4
Monte Carlo methods

4.1 Introduction

The Monte Carlo method was first developed by von Neumann, Ulam, and Metropolis at the end of the Second World War to study the diffusion of neutrons in fissionable material. The name 'Monte Carlo', coined by Metropolis in 1947 and used in the title of a paper describing the early work at Los Alamos (Metropolis and Ulam, 1949), derives from the extensive use of random numbers in the approach.

The method is based on the idea that a determinate mathematical problem can be treated by finding a probabilistic analogue which is then solved by a stochastic sampling experiment (von Neumann and Ulam, 1945). For example, the configurational energy of a liquid can be calculated by solving the coupled equations of motion of the atoms and averaging over time. Alternatively, one can set up an ensemble of states of the liquid, choosing individual states with the appropriate probability, and calculating the configurational energy by averaging uniformly over the ensemble. These sampling experiments involve the generation of random numbers followed by a limited number of arithmetic and logical operations, which are often the same at each step. These are tasks that are well suited to a computer and the growth in the importance of the method can be linked to the rapid development of these machines. The arrival of the MANIAC computer at Los Alamos in 1952 prompted the study of the many-body problem by Metropolis et al. (1953) and the development of the Metropolis Monte Carlo method (Wood, 1986), which is the subject of this chapter. Today, the Monte Carlo method is widely applied in all branches of the natural and social sciences and is, arguably, 'the most powerful and commonly used technique for analysing complex problems' (Rubinstein, 1981).

4.2 Monte Carlo integration

As outlined in Chapter 2, the Metropolis Monte Carlo method aims to generate a trajectory in phase space which samples from a chosen statistical ensemble. There are several difficulties involved in devising such a prescription and making it work for a system of molecules in a liquid. So we take care to introduce the Monte Carlo method with a simple example.

Consider the problem of finding the volume, V, of the solid bounded by the coordinate axes, and the planes $z = 1 + y$ and $2x + y = 2$. This is the volume below the dark-grey

Computer Simulation of Liquids. Second Edition. M. P. Allen and D. J. Tildesley.

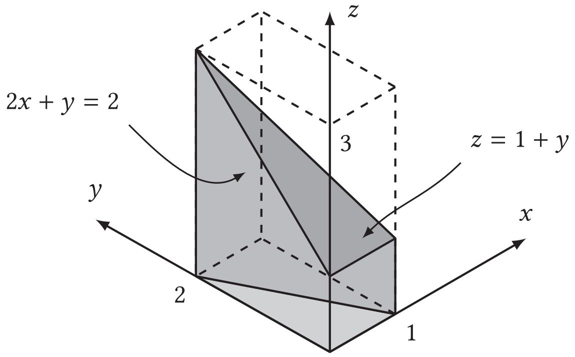

Fig. 4.1 The solid volume V, below the dark-grey triangle and above the light-grey triangle can be evaluated from the integral in eqn (4.1).

triangle in Fig. 4.1. The volume is exactly given by

$$V = \int_0^1 \mathrm{d}x \int_0^{2-2x} \mathrm{d}y\,(1+y) = \frac{5}{3}. \tag{4.1}$$

Here we consider two simple Monte Carlo methods to evaluate it numerically.

4.2.1 Hit and miss

In a Monte Carlo evaluation of the integral, the volume of interest, V, would be surrounded by a sampling region of a simple geometry whose volume is known. In this case we choose the rectangular box of volume $V_0 = 6$ indicated by the dashed lines in Fig. 4.1. A random position is chosen within the rectangular box $\mathbf{r}_\tau = (x_\tau, y_\tau, z_\tau)$; this is a shot, τ. If this shot is within the required volume, V, it is a hit. If a total of τ_{shot} shots are fired and τ_{hit} hits scored then

$$V = \frac{V_0\,\tau_{\text{hit}}}{\tau_{\text{shot}}}. \tag{4.2}$$

The key to this method is the generation of $3\tau_{\text{shot}}$ random numbers from a uniform distribution. A sample program to perform this integration is given in Code 4.1. `RANDOM_SEED()` and `RANDOM_NUMBER()` are built-in Fortran functions for generating uniform random numbers on $(0, 1)$. Random number generators are discussed briefly in Appendix E.

4.2.2 Sample mean integration

Hit-and-miss integration is conceptually easy to understand but the sample mean method is more generally applicable and offers a more accurate estimate for most integrals (Hammersley and Handscomb, 1964; Rubinstein, 1981). In this case the integral of interest

$$F = \int_{x_1}^{x_2} \mathrm{d}x\, f(x) \tag{4.3}$$

is rewritten as

$$F = \int_{x_1}^{x_2} \mathrm{d}x \left(\frac{f(x)}{\rho(x)}\right) \rho(x) \tag{4.4}$$

Code 4.1 Hit-and-miss integration

This program is also available online in the file `hit_and_miss.f90`.

```
PROGRAM hit_and_miss
  USE, INTRINSIC :: iso_fortran_env, ONLY : output_unit
  IMPLICIT NONE
  REAL                          :: v
  REAL, DIMENSION(3)            :: r, zeta
  REAL, DIMENSION(3), PARAMETER :: r_0 = [1.0, 2.0, 3.0]
  REAL,               PARAMETER :: v_0 = PRODUCT(r_0)
  INTEGER                       :: tau, tau_shot, tau_hit

  CALL RANDOM_SEED()
  tau_hit  = 0
  tau_shot = 1000000

  DO tau = 1, tau_shot
     CALL RANDOM_NUMBER ( zeta(:) )  ! uniform in range (0,1)
     r = zeta * r_0                  ! uniform in v_0
     IF (    r(2) < ( 2.0 - 2.0*r(1) ) .AND. &
          & r(3) < ( 1.0 + r(2) )    ) THEN ! in polyhedron
        tau_hit = tau_hit + 1
     END IF
  END DO
  v = v_0 * REAL ( tau_hit ) / REAL ( tau_shot )
  WRITE (UNIT=output_unit,FMT='(a,f10.5)') 'Estimate␣=␣', v
END PROGRAM hit_and_miss
```

where $\rho(x)$ is an arbitrary probability density function. Consider performing a number of trials τ, each consisting of choosing a random number ζ_τ, from the distribution $\rho(x)$ in the range (x_1, x_2). Then

$$F = \left\langle \frac{f(\zeta_\tau)}{\rho(\zeta_\tau)} \right\rangle_{\text{trials}} \tag{4.5}$$

where the brackets represent an average over all trials. A simple application would be to choose $\rho(x)$ to be uniform, that is,

$$\rho(x) = \frac{1}{(x_2 - x_1)} \qquad x_1 \leq x \leq x_2 \tag{4.6}$$

and then the integral F can be estimated as

$$F \approx \frac{(x_2 - x_1)}{\tau_{\text{max}}} \sum_{\tau=1}^{\tau_{\text{max}}} f(\zeta_\tau). \tag{4.7}$$

Code 4.2 Sample mean integration

This program is also available online in the file `sample_mean.f90`.

```
PROGRAM sample_mean
  IMPLICIT NONE
  REAL                          :: v, f
  REAL, DIMENSION(2)            :: r, zeta
  REAL, DIMENSION(2), PARAMETER :: r_0 = [1.0, 2.0]
  REAL,               PARAMETER :: a_0 = PRODUCT(r_0)
  INTEGER                       :: tau, tau_max

  CALL RANDOM_SEED()
  tau_max = 1000000

  f = 0.0
  DO tau = 1, tau_max
     CALL RANDOM_NUMBER ( zeta ) ! uniform in (0,1)
     r = zeta * r_0 ! uniform in xy rectangle
     IF ( r(2) < 2.0-2.0*r(1) ) THEN ! in xy triangle
        f = f + ( 1.0 + r(2) )       ! value of z
     END IF
  END DO
  v = a_0 * f / REAL ( tau_max )
  WRITE (UNIT=output_unit,FMT='(a,f10.5)') 'Estimate␣=␣', v
END PROGRAM sample_mean
```

The method can be readily generalized to multiple integrals. A Monte Carlo sample mean evaluation of the volume in Fig. 4.1 can be performed with the program in Code 4.2. In this case, the integration is carried out by selecting points in a rectangle in the xy plane. The function f to be summed is zero if the points lie outside the light-grey triangle, and equal to $z = 1 + y$ inside. The sample mean method can be used to calculate many of the multiple integrals of liquid state theory, for example the long-range correction to the three-body potential energy in eqn (2.149).

4.2.3 A direct evaluation of the partition function?

For the multidimensional integrals of statistical mechanics, the sample mean method, with a suitable choice of $\rho(x)$, is the only sensible solution. To understand this, we consider the evaluation of the configurational integral $Z_{NVT} = \int \mathrm{d}\mathbf{r} \exp(-\beta\mathcal{V})$, eqn (2.26), for a system of, say, $N = 100$ molecules in a cube of side L. The sample mean approach to this integral, using a uniform distribution could involve the following trials:

(a) pick a point at random in the 300-dimensional configuration space, by generating 300 random numbers, on $(-\frac{1}{2}L,\ \frac{1}{2}L)$, which, taken in triplets, specify the coordinates of each molecule;

(b) calculate the potential energy, $\mathcal{V}(\tau)$, and hence the Boltzmann factor for this configuration.

This procedure is repeated for many trials and the configurational integral is estimated using

$$Z_{NVT} \approx \frac{V^N}{\tau_{\max}} \sum_{\tau=1}^{\tau_{\max}} \exp\bigl(-\beta\mathcal{V}(\tau)\bigr). \tag{4.8}$$

In principle, the number of trials $\tau_{\max}$ may be increased until Z_{NVT} is estimated to the desired accuracy. Unfortunately, a large number of the trials would give a very small contribution to the average. In such a random configuration, molecules would overlap, $\mathcal{V}(\tau)$ would be large and positive, and the Boltzmann factor vanishingly small. An accurate estimation of Z_{NVT} for a dense liquid using a uniform sample mean method is not possible, although methods of this type have been used to examine the structural properties of the hard-sphere fluid at low densities (Alder et al., 1955). The difficulties in the calculation of Z_{NVT} apply equally to the calculation of ensemble averages such as

$$\bigl\langle\mathcal{A}\bigr\rangle_{NVT} = \frac{\int \mathrm{d}\mathbf{r}\mathcal{A}\exp\bigl(-\beta\mathcal{V}\bigr)}{\int \mathrm{d}\mathbf{r}\exp\bigl(-\beta\mathcal{V}\bigr)} \approx \frac{\sum_{\tau=1}^{\tau_{\max}} \mathcal{A}(\tau)\exp\bigl(-\beta\mathcal{V}(\tau)\bigr)}{\sum_{\tau=1}^{\tau_{\max}} \exp\bigl(-\beta\mathcal{V}(\tau)\bigr)}, \tag{4.9}$$

if we attempt to estimate the numerator and denominator separately by using the uniform sample mean method. However, at realistic liquid densities the problem might be solved using a sample mean integration where the random coordinates are chosen from a non-uniform distribution. This method of 'importance sampling' is discussed in the next section.

4.3 Importance sampling

Importance sampling techniques choose random numbers from a distribution $\rho(x)$, which allows the function evaluation to be concentrated in the regions of space that make important contributions to the integral. Consider the canonical ensemble. In this case the desired integral is

$$\langle\mathcal{A}\rangle = \int \mathrm{d}\boldsymbol{\Gamma}\rho_{NVT}(\boldsymbol{\Gamma})\mathcal{A}(\boldsymbol{\Gamma})$$

that is, the integrand is $f = \rho_{NVT}\mathcal{A}$. By sampling configurations at random, from a chosen distribution $\rho(\boldsymbol{\Gamma})$ we can estimate the integral as

$$\langle\mathcal{A}\rangle_{NVT} = \langle\mathcal{A}\,\rho_{NVT}/\rho\rangle_{\text{trials}}. \tag{4.10}$$

For most functions $\mathcal{A}(\boldsymbol{\Gamma})$, the integrand will be significant where $\rho_{NVT}(\boldsymbol{\Gamma})$ is significant. In these cases choosing $\rho(\boldsymbol{\Gamma}) = \rho_{NVT}(\boldsymbol{\Gamma})$ should give a good estimate of the integral. In this case

$$\langle\mathcal{A}\rangle_{NVT} = \langle\mathcal{A}\rangle_{\text{trials}}. \tag{4.11}$$

(This is not always the best choice, and sometimes we choose alternative distributions $\rho(\boldsymbol{\Gamma})$; see Section 9.2.3.)

Such a method, with $\rho(\boldsymbol{\Gamma}) = \rho_{NVT}(\boldsymbol{\Gamma})$ was originally developed by Metropolis et al. (1953). The problem is not solved, simply rephrased. The difficult job is finding a method of generating a sequence of random states so that by the end of the simulation each state has occurred with the appropriate probability. It turns out that it is possible to do this without ever calculating the normalizing factor for ρ_{NVT}, that is, the partition function (see eqns (2.11)–(2.13)).

The solution is to set up a Markov chain of states of the liquid, which is constructed so that it has a limiting distribution of $\rho_{NVT}(\boldsymbol{\Gamma})$. A Markov chain is a sequence of trials that satisfies two conditions:

(a) The outcome of each trial belongs to a finite set of outcomes, $\{\boldsymbol{\Gamma}_1, \boldsymbol{\Gamma}_2, \ldots\}$, called the state space.

(b) The outcome of each trial depends only on the outcome of the trial that immediately precedes it.

Two such states $\boldsymbol{\Gamma}_m$ and $\boldsymbol{\Gamma}_n$ are linked by a transition probability π_{mn} which is the probability of going from state m to state n. The properties of a Markov chain are best illustrated with a simple example. Suppose the reliability of your computer follows a certain pattern. If it is up and running on one day it has a 60 % chance of running correctly on the next. If however, it is down, it has a 70 % chance of also being down the next day. The state space has two components, up (↑) and down (↓), and the transition matrix has the form

$$\boldsymbol{\pi} = \begin{array}{cc} & \begin{array}{cc} \uparrow & \downarrow \end{array} \\ \begin{array}{c} \uparrow \\ \downarrow \end{array} & \begin{pmatrix} 0.6 & 0.4 \\ 0.3 & 0.7 \end{pmatrix} \end{array}. \tag{4.12}$$

If the computer is equally likely to be up or down to begin with, then the initial probability can be represented as a vector, which has the dimensions of the state space

$$\boldsymbol{\rho}^{(1)} = \begin{array}{c} \begin{array}{cc} \uparrow & \downarrow \end{array} \\ \begin{pmatrix} 0.5 & 0.5 \end{pmatrix} \end{array}. \tag{4.13}$$

The probability that the computer is up on the second day is given by the matrix equation

$$\boldsymbol{\rho}^{(2)} = \boldsymbol{\rho}^{(1)}\boldsymbol{\pi} = (0.45,\ 0.55) \tag{4.14}$$

that is, there is a 45 % chance of running a program. The next day would give

$$\boldsymbol{\rho}^{(3)} = \boldsymbol{\rho}^{(2)}\boldsymbol{\pi} = \boldsymbol{\rho}^{(1)}\boldsymbol{\pi}\boldsymbol{\pi} = \boldsymbol{\rho}^{(1)}\boldsymbol{\pi}^2 = (0.435, 0.565), \tag{4.15}$$

and a 43.5 % chance of success. If you are anxious to calculate your chances of running a program in the long run, then the limiting distribution is given by

$$\boldsymbol{\rho} = \lim_{\tau\to\infty} \boldsymbol{\rho}^{(1)}\boldsymbol{\pi}^{\tau}. \tag{4.16}$$

A few applications of eqn (4.16) show that the result converges to $\boldsymbol{\rho} = (0.4286, 0.5714)$. It is clear from eqn (4.16) that the limiting distribution, $\boldsymbol{\rho}$, must satisfy the eigenvalue equation

$$\boldsymbol{\rho}\boldsymbol{\pi} = \boldsymbol{\rho}, \qquad \text{or} \quad \sum_m \rho_m \pi_{mn} = \rho_n \quad \forall n, \tag{4.17}$$

with eigenvalue unity. $\boldsymbol{\pi}$ is termed a stochastic matrix since its rows add to 1

$$\sum_n \pi_{mn} = 1 \quad \forall m. \tag{4.18}$$

It is the transition matrix for an irreducible Markov chain. (An irreducible or ergodic chain is one where every state can eventually be reached from another state.) More formally, we note that the Perron–Frobenius theorem (Chung, 1960; Feller, 1957) states that an irreducible stochastic matrix has one left eigenvalue which equals unity, and the corresponding eigenvector is the limiting distribution of the chain. The other eigenvalues are less than unity and they govern the rate of convergence of the Markov chain. The limiting distribution, $\boldsymbol{\rho}$, implied by the chain is quite independent of the initial condition $\boldsymbol{\rho}^{(1)}$ (in the long run, it matters nothing if your computer is down today). In the case of a liquid, we must construct a much larger transition matrix, which is stochastic and ergodic (see Chapter 2). In contrast to the previous problem, the elements of the transition matrix are unknown, but the limiting distribution of the chain is the vector with elements $\rho_m = \rho_{NVT}(\boldsymbol{\Gamma}_m)$ for each point $\boldsymbol{\Gamma}_m$ in phase space. It is possible to determine elements of $\boldsymbol{\pi}$ which satisfy eqns (4.17) and (4.18) and thereby generate a phase space trajectory in the canonical ensemble. We have considerable freedom in finding an appropriate transition matrix, with the crucial constraint that its elements can be specified without knowing Q_{NVT}. A useful trick in searching for a solution of eqn (4.17) is to replace it by the unnecessarily strong condition of 'microscopic reversibility':

$$\rho_m \pi_{mn} = \rho_n \pi_{nm}. \tag{4.19}$$

Summing over all states m and making use of eqn (4.18) we regain eqn (4.17)

$$\sum_m \rho_m \pi_{mn} = \sum_m \rho_n \pi_{nm} = \rho_n \sum_m \pi_{nm} = \rho_n. \tag{4.20}$$

A suitable scheme for constructing a phase space trajectory in the canonical ensemble involves choosing a transition matrix which satisfies eqns (4.18) and (4.19) . The first such scheme was suggested by Metropolis et al. (1953) and is often known as the asymmetrical solution. If the states m and n are distinct, this solution considers two cases

$$\pi_{mn} = \alpha_{mn} \qquad \rho_n \geq \rho_m \qquad m \neq n \tag{4.21a}$$

$$\pi_{mn} = \alpha_{mn}(\rho_n/\rho_m) \qquad \rho_n < \rho_m \qquad m \neq n. \tag{4.21b}$$

It is also important to allow for the possibility that the liquid remains in the same state,

$$\pi_{mm} = 1 - \sum_{n \neq m} \pi_{mn}. \tag{4.21c}$$

In this solution $\boldsymbol{\alpha}$ is a symmetrical stochastic matrix, $\alpha_{mn} = \alpha_{nm}$, often called the underlying matrix of the Markov chain. The symmetric properties of $\boldsymbol{\alpha}$ can be used to show that for the three cases ($\rho_m = \rho_n$, $\rho_m < \rho_n$, and $\rho_m > \rho_n$) the transition matrix defined in eqn (4.21) satisfies eqns (4.18) and (4.19). It is worth stressing that it is the symmetric property of $\boldsymbol{\alpha}$ that is essential in satisfying microscopic reversibility in this case. Non-symmetrical $\boldsymbol{\alpha}$ matrices which satisfy microscopic reversibility or just the weaker condition, eqn (4.17), can be constructed but these are not part of the basic Metropolis recipe (Owicki and Scheraga, 1977b). These cases are considered in more detail in Chapter 9. This Metropolis solution only involves the ratio ρ_n/ρ_m and is therefore independent of Q_{NVT} which is not required to perform the simulations.

There are other solutions to eqns (4.18) and (4.19). The symmetrical solution (Wood and Jacobson, 1959; Flinn and McManus, 1961; Barker, 1965) is often referred to as Barker sampling:

$$\pi_{mn} = \alpha_{mn}\,\rho_n/(\rho_n + \rho_m) \qquad m \neq n \tag{4.22a}$$

$$\pi_{mm} = 1 - \sum_{n \neq m} \pi_{mn}. \tag{4.22b}$$

Equation (4.22) also satisfies the condition of microscopic reversibility.

If states of the fluid are generated using transition matrices such as eqns (4.21) and (4.22), then a particular property, $\langle\mathcal{A}\rangle_{\text{run}}$, obtained by averaging over the τ_{run} trials in the Markov chain, is related to the average in the canonical ensemble (Chung, 1960; Wood, 1968b)

$$\langle\mathcal{A}\rangle_{NVT} = \langle\mathcal{A}\rangle_{\text{run}} + \mathcal{O}\Big(\tau_{\text{run}}^{-1/2}\Big). \tag{4.23}$$

As mentioned in Chapter 2, we usually restrict simulations to the configurational part of phase space, calculate average configurational properties of the fluid, and add the ideal gas parts after the simulation.

Since there are a number of suitable transition matrices, it is useful to choose a particular solution which minimizes the variance in the estimate of $\langle\mathcal{A}\rangle_{\text{run}}$. Suitable prescriptions for defining the variance in the mean, $\sigma^2(\langle\mathcal{A}\rangle_{\text{run}})$ are discussed in Chapter 8. In particular, the statistical inefficiency (Section 8.4.1)

$$s = \lim_{\tau_{\text{run}}\to\infty} \tau_{\text{run}}\sigma^2(\langle\mathcal{A}\rangle_{\text{run}})/\sigma^2(\mathcal{A}) \tag{4.24}$$

measures how slowly a run converges to its limiting value. Peskun (1973) has shown that it is reasonable to order two transition matrices,

$$\boldsymbol{\pi}_1 \leq \boldsymbol{\pi}_2 \tag{4.25}$$

if each off-diagonal element of $\boldsymbol{\pi}_1$ is less than the corresponding element in $\boldsymbol{\pi}_2$. If this is the case, then

$$s(\langle\mathcal{A}\rangle\,,\,\boldsymbol{\pi}_1) \geq s(\langle\mathcal{A}\rangle\,,\,\boldsymbol{\pi}_2) \tag{4.26}$$

for any property $\mathcal{A}$. If the off-diagonal elements of $\boldsymbol{\pi}$ are large then the probability of remaining in the same state is small and the sampling of phase space will be improved. With the restriction that ρ_m and ρ_n are positive, eqns (4.21) and (4.22) show that the Metropolis solution leads to a lower statistical inefficiency of the mean than the Barker solution.

Valleau and Whittington (1977b) stress that a low statistical inefficiency is not the only criterion for choosing a particular $\boldsymbol{\pi}$. Since the simulations are of finite length, it is essential that the Markov chain samples a representative portion of phase space in a reasonable

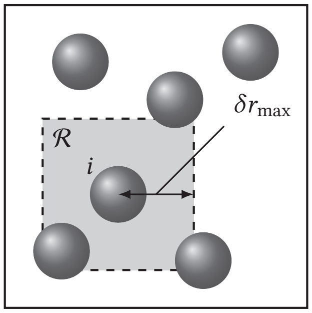

Fig. 4.2 State n is obtained from state m by moving atom i with a uniform probability to any point in the shaded region $\mathcal{R}$.

number of moves. All the results derived in this section depend on the ergodicity of the chain (i.e. that there is some non-zero multi-step transition probability of moving between any two allowed states of the fluid). If these allowed states are not connected, the MC run may produce a low s but in addition a poor estimate of the canonical average. When the path between two allowed regions of phase space is difficult to find, the situation is described as a bottleneck. These bottlenecks are always a worry in MC simulations but are particularly troublesome in the simulation of two-phase coexistence (Lee et al., 1974), in the simulation of phase transitions (Evans et al., 1984), and in simulations of ordinary liquids at unusually high density.

Where a comparison has been made between the two common solutions to the transition matrix, eqns (4.21) and (4.22), the Metropolis solution appears to lead to a faster convergence of the chain (Valleau and Whittington, 1977b). The Metropolis method becomes more favourable as the number of available states at a given step increases and as the energy difference between the states increases. (For two-state problems such as the Ising model the symmetric algorithm may be favourable (Cunningham and Meijer, 1976).) In the next section we describe the implementation of the asymmetric solution.

4.4 The Metropolis method

To implement the Metropolis solution to the transition matrix, it is necessary to specify the underlying stochastic matrix $\boldsymbol{\alpha}$. This matrix is designed to take the system from state m into any one of its neighbouring states n. In this chapter, we normally consider the use of a symmetric underlying matrix, that is, $\alpha_{mn} = \alpha_{nm}$. A useful but arbitrary definition of a neighbouring state is illustrated in Fig. 4.2. This diagram shows six atoms in a state m; to construct a neighbouring state n, one atom (i) is chosen at random and displaced from its position $\mathbf{r}_i^m$ with equal probability to any point $\mathbf{r}_i^n$ inside the square $\mathcal{R}$. This square is of side $2\delta r_{\max}$ and is centred at $\mathbf{r}_i^m$. In a three-dimensional example, $\mathcal{R}$ would be a small cube. On the computer there is a large but finite number of new positions, $N_{\mathcal{R}}$, for the atom i and in this case α_{mn} can be simply defined as

$$\alpha_{mn} = \begin{cases} 1/N_{\mathcal{R}} & \mathbf{r}_i^n \in \mathcal{R} \\ 0 & \mathbf{r}_i^n \notin \mathcal{R}. \end{cases} \tag{4.27}$$

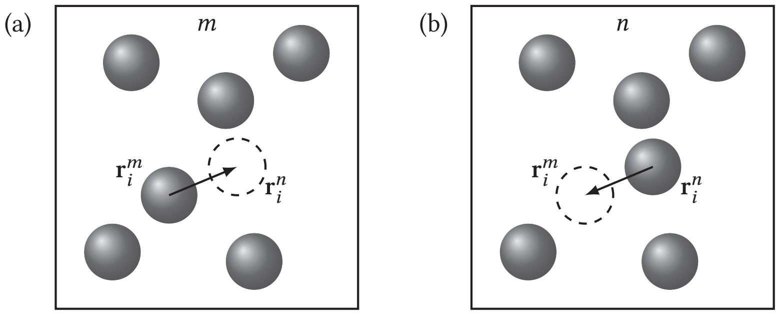

Fig. 4.3 (a) State n is generated from state m by displacing atom i from $\mathbf{r}_i^m$ to $\mathbf{r}_i^n$ (dashed circle). (b) The reverse move. To ensure microscopic reversibility, in the simple Metropolis method the probabilities of attempting the forward and reverse moves should be equal, $\alpha_{mn} = \alpha_{nm}$.

With this choice of $\boldsymbol{\alpha}$, eqn (4.21) is readily implemented. At the beginning of an MC move an atom is picked at random and given a uniform random displacement along each of the coordinate directions. The maximum displacement, δr_{max} is an adjustable parameter that governs the size of the region and controls the convergence of the Markov chain. The new position is obtained with the following code; `dr_max` is the maximum displacement δr_{max}, and the simulation box has unit length.

```
REAL, DIMENSION(3,n) :: r
REAL, DIMENSION(3)   :: ri, zeta

CALL RANDOM_NUMBER ( zeta )      ! uniform in range (0,1)
zeta = 2.0*zeta - 1.0            ! now in range (-1,+1)
ri(:) = r(:,i) + zeta * dr_max   ! trial move to new position
ri(:) = ri(:) - ANINT ( ri(:) ) ! periodic boundaries
```

The appropriate element of the transition matrix depends on the relative probabilities of the initial state m and the final state n. There are two cases to consider. If $\delta\mathcal{V}_{nm} = \mathcal{V}_n - \mathcal{V}_m \leq 0$ then $\rho_n \geq \rho_m$ and eqn (4.21a) applies. If $\delta\mathcal{V}_{nm} > 0$ then $\rho_n < \rho_m$ and eqn (4.21b) applies. (The symbol $\mathcal{V}_m$ is used as a shorthand for $\mathcal{V}(\boldsymbol{\Gamma}_m)$.) The next step in an MC move is to determine $\delta\mathcal{V}_{nm}$. The determination of $\delta\mathcal{V}_{nm}$ does not require a complete recalculation of the configurational energy of state m, just the changes associated with the moving atom. For example (see Fig. 4.3), the change in potential energy is calculated by computing the energy of atom i with all the other atoms before and after the move

$$\delta\mathcal{V}_{nm} = \left(\sum_{j=1}^{N} v(r_{ij}^n) - \sum_{j=1}^{N} v(r_{ij}^m) \right) \tag{4.28}$$

where the sum over the atoms excludes atom i. In calculating the change of energy, the explicit interaction of atom i with all its neighbours out to a cutoff distance r_c is considered. The contribution from atoms beyond the cutoff could be estimated using a mean field correction (see Section 2.8), but in fact the correction for atom i in the old and new positions is exactly the same in a homogeneous fluid, and does not need to be included explicitly in the calculation of $\delta\mathcal{V}_{nm}$.

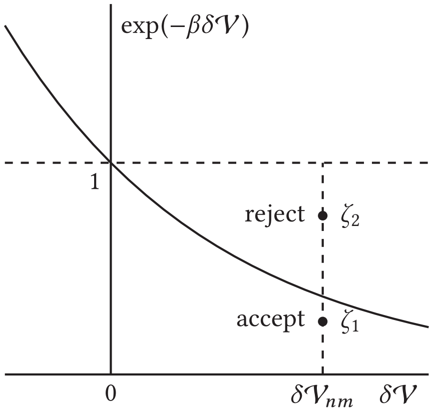

Fig. 4.4 Accepting uphill moves in the MC simulation

If the move is downhill in energy $\delta\mathcal{V}_{nm} \le 0$, then the probability of state n is greater than state m and the new configuration is accepted. The method of choosing trial moves ensures that the transition probability $\pi_{mn} = \alpha_{nm}$, the value required by eqn (4.21a).

If the move is uphill in energy $\delta\mathcal{V}_{nm} > 0$, then the move is accepted with a probability ρ_n/ρ_m according to eqn (4.21b). Again the factor α_{mn} is automatically included in making the move. This ratio can be readily expressed as the Boltzmann factor of the energy difference:

$$\frac{\rho_n}{\rho_m} = \frac{Z_{NVT}^{-1}\exp\bigl(-\beta\mathcal{V}_n\bigr)}{Z_{NVT}^{-1}\exp\bigl(-\beta\mathcal{V}_m\bigr)} = \frac{\exp\bigl(-\beta\mathcal{V}_n\bigr)\exp\bigl(-\beta\delta\mathcal{V}_{nm}\bigr)}{\exp\bigl(-\beta\mathcal{V}_n\bigr)} = \exp\bigl(-\beta\delta\mathcal{V}_{nm}\bigr). \tag{4.29}$$

To accept a move with a probability of $\exp(-\beta\delta\mathcal{V}_{nm})$, a random number ζ is generated uniformly on $(0,1)$. The random number is compared with $\exp(-\beta\delta\mathcal{V}_{nm})$. If it is less than $\exp(-\beta\delta\mathcal{V}_{nm})$ the move is accepted. This procedure is illustrated in Fig. 4.4. During the run, suppose that a particular uphill move, $\delta\mathcal{V}_{nm}$ is attempted. If at that point a random number ζ_1 is chosen (see Fig. 4.4), the move is accepted. If ζ_2 is chosen the move is rejected. Over the course of the run the net result is that energy changes such as $\delta\mathcal{V}_{nm}$ are accepted with a probability $\exp(-\beta\delta\mathcal{V}_{nm})$. If the uphill move is rejected, the system remains in state m in accord with the finite probability π_{mm} of eqn (4.21c). In this case, the atom is retained at its old position and the old configuration is recounted as a new state in the chain. This procedure can be summarized by noting that we accept any move (uphill or downhill) with probability $\min[1, \exp(-\beta\delta\mathcal{V}_{nm})]$, where the min function returns a value equal to the minimum of its arguments (as does the Fortran function with the same name).

A complete MC program for a fluid of Lennard-Jones atoms is given in Code 4.3. Here, we show the typical code for the heart of the program, the acceptance and rejection of moves. In this code, `pot_old` and `pot_new` are the potential energies of atom `i` summed over all its neighbours `j`, as in eqn (4.28). We also expect that, in calculating `pot_new`, a logical flag `overlap` is set if a significant molecular overlap is detected, that is, an interaction with a very high potential energy, which may be regarded as infinite for

Code 4.3 Monte Carlo NVT-ensemble for Lennard-Jones atoms

These files are provided online. The program `mc_nvt_lj.f90` controls the simulation, reads in the run parameters, selects moves, and writes out the results. It uses the routines in `mc_lj_module.f90` to evaluate the Lennard-Jones potential, and actually implement the moves, and utility modules (see Appendix A) for the Metropolis function, input/output and simulation averages.

```
! mc_nvt_lj.f90
! Monte Carlo, NVT ensemble
PROGRAM mc_nvt_lj

! mc_lj_module.f90
! Energy and move routines for MC simulation, LJ potential
MODULE mc_module
```

practical purposes. We use this to guard against a trial move with a very large value of $\delta\mathcal{V}_{nm}$ which might cause underflow problems in the computation of $\exp(-\beta\delta\mathcal{V}_{nm})$. The threshold should be high enough to guarantee that $\exp(-\beta\delta\mathcal{V}_{nm})$ is negligibly small at the chosen temperature; an advantage of including this in the energy calculation is that the program can immediately save time by setting the flag and leaping out of the energy loop.

```
IF ( .NOT. overlap ) THEN ! consider non-overlap only
  delta = ( pot_new - pot_old ) / temperature
  IF ( metropolis ( delta ) ) THEN   ! accept metropolis
    pot    = pot + pot_new - pot_old ! update potential
    r(:,i) = ri(:)                   ! update position
    moves  = moves + 1               ! update move counter
  END IF ! reject metropolis test
END IF ! reject overlap without calculation
```

Here `pot` holds the *total* potential energy of the system, $\mathcal{V}$, which changes by $\delta\mathcal{V}_{nm}$ if the move is accepted. The `metropolis` function simply returns a `.TRUE.` or `.FALSE.` result, using code like

```
REAL :: zeta
IF ( delta < 0.0 ) THEN ! downhill, accept
  metropolis = .TRUE.
ELSE
  CALL RANDOM_NUMBER ( zeta )     ! uniform in range (0,1)
  metropolis = EXP(-delta) > zeta ! metropolis test
END IF
```

In the function referred to by the program of Code 4.3 and other online programs, we include a further guard against underflow.

So far we have said little about the maximum allowed displacement of the atom, δr_{max}, which governs the size of the trial move. If this parameter is too small then a large fraction of moves are accepted but the phase space of the liquid is explored slowly, that is, consecutive states are highly correlated. If δr_{max} is too large then nearly all the trial moves are rejected and again there is little movement through phase space. In fact δr_{max} is often adjusted during the simulation so that about half the trial moves are rejected. This adjustment can be handled automatically using code similar to the following, at predefined intervals, for example, at the end of every sweep, assuming that move_ratio is the ratio of accepted to attempted moves during the sweep.

```
IF ( move_ratio > 0.55 ) THEN
  dr_max = dr_max * 1.05
ELSE IF ( move_ratio < 0.45 ) THEN
  dr_max = dr_max * 0.95
END IF
```

It is not clear that an acceptance ratio of 0.5 is optimum. A reported study of the parameter δr_{max} (Wood and Jacobson, 1959) suggests that an acceptance ratio of only 0.1 maximizes the root-mean-square displacement of atoms as a function of computer time. The root-mean-square displacement is one possible measure of the movement through phase space and the work suggests that a small number of large moves is most cost-effective. Few simulators would have the courage to reject nine out of ten moves on this limited evidence and an acceptance ratio of 0.5 is still common. This issue highlights a difficulty in assessing particular simulation methods. The work of Wood and Jacobson was performed on 32 hard spheres, at a particular packing fraction, on a first-generation computer. There is no reason to believe that their results would be the same for a different potential, at a different state point, on a different machine. The MC technique is time-consuming and since most researchers are more interested in new results rather than methodology there has been little work on the optimization of parameters such as δr_{max} and the choice of transition matrix.

In the original Metropolis method one randomly chosen atom is moved to generate a new state. The underlying stochastic matrix can be changed so that several or all of the atoms are moved simultaneously (Ree, 1970; Ceperley et al., 1977). $\delta\mathcal{V}_{nm}$ is calculated using a straightforward extension of eqn (4.28) and the move is accepted or rejected using the normal criteria. Chapman and Quirke (1985) have performed a simulation of 32 Lennard-Jones atoms at a typical liquid density and temperature. In this study, all 32 atoms were moved simultaneously, and an acceptance ratio of $\approx 30\,\%$ was obtained using $\delta r_{\text{max}} \approx 0.3\sigma$. Chapman and Quirke found that equilibration (see Chapter 5) was achieved more rapidly by employing multi-particle moves rather than single-particle moves. The relative efficiency of multi-particle and single-particle moves, as measured by their ability to sample phase space in a given amount of computer time, has not been subjected to systematic study.

A common practice in MC simulation is to select the atoms to move sequentially (i.e. in order of atom index) rather than randomly. This cuts down on the amount of random number generation and is an equally valid method of generating the correctly weighted states (Hastings, 1970). The length of an MC simulation is conveniently measured in ‘cycles’; that is, N trial moves whether selected sequentially or randomly. The computer

Code 4.4 Monte Carlo of hard spheres

These files are provided online. The program `mc_nvt_hs.f90` controls the simulation, reads in the run parameters, selects moves, and writes out the results. It uses the overlap routines in `mc_hs_module.f90`, and utility module routines (see Appendix A) for input/output and simulation averages.

```
! mc_nvt_hs.f90
! Monte Carlo, NVT ensemble, hard spheres
PROGRAM mc_nvt_hs

! mc_hs_module.f90
! Overlap routines for MC simulation, hard spheres
MODULE mc_module
```

time involved in an MC cycle is comparable (although obviously not equivalent) to that in an MD timestep.

The simulation of hard spheres is particularly easy using the MC method. The same Metropolis procedure is used, except that, in this case, the overlap of two spheres results in an infinite positive energy change and $\exp(-\beta\delta\mathcal{V}_{nm}) = 0$. All trial moves involving an overlap are immediately rejected since $\exp(-\beta\delta\mathcal{V}_{nm})$ would be smaller than any random number generated on $(0, 1)$. Equally, all moves that do not involve overlap are immediately accepted. As before in the case of a rejection, the old configuration is recounted in the average. As discussed in Section 2.4, one minor complication is that the pressure must be calculated by a box-scaling (or related) method. An example program is given in Code 4.4.

The importance sampling technique, as described, only generates states that make a substantial contribution to ensemble averages such as the energy. In practice we cannot sum over all the possible states of the fluid and so cannot calculate Z_{NVT}. Consequently, this is not a direct route to the 'statistical' properties of the fluid such as A, S, and μ. In the canonical ensemble there are a number of ways around this problem, such as thermodynamic integration and the particle insertion methods (see Section 2.4). It is also possible to estimate the free energy difference between the simulated state and a neighbouring state point, and a modification of the sampled distribution, so-called umbrella sampling or non-Boltzmann sampling can make this more efficient. A process of iterative refinement may allow the simulation to sample a greatly extended range of energies, and hence estimate the entropy. We return to these approaches in Chapter 9. Alternatively the problem can be tackled at root by conducting simulations in the grand canonical ensemble (Section 4.6), but as we shall see, this approach may have limited application to dense liquids.

4.5 Isothermal–isobaric Monte Carlo

An advantage of the MC method is that it can be readily adapted to the calculation of averages in any ensemble. Wood (1968a,b; 1970) first showed that the method could be extended to the isothermal–isobaric ensemble. This ensemble was introduced in Section 2.2,