

# The Monte Carlo method

## 10.1 Introduction

In Chapter 8 we saw how a classical many-particle system can be simulated by the MD method, in which the equations of motion are solved for all the particles involved. This enables us to calculate statistical averages of static and dynamic physical quantities. There exists another method, called the Monte Carlo (MC) method, for simulating classical many-particle systems by introducing artificial dynamics based on ‘random’ numbers.<sup>†</sup> The artificial dynamics used in the MC method prevent us from using it for determining dynamical physical properties in most cases, but for static properties it is very popular.

In fact, every numerical technique in which random numbers play an essential role can be called a ‘Monte Carlo’ method after the famous Mediterranean casino town, and we shall discuss the method not only as a tool for studying classical many-particle systems, but also as a way of dealing with the more general problem of calculating high-dimensional integrals. In fact, three main types of Monte Carlo simulations can be distinguished:

- *Direct Monte Carlo*, in which random numbers are used to model the effect of complicated processes, the details of which are not crucial. An example is the modelling of traffic where the behaviour of cars is determined in part by random numbers.
- *Monte Carlo integration*, which is a method for calculating integrals using random numbers. This method is efficient when the integration is over high-dimensional volumes (see below).
- *Metropolis Monte Carlo*, in which a sequence of distributions of a system is generated in a so-called Markov chain. This method allows us to study the

---

<sup>†</sup>As explained in Appendix B, computer-generated random numbers are not truly random, hence the quotes.

static properties of classical and quantum many-particle systems – the latter will be discussed in Chapter 12.

Direct Monte Carlo is a powerful method which can be applied to a wide variety of problems inside and outside physics. There is, however, not much to be said about this method as its implementation is as direct as the name suggests. The difficulty resides usually in the modelling aspect: how can certain phenomena be represented using random numbers – the implementation of the method is then usually rather straightforward. In the next section we shall briefly discuss MC integration. The Metropolis sampling method will be discussed in the remainder of this chapter. The MC techniques which will be discussed in this chapter are essential for much of the material covered in chapters 12 and 15 on quantum Monte Carlo methods and lattice field theory simulations.

A general reference on MC techniques is the book by Hammersley and Handscomb.<sup>1</sup> More detailed material concerning Metropolis Monte Carlo methods can be found in the book by Allen and Tildesley,<sup>2</sup> two review volumes by Binder<sup>3,4</sup> and the books by Kalos and Whitlock,<sup>5</sup> Binder and Heermann<sup>6</sup> and Barkema and Newman.<sup>7†</sup>

## 10.2 Monte Carlo integration

Suppose we want to calculate the integral of a smooth function  $f$  on the interval  $[a, b]$  on the real axis:

$$I = \int_a^b dx f(x). \quad (10.1)$$

Standard numerical methods for this problem are discussed in Section A.6, and they usually boil down to calculating the function on a set of equally spaced values  $x_i$  (except for Gaussian integration, where the points are not equidistant) and then evaluating the sum

$$I = \frac{(b-a)}{N} \sum_{i=1}^N w_i f(x_i), \quad (10.2)$$

where the weights  $w_i$  do not depend on  $f$  – they determine the accuracy of the method. Usually such methods are based on polynomial approximations of the integrand and their accuracy  $\sigma$  is expressed in terms of a power of the separation  $h$  of the integration points:  $\sigma \sim h^k \sim N^{-k}$ , where  $k$  is a positive integer. In Monte Carlo integration we also use Eq. (10.2), with the weights  $w_i$  all equal to 1 but the  $x_i$  now chosen *randomly!*

---

<sup>†</sup>Lecture notes by Frenkel<sup>8</sup> have been helpful in writing part of this chapter.

It will be clear that if the random coordinates  $x_i$  are homogeneously distributed on  $[a, b]$ , and if  $N$  is sufficiently large, the sum (10.2) yields a result close to the exact integral. We calculate the variance in the result:

$$\sigma^2 = \left\langle \left( \frac{b-a}{N} \sum_{i=1}^N f_i \right)^2 \right\rangle - \left( \left\langle \frac{b-a}{N} \sum_{i=1}^N f_i \right\rangle \right)^2. \quad (10.3)$$

The angular brackets denote an average over all possible realisations of the sequence of random coordinates  $x_i$ . Carrying out this average for the last term on the right hand side gives the square of the average  $\bar{f}$  of the function  $f$  on  $[a, b]$ . Splitting the sums in the first term on the right hand side into a sum with  $i = j$  and one with  $i \neq j$  leads after some manipulation to

$$\sigma^2 = \frac{(b-a)^2}{N} (\bar{f^2} - \bar{f}^2) \quad (10.4)$$

where the bar denotes an average of the function on  $[a, b]$ . We see that the error is proportional to the variance of  $f$  on  $[a, b]$ . The fact that  $\sigma \sim 1/\sqrt{N}$  is to be expected from the central limit theorem. This scaling is clearly unfavourable compared with standard quadrature methods using equidistant values for the  $x_i$ , which yields an error  $N^{-k}$  with  $k \geq 1$ . However, MC integration is more efficient in higher dimensions. To see this, let us consider standard numerical integration, with error  $\mathcal{O}(h^k)$ , for a  $d$ -dimensional integral. For simplicity we assume that the integration volume is a hypercube with side  $L$ . This contains  $N = (L/h)^d$  points and therefore the error in the result scales as  $N^{-k/d}$ . The error of the Monte Carlo integration however, is independent of  $d$  – it is still  $\mathcal{O}(N^{-1/2})$  since the central limit theorem does not depend on the dimension. Comparing this error with that of the standard method, we see that MC integration is more efficient than an order- $k$  algorithm when  $d > 2k$ .

This is a rather counter-intuitive result: we would expect that using a regular grid for calculating the integral would always be superior to the random distribution of points of the Monte Carlo method. The reason for the superiority of MC integration in higher dimensions is that in a sense the random distribution is more homogeneous than the regular grid. Consider for example a rectangular volume within the integration volume. A homogeneous distribution of the integration points would imply that the number of points within this rectangular volume should be approximately proportional to that volume. If we choose the rectangular volume to have sides parallel to the axes of the point grid used in the standard integration method, it is clear that on increasing the volume size, the number of points it contains increases stepwise whenever a facet of the volume moves through an array of integration points. In this respect random distributions are more homogeneous,

since for these distributions such steps in the number of points are extremely unlikely to occur. This heuristic argument can be formalised – see the review by James<sup>9</sup> and references therein.

Several methods have been devised to reduce the error of the Monte Carlo integration method; for a discussion see Ref. 9. We give a brief overview here. In order to distribute the points more homogeneously over the integration hypercube, it is possible to subdivide the latter into smaller, equally sized subvolumes and to choose an equal number of random points in each subvolume. This is called ‘stratified Monte Carlo’.

In many practical cases, the contributions to the integral from different regions in the integration volume vary strongly. The MC method samples the function homogeneously, so if the significant contributions to the integral come primarily from a small region within the integration volume, there will be only few MC points for sampling the function there, leading to large statistical errors. This effect shows up in (10.4) as a large variance of  $f$  over the volume. In order to reduce this contribution to the error, points are concentrated in the regions where  $f$  happens to be large (in absolute value). More precisely, let  $\rho(x)$  be a function on  $[a, b]$  which has more or less the shape of  $f$  in the sense that  $f/\rho$  is approximately constant over the interval. We furthermore require  $\rho$  to be normalised:

$$\int_a^b dx \rho(x) = 1. \quad (10.5)$$

We write the integral over  $f$  as follows:

$$\int_a^b dx f(x) = \int_a^b dx \rho(x) \left[ \frac{f(x)}{\rho(x)} \right]. \quad (10.6)$$

The function in square brackets is reasonably flat (as  $\rho$  is chosen to have more or less the shape of  $f$ ) and the weight  $\rho(x)$  in front of this function can be included in the integral by choosing the random points  $x_i$  with distribution  $\rho(x)$ . The Monte Carlo result for the integral is then still given by (10.2). This reduces the error in the result considerably as we can see by evaluating the variance (10.4) for this case:

$$\sigma^2 = \frac{1}{N} \left\{ \int_a^b \left[ \frac{f(x)}{\rho(x)} \right]^2 \rho(x) dx - \left( \int_a^b \left[ \frac{f(x)}{\rho(x)} \right] \rho(x) dx \right)^2 \right\}. \quad (10.7)$$

If we are able to choose  $\rho$  indeed such that  $f/\rho$  is approximately constant, then as a result of (10.5) the expression in the braces will indeed be much smaller than in the ‘crude sampling’ case. This method is called *importance sampling Monte Carlo*. *Adaptive Monte Carlo* methods aim also at concentrating the sampling points in those regions where  $f$  contributes significantly to the integral, but these methods

locate these regions by probing the function at random points and require no *a priori* knowledge on the function  $f$  as in the case of importance sampling.

Note that MC integration is not susceptible to correlations in the random generator. Correlations influence the order in which points the  $x_i$  are generated but this does not affect the sum. In fact, it is possible to generate artificial number sequences without attempting to achieve (pseudo-) randomness but which fill a high-dimensional volume very homogeneously so that they are suitable for integration. The resulting method is called ‘quasi-Monte Carlo’.<sup>9</sup>

### 10.3 Importance sampling through Markov chains

We now explain the importance sampling method for classical many-particle systems in the canonical or ( $NVT$ ) ensemble – extensions to other ensembles will be discussed later in this chapter. When calculating averages in the ( $NVT$ ) ensemble, the configurations should be weighted according to the Boltzmann factor

$$\rho(X) \propto \exp[-\beta E(X)], \quad \beta = 1/(k_B T). \quad (10.8)$$

This suggests that we apply MC integration with importance sampling, as the phase space over which we must integrate is high-dimensional. The method which first comes to mind for generating phase space points (configurations) with a Boltzmann distribution is the Von Neumann method discussed in Section B.3. In our case this would mean generating random configurations and accepting them with probability  $\exp[-\beta E(X)]$  where the energy scale is assumed to be such that the energy is always positive. However, as the number of configurations with a particular energy increases exponentially with energy, most of the randomly constructed states have very high energy – hence for finite temperature they will be accepted with a vanishingly small probability and we spend most of our time generating configurations which are then rejected, which is obviously very inefficient.

Another method would be to construct statistically independent configurations with a bias towards lower energies in accordance with the Boltzmann weight. However, recipes for such a construction are difficult and will certainly be very time consuming. Related to this method is the Metropolis method,<sup>10</sup> developed in 1953, which abandons the idea of constructing statistically independent configurations – rather, the configurations are constructed through a so-called Markov chain, in which each new configuration is generated with a probability distribution depending on the previous configuration.

Before considering Markov chains, we consider *truly random*, or *uncorrelated chains*. For an uncorrelated chain, the probability of occurrence of a particular

sequence of  $N$  objects  $X_1, \dots, X_N$  is statistically uncorrelated:

$$P_N(X_1, X_2, \dots, X_N) = P_1(X_1)P_1(X_2) \cdots P_1(X_N) \quad (10.9)$$

where  $P_1(X)$  is the independent probability of occurrence for the object  $X$  (this probability is assumed to be equal for each step). Truly random number sequences are examples of uncorrelated chains – see Appendix B. A Markov chain is different from uncorrelated chains – it is defined in terms of the transition probability  $T(X \rightarrow X')$  for having the object  $X'$  succeeding object  $X$  in the sequence. The probability of having a sequence of objects  $X_i$  then becomes

$$P_N(X_1, X_2, \dots, X_N) = P_1(X_1)T(X_1 \rightarrow X_2)T(X_2 \rightarrow X_3) \cdots T(X_{N-1} \rightarrow X_N). \quad (10.10)$$

The transition probabilities  $T(X \rightarrow X')$  are normalised:

$$\sum_{X'} T(X \rightarrow X') = 1. \quad (10.11)$$

As an example, consider a random walk on a two-dimensional square lattice. At every step, the random walker can jump from a point to each of its four nearest neighbours with equal probabilities  $1/4$ . This probability is, however, independent of *how* the walker got there, that is, which path he followed in order to arrive at the present point. An example of a non-Markovian (but correlated) sequence is given by the self-avoiding random walk in which the walker is not allowed to visit a site which has been visited in the past. Therefore, not only the last position determines the probability for being at a specific site, but this probability depends on the full history of the walk.

We want to generate a Markov chain of system configurations such that they have a distribution proportional to  $\exp[-\beta E(X)]$ , and this distribution should be independent of the position within the chain and independent of the initial configuration. Under certain conditions, a Markov chain yields indeed such an invariant distribution, at least for long times, as it needs some time to ‘forget’ the chosen initial distribution. We shall not go into details, nor give proofs, but summarise these conditions – they are: (i) every configuration which we want to be included in the ensemble should be accessible from every other configuration within a finite number of steps (this property is called *connectedness*, or *irreducibility*) and (ii) there should be no periodicity. Periodicity means that after visiting a particular configuration, it should not be possible to return to the same configuration except after  $t = nk$  steps,  $n = 1, 2, 3, \dots$ , where  $k$  is fixed. A Markov chain which satisfies these criteria is called *ergodic*.

The Metropolis Monte Carlo method consists of generating a Markov chain of configurations with the required invariant distribution, which in our case is the

Boltzmann distribution  $\exp(-\beta\mathcal{H})$ . We must therefore find a transition probability  $T(X \rightarrow X')$  which leads to a given stationary distribution  $\rho(X)$  (to keep the following analysis general, we use the arbitrary positive function  $\rho(X)$  rather than specify this to be the Boltzmann distribution). The number of possible configurations,  $N$ , is assumed to be finite as the computer memory in which they are to be stored is finite. The ‘matrix’  $T(X \rightarrow X')$  therefore has  $N^2$  elements, as opposed to only  $N$  elements of the ‘vector’  $\rho(X)$ . This means that there are many different solutions to this problem and we are allowed some freedom in finding one.

Let us introduce the function  $\rho(X, t)$  which gives us the probability of occurrence of configuration  $X$  at ‘time’, or Markov step,  $t$  (for an ergodic chain,  $\rho(x, t)$  becomes independent of  $t$  for large  $t$ ). The change in this function from one step to another is governed by two processes: (i) going from a configuration  $X$  at time  $t$  to some other configuration  $X'$  at  $t+1$ , leading to a decrease of  $\rho(X)$  and (ii) going from some configuration  $X'$  at time  $t$  to  $X$  at time  $t+1$ , thus causing an increase in  $\rho(X)$ . These mechanisms can be summarised in the formula

$$\rho(X, t+1) - \rho(X, t) = -\sum_{X'} T(X \rightarrow X')\rho(X, t) + \sum_{X'} T(X' \rightarrow X)\rho(X', t). \quad (10.12)$$

This equation is called the *master equation*. Remember that we are trying to find the stationary distribution, which is found by requiring  $\rho(X, t+1) = \rho(X, t)$ , so that we have

$$\sum_{X'} T(X \rightarrow X')\rho(X, t) = \sum_{X'} T(X' \rightarrow X)\rho(X', t). \quad (10.13)$$

We omit the  $t$ -dependence of  $\rho$  from now on. It is difficult to find the general solution to this equation, but a particular solution is recognised immediately:

$$T(X \rightarrow X')\rho(X) = T(X' \rightarrow X)\rho(X') \quad (10.14)$$

for all pairs of configurations  $X, X'$ . This solution is called *detailed balance* solution. Viewing the configurations  $X$  as buckets, each containing a certain amount  $\rho(X)$  of water, imagine that we make connections with pumps between the members of each pair of buckets. Water is pumped from bucket  $X$  to bucket  $X'$  with a pumping rate  $T(X \rightarrow X')\rho(X)$ . Condition (10.14) makes sure that the pumping rates between any pair of buckets  $X, X'$  is balanced: the flow from  $X$  to  $X'$  is equal to the flow from  $X'$  to  $X$ , so that obviously the volumes  $\rho(X)$  and  $\rho(X')$  do not change. As this holds for any pair of buckets, the whole set of water volumes in the buckets will remain stationary.

Let us now reformulate the detailed balance solution such as to make it suitable for practical purposes. We write the transition probability in the form

$$T(X \rightarrow X') = \omega_{XX'} A_{XX'}, \quad (10.15)$$

where the matrix  $\omega$  is symmetric:  $\omega_{XX'} = \omega_{X'X}$ . Furthermore, it satisfies  $0 \leq \omega_{XX'} \leq 1$ , and  $\sum_{X'} \omega_{XX'} = 1$ .  $A_{XX'}$  must lie between 0 and 1 for each pair  $XX'$ . Substituting this form of  $T$  into the detailed balance equation gives a detailed balance equation for  $A$ :

$$\frac{A_{XX'}}{A_{X'X}} = \frac{\rho(X')}{\rho(X)}. \quad (10.16)$$

In order to construct an algorithm, we use  $\omega_{XX'}$  as a *trial step probability* and  $A_{XX'}$  as an *acceptance probability*. This means that the algorithm proceeds in two stages. In the first stage, given a state  $X$ , we propose a new state  $X'$  with a probability given by  $\omega_{XX'}$ . In the second stage, we compare the weights of the old and the new one,  $\rho(X)$  and  $\rho(X')$ .  $A_{XX'}$  is chosen equal to 1 if  $\rho(X') > \rho(X)$ , and it is chosen equal to  $\rho(X')/\rho(X)$  if  $\rho(X') < \rho(X)$ . Obviously  $A_{XX'}$  satisfies condition (10.16). We accept the new state  $X'$  with a probability  $A_{XX'}$ , and we reject it with a probability  $1 - A_{XX'}$ . If the state  $X'$  is accepted, it replaces  $X$ ; if the new state  $X'$  is not accepted, the system remains in the state  $X$ . Note that if  $\rho(X') > \rho(X)$ , the state  $X'$  is always accepted. We can summarise the Metropolis algorithm as follows:

$$T(X \rightarrow X') = \omega_{XX'} A_{XX'}; \quad (10.17a)$$

$$\sum_{X'} \omega_{XX'} = 1; \quad \omega_{XX'} = \omega_{X'X}; \quad (10.17b)$$

$$\omega_{XX'} > 0, \quad \text{for all } X, X'; \quad (10.17c)$$

$$\text{If } \rho(X') < \rho(X): \quad A_{XX'} = \frac{\rho(X')}{\rho(X)}; \quad (10.17d)$$

$$\text{If } \rho(X') \geq \rho(X): \quad A_{XX'} = 1. \quad (10.17e)$$

The question now arises how we can accept a state with a probability  $A_{XX'} \leq 1$ , and reject it with probability  $1 - A_{XX'}$ . This is done by generating a random number  $r$  uniformly between 0 and 1. If  $r < A_{XX'}$ , the state is accepted, otherwise it is rejected. It is clear that if this procedure is carried out many times with the same probability  $A_{XX'}$ , the state will be accepted a fraction  $A_{XX'}$  of the total number of trials.

Note that because the configurations are generated in a Markov chain, they have correlations inherent to them. The theory of Markov chains guarantees that we arrive at the invariant distribution  $\rho$  for long times; however, it may take much longer than the available computer time to reach this distribution.

The total number of statistically independent configurations is given by the total number of steps divided by the correlation ‘time’, measured in Monte Carlo steps. Note that the number of steps is the *total* number of trials: do not fall into the trap of counting only the *successful* trials as MC steps. As we have generated a sequence of



configurations  $X$  with a statistical distribution  $\exp[-\beta E(X)]$ , the ensemble average of a physical quantity  $A$  is given by the ‘time average’

$$\bar{A} = \frac{1}{n - n_0} \sum_{v > n_0}^n A_v \quad (10.18)$$

where  $n_0$  is the number of steps used for equilibration. Note that the ‘time’  $n$  is not physical time. This average is exactly the same as for MD simulations discussed near the end of Section 8.2. For the determination of statistical errors in the resulting averages we refer to the discussion in Section 7.4.

In the next subsections we shall work out the canonical ensemble MC method in more detail for two examples of classical many particle systems, the Ising model and the monatomic fluid.

It should be noted that the detailed balance condition can be fulfilled by algorithms other than the Metropolis method. The *Barker algorithm*<sup>11, 12</sup> reads

$$T(X \rightarrow X') = \omega_{XX'} \frac{\rho(X')}{\rho(X) + \rho(X')} \quad \text{for } X \neq X', \quad (10.19a)$$

$$T(X \rightarrow X) = 1 - \sum_{X \neq X'} T(X \rightarrow X'). \quad (10.19b)$$

where  $\omega$  satisfies the same criteria as in the Metropolis method. It can easily be shown that this algorithm indeed satisfies the detailed balance condition. The Metropolis solution however turns out to be more efficient.<sup>2</sup>

We can generalise the Metropolis procedure, in which for some pairs  $X, X'$  of configurations,  $\omega_{XX'}$  is not equal to  $\omega_{X'X}$ . In that case, the acceptance criterion for an attempted move  $X \rightarrow X'$  must be replaced by

$$A_{XX'} = \min(1, q_{XX'}) \quad (10.20)$$

with

$$q_{XX'} = \frac{\omega_{X'X} \rho(X')}{\omega_{XX'} \rho(X)}. \quad (10.21)$$

This is called the *generalised Metropolis method*. In the context of Monte Carlo simulations, the method is also called *smart Monte Carlo*. It can easily be checked that the generalised Metropolis method satisfies detailed balance.

We conclude this section by mentioning another variant of the Metropolis method: the *heat-bath algorithm*. In this algorithm it is assumed that the trial step involves one or a few degrees of freedom, the remaining ones being kept fixed (this is the case in most Metropolis algorithms). The degrees of freedom which may change are denoted  $x$ , the remainder of the system, which is kept fixed, is then denoted symbolically as  $X - x$ . We assign now a new value to  $x$  according to

$$P(x) \propto \exp[-\beta \mathcal{H}(x|X - x)], \quad (10.22)$$

where  $\mathcal{H}(x|X-x)$  is the Hamiltonian as a function of  $x$ , with  $X-x$  kept fixed. It is easy to see that this procedure satisfies detailed balance, and that it is equivalent to applying an infinite number of Metropolis steps to  $x$  successively, with fixed  $X-x$ . The practical implementation of this method is often difficult, for reasons to be explained below; however, for lattice models it can be implemented rather straightforwardly.

### 10.3.1 Monte Carlo for the Ising model

The Ising model was discussed in Chapter 7. Here we consider the Metropolis algorithm for the Ising model in two dimensions – the extension to more than two dimensions is straightforward. In order to formulate the Monte Carlo method, we must make a choice for the matrix  $\omega_{XX'}$ . For the two-dimensional Ising model on an  $L \times L$  square lattice, we take

$$\begin{aligned}\omega_{XX'} &= 1/L^2 \text{ if } X \text{ and } X' \text{ differ by one spin;} \\ \omega_{XX'} &= 0 \text{ otherwise.}\end{aligned}\tag{10.23}$$

The realisation of the first stage of the Markov step – generating a trial configuration – is then easy: we select a spin at random, and the trial configuration is the present configuration with the selected spin turned over.

We then calculate the energy difference  $\Delta E(X \rightarrow X')$  between the old and the trial configuration:

$$\Delta E(X \rightarrow X') = E(X') - E(X).\tag{10.24}$$

This is easy as there are only nearest neighbour interactions: the energy difference depends only on the number of neighbours which are equal to the selected spin in the old configuration – this is therefore an integer number between 0 and 4. If the energy increases from the old to the new configuration,  $\Delta E(X \rightarrow X') > 0$ , the trial state is accepted with probability  $\exp[-\beta\Delta E(X \rightarrow X')]$ . If however the energy decreases, the trial state is always accepted as the new state. Implementation of this algorithm for the two- or three-dimensional Ising model is straightforward; the reader is encouraged to go through this exercise. The average number of steps between two updates of the same spin is equal to  $L^2$ . Therefore, the ‘time’ in a MC simulation is often expressed in units of *Monte Carlo steps per spin* (MCS), 1 MCS being equal to  $L^2$  trials.

– Programming exercise –

Write a program for simulating the nearest neighbour two-dimensional Ising model on a square lattice using the Metropolis Monte Carlo technique.

The following considerations should be borne in mind. It is convenient to have a variable representing the total energy of the system. Usually,  $\beta E$  is recorded rather than the energy  $E$  itself, as the behaviour of the system is fully determined by  $\beta J$ . This must be calculated at the beginning by performing a sweep over the whole lattice. If, however, the initial state is one in which all the spins are the same, the total energy for the  $L \times L$  lattice with only nearest neighbour interactions is simply given by  $-2\beta J L^2$ , where  $\beta J$  is the coupling constant. Every time a trial configuration is accepted as the new one, we add the energy difference (which is calculated at every step) to the total energy so that we always have the energy of the new state at our disposal. Similarly it is possible to keep track of the magnetisation during the simulation. The approach described here avoids having to calculate these quantities every now and then by summing over the entire lattice.

As the acceptance probability  $\exp[-\beta\Delta E(X \rightarrow X')]$  can assume only five different values, it is advisable to store these in an array in order to avoid calculating the exponential function over and over again. It is nice to display the lattice after every fixed number of MC steps on the computer screen and give the user the opportunity to change the temperature during the simulation. A magnetic field can also be included. Visual inspection should convince you that the phase transition for zero field takes place somewhere around  $\beta J \approx 0.44$ , although critical fluctuations make it difficult to locate this transition temperature with satisfactory precision. The specific heat can be determined as the variance of the energy – see Eq. (7.28) – and this should exhibit a peak near the critical temperature – see Figure 10.1.

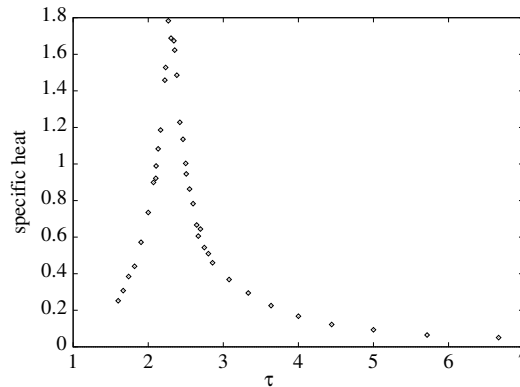


Figure 10.1: Specific heat per site in units of  $J/k_B$  of the Ising  $20 \times 20$  square lattice as a function of the coupling constant reduced temperature  $\tau = k_B T/J$ .

The initial configuration will be chosen either random (infinite temperature) or as one of the two ground states: all spins either + or -. In the first case,

if the temperature at which the system is simulated is lower than the transition temperature, spontaneous magnetisation may not always occur. In fact, several large regions of spin + or of spin – will be formed, separated by boundaries which are relatively smooth in order to minimise their energy. It will now take a very long time before one of the two spin values dominates. This undercooling effect can only be avoided by cooling the system gradually down to the desired temperature, taking care that the cooling rate is particularly slow near  $T_c$ . After passing a first order transition in a simulation, it is often impossible to arrive at the equilibrium state within a reasonable amount of time as a result of the fact that the system cannot overcome the free energy barrier separating the metastable from the stable phase. This can also be checked: below the critical temperature, the Ising model exhibits a first order transition triggered by the magnetic field. Going from a state with positive magnetisation and positive magnetic field to negative (but small) magnetic field, the magnetisation will not turn over. Making the field strongly negative will eventually pull the system over the free energy barrier.

*Check* Produce a graph of the specific heat (measured in units of  $J/k_B$ ) as a function of  $\tau = k_B T/J$  and compare it with Figure 10.1.

You could be tempted to calculate the magnetisation in order to compare this with Figure 7.3. As we have just mentioned, cooling the system down from a high temperature fails unless you cool very slowly through the critical region. However, even if you start with the low temperature phase and heat the system up, you will notice that the magnetisation vanishes for temperatures just below the transition temperature. The point is that in order to let the magnetisation flip from a positive value to a negative one, a domain wall separating the two phases must be built up. Below the transition temperature, domain walls have a positive wall tension, that is, they carry a free energy cost per unit length. Therefore, flipping the magnetisation in the *infinite* system requires an infinite amount of free energy so that this will never happen. In the finite system however, if the wall tension is still finite (and the temperature therefore still below the finite-size critical point), the energy barrier for a magnetisation flip is finite as the domain walls are necessarily finite – hence the magnetisation will fluctuate around a positive or negative equilibrium value for relatively long periods. But now and then it may switch sign, so that its long time average vanishes. Figure 10.2 shows the typical behaviour of the magnetisation in this phase. A histogram of the probability of occurrence for various magnetisations shows a double-peaked shape. These problems also show up in the determination of the magnetic susceptibility which can be expressed in terms of the variance of the magnetisation.

Several methods exist for avoiding this problem:

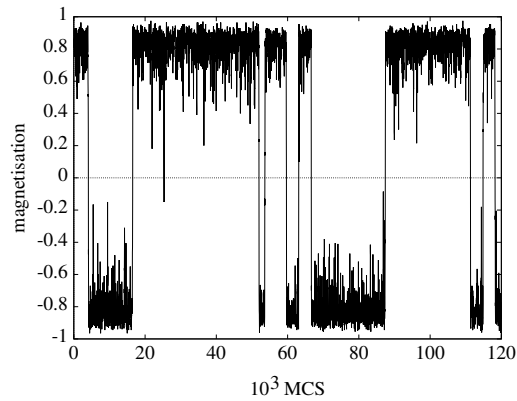


Figure 10.2: Magnetisation of a  $20 \times 20$  Ising lattice as a function of time for an effective coupling constant  $J/k_B T = 0.46$ . The graph clearly shows plateaux where the magnetisation fluctuates around positive or negative average values.

- A restriction to – say – positive magnetisation is built in. This option has the disadvantage that the system is distorted and the consequences of this are not *a priori* clear. Moreover the average magnetisation is always positive so it becomes hard, if not impossible, to see where it would vanish without this restriction.
- Making a plot of the magnetisation as a function of time and taking averages only on the plateaux where the magnetisation is either positive or negative. This method is, however, difficult to apply close to the transition, as the strong fluctuations there will make it difficult to distinguish these plateaux clearly.
- Making a histogram of the magnetisation as mentioned above and taking the peak positions as the value of the magnetisation. Again, distinguishing the peak will be difficult when critical fluctuations are strong.

It will be clear that measuring the magnetisation close to the phase transition is a difficult task which should be avoided if possible.

It should be noticed that the Ising model is formulated without a prescription for its dynamical evolution. Consequently, there exists no molecular dynamics method for the Ising model and Monte Carlo is the only simulation technique for spin systems such as the Ising model. Using the Metropolis method, the Ising model becomes dynamic in the sense that its configurations change with time. The kinetics of this behaviour have been studied, first because it is assumed that the dynamic evolution of Ising-like systems in nature is governed by processes not too dissimilar from Metropolis MC and second because the kinetics is relevant to

the reliability of the simulations, in particular near the critical point.<sup>13, 14</sup> Recall that the dynamical behaviour close to the phase transition is expressed in terms of the dynamic critical exponent,  $z$ , which describes the divergence of the correlation time – see Section 7.3.2.

Our choice for the matrix  $\omega_{XX'}$  is not the only possibility. We may allow for two (or more) spins flipping over at the same step, and these spins might be restricted to be opposite so that the total magnetisation does not change, giving us a constant-magnetisation algorithm. In all cases, instead of selecting the spins randomly, they may be chosen in a regular fashion, for example by scanning through the lattice row by row. In that case, one step in the Markov chain consists of a scan through the whole lattice.  $\omega_{XX'}$  is equal for *any* new configuration  $X'$  and it remains to show that the acceptance probability for each such new configuration is proportional to  $\exp[-\beta\Delta E(X \rightarrow X')]$ . This can be done using induction – see problem 10.1 and Ref. 15.

The heat-bath method (see above) can be implemented straightforwardly for the Ising model. Suppose the spin  $s_i$  selected by the random generator is surrounded by  $n_+$  + spins and  $n_- = 4 - n_+$  – spins. In that case the Hamiltonian for the spin  $s_i$  in the fixed neighbour configuration is given by

$$\mathcal{H}(s_i|S - s_i) = -J(2n_+ - 4)s_i. \quad (10.25)$$

Therefore,  $s_i$  is given the value + or – with probabilities

$$P_+ = \frac{e^{(2n_+-4)\beta J}}{e^{(2n_+-4)\beta J} + e^{-(2n_+-4)\beta J}} \quad \text{and} \quad (10.26a)$$

$$P_- = \frac{e^{-(2n_+-4)\beta J}}{e^{(2n_+-4)\beta J} + e^{-(2n_+-4)\beta J}}. \quad (10.26b)$$

Using the heat-bath instead of the Metropolis method results in a substantial decrease of the correlation time. We shall encounter this method again in Chapter 15, where we present results for the correlation times for the scalar lattice field model. We shall return to the Ising model in Section 15.5.1 where we discuss algorithms which are much more efficient near the critical point than the standard MC methods discussed here.

### 10.3.2 Monte Carlo simulation of a monatomic gas

The Metropolis MC technique enables us to calculate averages of static quantities. Therefore, the momentum degrees of freedom are irrelevant and we integrate these out as in Section 7.2.1, so that the Boltzmann factor depends only on the configurational potential energy:

$$\rho(R) = \exp[-\beta U(R)], \quad (10.27)$$

where  $R$  is the combined position coordinate  $\mathbf{r}_1, \dots, \mathbf{r}_N$ . The configurational potential energy  $U(R)$  is usually written as a sum of pair-potentials as in the previous chapters.

The Monte Carlo procedure for a monatomic gas proceeds as follows. The matrix  $\omega_{XX'}$  is chosen such that only one particle may be moved to a new position – it is selected at random and the remaining particles are kept fixed. The new position of the particle is chosen at random with a homogeneous distribution within a cube centred at the old position of the particle.<sup>†</sup> The energy difference is calculated and the trial configuration is accepted or rejected as usual, with an acceptance rate  $\exp[-\beta\Delta E(X \rightarrow X')]$ .

The displacement volume remains to be specified. Any cube should lead to the correct behaviour as the requirements of the Markov chain (connectedness and aperiodicity) and the detailed balance condition are fulfilled, but there will obviously be a particular cube size yielding optimum efficiency, where efficiency can be defined as the number of statistically independent configurations that we can generate in a fixed amount of computer time. Obviously, if the cube is chosen very small, particles are allowed to move over small distances only and it will take many MC steps to arrive at a statistically independent configuration. On the other hand, if the cube is chosen too large, the particles will on average make large moves. This changes the configuration of the system to such an extent that the energy will increase strongly in the large majority of cases. The probability that the trial move is accepted will then be vanishingly small and we spend most of our time generating configurations which are subsequently rejected, so the configuration changes very slowly in that case as well.

A widely adopted rule of thumb is to choose the cube such that the acceptance rate of trial states is on average somewhere between 0.4 and 0.6. Although for hard sphere systems it should perhaps be lower (around 0.1),<sup>11</sup> this rate is generally relied upon as being reasonably efficient.<sup>2</sup>

When programming the Metropolis method for the case of argon, much of the MD code (see Section 8.3) can be copied. The particles are again released from face-centred cubic (fcc) lattice positions. Every trial displacement should be performed respecting the periodic boundary conditions. Calling half the linear size of the cube  $d_{\max}$ , the trial displacement (without correcting for PBC) in the  $x$ -direction is given in terms of a random number  $0 \leq r < 1$  as

$$x_{\text{new}} - x_{\text{old}} = d_{\max}(2r - 1) \quad (10.28)$$

and similarly for  $y$  and  $z$ . A neighbour list may be kept as in the MD case – however the list must be constructed such that we have at our disposal all the neighbours (at

---

<sup>†</sup>The cubic shape is not essential – it is chosen for convenience.

a distance smaller than the neighbour list cut-off  $r_{\text{cut-off}}$ ) for every particle. This means that we need twice as much storage in comparison to the MD method where only the neighbours with an index higher than the particle under consideration are stored in the list. As in the Ising case, the total potential energy can be updated after every acceptance of a trial configuration. The same can be done with the virial. Physical properties to be calculated are the same as in MD simulations: the pressure, the configurational energy, and the pair correlation function. Any quantity dependent on the positions only can be determined – remember the momenta are not considered in the MC method.

– Programming exercise –

Code the Metropolis method for argon, and compare the results with those of the molecular dynamics program; see table 8.1.

The heat bath method turns out impractical in this example. If we were to use the pure heat bath method, we would have to allow the particle to move to any position in the system cell. This can only be done using the Von Neumann algorithm (see Section B.3), but as the vast majority of positions in the cell are unacceptable, this is very inefficient. One could imagine a ‘hybrid’ heat bath method, in which we move the particle to a new position in a small sphere or cube centred at the old position with a probability distribution determined by the conditional Hamiltonian  $\mathcal{H}(\mathbf{r}_i|R - \mathbf{r}_i)$ , using the symbolic notation of Eq. (10.22). This can only be done using the Von Neumann method (see Section B.3). The difference with the Metropolis method is that the new position is accepted with a probability which is independent of the previous one (except for the sphere or cube being centred on the old position). To apply the Von Neumann method, we should know the minimum of the conditional Hamiltonian, as the acceptance probability in the Von Neumann method may never exceed 1. We might guess a lower bound for this minimum, but this will often be much lower than the actual potential minimum, so that the acceptance rate becomes very small. Because of these difficulties, the heat-bath method is not used for atomic or molecular systems.

Just as in the case of the Ising model, we may sometimes suffer from critical slowing down due to a diverging the correlation time close to a critical point. Methods have been developed which dramatically reduce this effect. A recent breakthrough in this field is the Liu–Luijten algorithm.<sup>16</sup>

## 10.4 Other ensembles

The canonical ensemble is the most natural ensemble for MC simulations. It is, however, possible to simulate other ensembles by the Metropolis MC method. We



shall consider the isothermal-isobaric, or (*NPT*) ensemble, and the grand canonical ensemble. There exists also a microcanonical MC method,<sup>17</sup> but this is seldom used as it is of little practical importance.

#### 10.4.1 The (*NPT*) ensemble

The (*NPT*) ensemble is relevant because temperature and pressure are often kept fixed in experiments. A Monte Carlo method for this ensemble was first developed for hard sphere systems by Wood<sup>18, 19</sup> and later extended to smooth potentials by McDonald.<sup>20, 21</sup> We consider the latter case here.

The (*NPT*) ensemble average of a physical quantity  $A$  depending on the positions  $R = \mathbf{r}_1, \dots, \mathbf{r}_N$  is given as

$$\langle A \rangle_{(NPT)} = \frac{\int_0^\infty dV e^{-\beta PV} \int dR A(R) e^{-\beta U(R)}}{Q(N, P, T)}. \quad (10.29)$$

$Q$  is the partition function which is related to the Gibbs free energy:  $G = -k_B T \ln Q(N, P, T)$  – see Section 7.1. As the volume is allowed to vary, we extend the notion of a configuration to include the volume in addition to the set of particle positions,  $R = (\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_N)$ , with the restriction that the latter should all lie within that volume. A Markov chain must be constructed in which particle moves and volume changes are allowed. It is, however, impossible to change the volume independently from the particle positions, as a decrease of the volume would hardly be possible since particles quite close to the wall may fall outside the volume after such a contraction. Therefore, a change in volume must be accompanied by an appropriate rescaling of the particle positions.

To be more specific, let us consider a cubic volume  $L \times L \times L$  with the edges along the positive Cartesian axes. We scale the particle positions according to  $\mathbf{r}_i = \mathbf{s}_i L$  so that the positions  $\mathbf{s}_i$  lie within the unit cube. The average (10.29) can now be written as

$$\langle A \rangle_{(NPT)} = \frac{\int_0^\infty dV e^{-\beta PV} V^N \int dS A(LS) e^{-\beta U(LS)}}{Q(N, P, T)} \quad (10.30)$$

where  $S$  denotes the combined positions  $\mathbf{s}_1, \dots, \mathbf{s}_N$ . When changing the volume,  $S$  remains the same; the change in the real positions  $\mathbf{r}_i$  is accounted for by a change in the Jacobian  $V^N$  and the various factors  $L$  in (10.30). The Boltzmann weight of the (*NVT*) ensemble method is replaced by

$$\rho(V, S) = e^{-\beta PV} V^N e^{-\beta U(LS)}. \quad (10.31)$$

A step in the Metropolis Markov chain consists either of a particle move, which is performed exactly similar to the particle moves in the (*NVT*) ensemble method, or

a volume change. The calculation of the ratio of the weights before and after the volume change consists of calculating the change due to the potential energy,

$$\exp\{-\beta[U(L_{\text{new}}S) - U(L_{\text{old}}S)]\} \quad (10.32)$$

and multiplying this by the ratio of the terms involving the volume coordinate:

$$\exp[-\beta P(V_{\text{new}} - V_{\text{old}})] \left(\frac{V_{\text{new}}}{V_{\text{old}}}\right)^N. \quad (10.33)$$

The product of (10.32) and (10.33) defines the acceptance ratio of the new configuration according to the Metropolis recipe. Eppenga and Frenkel<sup>22</sup> have applied the method with the logarithm of the volume as the extra coordinate rather than the volume itself.

The calculation of the potential energy difference associated with a volume change is rather demanding – as all positions change, we must sum over all pairs of particles. If the potential can be written as a linear combination of powers  $(\sigma/r)^k$ , the calculation can be performed very fast, because the energy difference in the exponent of (10.32) due to a term  $(\sigma/r)^k$  can be evaluated as

$$\sigma^k \sum_{i < j} (L_{\text{after}} \mathbf{s}_{ij})^{-k} = \sigma^k \left[ \sum_{i < j} (L_{\text{before}} \mathbf{s}_{ij})^{-k} \right] (L_{\text{before}}/L_{\text{after}})^k, \quad (10.34)$$

that is, this contribution to the potential energy changes simply by an overall scaling factor! If the potential is a linear combination of such powers (e.g. the Lennard-Jones potential), this formula can be used, provided that the contributions to the total potential energy due to each power are stored separately.

A possible problem needs some attention. If the potential is cut off beyond a range  $r_{\text{cut-off}}$ , we must correct for this in the total energy using Eq. (8.18). The correlation function,  $g$ , occurring in this formula is usually replaced by 1, in which case this term does not contribute to the total energy difference in MC steps. Therefore, the cut-off distance is usually kept constant. In the special case of a potential consisting of a sum of powers, where we would like to calculate the potential energy difference using the simple scaling procedure just described, we would like to scale the cut-off with the linear system size,  $r_{\text{cut-off}} = L s_{\text{cut-off}}$ , so that (10.34) remains valid. It should be noted, however, that in that case the correction to the potential (which depends also on the density  $N/V$ , which changes at each rescaling) has to be included in the calculation of the energy difference in this case using Eq. (8.18).

We have the freedom to choose the relative frequency with which particle moves and volume changes are attempted. If the scaling method just described

for calculating the potential cannot be used because of invariant intramolecular configurations, or because of more complicated parametrisations of the potential, calculating the potential energy difference due to a volume change becomes quite expensive and volume changes should be attempted at a much lower rate than particle moves. If however the method of Eq. (10.34) is applicable, both types of changes can be attempted with equal probability.

#### 10.4.2 The grand canonical ensemble

In 1969, Norman and Filinov<sup>23</sup> developed a Metropolis Monte Carlo method for simulating many-particle systems in the grand canonical ensemble. In this case the temperature, the system volume and the chemical potential are given, and the particle number and pressure vary; their average values can be determined to establish the equation of state. As the particle number does not remain constant, creation and annihilation of particles should be possible. Let us write down the probability distribution of configurations in this ensemble. The grand canonical ensemble average of a configurational physical quantity  $A$  is given by

$$\langle A \rangle_{\mu VT} = \frac{\sum_{N=0}^{\infty} 1/N! e^{\beta\mu N} \Lambda^{-3N} \int dR_N A(R_N) e^{-\beta U(R_N)}}{Z_G(\mu, V, T)}; \quad (10.35)$$

$\Lambda = h/\sqrt{2\pi m k_B T}$  in the case of a monatomic gas – it results from integrating out the momentum degrees of freedom. We have attached a subscript  $N$  to the positional coordinate  $R$  because the number of particles is not fixed.

It is clear from (10.35) that the configurations are now defined by the number of particles  $N$  and by their positions  $R_N$ . The weight factor which replaces the Boltzmann factor of the  $(NVT)$  ensemble now becomes

$$\rho(N, R_N) = e^{-\beta U(R_N)} \Lambda^{-3N} / N! e^{\beta\mu N} \quad (10.36)$$

and the Metropolis algorithm can be applied directly, provided that in addition to the usual particle moves, we allow for particle creations at random positions and annihilations of randomly chosen particles. The algorithm for a Metropolis step then becomes:

- Decide whether the next trial configuration is constructed via a creation, annihilation, or a particle move according to the probabilities for these processes given by the matrix  $\omega_{XX'}$  (note that the trial rates for creation and annihilation should be equal in order to keep  $\omega_{XX'}$  symmetric). This choice can be made simply by dividing the interval  $[0, 1]$  up into three segments with sizes equal to the respective probabilities, generate a random number uniformly between 0 and 1, and then check in which segment this number lies.

- If a creation is attempted, a random position in the system is selected and the interactions between a new particle inserted at that position and the remaining ones are added to yield a potential energy difference  $\Delta U^+ = U(R_{N+1}) - U(R_N)$ . As the probability that the new particle ends up in the volume element  $d^3r_{N+1}$  is given by  $d^3r_{N+1}/V$ , we accept the creation with a probability

$$e^{-\beta\Delta U^+} \Lambda^{-3} V / (N+1) e^{\beta\mu}. \quad (10.37)$$

- If an annihilation is attempted, one of the existing particles is selected at random, and its interaction with the remaining particles,  $\Delta U^- = U(R_{N-1}) - U(R_N)$ , is calculated. The annihilation is then accepted with probability

$$e^{-\beta\Delta U^-} \Lambda^3 N / V e^{-\beta\mu}. \quad (10.38)$$

- Particle moves are processed similarly to the canonical case. Only the potential energy difference enters into the acceptance probability.

This form of grand canonical Monte Carlo was presented by Norman and Filinov.<sup>23</sup> Other approaches turn particles into ‘ghost’ particles instead of annihilating them,<sup>24,25</sup> and these ghosts can be switched on so that they enter the real life of the simulation again. The Norman-Filinov version is, however, more popular.

It is possible to change the relative rates of the creation and annihilation process and correcting for this by a suitable change in acceptance rates of the corresponding trial configurations. This was done for example by Saito and Müller-Krumbhaar<sup>26</sup> – see also problem 10.3. These methods may be useful in situations where the acceptance rate for particle creation in the standard method becomes exceedingly small due to a high value of the chemical potential.

It should be noticed that at high densities, insertion of new particles is likely to fail because the probability of spatial overlap between the new particle and the existing ones becomes very high. In this case, the Boltzmann factor is small, not as a result of a high value of the chemical potential but as a result of the interactions between the new particle and the existing ones. Methods have been devised for locating ‘cavities’ in such fluids and creating particles preferentially in these regions.<sup>27</sup>

More details on grand canonical Monte Carlo methods can be found in refs. 2 and 28.

### 10.4.3 The Gibbs ensemble

Studying the coexistence of different phases of the same material, and of different species which can transform into each other via chemical reactions, is difficult

using the ensembles defined up to now. We know that for coexistence to be possible, the different phases or species must have equal temperature, pressure and chemical potential. We shall use the name ‘species’ in the context of both chemical mixtures and phase transitions in the following. If we fix pressure and temperature, the chemical potentials of the different species will in general not be equal and one species will grow at the expense of the other, until either one of the two has disappeared or until the chemical potentials are equal. However, it is often very hard to achieve equilibrium in such cases since droplets surrounded by domain walls with a nonvanishing wall tension need a long time to disappear.

Panagiotopoulos<sup>29,30</sup> has developed a method in which two subsystems are considered with no interface between them so that they are free to exchange particles without having to overcome free energy barriers. The method is called ‘Gibbs ensemble method’. The method is quite simple. Consider a volume  $V$  which is divided into two subsystems with volumes  $V_1$  and  $V_2$  by a freely movable piston. The volume  $V = V_1 + V_2$  is fixed – therefore, the total system is described by the  $(NVT)$  ensemble. Furthermore, there is a virtual hole in the piston via which particles can move from one subsystem to the other. Most importantly: there are no interactions between the particles of the two subsystems, that is, if a particle moves from  $V_1$  to  $V_2$ , the energy difference it feels consists of the interactions with its partners in the new subsystem ( $V_2$ ) minus the interactions it felt in its previous subsystem ( $V_1$ ). These moves are actually executed in the Gibbs ensemble MC method, alongside the usual particle moves within the two subsystems and relative changes of the two subsystem volumes. In order to be able to vary the subsystem volumes, we scale the particle positions  $R_K = L_K S_K$  where the index  $K = 1, 2$  labels the subsystem and  $L_K = V_K^{1/3}$ . The weight of a configuration with subsystem volumes  $V_1$  and  $V_2 = V - V_1$ , subsystem particle numbers  $N_1$  and  $N_2$  and subsystem configurations  $S_1$  and  $S_2$  is given by

$$\rho(V_1, N_1, S_1, V_2, N_2, S_2) = \frac{V_1^{N_1}}{N_1!} \frac{V_2^{N_2}}{N_2!} e^{-\beta U(L_1 S_1)} e^{-\beta U(L_2 S_2)}. \quad (10.39)$$

This expression follows directly from the weight factors for the ensembles considered in the two previous subsections (the pressure and the chemical potential occur with  $V_1 + V_2$  and  $N_1 + N_2$  respectively, which are constant).

Equation (10.39) determines the detailed balance transition probabilities in a Metropolis algorithm:

- The matrix  $\omega_{XX'}$  is a probability for volume changes, particle transfers from subsystem 1 to 2 or vice versa, and for particle moves in either 1 or 2 respectively. The matrix elements for particle moves are chosen such as to allow for single particle moves only, with equal probability for each particle.

- Particle moves in each subsystem are processed with the probabilities based on the factor

$$\exp\left\{-\beta\left[U(LS_K^{\text{new}}) - U(LS_K^{\text{old}})\right]\right\}. \quad (10.40)$$

- For subsystem volume changes we find for the acceptance rate (see also subSection 10.4.1):

$$\prod_{K=1,2} \exp\left\{-\beta\left[U(L_K^{\text{new}} S_K) - U(L_K^{\text{old}} S_K)\right]\right\} \left(\frac{V_K^{\text{new}}}{V_K^{\text{old}}}\right)^{N_K}. \quad (10.41)$$

- Acceptance rates for particle transfers from subsystem 1 to subsystem 2 involve the energy difference  $\Delta U_K^\pm$  for removing (-) or adding (+) a particle to the subsystem analogous to the grand canonical MC method:

$$\exp\left[-\beta(\Delta U_2^+ - \Delta U_1^-)\right] \frac{N_1}{N_2 + 1} \frac{V_2}{V_1} \quad (10.42)$$

and similarly for transfers from subsystem 2 to subsystem 1.

These transition rates define the Gibbs ensemble Monte Carlo method. It should be noted that this method is still susceptible to the kind of problems described in connection with the grand canonical Monte Carlo method: moving a particle from one subsystem to another may have a prohibitively low acceptance rate at high densities, due to the large increase of configurational energy in most such attempts.

It is not necessary to separate the two subvolumes by a movable piston in order to arrive at equal pressures in both subsystems: it is also possible<sup>30</sup> to couple both subsystems to a ‘pressure bath’ with predefined pressure  $P$  similar to the ( $NPT$ ) method described in Section 10.4.1, and imposing no restriction on the total volume  $V_1 + V_2$ . This method is less suitable for phase coexistence as the coexistence occurs on a line in the ( $P, T$ ) diagram. Therefore we need to know exactly the location of that line, as we have to specify  $T$  and  $P$  in this method. In the original version, the system will move to the coexistence line by adjusting pressure and chemical potential simultaneously. In the case of chemical equilibrium, however, the coexistence region has a finite width and the constant ( $P, T$ ) version is useful there. The Gibbs ensemble method has become very popular for studying coexistence equilibria in recent years.<sup>29–32</sup>

### \*10.5 Estimation of free energy and chemical potential

In Section 7.1 we have already mentioned the difficulties involved in calculating free energies. We described briefly the method of thermodynamic integration.

Other methods have been proposed and these are easier to understand in the context of MC trials and this is the reason why we discuss them in this chapter. It should, however, be noted that the methods described below are not restricted to MC. Some are applicable within MD, especially in the canonical (*NVT*) MD method. In view of the equivalence of ensembles, microcanonical MD allows for using these methods too.<sup>33</sup>

The reader is referred to refs. 28 and 32 for more details concerning the material in this section.

### 10.5.1 Free energy calculation

The difficulty in free energy calculation is that it cannot be formulated directly as a ‘mechanical average’, that is, an ensemble average of functions of the coordinates  $\mathbf{r}_i$  (and  $\mathbf{p}_i$  in the case of MD). Rather, the free energy must be evaluated as a sum (or integral) over phase space. Clearly, both MD and MC methods sample that part of phase space from which the dominant contribution to the free energy arises; however, this does not provide an estimate of the phase space volume integral, as the frequency with which phase space points are visited is proportional to the Boltzmann weight, with an unknown proportionality factor. Moreover, it is questionable whether the number of points visited in a typical simulation would be sufficient to sample the relevant phase space volume adequately.

It is nevertheless possible to formulate the free energy *difference* between two systems with different interactions as a mechanical ensemble average. The first method for doing this is thermodynamic integration, described in Section 7.1.1. This method suffers from the fact that many simulations are needed in order to calculate free energy differences. We shall now describe a few alternative procedures for this purpose.

Suppose the two systems have interactions described by the potential functions  $U_0$  and  $U_1$ . We show now that the partition function of system 1 can be rewritten as a mechanical average with the Boltzmann factor of system 0. Writing the respective partition functions as  $Z_0$  and  $Z_1$  we have

$$\frac{Z_1}{Z_0} = \frac{\sum_X e^{-\beta U_1}}{\sum_X e^{-\beta U_0}} = \frac{\sum_X e^{-\beta U_0} e^{-\beta(U_1 - U_0)}}{\sum_X e^{-\beta U_0}} = \left\langle e^{-\beta(U_1 - U_0)} \right\rangle_0, \quad (10.43)$$

where  $\langle \dots \rangle_0$  represents an ensemble average in system 0.

Applications of (10.43) are restricted to systems for which the regions they tend to occupy in phase space have appreciable overlap. This can be seen by considering two systems with the same potential function but kept at different temperatures,  $U_1 = \alpha U_0$ . If  $\alpha < 1$ , system 1 is at a higher temperature than system 0. If we evaluate the expectation value in (10.43) using MC or MD, the result contains the

contributions to  $Z_1$  arising from the equilibrium states of the same system at lower temperature – hence the free energy estimate will be drastically wrong as a result of not taking the overwhelming majority of high-energy states contributing to  $Z_1$  into account. On the other hand, if  $\alpha < 0$ , the system will visit configurations with high energy and rarely assume one of the lower energy configurations which contribute to the partition function of system 1. Sufficient overlap between the phase space volume occupied by the two systems is essential. One way to achieve such overlap is to simulate a *chain* of systems with potentials  $U_p$  that interpolate between  $U_0$  and  $U_1$ , such that subsequent configurational potential functions  $U_p, U_{p+1}$  have sufficient overlap to give reliable results. This means that the amount of computer time needed is comparable to that of thermodynamic integration, which uses a chain of different thermodynamic parameters.

Torrie and Valleau have refined the method by adding an extra weight function  $W(X)$  to the average in (10.43), which pushes the system to a different region of phase space such as to reduce the overlap problem. Their method is called ‘umbrella sampling’ and we refer to their paper for a description of the method and examples.<sup>34, 35</sup>

Bennett<sup>33</sup> writes the free energy difference in another way, by defining the ‘Metropolis function’  $M(x)$  as  $M(x) = \min[1, \exp(-x)]$ . Then we have

$$\exp[-\beta(U_0 - U_1)] = M[\beta(U_0 - U_1)] / M[\beta(U_1 - U_0)] \quad (10.44)$$

from which it follows that

$$Z_0 \frac{\sum_X M[\beta(U_1 - U_0)] e^{-\beta U_0}}{Z_0} = Z_1 \frac{\sum_X M[\beta(U_0 - U_1)] e^{-\beta U_1}}{Z_1}. \quad (10.45)$$

The quotients in this expression are canonical ensemble averages corresponding to

$$\frac{Z_0}{Z_1} = \frac{\langle M[\beta(U_0 - U_1)] \rangle_1}{\langle M[\beta(U_1 - U_0)] \rangle_0}. \quad (10.46)$$

This equation is now used as follows. We perform two simulations, one with potential  $U_0$ , and one with potential  $U_1$ . We consider switching the potential in system 0 from  $U_0$  to  $U_1$  as a trial move for which we calculate the acceptance ratio, but which is in fact never carried out. Taking the average of the acceptance ratios of such moves gives the numerator of (10.46). Similarly, the denominator is given as the acceptance ratio of trial switches from  $U_1$  to  $U_0$ , evaluated in the system with potential  $U_1$ . Again, this method is reliable only for appreciable overlap between the two equilibrium distributions in phase space. Bennett’s method can also be extended with a weight function similar to that of Torrie and Valleau, and Bennett calculates the actual form of this function which gives the most accurate results.



Bennett furthermore describes an interpolation method to estimate the free energy difference even if the overlap between the distributions is very small. His paper should be consulted for details.<sup>33</sup>

### 10.5.2 Chemical potential determination

Determining the chemical potential is done using relation (7.13), which enables us to extract this thermodynamic quantity as a free-energy difference between two systems with  $N$  and  $N+1$  particles respectively. The exponential of this free energy difference can be written as the fraction of two partition functions:

$$\frac{Z_{N+1}}{Z_N} = \frac{N!\Lambda^{3N}}{(N+1)!\Lambda^{3(N+1)}} \frac{\int dR_{N+1} \exp[-\beta U(R_{N+1})]}{\int dR_N \exp[-\beta U(R_N)]} = \frac{V}{(N+1)\Lambda^3} \left\langle \frac{1}{V} \int d^3 r_{N+1} \exp(-\beta \Delta U_+) \right\rangle_N = e^{-\beta \mu} \quad (10.47)$$

where  $\Delta U_+$  is the energy of a particle inserted at  $\mathbf{r}_{N+1}$  into the  $N$ -particle system. The prefactor  $V\Lambda^{-3}/(N+1)$  of the expectation value on the right hand side is  $\exp(-\beta \mu_{\text{ideal}})$ , where  $\mu_{\text{ideal}}$  is the chemical potential of the ideal gas. The term within brackets gives the expectation value of the Boltzmann factor associated with the energy difference for the addition of a particle anywhere in the system. This expectation value is determined via trial additions which are regularly performed but never accepted. After each MC step we generate a position  $\mathbf{r}$  at random in the system and calculate the factor  $\exp[-\beta \Delta U(\mathbf{r}_{N+1})]$ . The factor  $1/V$  within the brackets ensures the proper evaluation of the average. These values are then used to calculate the required expectation value by dividing by the number of such trial additions. This method is called ‘Widom’s particle insertion method’.<sup>36</sup> The method works well, although problems arise for high densities. In that case the Boltzmann factor is very small for most trials, because the probability that the core of the new particle overlaps with one of the existing particles becomes very high. There have been refinements in which the particle insertions are biased towards the cavities in the fluid.<sup>22, 37, 38</sup>

Instead of particle insertions, particle removals could be used to find the chemical potential. In this case we would use the inverse of Eq. (10.47):

$$\frac{Z_{N-1}}{Z_N} = \frac{N!\Lambda^{3N}}{(N-1)!\Lambda^{3(N-1)}} \frac{\int dR_{N-1} \exp[-\beta U(R_{N-1})]}{\int dR_N \exp[-\beta U(R_N)]} = NV\Lambda^3 \left\langle \frac{1}{V} \int d^3 r_N \exp(\beta \Delta U_-) \right\rangle_N = e^{+\beta \mu}. \quad (10.48)$$

However, generalisation of the method proposed above to this case usually fails as the sampling of  $\exp(\beta\Delta U_-)$  is very inefficient. The reason is that we are trying to calculate  $\langle \exp(\beta\Delta U_-) \rangle$  whereas the Boltzmann weight factor used in the average includes a factor  $\exp(-\beta\Delta U_-)$ . The latter squeezes the high- $\Delta U$  contributions off and these contribute significantly to the average. Shing and Gubbins<sup>37</sup> have formulated an efficient method combining particle insertions and removals.

## 10.6 Further applications and Monte Carlo methods

### 10.6.1 Generating ensembles of polymers

An important topic in statistical mechanics is the behaviour of polymers: long and flexible chain-like molecules. These can be studied as a melt (a kind of liquid consisting of polymers) or in solution. A nice review on models, theory and Monte Carlo methods for this problem can be found in Ref. 39. We focus here on the problem of a dilute polymer solution. In that case, if we can somehow model the effect of the solvent in terms of a simple interaction, the problem reduces to studying ensembles of individual polymers in different conformations. If the solvent is good, then the free energy for a polymer segment which is on all sides embedded in the solvent is lower than that of a polymer segment which is close to another polymer segment, without solvent molecules in between them. This picture boils down to an effective repulsion of the polymer segments.

It is useful to make a model in which the important properties of the polymer are preserved, whereas the details of its structure on the atomic scale have disappeared from the description. This is related to the idea of universality (see Section 7.3.2): only a few major features of the interactions at small length scales influence the behaviour on longer length scales; the details do not matter. We therefore make a ‘mesoscopic’ model of the polymer. It is a chain consisting of beads: the beads represent segments of the polymer; the segments in turn represent groups of atoms. These atomic groups have a strong short-range interaction, as they are chemically bonded. Remote segments influence each other through a Van der Waals attraction, and they repel when they overlap. However, the solvent effect described above represents another type of repulsion which forces the beads to remain at a minimum distance of at least a few solvent layers. All these characteristics can be summarised in the following polymer model:

- The polymer consists of  $N$  beads, which are represented as point particles. Neighbouring beads have a fixed mutual distance. This is the only interaction between them.
- Remote beads feel a repulsion at short distances and a Van der Waals attraction

at longer distances. Their interaction can be modelled by a Lennard–Jones shape [see Eq. (7.33)].

Note that if we switch off the Lennard–Jones interaction, the polymer represents a random walk: this is called the *ideal chain* case. If, on the other hand, the beads repel each other, we are dealing with a self-avoiding walk (SAW) as the chain cannot cross itself. Often, polymers are studied on a lattice, which is an even further restriction of the model, but asymptotic (large length scale) behaviour is not sensitive to this. For a review of lattice algorithms, see Ref. 40. Here we shall discuss algorithms for simulating the off-lattice model. Most of the methods used in this field have a very similar counterpart in the lattice case.

Note that the behaviour of the polymer is independent of temperatures for high temperatures. This is because the Lennard–Jones interaction is then dominated by the repulsive term, which is always noticeable, even when the temperature is very high. The quantities which are of interest are the end-to-end distance, which is the distance from the first to the last bead, and the radius of gyration [see Problem 1.2]. We shall restrict ourselves here to studying the end-to-end distance as a function of  $N$  for polymers in two dimensions.

Our aim is to generate (an ensemble of) polymer configurations which are distributed according to the canonical distribution at a given temperature. Then we can calculate physical properties for this temperature as an average over the ensemble. Now let us consider possible steps in a standard Metropolis algorithm. We cannot move a single bead, as we should keep the distance to its neighbours constant. An obvious alternative is to choose a bead at random and change the angle between this bead and its two neighbours by some amount. If the polymer is ‘curled up’, which turns out to be not unlikely in typical simulations, there are very few such moves that will be accepted. Algorithms based on this type of moves are called ‘pivot algorithms’: the selected bead acts as a pivot for changing bending or (in three dimensions) dihedral angles.

Another approach is inspired on type of motion of polymers: reptation. This is a snake-type motion which, in the case of the Monte Carlo algorithm, works as follows. A trial step consists of removing the This means that for two replicas  $n$  and  $m$  in our collection, the temperatures are exchanged. In fact we can exchange either the temperature or the configurations. last bead (the ‘tail’) of the polymer and this is attached to the head. Acceptance of this move proceeds according to the standard Metropolis criterion.

The previous methods are typical Metropolis methods, in the sense that every conformation is strongly correlated with the previous one, and these correlations may result in extremely slow relaxation. Another approach tries to avoid this correlation by generating a new polymer at each step. This must be done

quite carefully as simply adding beads will typically result in very improbable conformations. This is because there will certainly be overlaps or crossings in this method. We should therefore add new segments to the polymer in a careful way, avoiding high-energy conformations.

A method in which this is done is the *Rosenbluth algorithm*.<sup>41</sup> We start with two beads on positions (0,0) and (1,0) (the distance between the beads is taken to be 1). Now we add the third one. This is characterised by the angle made by the three beads. The Metropolis algorithm suggests that it would be wise to add the third one with a distribution  $\exp[-E(\theta)/(k_B T)]$ , where  $E(\theta)$  is the interaction energy of the third bead (of angle  $\theta$ ) with the first two. This is however difficult to do if we do not want to use many trial steps. Therefore we discretise the space of possible  $\theta$ -values to obtain a finite number of, say, six different values of  $\theta$ , with a random offset and spaced by  $2\pi/6$ . Now we calculate the weights  $w_j^{(l)} = \exp[-E(\theta_j)/(k_B T)]$  for these six  $\theta$ -values numbered by  $j$  ( $l$  denotes the bead we are adding), and their sum  $W^{(l)}$ :

$$W^{(l)} = \sum_j \exp[-E(\theta_j)/(k_B T)]. \quad (10.49)$$

We accept angle  $j$  with probability  $w_j^{(l)}/W^{(l)}$ . This is done by the ‘roulette-wheel’ algorithm. In fact, what we do is divide up the interval [0,1] into six segments of size  $w_j^{(l)}/W^{(l)}$ . Then we calculate a uniform random number between 0 and 1, and we check which segment  $j$  it corresponds to. Then we choose the corresponding  $\theta_j$ . The final Boltzmann weight of the polymer is  $\exp[-E_{\text{total}}/(k_B T)]$ , where  $E_{\text{total}}$  is the total energy of the chain. Some reflexion may convince you that this is equivalent to the product of the  $w_j$ :

$$\exp[-E_{\text{total}}/(k_B T)] = \prod_{l=3}^N w_j^{(l)} \quad (10.50)$$

where  $j$  denotes the choice which has been made for the angle  $\theta$  at step  $l$ .

We choose as parameters of the Lennard–Jones potential  $\epsilon = 0.25$ , and  $\sigma = 0.8$  (the distance between the polymers is taken to be 1). If the algorithm would do what we want it to do, it should generate the polymers distributed according to (10.50). Calculating the actual probability for a particular configuration to occur, we find that it is

$$P \sim \prod_{l=3}^N \frac{w_j^{(l)}}{W^{(l)}}. \quad (10.51)$$

We see that we are off by a factor  $\prod_l W^{(l)}$  – we must correct for this by storing this number into a *weight factor* for calculating properties of the polymers in the generating ensemble. The weighting factor thus compensates for the fact that the

actual probability of occurrence does not match (10.50). The method described here is known as ‘method A’. If we would take all  $\theta$  angles with an equal probability of  $1/N_\theta$ , where  $N_\theta$  is the number of angles (six in our example), then we should take for the weights of each polymer  $\prod_{l=1}^N w_j^{(l)}$  (check this!). This is known as ‘method B’.

An interesting quantity to look at is the end-to-end distance as a function of the number of beads  $N$ . In fact, the Rosenbluth algorithm is extremely useful for this case as it generates polymer populations for *all* lengths up to the maximum set in the simulation. We have seen that the polymer corresponds to a self-avoiding random walk in two dimensions. It turns out that the end-to-end length scales with  $N$  as

$$R \propto N^\nu, \quad (10.52)$$

where  $\nu$  is an exponent which is 0.75 in two dimensions. The crosses with errorbars in Figure 10.3 show the scaling behaviour as determined in the Rosenbluth algorithm. For this figure, 10000 polymers were grown up to a size of 250 beads. We see that the simulation reproduces the scaling behaviour well for small sizes, but when the sizes exceed 100, the errors become rather large. This seems surprising as the population sizes are equal for all polymer lengths, and the statistics for long polymers should only become better (unless the fluctuations in  $R$  increase with polymer length). The reason behind these fluctuations is the fact that the algorithm generally does not suppress high-energy configurations sufficiently; they will be accepted, but with a low weight. This effect becomes more pronounced for long chains – therefore we have a large population with only a few polymers dominating the average, and most members of the population not contributing to better statistics.

A solution to this problem is offered in the ‘pruned-enriched Rosenbluth method’ (PERM) by Grassberger,<sup>42</sup> in which populations evolves towards a more balanced distribution by removing the ‘bad’ configurations from it and replacing them by copies of the good ones which can then further evolve independently in the simulation. Before going into details, we consider the standard Rosenbluth algorithm formulated in a recursive implementation:

```

Set PolWeight to 1;
ROUTINE AddBead(Polymer, PolWeight, L)
  Calculate the weights  $w_j^L$  and their product  $W^L$ ;
  Add bead number L;
  PolWeight = PolWeight *  $W^L$ ;
  IF L < N THEN
    AddBead(Polymer, PolWeight, L+1);

```

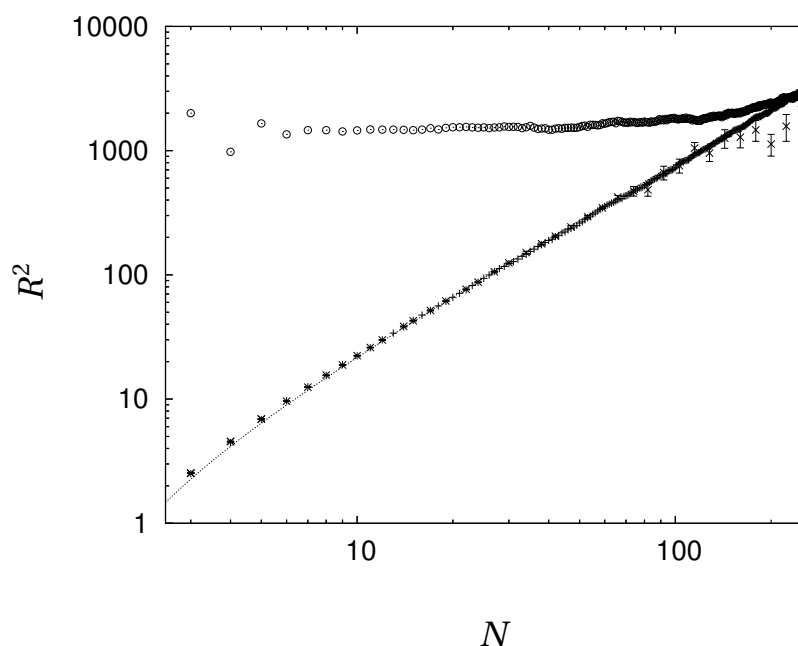


Figure 10.3: Scaling behaviour of the square of the end-to-end length  $R$  as a function of  $N$  on a semi-log scale. Crosses with error bars: Rosenbluth algorithm. Double crosses: PERM algorithm. The circles are the population sizes in the PERM algorithm. The drawn line has the form  $a(N-1)^{0.75}$ , where  $a$  is chosen to fit the data.

```

END IF
END ROUTINE AddBead.

```

A careful read of this pseudocode should convince you that this routine is equivalent to the Rosenbluth algorithm described above. The PERM algorithm cleverly uses this recursive type of procedure in order to perform pruning (i.e. the removal of weak members of the population) and enrichment (the proliferation of the strong members) in a single recursive procedure without having to analyse an entire population. Note that by ‘weak’ we denote conformations having low weight and vice-versa for ‘strong’.

Now consider a polymer which has a relatively large weight. This is ‘enriched’ in the following way. We create two members of this polymer at the next step (i.e. when adding the next bead) and give them each a weight which is half of the weight of their ‘parent’. This causes the total weight of this conformation in the

population to be constant. Suppose we have an upper limit (called ‘UpLim’) of the weight above which polymers are enriched. The recursive code would then contain the following lines:

```

IF (PolWeight>UpLim) THEN
  NewWeight = 0.5*PolWeight;
  AddBead(Polymer, NewWeight, L+1);
  NewWeight = 0.5*PolWeight;
  AddBead(Polymer, NewWeight, L+1);
END IF

```

This does precisely what we want. Note the copy we have made of PolWeight into NewWeight. This is necessary when the call to AddBead would change its PolWeight variable on exit; this depends on the type of call and on the possibilities offered by the computer language.

Next we consider pruning. This means we must remove weak members from the population, but we are not allowed to change the distribution. This is realised by removing weak polymers with probability 1/2, and multiplying the weight of those who happen to be not removed by a factor of 2. The criterion for removal is determined by a lowest weight LowLim. This is done in the following piece of code:

```

IF (PolWeight<LowLim) THEN
  Choose random number R uniformly between 0 and 1;
  IF (R<0.5) THEN
    NewWeight = 2*PolWeight;
    AddBead(Polymer, NewWeight, L+1);
  END IF;
END IF

```

The choice of UpLim and LowLim determine whether the population will grow, shrink or remain stationary. The right choice depends on the average weight ‘AvWeight’ at step  $L$  of the procedure. This average is updated for every new polymer reaching this length. UpLim and LowLim are then taken as multiples of the ratio of this average weight and the weight ‘Weight3’ at the shortest length (3 beads):

$$\text{UpLim} = \alpha * \text{AvWeight}/\text{Weight3}. \quad (10.53)$$

and similar for LowLim. A good value for  $\alpha$  for UpLim is 2, for LowLim we take 1.2. The values of  $\alpha$  may depend on the level  $L$ . It is possible to remove this dependence by multiplying the polymer weight by a constant at each addition step. In this example that constant should be near  $1/(0.75N_\theta)$ , where  $N_\theta$  is the number of choices for the angle  $\theta$  (6 in the example).

– Programming exercise –

You can now try to code the PERM algorithm. If you have done it correctly you should be able to reproduce the double crosses in Figure 10.3. As you can see, they fall on top of the theoretical fit which has  $\nu = 3/4$ .

It should be mentioned that there exist many more methods than described in this subsection. In particular we mention the configurational bias Monte Carlo (CBMC) method by.<sup>43</sup>

### 10.6.2 Tempering and replica exchange

If we want to simulate a disordered system at low temperatures, we run into the problem that many low-energy states exist. Once the system finds itself in phase space near such a low-energy state, it can escape only with great difficulty: the system is trapped. In order to sample the phase space, we must visit a large set of minima accessible at the relevant temperature, and therefore we should somehow ‘push’ the system over the barriers separating the minima. The problem which arises is that pushing the system over the barrier is likely to destroy the detailed balance condition. Methods preserving detailed balance and moving the system efficiently through phase space are the *simulated tempering Monte Carlo*,<sup>44</sup> the *replica Monte Carlo*<sup>45, 46</sup> and the *replica exchange*<sup>47</sup> method. All these methods are based on the idea that the system can move from one minimum to another by repetitive heating and cooling. The problem is to heat and cool efficiently such that moves to different temperatures have a reasonable acceptance probability and that overall detailed balance is preserved. We briefly describe the replica exchange method in this section.

$M$  replicas of the system under consideration are simulated in parallel, each a different (inverse) temperature  $\beta_1, \beta_2, \dots, \beta_M$ . We take  $\beta_1 < \beta_2 < \dots < \beta_M$ . We now alternate a number of ordinary MC steps for each of the replicas by an exchange of the replica configurations, or, equivalently, of the replica temperatures. Exchanging the temperatures just involves assigning different numbers to the temperatures in the two replicas but the temperatures are then no longer ordered. Exchanging the configurations will be more time-consuming.

The partition function describing the collection of replicas is given as

$$Z = \prod_{m=1}^M Z(\beta_m) = Z(\beta_1)Z(\beta_2)\dots Z(\beta_M). \quad (10.54)$$



This can also be written as

$$Z = \sum_{X_m; m=1, \dots, M} \exp\left(-\sum_{m=1}^M \beta_m E_{X_m}\right), \quad (10.55)$$

where  $X_m$  denotes a system configuration of replica  $m$ ). We can interpret this partition function as that of a large system, encompassing all replicas, with a ‘Boltzmann factor’

$$P(\beta_1, X_1; \dots; \beta_M, X_M) = \exp\left(-\sum_{m=1}^M \beta_m E_{X_m}\right). \quad (10.56)$$

Note that the  $\beta_m$  are stochastic variables in this formulation. They can exchange their values but are subject to the condition that the set of values taken on by the  $\beta_m$  remains the same. For this ensemble we perform a Metropolis Monte Carlo algorithm. Suppose we let the replicas evolve independently according to the standard Metropolis algorithm. Then obviously detailed balance is satisfied. After a number of steps however we exchange the configurations (or the temperatures) pairwise according to

$$\beta_n, X_n; \beta_m, X_m \rightarrow \beta_n, X_m; \beta_m, X_n. \quad (10.57)$$

This means that a low-temperature replica receives a high(er) temperature configuration. At higher temperatures, in particular above the phase transition temperature, the system moves easily over all the free energy barriers in phase space. If the low-temperature replicas move to such high temperatures and back, their configuration will in general have moved to different (free) energy minima.

We calculate the expectation values of physical quantities  $A$  at any of the temperatures by averaging over all replica configurations at that particular temperature (perhaps omitting the first few MCS just after a temperature change). Note that exchanges are performed only between adjacent temperatures ( $\beta_m$  and  $\beta_{m+1}$  if the temperatures are ordered), as the acceptance rate decreases exponentially with temperature difference.

Let us calculate the transition probability for temperature exchange. The ratio between a forward- and backward move is given by

$$\frac{T(\beta_n, X_n; \beta_m, X_m \rightarrow \beta_n, X_m; \beta_m, X_n)}{T(\beta_n, X_m; \beta_m, X_n \rightarrow \beta_n, X_n; \beta_m, X_m)} = \exp[-\beta_m (E_{X_n} - E_{X_m}) - \beta_n (E_{X_n} - E_{X_m})] = \exp[-(\beta_m - \beta_n) (E_{X_n} - E_{X_m})]. \quad (10.58)$$

This implies that a trial step in which the temperatures are exchanged, is accepted with a probability  $\min\{1, \exp[-(\beta_m - \beta_n) (E_{X_n} - E_{X_m})]\}$ .

For the method to be successful it is necessary that all Metropolis steps have a reasonable acceptance. It is clear that the acceptance rate is sensitive to the difference between adjacent temperatures. For the acceptance rates to be of order 1, the exponent

$$-(\beta_m - \beta_n)(E_{X_n} - E_{X_m}) \quad (10.59)$$

should be of order 1. Note that before the exchange,  $E_{X_n}$  will be close to the equilibrium energy at temperature  $\beta_n$ , and similarly for  $E_{X_m}$ . This implies that their difference will be approximately

$$E_{X_n} - E_{X_m} \approx C(\beta_m - \beta_n), \quad (10.60)$$

where  $C$  is the (total) specific heat, evaluated near the temperatures  $\beta_m$  and  $\beta_n$  [we assume that  $C(\beta)$  does not vary too strongly between both temperatures – note that this is not justified if the temperatures are close to a critical phase transition]. We see that for the acceptance rate to be of order 1, we should have

$$\Delta \equiv (\beta_m - \beta_n)(E_{X_n} - E_{X_m}) \approx C(\beta_m - \beta_n)^2 = \mathcal{O}(1). \quad (10.61)$$

From the fact that the total specific heat is an extensive quantity, we see that

$$\beta_m - \beta_n \sim \frac{1}{\sqrt{N}}. \quad (10.62)$$

In practice, the set of  $\beta$  values is determined dynamically, such as to make the acceptance rate approximately constant. This is done as follows. We start with a set of temperatures  $\{\beta_n\}$ . For these we perform a number of MCS and replica exchanges. The acceptance rates  $p_m$  for temperature  $\beta_m$  are stored and then the latter are updated according to the recipe:

$$\begin{aligned} \beta_1^{\text{new}} &= \beta_1; \\ \beta_m^{\text{new}} &= \beta_{m-1}^{\text{new}} + (\beta_m - \beta_{m-1}) \frac{p_m}{p_{\text{target}}}, \end{aligned}$$

where  $p_{\text{target}}$  is the ‘target’ acceptance rate which is taken equal for all temperatures. In this way we ensure that the replicas will cycle through all temperatures.

This method has been applied very successfully to spin-glasses and many other examples. For a review, see Ref. 48.

### 10.6.3 Walking in a rough landscape

In everyday life, we often encounter the problem of finding the optimum solution to a problem among many candidate solutions. An example is the well-known

‘travelling salesperson problem’ (TSP) which consists of finding the shortest path connecting a set of cities to be visited by a salesperson. This problem is related to design problems in electronic circuits, where the wiring must be done as efficiently as possible. The TSP is an example of so-called combinatorial optimisation problems, as the aim is to find the optimum among all permutations of the cities. Another type of problems is formulated in a continuum phase space. An example is finding the minimum-energy conformation of a polymer. In particular, the problem of protein folding has received much attention. Although the equilibrium conformations of a protein are usually determined by the minimum of the free energy, in many approaches it is a (sometimes phenomenological) potential energy which is to be minimised.

The problems described here have something in common: it is possible to find many good solutions, but there is only one, or at most a few optimal ones. To define the problem, we should first specify what makes a solution the ‘best’. This is done by assigning to each possible candidate solution a ‘merit function’ (or ‘fitness function’). In the case of the TSP, the merit function is the length of a path. In the case of the polymer conformation it is the potential energy. The optimum solution is defined as the one with the lowest value of the merit function (if the problem is to find the maximum of some quantity we choose the negative of that quantity to be the merit function).

Now we can define the problems in a more abstract way. It is convenient to consider continuum problems. The candidate solutions (for example the possible conformations) form a phase space, and the merit function has some complicated shape on that space – it contains many valleys and mountains, which can be very steep. The solution we seek corresponds to the lowest valley in the landscape. Note that the landscape is high-dimensional. You may think, naively, that a standard numerical minimum finder can solve this problem for you. However, this is not the case as such an algorithm always needs a starting point, from which it finds the nearest *local* minimum, which is not necessarily the best you can find in the conformation space. The set of points which [see Section A.4] would go to one particular local minimum when fed into a steepest descent or other minimum-finder, is called the *basin of attraction* of that minimum. Once we are in the basin of attraction of the global minimum we can easily find this global minimum; the problem is to find its basin of attraction.

There exist many methods for dealing with this problem. In this section we review a few of these only briefly. One method is to generate configurations at random or on a regular grid in the (high-dimensional) phase space, and then finding the nearest local minima for all these point using a standard function minimisation such as the conjugate gradient method [see Section A.4]. It is however possible to have the simulation let the system probe preferentially those regions where the

energy is low. Many of these approaches are based on the ideas presented earlier in this chapter. Note that we simply want to find a (near)-global minimum of the merit function – detailed balance is no longer a concern.

Suppose that you were to find the minimum energy of a polymer studied in the previous subsection. You could have generated a low-temperature ensemble and then for each conformation you encounter, find the nearest local minimum. This has been done by Grassberger for polymer chains consisting of two different types of beads.<sup>49</sup> This method is closely related to *simulated annealing*<sup>50</sup> which is a special version of the Metropolis algorithm. In simulated annealing, the merit function is viewed as an energy. Applying the standard Metropolis ensemble, we would generate configurations distributed according to the Boltzmann factor  $\exp[-E/(k_B T)]$ . In simulated annealing, we slowly cool the system down. The idea behind this is that at higher temperatures, the system can easily hop over the barriers, thereby probing a large part of the phase space. Cooling the system down, it samples only the lower energy domains and finally ends up in one of the deepest valleys. It is obviously efficient to use various samples of the system, so that in fact a population of systems is cooled down and we finally take the one which has reached the lowest energy. Alternatively, one could heat up the system again and then anneal it once more and so on. It is always advisable in these simulations to combine the stochastic algorithm with a deterministic minimum finder such as the conjugate gradient method, in order to efficiently find the deepest point of an attraction basin from each point visited.

The idea behind these method is called ‘configurational search’ for obvious reasons. Another concept in this field is ‘energy sculpting’. This trick tries to overcome the difficulty that a method which focuses on the low-energy parts of the phase space, will automatically avoid the (sometimes) high (free) energy barriers separating the different energy minima. This is done by replacing the energy (i.e. the merit function) by a modified one. An extreme version of this is the *basin hopping* method<sup>51</sup> by Doye and Wales which was used to find optimal conformations of Lennard–Jones clusters. In this method, the energy of a point is replaced by the energy minimum of the attraction basin the point is in. This implies that all points in an attraction basin have the same (modified) energy. The energy steps up or down when moving from one attraction basin to another – the barriers between the basins are entirely removed.

Finally, another method deserves mention: the genetic algorithm (GA).<sup>52, 53</sup> This algorithm is inspired by the ideas of evolution theory and employs these to find optimal solutions to combinatorial or continuous problems. These algorithms are based on encoding any point in phase space into a linear chain. This can be a binary chain. For example, let us suppose that our merit function  $f$  depends on  $N$  variables

$x_j$ :

$$f = f(x_1, \dots, x_N). \quad (10.63)$$

We restrict the range of acceptable values  $x_j$  to some finite interval. Within this interval, we allow for a number (256 or 512, say) of different equidistant values. These values can be coded as a bit-string. It also possible to run the algorithm with the string of reals  $x_1, \dots, x_N$ . The algorithm now manipulates a population of such strings. We start with a pool of  $M$  individuals. Then we do the following steps.

```

WHILE No acceptable solution found DO
  Calculate fitness (merit function) of all individuals;
  Enrich the fit ones by letting them create identical clones;
  Weed out the individuals with low fitness;
  Mate randomly chosen pairs of the population and
    and have them create offspring;
  Mutate;
END WHILE.

```

The enrichment through cloning and the weeding of low-fitness individuals is carried out along the same lines as the PERM algorithm discussed above. The size of the population should remain more or less constant in this process. Mating is a process in which the two members of a pair of chains are cut into two pieces at some randomly chosen chain position. The left piece cut off from chain 1 is then connected to the right part of chain 2 and vice versa.

Mutation is a process in which one of the bits is chosen at random and then flipped. This is necessary to keep variety in the population. The necessity for this is seen by considering the case in which some segment is the same in all chains. This segment would always remain the same in all steps except when mutation takes place. This interesting method has been used for a large variety of problems.<sup>52</sup>

### **\*10.7 The temperature of a finite system**

We conclude this chapter with an intriguing aspect of the simulation of finite systems. In molecular dynamics, the microcanonical ensemble is the ‘default ensemble’, as the solution of the equations of motion leaves the total mechanical energy constant. On the other hand, in Monte Carlo, the Metropolis algorithm naturally leads to the canonical ensemble. We know from equilibrium statistical mechanics that the different ensembles are ‘equivalent’, which means that physical quantities evaluated with the same values of the thermodynamic quantities in different ensembles are the same up to corrections of order  $1/N$ . For finite

systems, the two can therefore not be compared to very high accuracy. However, with a careful analysis of the proper definitions of the temperature, we can make comparisons between the two as we shall now show.

Traditionally, the temperature in microcanonical molecular dynamics is calculated from the equipartition theorem. There is however a subtlety in that the number of degrees of freedom for an  $N$ -particle system is not  $3N$ , but  $3N-3$ , where the three degrees of freedom of the centre of mass must be subtracted as they are fixed in the molecular dynamics algorithm – only the *internal* momenta contribute to the temperature. This procedure thus leads to the expression

$$k_B T = 2 \frac{\langle K \rangle}{3N-3} \quad (10.64)$$

for the temperature ( $K$  is the total kinetic energy). Although this expression is widely used, it is not correct within the microcanonical ensemble if we use the thermodynamic definition of temperature

$$\frac{1}{T} = \left( \frac{\partial S}{\partial E} \right)_{N,V}. \quad (10.65)$$

where  $S = k_B \ln \Omega$ , together with standard expression for  $\Omega$ :

$$\Omega = k_B \ln \left[ \sum_{\text{all states}} \delta(E - H_{\text{state}}) \right], \quad (10.66)$$

where  $H_{\text{state}}$  is the Hamiltonian of the system.

For a system of  $N$  particles in three dimensions, taking into account that the total momentum is conserved, this leads to

$$\Omega = \frac{1}{h^{3N-3} N!} \int \delta[E - H(P, R)] d^{3N-3} P d^{3N} R. \quad (10.67)$$

Note however, that other expressions for the entropy can also be used.<sup>54</sup> Working out the derivative of this entropy with respect to the energy as prescribed by (10.65) does not seem very easy, but it can be done when we first use the explicit quadratic expression for the kinetic energy in the expression for the entropy:

$$\Omega = \frac{1}{h^{3N-3} N!} \int \delta \left[ E - \frac{P^2}{2m} - V(R) \right] d^{3N-3} P d^{3N} R. \quad (10.68)$$

Now we write  $d^{3N-3} P = \omega(3N-3) p^{3N-4} dp$  where  $p = |P|$  and  $\omega(3N-3)$  is the surface of a hypersphere in  $3N-3$  dimensions. Furthermore, substituting  $K = p^2/(2m)$ , the integral becomes

$$\Omega = \frac{1}{h^{3N-3} N!} \int \delta[E - K - V(R)] \omega(3N-3) (2mK)^{(3N-3)/2} \frac{dK}{2K} d^{3N} R. \quad (10.69)$$

Working out the delta function then leads to

$$\Omega = \frac{\omega(3N-3)}{h^{3N-3}N!} \int [2m(E-V)]^{(3N-5)/2} d^{3N}R. \quad (10.70)$$

Taking the derivative of  $S = k_B \ln \Omega$  with respect to energy then leads to the following expression for the temperature:

$$\frac{1}{k_B T} = \frac{1}{k_B} \frac{\partial S}{\partial E} = \frac{3N-5}{2} \frac{\int (E-V)^{(3N-7)/2} d^{3N}R}{\int (E-V)^{(3N-5)/2} d^{3N}R} = \frac{3N-5}{2} \left\langle \frac{1}{E-V} \right\rangle = \frac{3N-5}{2} \left\langle \frac{1}{K} \right\rangle. \quad (10.71)$$

The difference between this expression for the temperature and that obtained from the equipartition theorem is of the order of  $1/N$ , so it is not obvious why people bother about subtracting the 3 from the total number of degrees of freedom when the adopted convention is already off of the same order. However, the good reason for sticking to this convention is that it is adopted in most of the MD codes.

We now consider the question how properties obtained in, say, the canonical ensemble, compare with those obtained in the microcanonical ensemble. When we calculate the expectation value of some physical property in one ensemble for particular values of the system parameters, we can measure the expectation values of the variables conjugate to the system parameters and then evaluate the expectation value of the property at hand in any other ensemble. As we have already noted in the beginning of this section, the differences between values obtained in different ensembles will be of the order of  $1/N$ . To be specific, we may perform a molecular dynamics simulation in the microcanonical ensemble at some energy  $E^*$  and calculate some configurational average. If on the other hand we calculate the same configurational average in a canonical Monte Carlo simulation, we should find differences of order  $1/N$ . We shall now show that it is possible to calculate this difference analytically if the quantity under consideration is the total energy.

The expectation value of the total energy in the canonical ensemble is

$$\langle E \rangle = \frac{\sum_{\text{states } i} E_i e^{-\beta E_i}}{\sum_{\text{states } i} e^{-\beta E_i}}. \quad (10.72)$$

The sum can be rewritten when we collect terms with the same energy. According to the definition of entropy we have  $\exp[S(E)/k_B]$  states at energy  $E$ :

$$\langle E \rangle = \frac{\int E e^{-\beta E + S(E)/k_B} dE}{\int e^{-\beta E + S(E)/k_B} dE}. \quad (10.73)$$

We can approximate the term  $\rho(E) = \exp[-\beta E + S(E)/k_B]$  as follows:

$$\rho(E) = \exp \left[ -\beta E^* + S(E^*)/k_B + \frac{\Delta E^2}{2k_B} \frac{\partial^2 S(E^*)}{\partial E^2} + \Delta E \frac{\partial S(E^*)}{\partial E} + \frac{\Delta E^3}{6k_B} \frac{\partial^3 S(E^*)}{\partial E^3} \right]. \quad (10.74)$$

This is a Taylor expansion of the exponent around its maximum  $E^*$  which satisfies

$$\beta = \frac{1}{k_B} \frac{\partial S(E^*)}{\partial E}. \quad (10.75)$$

We now introduce the two parameters

$$\alpha = \frac{1}{k_B} \frac{\partial^2 S(E^*)}{\partial E^2}; \quad \gamma = \frac{1}{k_B} \frac{\partial^3 S(E^*)}{\partial E^3}. \quad (10.76)$$

The expectation value of the energy can be evaluated straightforwardly in terms of these parameters:

$$\langle E \rangle = \frac{\int_{-\infty}^{\infty} e^{-\alpha \Delta E^2} (1 + \gamma \Delta E^3) (E^* + \Delta E) d\Delta E}{\int_{-\infty}^{\infty} e^{-\alpha \Delta E^2} (1 + \gamma \Delta E^3) d\Delta E}. \quad (10.77)$$

The leading term is simply  $E^*$ . Realising that only even powers of  $\Delta E$  survive in the Gaussian integrals, we obtain for the correction

$$\langle E \rangle = E^* + \frac{\int_{-\infty}^{\infty} e^{-\alpha \Delta E^2} \gamma \Delta E^4 d\Delta E}{\int_{-\infty}^{\infty} e^{-\alpha \Delta E^2} d\Delta E} = E^* - \frac{3\gamma}{4\alpha^2}. \quad (10.78)$$

The derivatives  $\alpha$  and  $\gamma$  can be determined in a way similar to that used to find the temperature calculation above, with the results

$$\alpha = \frac{3N-5}{4} \left[ (3N-7) \left\langle \frac{1}{K^2} \right\rangle - (3N-5) \left\langle \frac{1}{K} \right\rangle^2 \right], \quad (10.79)$$

and

$$\gamma = \frac{(3N-5)(3N-7)(3N-9)}{8} \left\langle \frac{1}{K^3} \right\rangle - 3 \frac{(3N-5)^2(3N-7)}{8} \left\langle \frac{1}{K^2} \right\rangle \left\langle \frac{1}{K} \right\rangle + 1 \frac{(3N-5)^3}{4} \left\langle \frac{1}{K} \right\rangle^3. \quad (10.80)$$

From the definitions (10.76) We see that  $\alpha \sim 1/N$  (which ensures the energy fluctuations to be of order  $1/\sqrt{N}$ ), whereas  $\gamma \sim 1/N^2$ . Therefore, the relative correction to the energy  $3\gamma/(4\alpha^2)$  is still of order  $1/N$ . This is significant when  $N$  is not too large.



Armed with these expressions it is possible to relate the microcanonical energy  $E^*$  to the canonical one. We now give results for a test run involving only 8 particles, as for this number the differences are very clear. Accurate simulations for particles with Gaussian repulsion  $V(r) = \exp(-4r^2)$  have been performed. We have used this potential because it is smooth and does not suffer from the periodicity (it decays rapidly). The kinetic energy  $K$  and the expectation values of  $1/K$ ,  $1/K^2$  and  $1/K^3$  are determined in a molecular dynamics simulation at constant energy in order to calculate  $\alpha$  and  $\gamma$  according to (10.79) and (10.80).

Using a target temperature (according to the equipartition theorem) of 1.0, a (measured) temperature of 1.043 [using the correct definition (10.71)], and a total energy of  $10.913 \pm 0.003$  (in reduced units) have been found in our simulation. This is the energy  $E^*$ . In the canonical ensemble, we should add the correction  $3\gamma^2/(4\alpha^2)$ . Adding the correction, we obtain an energy of  $11.94 + 0.03$  in natural units. An MC calculation with a temperature of 1.043 gives  $E = 11.96$ , in good agreement with the prediction. Note that the statistical errors in  $\alpha$  and  $\gamma$  are difficult to calculate because the inverse powers of the kinetic energy are correlated quantities. The best method is to calculate the correction from several independent simulations and infer the error from these results, or by data blocking of the correction over a long run (see Section 7.4). Note that in order to calculate the canonical kinetic energy,  $K = 3(N-1)k_B T/2$  should be used rather than  $3Nk_B T/2$  in order to relate the energy to the molecular dynamics energy which has three degrees of freedom less (in the canonical ensemble, the equipartition is satisfied even for small particle numbers).

Obviously, the practical value of this calculation is limited – the main point is to show that careful analysis is necessary and possible for systems consisting of small numbers of particles (such as droplets).

### Exercises

10.1 Consider a MC algorithm for the two-dimensional Ising model in which the sites are scanned in lexicographic order, that is, each row is scanned from left to right, starting with the top row and proceeding towards the bottom row. We want to show that this method satisfies the detailed balance criterion. A sweep through the entire lattice is considered as a single step in the Markov chain.

(a) Explain why this Markov chain is ergodic.

The proof that the transition probabilities satisfy detailed balance is done by recursion. Suppose that the lattice contains  $N$  sites, and that for the lattice containing  $N-1$  sites the algorithm satisfies detailed balance.

(b) Show that if the  $N$ -th spin is flipped according to the usual Metropolis algorithm, the sweep over the lattice with  $N$  sites satisfies detailed balance.

10.2 [C] Code the heat-bath algorithm for the Ising model and analyse the correlation time (see section 7.4), in particular close to the critical temperature. Compare the results with the standard Metropolis algorithm.

10.3 Consider the Norman-Filinov method for a system with a large chemical potential. From Eqs. (10.37) and (10.38) we see that in that case the acceptance rate for creation is much smaller than that for annihilation. Therefore we use a generalised Metropolis method, in which creations are tried much more often than annihilations. Suppose the trial probabilities for creation and annihilation are  $P_C$  and  $P_A$  respectively. Show that the acceptances should be modified as follows:

- The acceptance probability for creation is

$$P_{\text{acc}} = \min(1, q_{XX'})$$

with

$$q_{XX'} = e^{-\beta\Delta U^+} \Lambda^{-3} V / (N+1) e^{\beta\mu} \frac{P_A}{P_C}.$$

- Similarly for annihilation:

$$P_{\text{acc}} = \min(1, q_{XX'})$$

with

$$q_{XX'} = e^{\beta\Delta U^-} \Lambda^3 N / V e^{-\beta\mu} \frac{P_C}{P_A}.$$

Show that this modification can be implemented by a suitable shift in the chemical potential. Find this shift.

10.4 [C] Consider the methane ( $\text{CH}_4$ ) molecule of problem 8.12. In that problem we have given the potential energy of the molecule in terms of stretching and bending terms.

Write a Monte Carlo simulation for simulating this molecule at a given temperature. Compare the results with those obtained in problem 8.12.

## References

- [1] J. M. Hammersley and D. C. Handscomb, *Monte Carlo Methods*. London, Methuen, 1964.
- [2] M. P. Allen and D. J. Tildesley, *Computer Simulation of Liquids*. Oxford, Oxford University Press, 1989.
- [3] K. Binder, ed., *Applications of the Monte Carlo Method in Statistical Physics*, vol. 36 of *Topics in Current Physics*. Berlin, Springer, 1984.
- [4] K. Binder, ed., *Monte Carlo Methods in Statistical Physics*, vol. 7 of *Topics in Current Physics*. Berlin, Springer, second ed., 1986.
- [5] M. H. Kalos and P. A. Whitlock, *Monte Carlo Methods*. New York, John Wiley, 1986.
- [6] K. Binder and D. W. Heermann, eds., *Monte Carlo Simulation in Statistical Physics*. New York, Springer, 1988.
- [7] G. T. Barkema and M. E. J. Newman, *Monte Carlo Methods in Statistical Physics*. Oxford, Oxford University Press, 1999.
- [8] D. Frenkel, *Monte Carlo Simulations*. Utrecht, Van 't Hoff laboratory, University of Utrecht, The Netherlands, 1988.
- [9] F. James, 'Monte Carlo theory and practice,' *Rep. Prog. Phys.*, **43**, 1145–1189, 1980.
- [10] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, 'Equation of state calculations by fast computing machines,' *J. Chem. Phys.*, **21**, 1087–1092, 1953.
- [11] W. W. Wood and J. D. Jacobsen, *Monte Carlo calculations in statistical mechanics*, pp. 261–269. Proceedings of the Western Joint Computer Conference, New York, San Francisco Institute of Radio Engineers, 1959.
- [12] J. A. Barker, 'Monte Carlo calculations of the radial distribution functions for a proton-electron plasma,' *Aust. J. Phys.*, **18**, 119–133, 1965.
- [13] P. C. Hohenberg and B. I. Halperin, 'Theory of dynamic critical phenomena,' *Rev. Mod. Phys.*, **49**, 435–479, 1977.
- [14] K. Kawasaki, 'Kinetics of Ising models,' vol. 2 of *Phase Transitions and Critical Phenomena*, London, Academic Press, 1972.
- [15] W. K. Hastings, 'Monte Carlo methods using Markov chains, and their applications,' *Biometrika*, **57**, 97–109, 1970.
- [16] J. Liu and E. Luijten, 'Rejection-free geometric cluster algorithm for complex fluids,' *Phys. Rev. Lett.*, **92**, 035504 (4 pages), 2004.
- [17] M. Creutz, *Quarks, Gluons and Lattices*. Cambridge, Cambridge University Press, 1983.
- [18] W. W. Wood, 'Monte Carlo calculations for hard disks in the isothermal-isobaric ensemble,' *J. Chem. Phys.*, **48**, 415–434, 1968.
- [19] W. W. Wood, ' $NpT$ -ensemble Monte Carlo calculations for the hard disk fluid,' *J. Chem. Phys.*, **52**, 729–741, 1970.
- [20] I. R. McDonald, 'Monte Carlo calculations for one- and two-component fluids in the isothermal-isobaric ensemble,' *Chem. Phys. Lett.*, **3**, 241–243, 1969.

- [21] I. R. McDonald, 'NpT-ensemble Monte Carlo calculations for binary liquid mixtures,' *Mol. Phys.*, **23**, 41–58, 1972.
- [22] R. Eppenga and D. Frenkel, 'Monte Carlo study of the isotropic and nematic phases of infinitely thin hard platelets,' *Mol. Phys.*, **52**, 1303–1334, 1984.
- [23] G. E. Norman and V. S. Filinov, 'Investigations of phase transitions by a Monte Carlo method,' *High Temp. (USSR)*, **7**, 216–222, 1969.
- [24] Z. W. Salsburg, J. D. Jacobson, W. Fickett, and W. W. Wood, 'Application of the Monte Carlo method to the lattice gas model. I. Two dimensional triangular lattice,' *J. Chem. Phys.*, **30**, 65–72, 1959.
- [25] D. A. Chesnut, 'Monte Carlo calculations for the two-dimensional triangular lattice gas: supercritical region,' *J. Chem. Phys.*, **39**, 2081–2084, 1963.
- [26] Y. Saito and H. Müller-Krumbhaar, '2-Dimensional Coulomb gas – a Monte Carlo study,' *Phys. Rev. B*, **23**, 308–315, 1981.
- [27] M. Mezei, 'A cavity-based ( $TV\mu$ ) Monte Carlo method for the computer simulation of fluids,' *Mol. Phys.*, **40**, 901–906, 1980.
- [28] D. Frenkel, 'Free energy computation and first-order phase transitions,' in *Molecular Dynamics Simulation of Statistical Mechanical Systems* (G. Ciccotti and W. G. Hoover, eds.), vol. 97 of *Proceedings of the International School of Physics "Enrico Fermi", Varenna 1985*, pp. 151–188, Amsterdam, North-Holland, 1986.
- [29] A. Z. Panagiotopoulos, 'Direct determination of phase coexistence properties of fluids by Monte Carlo simulation in a new ensemble,' *Mol. Phys.*, **61**, 813–826, 1987.
- [30] A. Z. Panagiotopoulos, N. Quirke, and D. J. Tildesley, 'Phase-equilibria by simulation in the Gibbs ensemble – Alternative derivation, generalization and application to mixture and membrane equilibria,' *Mol. Phys.*, **63**, 527–545, 1988.
- [31] A. Z. Panagiotopoulos, 'Adsorption and capillary condensation of fluids in cylindrical pores by Monte Carlo simulation in the Gibbs ensemble,' *Mol. Phys.*, **62**, 701–719, 1987.
- [32] D. Frenkel and B. Smit, *Understanding Molecular Simulation*. San Diego, Academic Press, 1996.
- [33] C. H. Bennett, 'Efficient estimation of free energy differences from Monte Carlo data,' *J. Comput. Phys.*, **22**, 245–268, 1976.
- [34] G. M. Torrie and J. P. Valleau, 'Nonphysical sampling distributions in Monte Carlo free energy estimation: umbrella sampling,' *J. Comp. Phys.*, **23**, 187–199, 1977.
- [35] G. M. Torrie and J. P. Valleau, 'Monte Carlo study of a phase separating liquid mixture by umbrella sampling,' *J. Chem. Phys.*, **66**, 1402–1408, 1977.
- [36] B. Widom, 'Some topics in the theory of fluids,' *J. Chem. Phys.*, **39**, 2808–2812, 1963.
- [37] K. S. Shing and K. E. Gubbins, 'The chemical potential in dense fluids and fluid mixtures via computer simulation,' *Mol. Phys.*, **46**, 1109–1128, 1982.
- [38] M. Fixman, 'Direct simulation of the chemical potential,' *J. Chem. Phys.*, **78**, 4223–4226, 1983.
- [39] J. Baschnagel, J. P. Wittmer, and H. Meyer, 'Monte Carlo simulation of polymers:

- coarse-grained models,' in *Computational Soft Matter: From Synthetic Polymers to Proteins. Lecture notes* (N. Attig, K. Binder, H. Grubmüller, and K. Kremer, eds.), vol. 23 of *NIC*, (Jülich), pp. 83–140, John von Neumann Institute for Computing, 2004.
- [40] K. Kremer and K. Binder, 'Monte Carlo simulation of lattice models for macromolecules,' *Comp. Phys. Comm.*, **7**, 259–310, 1988.
- [41] M. N. Rosenbluth and A. W. Rosenbluth, 'Monte Carlo calculation of the average extension of molecular chains,' *J. Chem. Phys.*, **23**, 356–359, 1955.
- [42] P. Grassberger, 'Pruned-enriched Rosenbluth method: Simulation of  $\theta$  polymers of chain length up to 1 000 000,' *Phys. Rev. E*, **56**, 3682–3693, 1997.
- [43] J.I.Siepmann and D. Frenkel, 'Configurational-bias Monte Carlo: a new sampling scheme for flexible chains,' *Molec.Phys.*, **75**, 59–70, 1992.
- [44] E. Marinari and G. Parisi, 'Simulated tempering: A new Monte Carlo scheme,' *Europhys. Lett.*, **19**, 451–458, 1992.
- [45] R. H. Swendsen and J.-S. Wang, 'Replica Monte Carlo simulation of spin-glasses,' *Phys. Rev. Lett.*, **57**, 2607–2609, 1986.
- [46] C. Geyer, 'Markov Chain Monte Carlo Maximum Likelihood,' in *Computing Science and Statistics: the 23rd symposium on the interface* (E. Keragimas, ed.), (Vietri Sul Mare, Salerno, Italy), pp. 156–163, 1991.
- [47] K. Hukushima and K. Nemoto, 'Exchange Monte Carlo method and application to spin glass simulation,' *J. Phys. Soc. Jpn.*, **65**, 1604–1608, 1996.
- [48] D. J. Earl and M. W. Deem, 'Parallel tempering: theory, applications, and new perspectives.' physics/0508111, 2005.
- [49] H.-P. Hsu, V. Mehra, and P. Grassberger, 'Structure optimization in an off-lattice protein model,' *Phys. Rev. E*, **68**, 037703 (4 pages), 2003.
- [50] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, 'Optimization by simulated annealing,' *Science*, **220**, 671–680, 1983.
- [51] D. J. Wales and J. P. K. Doye, 'Global optimization by basin-hopping and the lowest energy structures of Lennard-Jones clusters containing up to 110 atoms,' *J.Phys. Chem. A*, **101**, 5111–5116, 1997.
- [52] C. R. Reeves and J. E. Rowe, *Genetic algorithms: principles and perspectives*. Dordrecht, Kluwer, 2003.
- [53] J. H. Holland, *Adaptation in natural and artificial design*. Ann Arbor, The University of Michigan Press, 1975.
- [54] K. Huang, *Statistical Mechanics*. New York, John Wiley, second ed., 1987.