the short-range interactions. However, the use of this assumption can give rise to serious problems, particularly for techniques such as the Gibbs ensemble Monte Carlo simulation (see Section 8.12) where two coupled simulation boxes of different dimensions are involved. The boxes contain identical particles, but this would be compromised by the use of different non-bonded cutoffs and long-range corrections.

This simple scaling method cannot be used when simulating molecules, for a change in the scaled coordinates would have the effect of introducing large and energetically unfavourable changes in the internal coordinates, such as the bond lengths. It is therefore necessary to recalculate the total interaction energy of the system each time a volume change is made. This is computationally expensive to do, but it is in any case advisable to change the volume relatively infrequently compared to the rate at which the particles are moved. One way to speed up the energy calculation associated with a volume change is to write the potential energy change as a Taylor series expansion of the box size.

The criterion used to accept or reject a new configuration is slightly different for the isothermal-isobaric simulation than for a simulation in the canonical ensemble. The following quantity is used:

$$\Delta H(\mathbf{r}^N) = \mathcal{V}_{\text{new}}(\mathbf{r}^N) - \mathcal{V}_{\text{old}}(\mathbf{r}^N) + P(V_{\text{new}} - V_{\text{old}}) - Nk_{\text{B}}T \ln \left(\frac{V_{\text{new}}}{V_{\text{old}}} \right) \quad (8.78)$$

If ΔH is negative then the move is accepted; otherwise, $\exp(-\Delta H/k_{\text{B}}T)$ is compared to a random number between 0 and 1 and the move accepted according to:

$$\text{rand}(0, 1) \leq \exp(-\Delta H/k_{\text{B}}T) \quad (8.79)$$

To check that an isothermal–isobaric simulation is working properly, the pressure can be calculated from the virial as outlined in Section 6.2.3, including the appropriate long-range correction (which will not, of course, be constant as the volume of the box changes). Its value should be equal to the input pressure that appears in Equation (8.78).

8.9.1 Grand Canonical Monte Carlo Simulations

In the grand canonical ensemble the conserved properties are the chemical potential, the volume and the temperature. It can sometimes be more convenient to perform a grand canonical simulation at constant activity, z , which is related to the chemical potential μ by.

$$\mu = k_{\text{B}}T \ln \Lambda^3 z \quad (8.80)$$

where Λ is the de Broglie wavelength given by $\Lambda = \sqrt{h^2/2\pi mk_{\text{B}}T}$.

The key feature about the grand canonical Monte Carlo method is that the number of particles may change during the simulation. There are three basic moves in a grand canonical Monte Carlo simulation:

1. A particle is displaced, using the usual Metropolis method.
2. A particle is destroyed.
3. A particle is created at a random position.

The probability of creating a particle should be equal to the probability of destroying a particle. To determine whether to accept a destruction move the following quantity is calculated:

$$\Delta D = \frac{[\psi_{\text{new}}(\mathbf{r}^N) - \psi_{\text{old}}(\mathbf{r}^N)]}{k_{\text{B}}T} - \ln \left(\frac{N}{zV} \right) \quad (8.81)$$

For a creation step the equivalent quantity is:

$$\Delta C = \frac{[\psi_{\text{new}}(\mathbf{r}^N) - \psi_{\text{old}}(\mathbf{r}^N)]}{k_{\text{B}}T} - \ln \left(\frac{zV}{N+1} \right) \quad (8.82)$$

If ΔD or ΔC is negative then the move is accepted; if positive, then the exponential $\exp(-\Delta D/k_{\text{B}}T)$ or $\exp(-\Delta C/k_{\text{B}}T)$ as appropriate is calculated and compared with a random number between 0 and 1 in the usual way.

It is important that the possibility of creating a new particle is the same as the probability of destroying an old one. The ratio of particle creation/destruction moves to translation/rotation moves can vary, but the most rapid convergence is often achieved if all types of move occur with approximately equal frequency.

In grand canonical Monte Carlo simulations of liquids there can be some practical problems in achieving statistically accurate results. This is because the probability of achieving a successful creation or destruction step is often very small. Creation steps fail because the fluid is so dense that it is difficult to insert a new particle without causing significant

overlaps with neighbouring particles. Destruction steps fail because the particles in a fluid often experience significant attractive interactions, which are lost when the particle is removed. These problems are particularly acute for long-chain molecules. However, some of the newer Monte Carlo techniques such as the configurational bias Monte Carlo method do enable such systems to be simulated and accurate results obtained. These techniques will be discussed in Section 8.11.

8.9.2 Grand Canonical Monte Carlo Simulations of Adsorption Processes

One application of the grand canonical Monte Carlo simulation method is in the study of the adsorption and transport of fluids through porous solids. Mixtures of gases or liquids can be separated by the selective adsorption of one component in an appropriate porous material. The efficacy of the separation depends to a large extent upon the ability of the material to adsorb one component in the mixture much more strongly than the other component. The separation may be performed over a range of temperatures and so it is useful to be able to predict the adsorption isotherms of the mixtures.

A typical example of such a calculation is the simulation of Cracknell, Nicholson and Quirke [Cracknell *et al.* 1994] who studied the adsorption of a mixture of methane and ethane onto a microporous graphite surface. Four types of move were employed in their simulations: particle moves, particle deletions, particle creations and attempts to exchange particles. Methane was modelled as a single Lennard-Jones particle and ethane as two Lennard-Jones particles separated by a fixed bond length. The graphite surfaces were modelled as Lennard-Jones atoms with a slit-shaped pore being constructed from two layers of graphite separated by an appropriate distance. Triangle-shaped pores can also be used. The simulations were used to calculate the selectivity of the solid for the two components as the ratio of the mole fractions in the pore to the ratio of the mole fractions in the bulk. The selectivity was determined as a function of the pressure for different pore sizes to give some indication of the effect of changing the physical nature of the solid. The pressure can be calculated directly from the input chemical potential using the following standard relationship (for an ideal gas):

$$P = \{\exp(\mu/k_B T)k_B T\}/\Lambda^3 \quad (8.83)$$

The selectivity showed a complicated dependence upon the pore size (Figure 8.17).

The selectivity is best interpreted by considering the interactions between ethane molecules and the walls of the pore. For the smallest pore sizes, the molecules are restricted to the centre of the pore and the ethane molecules are forced to lie flat. As the pore size increases, it becomes possible for ethane to adopt a particularly favourable orientation perpendicular to the walls, with each methyl group being in a potential energy minimum for interaction with the pore atoms. This particular pore size ($2.5\sigma_{\text{CH}_4}$) thus has the greatest selectivity for ethane over methane. As the pore size increases further the distribution of ethane becomes more complex, with some layers of ethane lying flat on the pore wall and some in the centre of the pore, with ethane molecules spanning the space between. These arrangements are shown in Figure 8.18.

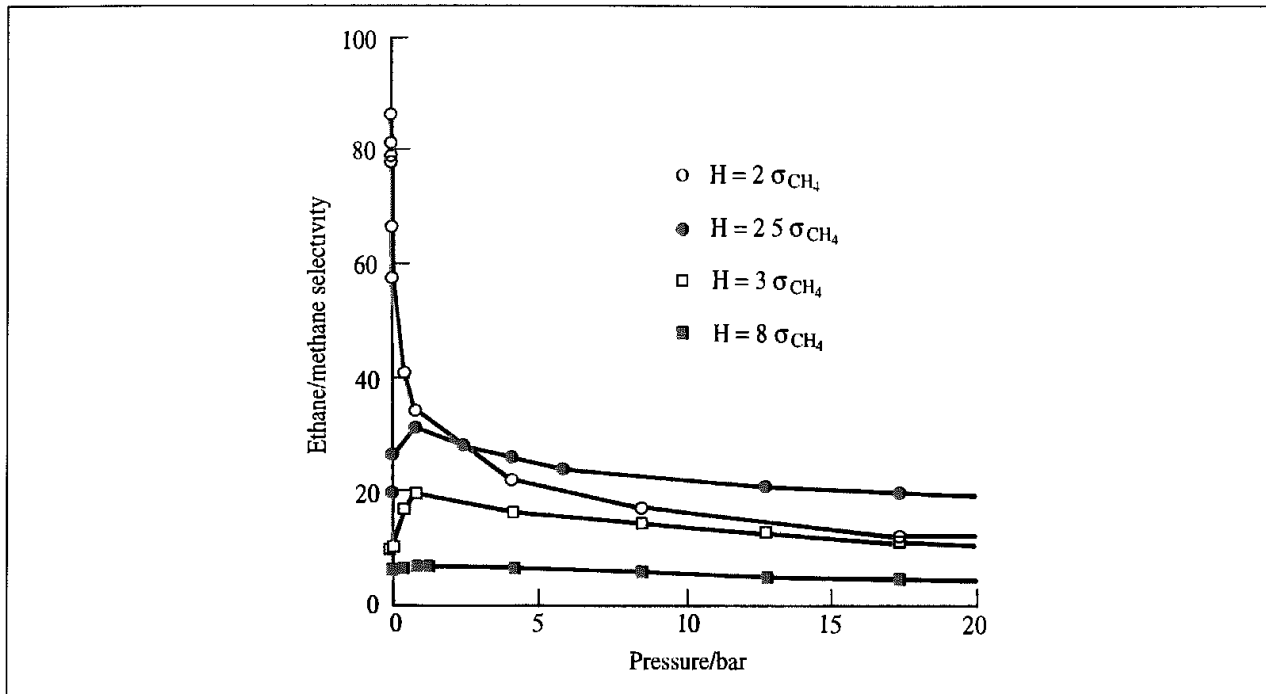


Fig 8.17 Ethane/methane selectivity calculated from grand canonical Monte Carlo simulations of mixtures in slit pores at a temperature of 296 K. The selectivity is defined as the ratio of the mole fractions in the pore to the ratio of the mole fractions in the bulk. H is the slit width defined in terms of the methane collision diameter σ_{CH_4} . (Figure redrawn from Cracknell R F, D Nicholson and N Quirke 1994. *A Grand Canonical Monte Carlo Study of Lennard-Jones Mixtures in Slit Pores; 2. Mixtures of Two-Centre Ethane with Methane* Molecular Simulation 13 161-175.)

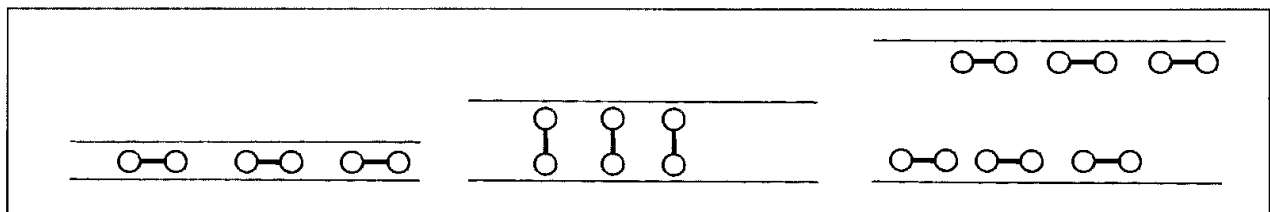


Fig 8.18 Schematic illustration of the arrangements of ethane molecules in slits of varying sizes. In the slit of width $(2.5\sigma_{\text{CH}_4})$ each methyl group is able to occupy a potential minimum from the pore (middle).

8.10 Calculating the Chemical Potential

In a grand canonical simulation the chemical potential is constant. One may also wish to determine how the chemical potential varies during a simulation. The chemical potential is usually determined using an approach due to Widom [Widom 1963], in which a 'test' particle is inserted into the system and the resulting change in potential energy is calculated. The Widom approach is applicable to both molecular dynamics and Monte Carlo simulations. Consider a system containing $N - 1$ particles, into which we insert another particle at a random position. The inserted particle causes the internal potential energy to change by an amount $\psi(\mathbf{r}^{\text{test}})$, i.e. $\psi(\mathbf{r}^N) = \psi(\mathbf{r}^{N-1}) + \psi(\mathbf{r}^{\text{test}})$. Then the configurational

integral for the $N\mathcal{V}$ particle system is given by:

$$Z_N = \int d\mathbf{r}^N \exp[-\mathcal{V}(\mathbf{r}^N)/k_B T] \quad (8.84)$$

or

$$Z_N = \int d\mathbf{r}^N \exp[-\mathcal{V}(\mathbf{r}^{\text{test}})/k_B T] \exp[-\mathcal{V}(\mathbf{r}^{N-1})/k_B T] \quad (8.85)$$

By substituting unity in the form Z_{N-1}/Z_{N-1} it is possible to show that $Z_N = Z_{N-1} V \langle \exp[-\mathcal{V}(\mathbf{r}^{\text{test}})/k_B T] \rangle$.

The excess chemical potential, that is the difference between the actual value and that of the equivalent ideal gas system, is given by:

$$\mu_{\text{excess}} = -k_B T \ln \langle \exp[-\mathcal{V}(\mathbf{r}^{\text{test}})/k_B T] \rangle \quad (8.86)$$

The excess chemical potential is thus determined from the average of $\exp[-\mathcal{V}(\mathbf{r}^{\text{test}})/k_B T]$. In ensembles other than the canonical ensemble the expressions for the excess chemical potential are slightly different. The ghost particle does not remain in the system and so the system is unaffected by the procedure. To achieve statistically significant results many Widom insertion moves may be required. However, practical difficulties are encountered when applying the Widom insertion method to dense fluids and/or to systems containing molecules, because the proportion of insertions that give rise to low values of $\mathcal{V}(\mathbf{r}^{\text{test}})$ falls dramatically. This is because it is difficult to find a 'hole' of the appropriate size and shape.

8.11 The Configurational Bias Monte Carlo Method

Various techniques have been developed to tackle the problem of calculating the chemical potential in cases where the routine Widom method does not give converged results. Of these methods, the configurational bias Monte Carlo (CBMC) method, which was originally introduced by Siepmann [Siepmann 1990], is particularly exciting as it can be applied to assemblies of chain molecules. The configurational bias Monte Carlo method also provides a way to overcome the problems associated with Monte Carlo simulations of assemblies of chain molecules, where many proposed moves are rejected because of high-energy overlaps. The problem of calculating the chemical potential in such cases is clear from the following example. The probability of successfully inserting a single monomer into a fluid of typical liquid density is of the order of 0.5%, or 1 in 200. If one wishes to insert a molecule consisting of n such monomers, the probability is thus approximately 1 in 200^n . For an eight-segment molecule, this probability is less than 1 in 10^{18} , making such calculations impractical. The configurational bias Monte Carlo simulation technique can dramatically improve the chances of making a successful insertion.

The essence of the configurational bias Monte Carlo method is that a growing molecule is preferentially directed (i.e. biased) towards acceptable structures. The effects of these biases can then be removed by modifying the acceptance rules. The configurational bias methods are based upon work published in 1955 by Rosenbluth and Rosenbluth

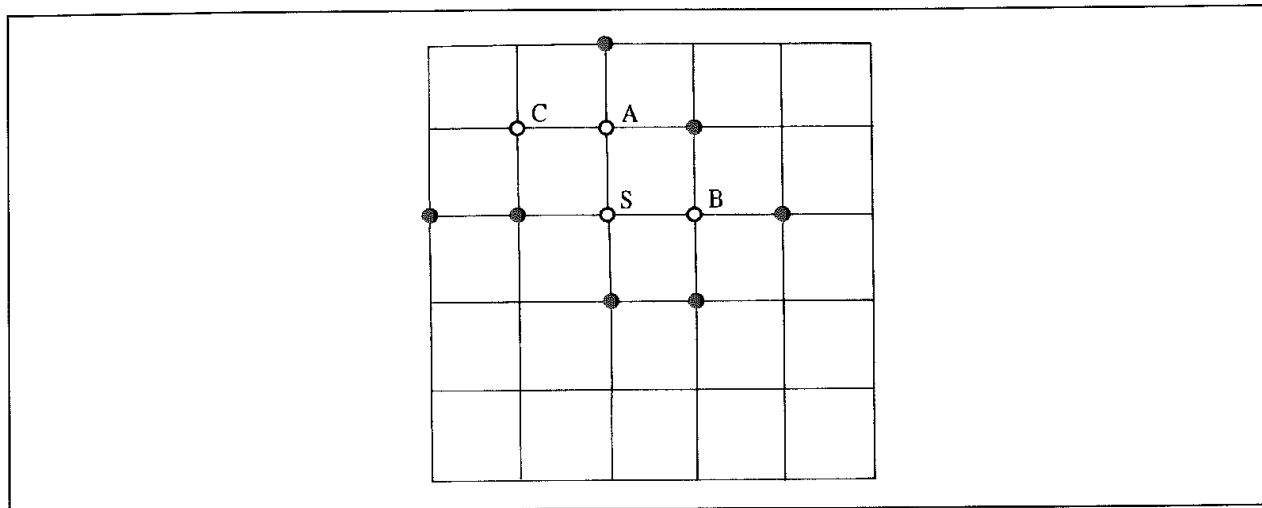


Fig. 8.19 The insertion of a three-unit molecule onto the lattice shown, starting at point S, can be achieved in only one way (see text). (Figure adapted from Siepmann J I 1990 *A Method for the Direct Calculation of Chemical Potentials for Dense Chain Systems*. *Molecular Physics* **70** 1145–1158.)

[Rosenbluth and Rosenbluth 1955] and can be applied to both lattice models and to systems with arbitrary molecular potentials and conformations. The method is most easily explained using a two-dimensional lattice model. Suppose we wish to insert a three-unit molecule onto the lattice shown in Figure 8.19. First we consider how the conventional approach would tackle this problem. The initial step is to select a lattice point at random. Suppose we select the lattice point labelled S in Figure 8.19. We then choose one of the four neighbours of S at random. Of the four neighbouring sites, two are occupied and two are free (A and B in Figure 8.19). There is thus a 50% probability that the move will be rejected at this stage. If we select site B then the molecule can be grown no further as all of the adjacent sites are filled. Were we to grow onto A then we would select one of the three neighbouring sites at random. Of these only site C is available. On average, only one trial in twelve will successfully grow a molecule from S using a conventional Monte Carlo algorithm.

Let us now consider how the configurational bias Monte Carlo method would deal with this problem. Again, the first site (S) is chosen at random. We next consider where to place the second unit. The sites adjacent to S are examined to see which are free. In this case, only two of the four sites are free. One of these free sites is chosen at random. Note that the conventional Monte Carlo procedure selected from all four adjoining sites at random, irrespective of whether it is occupied or not. A *Rosenbluth weight* for the move is then calculated. The Rosenbluth weight for each step i is given by:

$$W_i = \frac{n'}{n} W_{i-1} \quad (8.87)$$

where W_{i-1} is the weight for the previous step ($W_0 = 1$), n' is the number of available sites and n is the total number of neighbouring sites (not including the one occupied by the previous unit). In the case of our lattice, $W_1 = 2/4 = 1/2$. If site B is chosen then there is no site available for the third unit, and so the attempt has to be abandoned. If site A is chosen, its adjacent sites are examined to see which are free. In this case there is only one

free site where the third and final unit can be placed. The Rosenbluth weight for this step is $1/3 \times 1/2 = 1/6$. The overall statistical weight for the move is obtained by multiplying the number of successful trials by the Rosenbluth weight of each trial; as half the trials succeed, the statistical weight is therefore $1/2 \times 1/6 = 1/12$. This is exactly the same result that would be obtained with a conventional sampling scheme, though recall that in a conventional scheme only one trial in twelve results in a successful insertion. By contrast, with the configurational bias method the proportion of successful trials is one in two.

The configurational bias algorithm can be extended to take account of intermolecular interactions between the growing chain and its lattice neighbours. If the energy of segment i when occupying a particular site Γ is $v_{\Gamma}(i)$ then that site is chosen with a probability given by:

$$p_{\Gamma}(i) = \frac{\exp[-v_{\Gamma}(i)/k_{\text{B}}T]}{Z_i} \quad (8.88)$$

Z_i is the sum of the Boltzmann factors for all of the b positions considered:

$$Z_i = \sum_{\Gamma=1}^b \exp[-v_{\Gamma}(i)/k_{\text{B}}T] \quad (8.89)$$

The site can be chosen using a biased roulette wheel algorithm, in which the interval between 0 and 1 is divided into b adjacent segments each with a size proportional to the probabilities $p_1(i), p_2(i), \dots, p_b(i)$ (Figure 8.20). The site within whose interval a random number between 0 and 1 lies is the one chosen. The chain is thus biased towards those sites with a higher Boltzmann weighting; the sum of the Boltzmann factors plays the role of n' in Equation (8.87). The Rosenbluth weight for the entire chain (of length l) can be calculated as:

$$W_l = \exp[-\mathcal{V}_{\text{tot}}(l)/k_{\text{B}}T] \prod_{i=2}^l \frac{Z_i}{b} \quad (8.90)$$

where $\mathcal{V}_{\text{tot}}(l)$ is the total energy of the chain, equal to the sum of the individual segment energies $v_{\Gamma}(i)$. The average Rosenbluth weight is directly related to the excess chemical potential:

$$\mu_{\text{ex}} = k_{\text{B}}T \ln \langle W_l \rangle \quad (8.91)$$

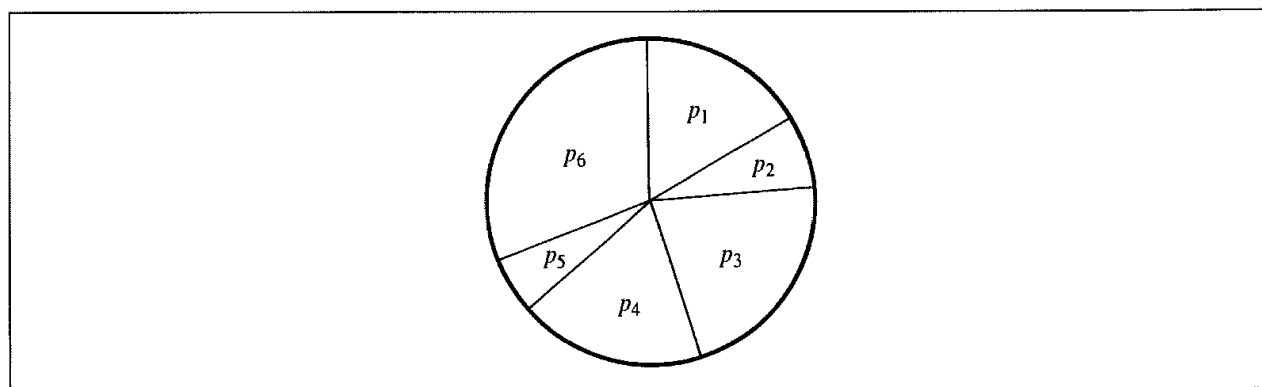


Fig 8.20: A biased roulette wheel chooses states according to their probabilities

If a segment has a zero Rosenbluth weight then growth of the chain is terminated. However, such chains must still be included in the averaging used to determine the excess chemical potential.

So far, we have only considered a fixed number of neighbouring sites for each segment. The method can be extended to cover fully flexible chains, where the set of possible neighbouring positions is infinite [De Pablo *et al.* 1992, 1993]. When growing each segment, a subset containing k random directions is chosen. These trial directions need not be uniformly distributed in space. For each of these orientations the energy $v_{\Gamma}(i)$ is calculated and so is the Boltzmann factor. An orientation is then chosen with probability:

$$p_{\Gamma}(i) = \frac{\exp[-v_{\Gamma}(i)/k_{\text{B}}T]}{\sum_{\Gamma=1}^k \exp[-v_{\Gamma}(i)/k_{\text{B}}T]} \quad (8.92)$$

The Rosenbluth factor is accumulated as follows:

$$W_i = W_{i-1} \frac{1}{k} \sum_{\Gamma=1}^k \exp[-v_{\Gamma}(i)/k_{\text{B}}T] \quad (8.93)$$

To implement this method it is necessary to determine the appropriate number of trial directions, k . If $k = 1$ then the method is equivalent to the original Widom particle insertion method. If k is too large then too much time is taken calculating the Rosenbluth factors for trial positions that are very close in phase space. Frenkel and colleagues have investigated how the choice of k influences the accuracy of the results and the efficiency with which those results were obtained [Frenkel *et al.* 1991]. The system they examined was of a flexible chain containing up to 20 segments in a moderately dense atomic fluid. The conventional particle insertion method failed completely for this system. Not surprisingly, the results showed that as the length of the chain increases so the number of random orientations that need to be considered also increases. At least four trial orientations were used at each step and k was chosen to increase logarithmically with the number of segments to be grown. The limiting value of k was considered to be reached when so many trials were required that the configurational bias method was no more efficient than alternative methods of regrowing chains, such as reptation algorithms. For example, for a 6-segment chain the proportion of accepted configurations (once the initial monomer had been inserted successfully) was 0.000 01% for $k = 1$, 3.2% for $k = 10$ and 35% for $k = 50$. For a 20-segment chain the proportion of accepted configurations was 0.0001% for $k = 20$, 0.66% for $k = 50$ and 2.0% for $k = 100$.

The Rosenbluth algorithm can also be used as the basis for a more efficient way to perform Monte Carlo sampling for fully flexible chain molecules [Siepmann and Frenkel 1992], which, as we have seen, is difficult to do as bond rotations often give rise to high energy overlaps with the rest of the system.

The configurational bias Monte Carlo method involves three types of move. Two of these are translational or rotational moves of the entire molecule, which are performed in the conventional way. The third type of move is a conformational change. A chain is selected at random and one of the segments within it is also randomly chosen. That part of the chain that lies above or below the segment (chosen with equal probability) is discarded and an

attempt is made to regrow the discarded portion. Let us consider first the case where each segment is restricted to a given number of discrete orientations, either because the chain is restricted to a lattice or because the model discretely samples the conformational space (e.g. it only permits *gauche* and *trans* conformations to a hydrocarbon chain). At each stage, the Boltzmann weights of the b discrete conformations are determined and one of the sites is chosen with a probability given by Equation (8.92). The Rosenbluth weight is determined for the growing chain using Equation (8.93).

Having generated a trial conformation it must be decided whether to accept it or not. To do this a random number is generated in the range 0–1 and compared with the ratio of the Rosenbluth weights for the trial conformation ($W_{l,\text{trial}}$) and the old conformation ($W_{l,\text{old}}$). The new chain is then accepted using the following criterion:

$$\text{rand}(0, 1) \leq \frac{W_{l,\text{trial}}}{W_{l,\text{old}}} \quad (8.94)$$

A similar approach can be adopted with continuous chains. Here it is also possible to enhance the sampling by guiding the choice of trial sites towards those with a particularly favourable intramolecular energy. This can be achieved by generating random vectors on the surface of a sphere of unit radius for each segment. The potential energy (angle-bending and torsional) for a bond directed along this vector is calculated. The vector is then accepted or rejected using the Metropolis criterion. If it is accepted, the vector is scaled to the appropriate bond length. This procedure continues until the desired number of trial sites have been generated. A trial site is then selecting using Boltzmann factors, which only consider the intra- and intermolecular non-bonded interactions of the sites with the chain and with the rest of the system. The Rosenbluth weights are similarly calculated and the move is accepted according to the ratio of the new and old Rosenbluth weights. Again, the correct choice of the number of trial sites is crucial to the efficiency of the method.

For branched molecules some modifications are required to the configurational bias method as described so far. This is because there may be bond angles which share the same central atom and torsion angles which have the same central two atoms in common. Thus in 2-methylalkanes the bond angles to the two terminal methyl groups share the 2-carbon atom. In 3,4-dimethylhexane there is a potential torsion problem. In the 'standard' configurational bias method one of the methyl groups would be grown, followed by the second. What is sometimes observed, however, is that the distributions of these bond angles is not equal (as it should be, as they are equivalent). Two possible ways to deal with this problem are to grow both atoms simultaneously [Dijkstra 1997] or to use a small Monte Carlo simulation to generate the trial positions [Vlugt *et al.* 1999]. When there are multiple torsion angles these two methods are not suitable; indeed, for a molecule such as 2,3-dimethylbutane the entire molecule must be generated in a single step. Martin and Siepmann suggested that it was possible to decouple the selection of the different energy terms [Martin and Siepmann 1999]. Suppose that the Lennard-Jones, torsional and bond-angle terms are decoupled. Then the probability of generating a particular configuration is given by:

$$P = \prod_{n=1}^{n_{\text{step}}} \left[\frac{\exp(-v_{\text{LJ}}(i)/k_{\text{B}}T)}{W_{\text{L}}(n)} \right] \left[\frac{\exp(-v_{\text{tor}}(j)/k_{\text{B}}T)}{W_{\text{T}}(n)} \right] \left[\frac{\exp(-v_{\text{bend}}(k)/k_{\text{B}}T)}{W_{\text{B}}(n)} \right] \quad (8.95)$$

The relevant Rosenbluth weights are:

$$W_L(n) = \sum_{i=1}^{n_{LJ}} \exp(-v_{LJ}(i)/k_B T) \quad (8.96)$$

$$W_T(n) = \sum_{j=1}^{n_{tor}} \exp(-v_{tor}(j)/k_B T) \quad (8.97)$$

$$W_B(n) = \sum_{k=1}^{n_{bend}} \exp(-v_{bend}(k)/k_B T) \quad (8.98)$$

where n_{LJ} , n_{tor} and n_{bend} are the number of trial sites for the Lennard-Jones, torsional and angle-bending interactions, respectively. Under these conditions, the move is accepted with a probability:

$$P_{acc} = \min \left[1, \frac{\prod_{n=1}^{n_{step}} W_L(n)_{new} W_T(n)_{new} W_B(n)_{new}}{\prod_{n=1}^{n_{step}} W_L(n)_{old} W_T(n)_{old} W_B(n)_{old}} \right] \quad (8.99)$$

The advantage of this decoupling method is that a large number of trial sites can be chosen for the computationally less expensive bond angle selection without increasing the cost of performing the other selections. Once the bond angle distribution has been chosen by this biased method, it is used as input to a biased selection of the torsional and Lennard-Jones interactions. An extension to this decoupling procedure involves grouping the torsional and Lennard-Jones together and having each biased selection of bond angles send many possible conformations forward to the next step. This coupling and decoupling of terms is claimed to provide a great deal of flexibility when designing a configurational bias scheme for any particular molecule and would also be applicable to force field models that included additional terms such as bond stretching or cross terms.

8.11.1 Applications of the Configurational Bias Monte Carlo Method

The CBMC method has been used to investigate a number of systems involving long-chain alkanes. Siepmann and McDonald examined a monolayer of 90 $\text{CH}_3(\text{CH}_2)_{15}\text{SH}$ molecules chemisorbed onto a gold surface [Siepmann and McDonald 1993a, b]. The thiol group forms a bond with the gold surface atoms, thus producing a high degree of surface ordering of the adsorbed molecules. Spectroscopic experiments indicated that the chains were tilted relative to the surface and adopted a predominantly *trans* conformation for the alkyl links. Both discrete and continuous versions of the configurational bias Monte Carlo method were employed; in the discrete model, each CH_2CH_2 segment was restricted to the *trans* and two *gauche* conformations. In the continuous simulation, six trial sites were used for each segment. The molecules were initially placed on a triangular lattice in an extended conformation perpendicular to the surface.

In the structure obtained at the end of the simulation the chains were ordered in an approximately hexagonal pattern. During equilibration *gauche* conformations were introduced into the alkyl chains, causing the system to tilt. However, once the molecules had all tilted the

gauche defects were gradually squeezed out to give chains with predominantly *trans* links. The final configuration is shown in Figure 8.21 (colour plate section)

The configurational bias Monte Carlo method has also been used to investigate the adsorption of alkanes in zeolites. Such systems are of especial interest in the petrochemical industry. One interesting experimental result obtained for the zeolite silicalite was that short-chain alkanes (C_1 to C_5) and long-chain alkanes (C_{10}) have simple adsorption isotherms but hexane and heptane show kinked isotherms. Such systems are obvious candidates for theoretical investigations because the experimental data is difficult and time-consuming to obtain. Moreover, the simulation can often provide a detailed molecular explanation for the observed behaviour. The simulation of such systems is difficult using conventional methods; the Monte Carlo method suffers from the problems of low acceptance ratios or a very slow exploration of phase space, and long simulation times would be required with molecular dynamics as the diffusion of long-chain alkanes is very slow. The configurational bias Monte Carlo method enabled effective and efficient simulations to be performed, providing both thermodynamic properties and the spatial distribution of the molecules within the zeolite [Smit and Siepmann 1994; Smit and Maesen 1995]. The adsorption isotherms (i.e. the number of molecules adsorbed as a function of the applied pressure) were calculated using grand canonical simulations in which the zeolite was coupled to a reservoir at constant temperature and chemical potential.

Silicalite has both straight and zig-zag channels, which are connected via intersections (Figure 8.22). An analysis of the configurations showed that the distribution of a short alkane, such as butane, was approximately equal between the two types of channel. However, as the length of the alkane chain was increased, so there was a greater probability of finding it in a straight channel than in a zig-zag channel. Hexane is an interesting case, for its length is almost equal to the period of the zig-zag channels. At low pressure the hexane molecules move freely in the zig-zag channels and occupy the intersections for part of the time. To fill the zeolite with hexane it is first necessary for the alkane molecules to occupy just the zig-zag channels and not the intersections. This is accompanied by a loss of entropy, which must be compensated for by a higher chemical potential and so gives the kinked isotherm. The straight channels can then be filled with hexane. Different behaviour

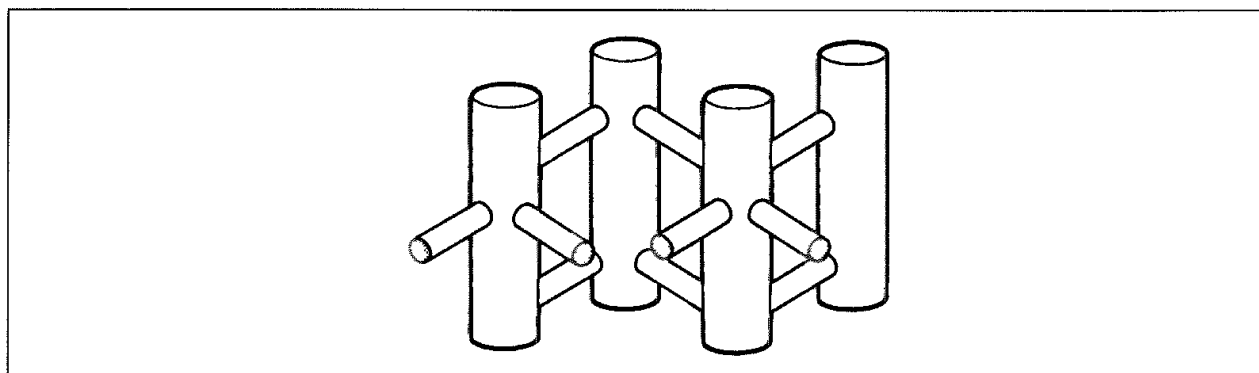


Fig 8.22 Schematic structure of the zeolite silicalite showing the straight and zig-zag channels (Figure adapted from Smit B and J I Siepmann 1994 *Simulating the Adsorption of Alkanes in Zeolites* Science 264 1118–1120)

is observed for smaller alkanes because more than one molecule can occupy the zig-zag channels. Longer alkanes always partially occupy the intersection and so there is no benefit from freezing the molecules in the zig-zag channels. It is also possible to simulate the behaviour of branched alkanes [Vlugt *et al.* 1998] and compare these with their linear equivalents. Thus, whereas *n*-butane has an equal probability of being in either channel, isobutane has a preference for the intersection. Moreover, once all the intersections are full it requires considerable energy to place isobutane elsewhere in the zeolite. This requires a much higher pressure, giving rise to an inflection in the adsorption isotherms

8.12 Simulating Phase Equilibria by the Gibbs Ensemble Monte Carlo Method

The most 'obvious' way to investigate phase equilibria is to set up an appropriate system with a conventional simulation technique. Unfortunately, simulations of systems with more than one phase usually require inordinate amounts of computer time. There are several reasons why the use of conventional simulation methods to investigate phase equilibria is difficult. First, it would take a very long time to equilibrate such a system, which would need to separate into its two phases (e.g. liquid and vapour). The properties of the fluid in the interfacial region differ substantially from the properties in the bulk and so to obtain a 'bulk' measurement all of the interfacial atoms must be ignored. Smit has calculated the percentage of the number of particles in the interfacial region for systems of varying sizes, these percentages range from 10% in the interfacial region for a system of 50 000 particles to 95% for a system of 100 particles [Smit 1993]. To simulate phase equilibria directly would thus require long simulations to be performed on systems containing many particles.

The Gibbs ensemble Monte Carlo simulation method, invented by Panagiotopoulos [Panagiotopoulos 1987], enables phase equilibria to be studied directly using small numbers of particles. Rather than trying to form an interface within a single simulation, two simulation boxes are used, each representing one of the two phases. There is no physical interface between the two boxes, which are subject to the usual periodic boundary conditions (Figure 8.23). Three types of move are possible. The first type of move comprises particle displacements within each box, as in a conventional Monte Carlo simulation. The second type of move involves volume changes of the two boxes by equal and opposite amounts so that the total volume of the system remains constant. The third type of move involves the removal of a particle from one box and its attempted placement in the other box. This is identical to the Widom insertion method for calculating the chemical potential. Indeed, as the energy of the inserted particle must be calculated, it is possible to determine the chemical potential in the Gibbs ensemble without any additional computational cost. These three types of move are often performed in strict order, but it may be better to choose each type of move at random, ensuring that, on average, the appropriate numbers of each type of move are made.

The properties of the Gibbs ensemble Monte Carlo simulation method have been examined in great detail using simple systems such as the Lennard-Jones fluid and simple gases. A

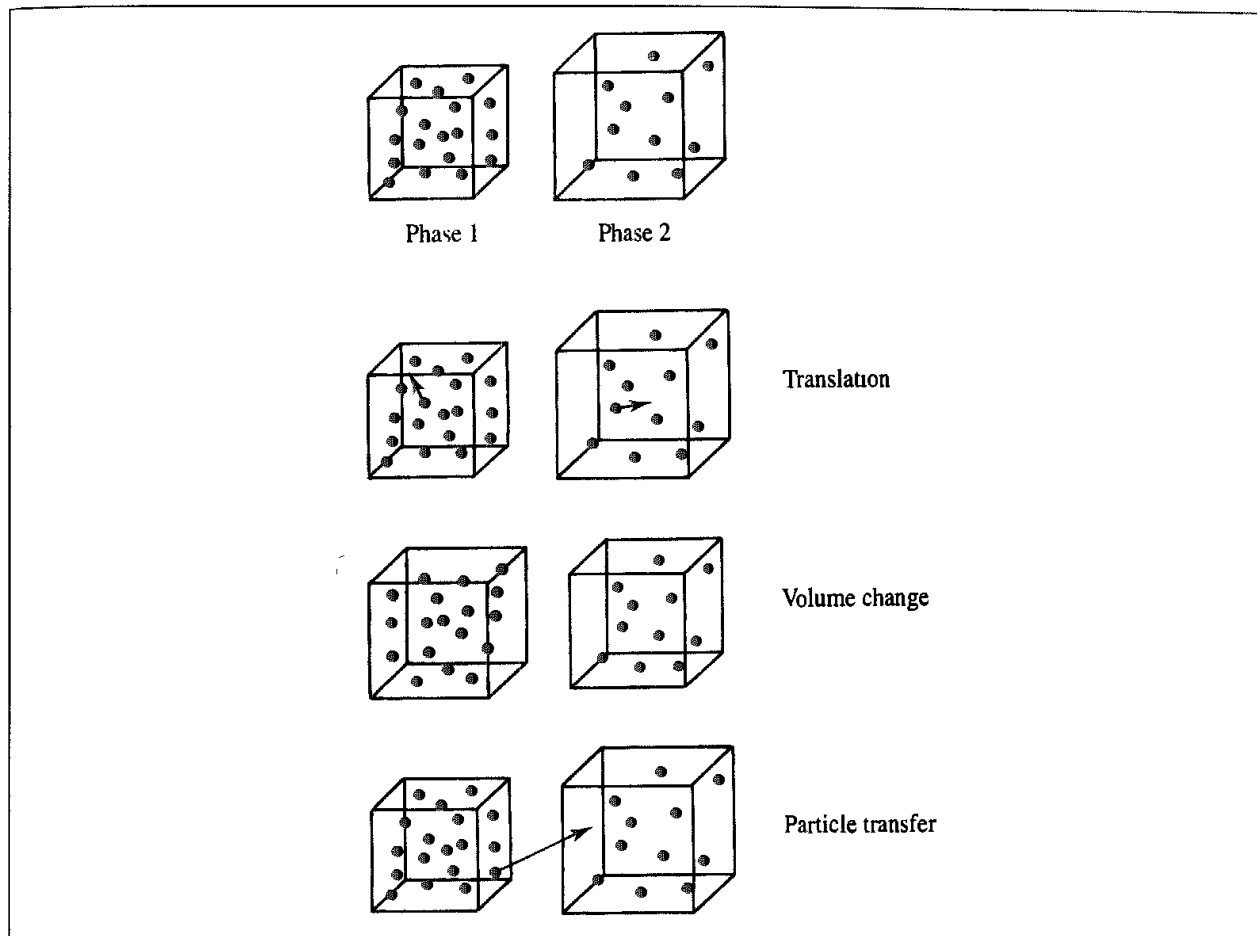


Fig 8 23 The Gibbs ensemble Monte Carlo simulation method uses one box for each of the two phases. Three types of move are permitted translations within either box; volume changes (keeping the total volume constant) and transfer of a particle from one box to the other

particularly exciting development is the use of the configurational bias Monte Carlo method in conjunction with the Gibbs ensemble method to construct the phase diagrams of complex, long-chain molecules. For example, the vapour-liquid phase equilibria of *n*-pentane and *n*-octane have been investigated by Siepmann, Karaborni and Smit using this combined approach on systems containing 200 pentane or 160 octane molecules [Siepmann *et al.* 1993a]. The calculated properties of these two systems agreed very well with the available experimental data, particularly for the shorter alkane. Their studies were subsequently extended to much longer alkanes (up to C_{48}) [Siepmann *et al.* 1993b]. One particularly noteworthy result was that the density at the critical point increased with the length of the carbon chain up to *n*-octane but then *decreased* as the chain increased in length. Until shortly before the simulations were performed it had been assumed that the critical density for longer chains could be extrapolated from the experimental data obtained with short chains under the assumption that the critical density increased with the length of the chain. Later experiments were able to examine longer chains and did indeed demonstrate that the critical point density passed through a maximum at octane and then decreased for shorter chain lengths.

8.13 Monte Carlo or Molecular Dynamics?

In principle, the modeller has the choice of using either the Monte Carlo or molecular dynamics technique for a given simulation. In practice one technique must be chosen over the other. Sometimes the decision is a trivial one, for example because a suitable program is readily available. In other cases there are clear reasons for choosing one method instead of the other. For example, molecular dynamics is required if one wishes to calculate time-dependent quantities such as transport coefficients. Conversely, Monte Carlo is often the most appropriate method to investigate systems in certain ensembles; for example, it is much easier to perform simulations at exact temperatures and pressures with the Monte Carlo method than using the sometimes awkward and ill-defined constant temperature and constant pressure molecular dynamics simulation methods. The Monte Carlo method is also well suited to certain types of models such as the lattice models.

The two methods can differ in their ability to explore phase space. A Monte Carlo simulation often gives much more rapid convergence of the calculated thermodynamic properties of a simple molecular liquid (modelled as a rigid molecule), but it may explore the phase space of large molecules very slowly due to the need for small steps unless special techniques such as the configurational bias Monte Carlo method are employed. However, the ability of the Monte Carlo method to make non-physical moves can significantly enhance its capacity to explore phase space in appropriate cases. This may arise for simulations of isolated molecules, where there are a number of minimum energy states separated by high barriers. Molecular dynamics may not be able to cross the barriers between the conformations sufficiently often to ensure that each conformation is sampled according to the correct statistical weight. Molecular dynamics advances the positions and velocities of all the particles simultaneously and it can be very useful for exploration of the local phase space whereas the Monte Carlo method may be more effective for conformational changes, which jump to a completely different area of phase space.

Given that the two techniques in some ways complement each other in their ability to explore phase space, it is not surprising that there has been some effort to combine the two methods. Some of the techniques that we have considered in this chapter and in Chapter 7 incorporate elements of the Monte Carlo and molecular dynamics techniques. Two examples are the stochastic collisions method for performing constant temperature molecular dynamics, and the force bias Monte Carlo method. More radical combinations of the two techniques are also possible.

An obvious way to combine Monte Carlo with molecular dynamics is to use each technique for the most appropriate part of a simulation. For example, when simulating a solvated macromolecule, the equilibration phase is usually performed in a series of stages. In the first stage, the solute is kept fixed while the solvent molecules (and any ions, if present) are allowed to move under the influence of the solute's electrostatic field. This solvent equilibration may often be performed more effectively using a Monte Carlo simulation as the solvent and ions do not have any appreciable conformational flexibility. To simulate the whole system, molecular dynamics is then the most appropriate method. Such a protocol has been used to perform long simulations of DNA molecules [Swaminathan *et al.* 1991].

A variety of hybrid molecular dynamics/Monte Carlo methods have been devised, in which the simulation algorithm alternates between molecular dynamics and Monte Carlo. The aim of such methods is to achieve better sampling, and thereby more rapid convergence of thermodynamic properties. *In extremis*, each molecular dynamics (or stochastic dynamics) step is followed by a Monte Carlo step, the velocities being unaffected by acceptance or rejection of the Monte Carlo step. Such a method has been devised by Guarnieri and Still [Guarnieri and Still 1994]. An alternative is to perform a block of molecular dynamics steps to generate a new state, which is then accepted or rejected on the basis of the total energy (potential plus kinetic) using the usual Metropolis criterion. If the new coordinates are rejected then the original coordinates from the start of the block are restored and molecular dynamics is run again, but with an entirely new set of velocities that is chosen from a Gaussian distribution. This approach is very similar to the stochastic collisions method for temperature control discussed in Section 7.7.1 but with the addition of a Monte Carlo acceptance or rejection step [Duane *et al.* 1987]. A simulation using this hybrid algorithm samples from the canonical ensemble (constant temperature) and was shown by Clamp and colleagues to be more effective than conventional molecular dynamics or Monte Carlo methods for exploring the phase space of both simple model systems and proteins [Clamp *et al.* 1994].

Appendix 8.1 The Marsaglia Random Number Generator

The Marsaglia random number generator [Marsaglia *et al.* 1990] is known as a *combination generator* because it is constructed from two different generators. It has a period of about 2^{144} . The first generator is a lagged Fibonacci generator that performs the following binary operation on two real numbers x and y :

$$x \bullet y = x - y \text{ if } x \geq y; \quad x \bullet y = x - y - 1 \text{ if } x < y \quad (8.100)$$

The values x and y are chosen from numbers earlier in the sequence, so that the n th value in the sequence is calculated by:

$$x_n = x_{n-r} \bullet x_{n-s} \quad (8.101)$$

r and s are the *lags*, which are chosen to give numbers that are satisfactorily random and have a long period. Marsaglia chose $r = 97$ and $s = 33$. The algorithm does therefore require the last 97 numbers to be stored at all stages.

The second generator is an arithmetic sequence method that generates random numbers using the following mathematical operation:

$$c \circ d = c - d \text{ if } c \geq d, \quad c \circ d = c - d + 16777213/16777216 \text{ if } c < d \quad (8.102)$$

The n th value in this sequence is given by:

$$c_n = c_{n-1} \circ (7654321/16777216) \quad (8.103)$$

The n th number, U_n , in the combined sequence is then obtained as

$$U_n = x_n \circ c_n \quad (8.104)$$

The c sequence requires one initial seed value and the x sequence requires 97 initial seeds (which should themselves be reasonably random). These can be supplied by the user but in the published algorithm these 97 values were obtained from another combination generator comprising a lagged Fibonacci generator and a congruential algorithm.

Further Reading

- Adams D J 1983. Introduction to Monte Carlo Simulation Techniques In Perran J W (Editor) *Physics of Superionic Conductors and Electrode Materials*. New York, Plenum, pp 177–195.
- Allen M P and D J Tildesley 1987 *Computer Simulation of Liquids* Oxford, Oxford University Press.
- Colbourn E A (Editor) 1994 *Computer Simulation of Polymers* Harlow, Longman.
- Frenkel D Monte Carlo Simulations: A Primer In van Gunsteren W F, P K Weiner and A J Wilkinson (Editors). *Computer Simulation of Biomolecular Systems* Volume 2 Leiden, ESCOM, pp. 37–66
- Galaiatsatos V 1995 Computational Methods for Modelling Polymers. An Introduction. In Lipkowitz K B and D B Boyd (Editors) *Reviews in Computational Chemistry* Volume 6. New York, VCH Publishers, pp. 149–208.
- Kaols M H and P A Whitlock 1986. *Monte Carlo Methods, Volume 1. Basics*. New York, John Wiley & Sons.
- Kermer K 1993 Computer Simulation of Polymers. In Allen M P and D J Tildesley (Editors) *Computer Simulation in Chemical Physics*. Dordrecht, Kluwer, NATO ASI Series 397.397–459
- Rubinstein R Y 1981. *Simulation and Monte Carlo Methods*. New York, John Wiley & Sons.

References

- Barker J A and R O Watts 1969 Structure of Water, A Monte Carlo Calculation *Chemical Physics Letters* 3:144–145.
- Baschnagel J, K Binder, W Paul, M Laso, U Suter, I Batoulis, W Jilge and T Bürger 1991. On the Construction of Coarse-Grained Models for Linear Flexible Polymer Chains – Distribution Functions for Groups of Consecutive Monomers. *Journal of Chemical Physics* 95:6014–6025.
- Clamp M E, P G Baker, C J Stirling and A Brass 1994 Hybrid Monte Carlo: An Efficient Algorithm for Condensed Matter Simulation. *Journal of Computational Chemistry* 15:838–846.
- Cracknell R F, D Nicholson and N Quirke 1994 A Grand Canonical Monte Carlo Study of Lennard-Jones Mixtures in Slit Pores; 2 Mixtures of Two-Centre Ethane with Methane *Molecular Simulation* 13:161–175.
- De Pablo J J, M Laso, J I Siepmann and U W Suter 1993 Continuum-Configurational Bias Monte Carlo Simulations of Long-chain Alkanes. *Molecular Physics* 80 55–63
- De Pablo J J, M Laso, and U W Suter 1992. Estimation of the Chemical Potential of Chain Molecules by Simulation. *Journal of Chemical Physics* 96:6157–6162.
- Dijkstra M 1997 Confined Thin Films of Linear and Branched Alkanes. *Journal of Chemical Physics* 107 3277–3288.
- Duane S, A D Kennedy and B J Pendleton 1987. Hybrid Monte Carlo *Physics Letters* B195:216–222
- Flory P J 1969. *Statistical Mechanics of Chain Molecules*. New York, Interscience.
- Frantz D D, D L Freeman and J D Doll 1990 Reducing Quasi-ergodic Behavior in Monte Carlo Simulations by J-walking Applications to Atomic Clusters *Journal of Chemical Physics* 93:2769–2784
- Frenkel D D, C A M Mooij and B Smit 1991 Novel Scheme to Study Structural and Thermal Properties of Continuously Deformable Materials. *Journal of Physics Condensed Matter* 3:3053–3076.

- Guarnieri F and W C Still 1994. A Rapidly Convergent Simulation Method Mixed Monte Carlo/Stochastic Dynamics. *Journal of Computational Chemistry* **15** 1302–1310
- Marsaglia G, A Zaman and W W Tsang 1990. Towards a Universal Random Number Generator *Statistics and Probability Letters* **8**:35–39.
- Martin M G and J I Siepmann 1999. Novel Configurational-bias Monte Carlo Method for Branched Molecules Transferable Potentials for Phase Equilibria. 2 United-atom Description of Branched Alkanes *Journal of Physical Chemistry* **103**:4508–4517
- Metropolis N, A W Rosenbluth, M N Rosenbluth, A H Teller and E Teller 1953 Equation of State Calculations by Fast Computing Machines. *Journal of Chemical Physics* **21**:1087–1092
- Okamoto Y and U H E Hansmann 1995. Thermodynamics of Helix-coil Transitions Studied by Multicanonical Algorithms. *Journal of Physical Chemistry* **99**:11276–11287
- Panagiotopoulos A Z 1987. Direct Determination of Phase Coexistence Properties of Fluids by Monte Carlo Simulation in a New Ensemble. *Molecular Physics* **61**:813–826
- Pangali C, M Rao and B J Berne 1978 On a Novel Monte Carlo Scheme for Simulating Water and Aqueous Solutions *Chemical Physics Letters* **55** 413–417.
- Rao M and B J Berne 1979 On the Force Bias Monte Carlo Simulation of Simple Liquids. *Journal of Chemical Physics* **71**:129–132
- Rosenbluth M N and A W Rosenbluth 1955. Monte Carlo Calculation of the Average Extension of Molecular Chains. *Journal of Chemical Physics* **23**:356–359
- Rosky P J, J D Doll and H L Friedman 1978. Brownian Dynamics as Smart Monte Carlo Simulation. *Journal of Chemical Physics* **69**:4628–4633.
- Senderowitz H, F Guarnieri and W C Still 1995. A Smart Monte Carlo Technique for Free Energy Simulations of Multicanonical Molecules. Direct Calculations of the Conformational Populations of Organic Molecules. *Journal of the American Chemical Society* **117** 8211–8219
- Sharp W E and C Bays 1992 A Review of Portable Random Number Generators. *Computers and Geosciences* **18**:79–87
- Siepmann J I 1990. A Method for the Direct Calculation of Chemical Potentials for Dense Chain Systems. *Molecular Physics* **70**:1145–1158.
- Siepmann J I and D Frenkel 1992 Configurational Bias Monte Carlo: A New Sampling Scheme for Flexible Chains *Molecular Physics* **75**:59–70.
- Siepmann J I, S Karabornu and B Smit 1993a Vapor-Liquid Equilibria of Model Alkanes *Journal of the American Chemical Society* **115** 6454–6455.
- Siepmann J I, S Karabornu and B Smit 1993b. Simulating the Crucial Behaviour of Complex Fluids *Nature* **365**:330–332
- Siepmann J I and I R McDonald 1993a Domain Formation and System-size Dependence in Simulations of Self-assembled Monolayers *Langmuir* **9**:2351–2355.
- Siepmann J I and I R McDonald 1993b Monte Carlo Study of the Properties of Self-assembled Monolayers Formed by Adsorption of $\text{CH}_3(\text{CH}_2)_{15}\text{SH}$ on the (111) Surface of Gold. *Molecular Physics* **79**:457–473.
- Smit B 1993 Computer Simulation in the Gibbs Ensemble. In Allen M P and D J Tildesley (Editors) *Computer Simulation in Chemical Physics* Dordrecht, Kluwer. NATO ASI Series 397, pp. 173–210
- Smit B and T L M Maesen 1995. Commensurate 'Freezing' of Alkanes in the Channels of a Zeolite. *Nature* **374**:42–44.
- Smit B and J I Siepmann 1994. Simulating the Adsorption of Alkanes in Zeolites *Science* **264** 1118–1120.
- Stephenson G 1973 *Mathematical Methods for Science Students*. London, Longman
- Swaminathan S, G Ravishanker and D L Beveridge 1991. Molecular Dynamics of B-DNA Including Water and Counterions - A 140-ps Trajectory for d(CGCGAATTCGCG) Based on the Gromos Force Field *Journal of the American Chemical Society* **113**:5027–5040

- Verdier P H and W H Stockmayer 1962. Monte Carlo Calculations on the Dynamics of Polymers in Dilute Solution. *Journal of Chemical Physics* **36**:227-235
- Vesely F J 1982. Angular Monte Carlo Integration Using Quaternion Parameters: A Spherical Reference Potential for CCl₄ *Journal of Computational Physics* **47** 291-296
- Vlugt T J H, R Krishna and B Smit 1999 Molecular Simulations of Adsorption Isotherms for Linear and Branched Alkanes and Their Mixtures in Silicalite *Journal of Physical Chemistry* **103**:1102-1118
- Vlugt T J H, W Zhu, F Kapteijn, J A Moulijn, B Smit and R Krishna 1998 Adsorption of Linear and Branched Alkanes in the Zeolite Silicalite-1. *Journal of the American Chemical Society* **120**:5599-5600.
- Widom B 1963. Topics in the Theory of Fluids. *Journal of Chemical Physics* **39** 2808-2812
- Xu H and B J Berne 1999. Multicanonical Jump Walking: A Method for Efficiently Sampling Rough Energy Landscapes. *Journal of Chemical Physics* **110** 10299-10306