3 Simple sampling Monte Carlo methods

3.1 INTRODUCTION

Modern Monte Carlo methods have their roots in the 1940s when Fermi, Ulam, von Neumann, Metropolis and others began considering the use of random numbers to examine different problems in physics from a stochastic perspective (Cooper, 1989); this set of biographical articles about S. Ulam provides fascinating insight into the early development of the Monte Carlo method, even before the advent of the modern computer). Very simple Monte Carlo methods were devised to provide a means to estimate answers to analytically intractable problems. Much of this work is unpublished and a view of the origins of Monte Carlo methods can best be obtained through examination of published correspondence and historical narratives. Although many of the topics which will be covered in this book deal with more complex Monte Carlo methods which are tailored explicitly for use in statistical physics, many of the early, simple techniques retain their importance because of the dramatic increase in accessible computing power which has taken place during the last two decades. In the remainder of this chapter we shall consider the application of simple Monte Carlo methods to a broad spectrum of interesting problems.

3.2 COMPARISONS OF METHODS FOR NUMERICAL INTEGRATION OF GIVEN FUNCTIONS

3.2.1 Simple methods

One of the simplest and most effective uses for Monte Carlo methods is the evaluation of definite integrals which are intractable by analytic techniques. (See the book by Hammersley and Handscomb (1964) for more mathematical details.) In the following discussion, for simplicity we shall describe the methods as applied to one-dimensional integrals, but it should be understood that these techniques are readily extended, and often most effective, when applied to multidimensional integrals. In the simplest case we wish to obtain the integral of $f(x)$ over some fixed interval:

$$y = \int_a^b f(x)dx. \tag{3.1}$$

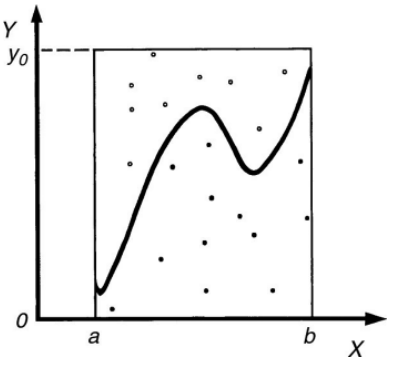

Fig. 3.1 Simple representation of 'hit-or-miss' Monte Carlo integration of a function $f(x)$, given by the solid curve, between $x = a$ and $x = b$. N points are randomly dropped into the box, N_o of them fall below the curve. The integral is estimated using Eqn. (3.2).

In Fig. 3.1 we show a pictorial representation of this problem. A straightforward Monte Carlo solution to this problem via the 'hit-or-miss' (or acceptance–rejection) method is to draw a box extending from a to b and from 0 to y_o where $y_o > f(x)$ throughout this interval. Using random numbers drawn from a uniform distribution, we drop N points randomly into the box and count the number, N_o, which fall below $f(x)$ for each value of x. An estimate for the integral is then given by the fraction of points which fall below the curve times the area of the box, i.e.

$$y_{est} = (N_o/N) \times [y_o(b - a)]. \tag{3.2}$$

This estimate becomes increasingly precise as $N \to \infty$ and will eventually converge to the correct answer. This technique is an example of a 'simple sampling' Monte Carlo method and is obviously dependent upon the quality of the random number sequence which is used. Independent estimates can be obtained by applying this same approach with different random number sequences and by comparing these values the precision of the procedure can be ascertained. An interesting problem which can be readily attacked using this approach is the estimation of a numerical value for π. The procedure for this computation is outlined in the example described below.

Example

How can we estimate the value of π using simple sampling Monte Carlo? Choose N points randomly in the xy-plane so that $0 < x < 1$ and $0 < y < 1$. Calculate the distance from the origin for each point and count those which are less than a distance of 1 from the origin. The fraction of the points which satisfy this condition, N_o/N, provides an estimate for the area of one-quarter of a circle so that $\pi \approx 4N_o/N$. This procedure may be repeated multiple times and the variance of the different results may be used to estimate the error. Here are some sample results for a run with 10 000 points. Note that on the right we show estimates based on up to the first 700 points; these results appear to have converged to the wrong answer but the apparent difficulty is really due simply to the use of too few points. This lesson should not be forgotten!

N	Result		
1000	3.0800		
2000	3.0720	100	3.1600
3000	3.1147	200	3.0400
4000	3.1240	300	3.1067
5000	3.1344	400	3.0800
6000	3.1426	500	3.0560
7000	3.1343	600	3.0800
8000	3.1242	700	3.0743
9000	3.1480		
10 000	3.1440		

A variation of this approach is to choose the values of x in a regular, equidistant fashion. The advantage of this algorithm is that it requires the use of fewer random numbers. For functions with very substantial variations over the range of interest, these methods are quite likely to converge slowly, and a different approach must be devised.

Another type of simple Monte Carlo method is termed the 'crude method'. In this approach we choose N values of x randomly and then evaluate $f(x)$ at each value so that an estimate for the integral is provided by

$$y_{\text{est}} = \frac{1}{N} \sum_i f(x_i) \tag{3.3}$$

where, again, as the number of values of x which are chosen increases, the estimated answer eventually converges to the correct result. In a simple variation of this method, one can divide the interval into a set of unequal subintervals and perform a separate Monte Carlo integration for each subinterval. In those regions where the function is large the sampling can be extensive and less effort can be expended on those subintervals over which the function is small.

3.2.2 Intelligent methods

Improved methods may be broadly classified as 'intelligent' Monte Carlo methods. In one technique, the 'control variate method', one selects a known, integrable function $f'(x)$ which has a relatively similar functional dependence on x and only integrates the difference $[f'(x) - f(x)]$ by some Monte Carlo method, i.e.

$$y_{\text{est}} = F' + \int_a^b [f(x) - f'(x)]dx \tag{3.4}$$

where $F' = \int_a^b f'(x)dx$. The final estimate for y_{est} can be improved without additional numerical effort by an intelligent choice of $f'(x)$.

Instead of selecting all points with equal probability, one can choose them according to the anticipated importance of the value of the function at that

point to the integral $p(x)$ and then weight the contribution by the inverse of the probability of choice. This is one of the simplest examples of the class of Monte Carlo methods known as 'importance sampling' which will be discussed in much greater detail in the next chapter. Using importance sampling an estimate for the integral is given by

$$y_{\text{est}} = \sum_i p^{-1}(x_i) f(x_i). \tag{3.5}$$

For functions which vary wildly over the interval of interest, this approach allows us to increase the sampling in the region in which the contribution to the integral is particularly large. Since the values of x are no longer chosen with equal probability, we begin to see the need for sequences of random numbers which are not drawn from a uniform sequence. Obviously for oddly behaved functions some expertise is needed in choosing $p(x)$, but this can be done iteratively by first carrying out a rough Monte Carlo study and improving the choice of sampling method. Intelligent importance sampling is far more effective in improving convergence than the brute force method of simply generating many more points.

In Chapter 7 we will show how a completely different Monte Carlo approach, called 'Wang–Landau sampling', can be used to provide accurate estimates of multidimensional integrals. We do not discuss this here because the reader will benefit from a presentation about the algorithm before considering its implementation to the problem of numerical integration.

Problem 3.1 Suppose $f(x) = x^{10} - 1$. Use a 'hit-or-miss' Monte Carlo simulation to determine the integral between $x = 1$ and $x = 2$.

Problem 3.2 Suppose $f(x) = x^{10} - 1$. Use an importance sampling Monte Carlo simulation to determine the integral between $x = 1$ and $x = 2$.

Problem 3.3 Estimate π using the methods described above with $N = 100\,000$ points. What is the error of your estimate? Does your estimate agree with the correct answer?

3.3 BOUNDARY VALUE PROBLEMS

There is a large class of problems which involve the solution of a differential equation subject to a specified boundary condition. As an example we consider Laplace's equation

$$\nabla^2 u = \partial^2 u/\partial x^2 + \partial^2 u/\partial y^2 = 0 \tag{3.6}$$

where the function $u(r) = f$ on the boundary. Equation (3.6) can be re-expressed as a finite difference equation, if the increment Δ is sufficiently small,

$$\begin{aligned} \nabla^2 u = [&u(x + \Delta, y) + u(x - \Delta, y) + u(x, y + \Delta) \\ &+ u(x, y - \Delta) - 4u(x, y)]/\Delta^2 = 0 \end{aligned} \tag{3.7}$$

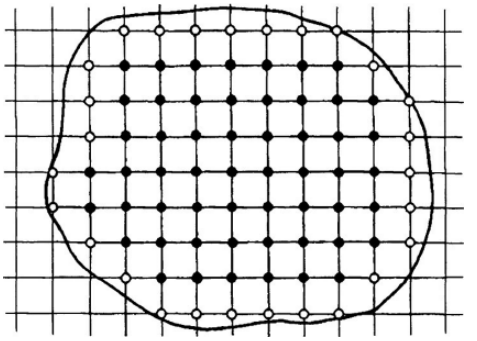

Fig. 3.2 Schematic representation of a grid superimposed upon the region of space which contains the boundary of interest. Closed circles show 'interior' positions; open circles show boundary points.

or

$$u(x, y) = [u(x + \Delta, y) + u(x - \Delta, y) + u(x, y + \Delta) + u(x, y - \Delta)]/4. \tag{3.8}$$

If we examine the behavior of the function $u(r)$ at points on a grid with lattice spacing Δ, we may give this equation a probabilistic interpretation. If we consider a grid of points in the xy-plane with a lattice spacing of Δ, then the probability of a random walk returning to the point (x, y) from any of its nearest neighbor sites is $1/4$. If we place the boundary on the grid, as shown in Fig. 3.2, a random walk will terminate at a boundary point (x', y'), where the variable u has the value

$$u(x', y') = f(x', y'). \tag{3.9}$$

One can then estimate the value of $u(x, y)$ by executing many random walks which begin at the point (x, y) as the average over all N walks which have been performed:

$$u(x, y) \approx \frac{1}{N} \sum_i f(x_i', y_i'). \tag{3.10}$$

After a large number of such walks have been performed, a good estimate of $u(x, y)$ will be produced, but the estimate will depend upon both the coarseness of the grid as well as the number of random walks generated.

Example

Consider two concentric, circular conductors in a plane which are placed into the center of a square box which is 20 cm on a side. The inner conductor has a radius of 4 cm and carries a potential of 2 V; the outer conductor has a radius of 16 cm and has a potential of 4 V. What is the potential halfway between the two conductors? Consider a square box with an $L \times L$ grid. Execute N random walks and follow the estimates for the potential as a function of N for different grid sizes L. Note that the variation of the estimates with grid size is not simple.

N	$L = 10$	$L = 20$	$L = 40$	$L = 80$	$L = 160$	$L = 320$
500	3.6560	3.3000	3.2880	3.3240	3.3400	3.3760V
5 000	3.5916	3.2664	3.3272	3.3372	3.3120	3.3172
10 000	3.6318	3.2886	3.3210	3.3200	3.3128	3.3222
50 000	3.6177	3.2824	3.3149	3.3273	3.3211	3.3237
100 000	3.6127	3.2845	3.3211	3.3240	3.3243	3.3218

exact value = 3.3219 V.

Of course, with these comments and the preceding example we only wish to provide the flavor of the idea – more detailed information can be found in a comprehensive book (Sabelfeld, 1991).

3.4 SIMULATION OF RADIOACTIVE DECAY

One of the simplest examples of a physical process for which the Monte Carlo method can be applied is the study of radioactive decay. Here one begins with a sample of N nuclei which decay at rate λ sec^{-1}. We know that the physics of the situation specifies that the rate of decay is given by

$$dN/dt = -\lambda N, \tag{3.11}$$

where the nuclei which decay during the time interval dt can be chosen randomly. The resultant time dependence of the number of undecayed nuclei is

$$N = N_o e^{-\lambda t} \tag{3.12}$$

where N_o is the initial number of nuclei and λ is related to the 'half-life' of the system. In the most primitive approach, the position of the nuclei plays no role and only the number of 'undecayed' nuclei is monitored. Time is divided into discrete intervals, and each undecayed nucleus is 'tested' for decay during the first time interval. The number of undecayed nuclei is determined, time is then incremented by one step, and the process is repeated so that the number of undecayed nuclei can be determined as a function of time. The time discretization must be done intelligently so that a reasonable number of decays occur in each time step or the simulation will require too much CPU time to be effective. On the other hand, if the time step is chosen to be too large, then so many decays occur during a given interval that there is very little time resolution. This entire process may be repeated many times to obtain a series of independent 'experiments' and the mean value of N, as well as an error bar, may be determined for each value of time. Note that since each 'sample' is independent of the others, measurements for each value of time are uncorrelated even though there may be correlations between different times for a single sample. The extension to systems with multiple decay paths is straightforward.

Problem 3.4 Given a sample with 10 000 radioactive nuclei each of which decays at rate p per second, what is the half-life of the sample if $p = 0.2$?

(Hint: The most accurate way to determine the half-life is not to simply determine the time which it takes for each sample to decay to half its original size. What does physics tell you about the expected nature of the decay for all times?)

3.5 SIMULATION OF TRANSPORT PROPERTIES

3.5.1 Neutron transport

Historically the examination of reactor criticality was among the first problems to which Monte Carlo methods were applied. The fundamental question at hand is the behavior of large numbers of neutrons inside the reactor. In fact, when neutrons traveling in the moderator are scattered, or when a neutron is absorbed in a uranium atom with a resultant fission event, particles fly off in random directions according to the appropriate differential cross-sections (as the conditional probabilities for such scattering events are called). In principle, these problems can be described by an analytic theory, namely integro-differential equations in a six-dimensional space (Davison, 1957); but this approach is rather cumbersome due to the complicated, inhomogeneous geometry of a reactor that is composed of a set of fuel elements surrounded by moderator, shielding elements, etc. In comparison, the direct simulation of the physical processes is both straightforward and convenient. (Note that such types of simulations, where one follows the trajectories of individual particles, belong to a class of methods that is called 'event-driven Monte Carlo'.)

To begin with we consider a neutron with energy E that is at position $\mathbf{r}$ at time t and moving with constant velocity in the direction of the unit vector $\mathbf{u}$. The neutron continues to travel in the same direction with the same energy until at some point on its straight path it collides with some atom of the medium. The probability that the particle strikes an atom on an infinitesimal element of its path is $\sigma_c \delta s$, where σ_c is the cross-section for the scattering or absorption event. The value of σ_c depends on E and the type of medium in which the neutron is traveling. If we consider a path of length s which is completely inside a single medium (e.g. in the interior of a uranium rod, or inside the water moderator, etc.), the cumulative distribution of the distances s that the particle travels before it hits an atom of the medium is $P_c(s) = 1 - \exp(-\sigma_c s)$.

In the Monte Carlo simulation we now simply keep track of the particles from collision to collision. Starting from a state $(E, \mathbf{u}, \mathbf{r})$, we generate a distance s with the probability $P_c(s)$ (if the straight line from $\mathbf{r}$ to $\mathbf{r} + s\mathbf{u}$ does not intersect any boundary between different media). Now the particle has a collision at the point $\mathbf{r}' = \mathbf{r} + s\mathbf{u}$. If there is a boundary, one only allows the particle to proceed up to the boundary. If this is the outer boundary, this means that the neutron has escaped to the outside world and it is not considered further. If it is an interior boundary between regions, one repeats the above procedure, replacing $\mathbf{r}$ by the boundary position, and adjusts σ_c to be the appropriate value for the new region that the neutron has entered. This procedure is

valid because of the Markovian character of the distribution $P_c(s)$. Note that E determines the velocity $\mathbf{v}$ of each neutron, and thus the time t' of the next event is uniquely determined. The collision process itself is determined by an appropriate differential cross-section, e.g. for an inelastic scattering event it is $d^2\sigma / d\Omega\, d\omega$, where Ω is the solid angle of the scattering (with the z-axis in the direction of $\mathbf{u}$) and $\hbar\omega = E' - E$ the energy change. These cross-sections are considered to be known quantities because they can be determined by suitable experiments. One then has to sample E' and the angles $\Omega = (\theta, \varphi)$ from the appropriate conditional probability.

Now, one problem in reactor criticality is that the density function $\rho(E, \mathbf{u}, r)$ will develop in time with a factor $\exp[\mu(t' - t)]$: if $\mu > 0$, the system is supercritical, whereas if $\mu < 0$, it is subcritical. In order to keep the number of tracks from either decreasing or increasing too much, reweighting techniques must be used. Thus, if μ is rather large, one randomly picks out a neutron and discards it with probability p. Otherwise, the neutron is allowed to continue, but its weight in the sample is increased by a factor $(1 - p)^{-1}$. The value p can be adjusted such that the size of the sample (i.e. the number of neutron tracks that are followed) stays asymptotically constant.

3.5.2 Fluid flow

The direct simulation Monte Carlo method (Bird, 1987; Watanabe *et al.*, 1994) has proven to be useful for the simulation of fluid flow from an atomistic perspective. The system is divided into a number of cells, and trajectories of particles are followed for short time intervals by decoupling interparticle collisions. Collision subcells are used in which interparticle collisions are treated on a probabilistic basis. The size of the collision subcells must be monitored so that it is smaller than the mean free path of the particles; otherwise atomistic information is lost. (Thus, the method is well suited to the study of gases but should not be expected to work well for very dense fluids.) This method has succeeded in delivering information about a number of different systems. For example, this technique produces vortices in a flow field. The direct simulation Monte Carlo method has also been used to study the transition from conduction to convection in a Rayleigh–Bénard system, complete with the formation of convection rolls, as the bottom plate is heated. The results for this problem compared quite favorably with those from solution of the Navier–Stokes equation. Typically a system of 40×20 sampling cells each of which contained 5×5 collision cells was used. Each collision cell contained between 16 and 400 particles. One result of this study was the discovery that semi-slip boundary conditions at the top and bottom are inadequate; instead strict diffuse boundary conditions must be used.

3.6 THE PERCOLATION PROBLEM

A geometric problem which has long played a significant role in statistical mechanics is that of 'percolation'. Percolation processes are those in which, by

the random addition of a number of objects, a contiguous path which spans the entire system is created. In general, particles may be distributed continuously in space and the overlap between particles determines the connected paths; however, for our purposes in the first part of this discussion we shall confine ourselves to lattice systems in which the random creation of bonds eventually leads to a connected 'cluster' which spans the lattice. We shall briefly discuss some aspects of percolation here. Percolation has a long history of study by various numerical methods, and for the reader who is interested in obtaining a more thorough knowledge of various aspects of percolation theory, we emphasize that other literature will provide further information (Stauffer and Aharony, 1994).

3.6.1 Site percolation

A lattice is composed of a periodic array of potential occupation sites. Initially the lattice is empty, i.e. none of the sites is actually occupied. Sites are then randomly occupied with probability p and clusters are formed of occupied sites which are neighbors, i.e. bonds are drawn between all occupied nearest neighbor sites. The smallest cluster can then be a single site if none of the nearest neighbor sites is occupied. Two different properties of the system can be determined directly. First of all, for each value of p the probability P_{span} of having a spanning, or 'infinite' cluster may be determined by generating many realizations of the lattice and counting the fraction of those cases in which a spanning cluster is produced. As the lattice size becomes infinite, the probability that a spanning cluster is produced becomes zero for $p < p_c$ and unity for $p > p_c$. Another important quantity is the order parameter M which corresponds to the fraction of occupied sites in the lattice which belong to the infinite cluster. The simplest way to determine M through a simulation is to generate many different configurations for which a fraction p of the sites is occupied and to count the fraction of states for which an infinite cluster appears. For relatively sparsely occupied lattices M will be zero, but as p increases eventually we reach a critical value $p = p_c$ called the 'percolation threshold' for which $M > 0$. As p is increased still further, M continues to grow. The behavior of the percolation order parameter near the percolation threshold is given by an expression which is reminiscent of that for the critical behavior of the order parameter for a temperature induced transition given in Section 2.1.2:

$$M = B(p - p_c)^{\beta} \tag{3.13}$$

where $(p - p_c)$ plays the same role as $(T_c - T)$ for a thermal transition. Of course, for a finite L^d lattice in d-dimensions the situation is more complicated since it is possible to create a spanning cluster using just dL bonds as shown in Fig. 3.3. Thus, as soon as $p = d/L^{d-1}$ the percolation probability becomes non-zero even though very few of the clusters percolate. For random placement of sites on the lattice, clusters of all different sizes are formed and percolation clusters, if they exist, are quite complex in shape. (An example is shown in Fig. 3.3b.) The characteristic behavior of M vs. p is shown for a range of

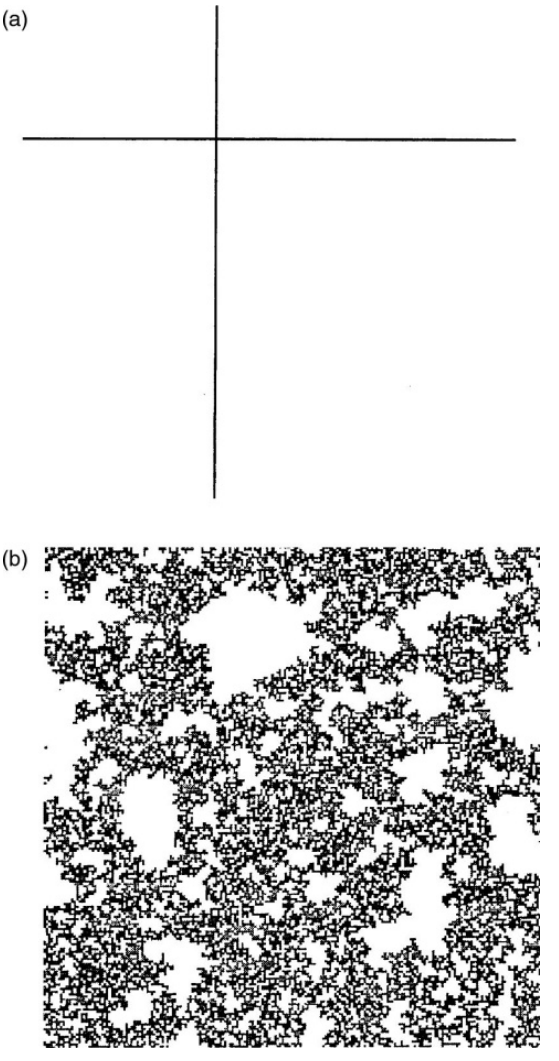

Fig. 3.3 Site percolation clusters on an $L \times L$ lattice: (a) simplest 'infinite cluster'; (b) random infinite cluster.

lattice sizes in Fig. 3.4. As the lattice size increases, the finite size effects become continuously smaller. We see that M (defined as P_∞ in the figure) rises smoothly for values of p that are distinctly smaller than p_c rather than showing the singular behavior given by Eqn. (3.13). As L increases, however, the curves become steeper and steeper and eventually Eqn. (3.13) emerges for macroscopically large lattices. Since one is primarily interested in the behavior of macroscopic systems, which clearly cannot be simulated directly due to limitations on CPU time and storage, a method must be found to extrapolate the results from lattice sizes L which are accessible to $L \to \infty$. We will take up this issue again in detail in Chapter 4. The moments of the cluster size

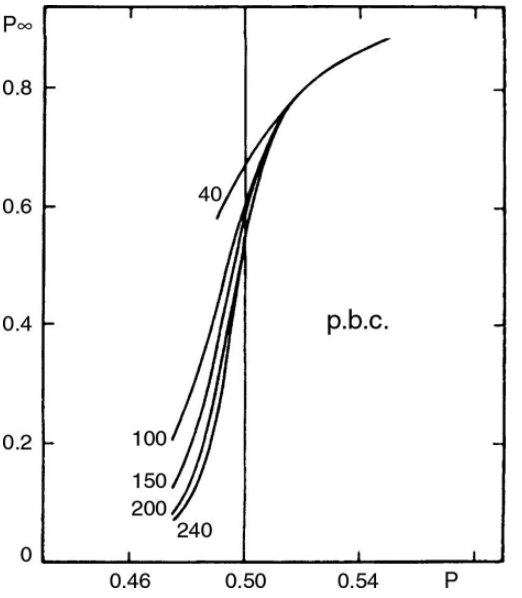

Fig. 3.4 Variation of the percolation order parameter M with p for bond percolation on $L \times L$ lattices with periodic boundary conditions (p.b.c.). The solid curves show finite lattice results and the vertical line shows the percolation threshold. From Heermann and Stauffer (1980).

distribution also show critical behavior. Thus, the equivalent of the magnetic susceptibility may be defined as

$$\chi = \sum_{c} s^2 n(s), \tag{3.14}$$

where $n(s)$ is the number of clusters of size s and the sum is over all clusters. At the percolation threshold the cluster size distribution $n(s)$ also has characteristic behavior

$$n(s) \propto s^{-\tau}, \qquad s \to \infty, \tag{3.15}$$

which implies that the sum in Eqn. (3.14) diverges for $L \to \infty$.

The implementation of the Monte Carlo method to this problem is, in principle, quite straightforward. For small values of p it is simplest to begin with an empty lattice, and randomly fill the points on the lattice, using pairs (in two dimensions) of random integers between 1 and L, until the desired occupation has been reached. Clusters can then be found by searching for connected pairs of nearest neighbor occupied sites. For very large numbers of occupied sites it is easiest to start with a completely filled lattice and randomly empty the appropriate number of sites. In each case it is necessary to check that a point is not chosen twice, so in the 'interesting' region where the system is neither almost empty nor almost full, this method becomes inefficient and a different strategy must be found. Instead one can go through an initially empty lattice, site by site, filling each site with probability p. At the end of this sweep the actual concentration of filled sites is liable to be different from p, so a few sites will need to be randomly filled or emptied until the desired value of p is reached. After the desired value of p is reached the properties of the system are determined. The entire process can be repeated many times so that we can

Fig. 3.5 Labeling of clusters for site percolation on a square lattice. The question mark shows the 'conflict' which arises in a simple labeling scheme.

obtain mean values of all quantities of interest as well as determine the error bars of the estimates.

Problem 3.5 Consider an $L \times L$ square lattice with $L = 16$, 32, and 64. Determine the percolation probability for site percolation as a function of p. Estimate the percolation threshold.

3.6.2 Cluster counting: the Hoshen–Kopelman algorithm

In order to identify the clusters in a system and to determine the largest cluster and see if it is a percolating cluster, a rapid routine must be devised. A very fast 'single-pass' routine by Hoshen and Kopelman (1976) is simple to implement and quite efficient. It is rather easy to identify clusters by going through each row of the lattice in turn and labeling each site which is connected to a nearest neighbor with a number. Thus the cluster label $L_{i,j} = n$ for each occupied site, where n is the cluster number which is assigned when looking to see if previously inspected sites are nearest neighbors or not. This process is shown for the first row of a square lattice in Fig. 3.5. The difficulty which arises from such a direct approach becomes obvious when we consider the third row of the lattice at which point we realize that those sites which were initially assigned to cluster 1 and those assigned to cluster 2 actually belong to the same cluster. A second pass through the lattice may be used to eliminate such errors in the cluster assignment, but this is a time consuming process. The Hoshen–Kopelman method corrects such mislabeling 'on the fly' by introducing another set of variables known as the 'labels of the labels', N_n. The 'label of the label' keeps track of situations in which we discover that two clusters actually belong to the same cluster, i.e. that an occupied site has two nearest neighbors which have already been assigned different cluster numbers. When this happens the 'label of the label' which is larger is set to the negative of the value of the smaller one (called the 'proper' label) so that both 'clusters' are identified as actually belonging to the same cluster and the proper label is set equal to the total size of the cluster. Thus in Fig. 3.5 we see that after examination of the third row has been completed, $N_1 = 7$, and $N_2 = -1$. The Hoshen–Kopelman method finds a wide range of application beyond the simple percolation problem mentioned here.

Of course, there are many other properties of the clusters which are interesting. As an example we mention the 'backbone', which is that portion of the cluster which forms a connected path with no dangling ends between the two most distant points. This information is lost during implementation of

the Hoshen–Kopelman algorithm, but other types of 'depth first' and 'breadth first' searches may be used, see e.g. Babalievski (1998), which retain more information. These generally sacrifice the very efficient use of storage in order to keep more detail.

Problem 3.6 Use the Hoshen–Kopelman method to determine the cluster size dependence for the site dilution problem with $L = 64$ and $p = 0.59$. Can it be described in terms of a power law?

3.6.3 Other percolation models

The simplest variation of the percolation model discussed above is the case where the bonds are thrown on the lattice randomly and clusters are formed directly from connected bonds. All of the formalism applied to the site problem above is also valid, and 'bond percolation' problems have been studied quite extensively in the past. The major difference is that clusters, defined in terms of connected lattice size, may have a minimum size of 2. A physical motivation for the study of such models comes from the question of the nature of the conductivity of disordered materials ('random resistor networks'). Another class of models results if we remove the restriction of a lattice and allow particles to occupy positions which vary continuously in space. 'Continuum percolation', as it is called, suffers from the added complication that tricks which can sometimes be used on lattice models cannot be applied (Meester and Roy, 1996). An important aspect of continuum percolation is the dependence upon the shapes of the objects whose percolation is being considered, e.g. spheres, rods, or platelets. Such questions arise when materials are produced where carbon nanotubes or graphene sheets, randomly placed in a polymeric matrix, may provide an electrical 'percolation conductivity' (Schilling *et al.*, 2007; Mathew *et al.*, 2012). A quite different process is known as 'invasion percolation'; its invention was prompted by attempts to understand flow in porous media by Wilkinson and Willemsen (1983). Random numbers are assigned to each site of a lattice. Choose a site, or sites, on one side of the lattice and draw a bond to the neighbor which has the lowest random number assigned to it. (The growing cluster represents the invading fluid with the remainder of the sites representing the initial, or defending, fluid.) This process continues until the cluster reaches the other side (i.e. the exit).

3.7 FINDING THE GROUNDSTATE OF A HAMILTONIAN

For systems with Hamiltonians the groundstate is usually a relatively unique, minimum energy state. If the groundstate of a system is not known, a simple Monte Carlo simulation can be used to find states of low energy, and hopefully that of lowest energy. For purposes of discussion we will consider a system of Ising spins. Some initial, randomly chosen state of the system is selected

and then one proceeds through the lattice determining the change in energy of the system if the spin is overturned; if the energy is lowered the spin is overturned, otherwise it is left unchanged and one proceeds to the next site. The system is swept through repeatedly, and eventually no spin-flips occur; the system is then either in the groundstate or in some metastable state. This process can be repeated using different initial configurations, and one tests to see if the same state is reached as before or if a lower energy state is found. For systems with very complicated energy landscapes (i.e. the variation of the energy as some parameter x is changed) there may be many energy minima of approximately the same depth and a more sophisticated strategy will have to be chosen. This situation will be discussed in the next chapter. In some cases relatively non-local metastable structures, e.g. anti-phase domains, are formed and cannot be removed by single spin-flips. (Anti-phase domains are large regions of well ordered structures which are 'shifted' relative to each other and which meet at a boundary with many unsatisfied spins.) It may then be helpful to introduce multiple spin-flips or other algorithmic changes as a way of eliminating these troublesome defects. In all cases it is essential to begin with diverse initial states and check that the same 'groundstate' is reached.

An interesting class of problems are those in which the free energy landscape is rough, i.e. there are many free energy minima that are well separated in parameter space so that it is easy to become trapped in a 'local minimum'. In some cases the problem is made even more difficult because there are not only energetic barriers but also entropic constraints. This important and fascinating topic will be discussed in later chapters in which the more sophisticated simulational methods that are essential for studying such problems are presented.

Example

Consider an $L \times L$ Ising square lattice in which all spins to the left of a diagonal are initially up and all those to the right are down. All portions of the system are in their lowest energy state except for those spins which are in the domain wall between the up-spin and down-spin regions. Since the spins in the domain wall have equal numbers of up and down neighbors they cannot lower their energy by overturning, but if we allow those spins to flip with 50% probability, we provide the method with a way of eventually eliminating the domain structure.

3.8 GENERATION OF 'RANDOM' WALKS

3.8.1 Introduction

In this sub-section we shall briefly discuss random walks on a lattice which is a special case of the full class of random walks. A random walk consists of a connected path formed by randomly adding new bonds to the end of the existing walk, subject to any restrictions which distinguish one kind of random

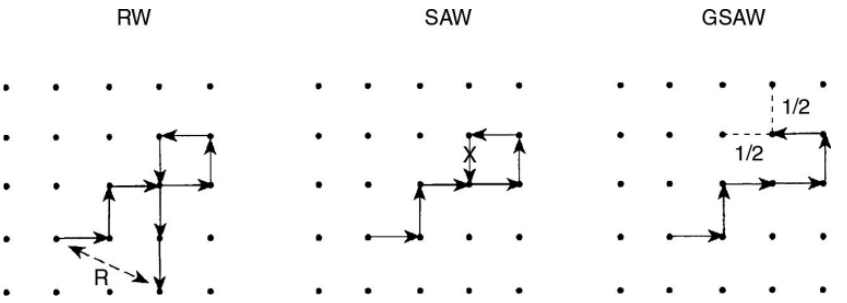

Fig. 3.6 Examples of different kinds of random walks on a square lattice. For the RW every possible new step has the same probability. For the SAW the walk dies if it touches itself. The GSAW walker recognizes the danger and takes either of the two steps shown with equal probability.

walk from another. The mean-square end-to-end distance $\langle R^2 \rangle$ of a walk with N steps may diverge as N goes to infinity as (de Gennes, 1979)

$$\langle R^2(N) \rangle = aN^{2\nu}(1 + bN^{-\Delta} + \cdots) \tag{3.16}$$

where ν is a 'critical exponent' that determines the universality class. Here a and b are some 'non-universal' constants which depend on the model and lattice structure chosen and Δ is a 'correction to scaling' exponent. In such cases there is a strong analogy to critical behavior in percolation or in temperature-driven transitions in systems of interacting particles. The equivalent of the partition function for a system undergoing a temperature-driven transition is given by the quantity Z_N, which simply counts the number of distinct random walks on the lattice and which behaves as

$$Z_N \propto N^{\gamma - 1} q_{\text{eff}}^N \tag{3.17}$$

as $N \to \infty$. γ is another critical exponent and q_{eff} is an effective coordination number which is related to the exchange constant in a simple magnetic model. The formalism for describing this geometric phenomenon is thus the same as for temperature-driven transitions, even including corrections to scaling in the expression for the mean-square end-to-end distance as represented by the term in $N^{-\Delta}$ in Eqn. (3.16). The determination of ν and γ for different kinds of walks is essential to the classification of these models into different universality classes. We now know that the lattice dimensionality as well as the rules for the generation of walks affect the critical exponents and thus the universality class (Kremer and Binder, 1988). Examples of several kinds of walks are shown in Fig. 3.6.

3.8.2 Random walks

For simple, random walks (RW) the walker may cross the walk an infinite number of times with no cost. In d dimensions the end-to-end distance diverges with the number of steps N according to

$$\sqrt{\langle R^2(N) \rangle} \propto N^{\frac{1}{2}}. \tag{3.18}$$

A simulation of the simple random walk can be carried out by picking a starting point and generating a random number to determine the direction of each subsequent, additional step. After each step the end-to-end distance can be calculated. Errors may be estimated by carrying out a series of independent random walks and performing a statistical analysis of the resultant distribution. Thus, the simple RW has the trivial result $\nu = 1/2$ but is not really very useful in understanding physical properties of polymers in dilute solution; but random walks have great significance for the description of diffusion phenomena – the number of steps N is then related to time.

At this point we briefly mention a simple variant of the RW for which the choice of the $(n+1)$ step from the nth step of a return to the point reached at the $(n-1)$ step, i.e. an 'immediate reversal', is forbidden. Although for this so-called 'non-reversal random walk' (NRRW) the exponents remain unchanged, i.e. $\nu = 1/2, \gamma = 1$, as for the ordinary RW, prefactors change. This means that in Eqn. (3.17) $q_{\text{eff}} = (q-1)$ for the NRRW whereas $q_{\text{eff}} = q$ for the ordinary RW, etc. This NRRW model represents, in fact, a rather useful approach for the modeling of polymer configurations in dense melts, and since one merely has to keep track of the previous step and then choose one of the remaining $q-1$ possibilities, it is straightforward to implement. Furthermore, this NRRW model is also a good starting point for the simulation of 'self-avoiding walks', a topic to which we shall turn in the next section.

Problem 3.7 Perform a number of random walk simulations to estimate the value of ν for a simple random walk on a square lattice. Give error bars and compare your result with the exact answer in Eqn. (3.18).

3.8.3 Self-avoiding walks

In contrast to the simple random walk, for a self-avoiding walk (SAW), the walker dies when attempting to intersect a portion of the already completed walk. (Immediate reversals are inherently disallowed.) There has been enormous interest in this model of SAWs since this is the generic model used to probe the large scale statistical properties of the configurations of flexible macromolecules in good solvents. Although it is possible to carry out an exact enumeration of the distribution of walks for small N, it is in general not possible to extract the correct asymptotic behavior for the range of N which is accessible by this method. Monte Carlo methods have also been used to study much larger values of N for different kinds of walks, but even here very slow crossover as a function of N has complicated the analysis. After each step has been added, a random number is used to decide between the different possible choices for the next step. If the new site is one which already contains a portion of the walk, the process is terminated at the Nth step. Attrition becomes a problem and it becomes difficult to generate large numbers of walks with large N. The most simple minded approach to the analysis of the data is to simply make a plot of $\log\langle R^2(N)\rangle$ vs. $\log N$ and to calculate ν from the slope. If corrections to scaling are present, the behavior of the data may become quite subtle and

a more sophisticated approach is needed. The results can instead be analyzed using traditional 'ratio methods' which have been successful in analyzing series expansions. In this manner we can calculate an 'effective exponent' by forming the ratio

$$\nu(N) = \frac{1}{2}\frac{\ln[\langle R^2(N+i)\rangle/\langle R^2(N-i)\rangle]}{\ln\,[(N+i)/(N-i)]} \tag{3.19}$$

for different values of $i \ll N$. The values of i must be chosen to be large enough to help eliminate 'short time' noise in the comparison of nearby values but small enough that the effects of correction terms do not infect the effective exponent estimate. The effective exponent is then related to the true value, i.e. for $N = \infty$, by

$$\nu(N) = \nu - 1/2\; bN^{-\Delta} + \cdots. \tag{3.20}$$

Thus, by extrapolation to $N \to \infty$ we can extract a rather accurate estimate for the (asymptotic) exponent. This method, which is introduced here for convenience, is not restricted to SAWs and can be applied to many problems involving enumeration. For SAWs the current estimates for ν are (Kremer and Binder, 1988)

$$\nu = 3/4, \qquad d = 2, \tag{3.21a}$$
$$\nu \approx 0.588, \qquad d = 3, \tag{3.21b}$$
$$\nu = 1/2, \qquad d \geq 4. \tag{3.21c}$$

The exponent γ is also of great interest and numerical estimates can be made by comparing the values of the 'partition function' which are obtained for two successive values of N, i.e. using Eqn. (3.17):

$$\ln\left(\frac{Z(N)}{Z(N-1)}\right) = \ln q_{\text{eff}} + (\gamma - 1)/N. \tag{3.22}$$

Here, too, a more sensitive analysis can be made by using 'symmetric' values in step number by looking not only at $Z(N)$ but also $Z(N+i)$ and $Z(N-i)$ so that

$$\ln\frac{Z(N)}{Z(N-i)} - \ln\frac{Z(N+i)}{Z(N)} = (\gamma-1)\ln\frac{N^2}{(N-i)(N+i)} \to (1-\gamma)i^2/N^2. \tag{3.23}$$

Once again, i must be chosen to be sufficiently large that the effects of 'short time' fluctuations are minimized but small enough that curvature effects do not enter.

Although these techniques are very straightforward, many research problems of current interest remain that one can solve with them. For example, consider the case of a star polymer adsorbed with its core on a wall as shown in Fig. 3.7a. While in two dimensions we expect that the size of a polymer scales with the number of monomers as $R \sim N^{\nu} = N^{3/4}$, for a star polymer we have

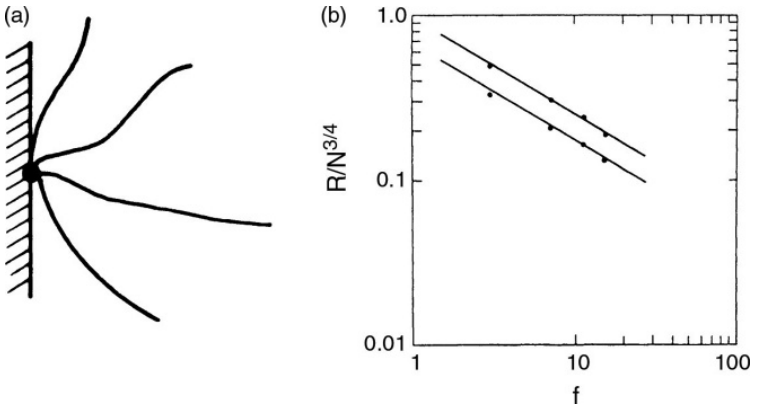

Fig. 3.7 (a) A two-dimensional star polymer consisting of $f = 4$ flexible arms covalently linked together in a core (dot) adsorbed at a one-dimensional repulsive wall (shaded). (b) Log–log plot of $R/N^{3/4}$ for center-adsorbed stars plotted vs. the number of arms, where $N = fl$ is the total number of monomers, $l = 50$ is the number of monomers per arm, and R is the center-end distance of the arms (upper set of points) or the mean distance of a monomer from the center (lower set of points). Straight lines illustrate agreement with the theoretical prediction $R/N^{3/4} \propto f^{-1/2}$. From Ohno and Binder (1991).

the additional question of how the number of arms f affects the scaling in the macromolecular object. This question was studied using a simple sampling Monte Carlo method by Ohno and Binder (1991). To remedy the attrition problem mentioned earlier, they used a variation of simple sampling known as the enrichment method. Treating each arm as a self-avoiding walk on a lattice with q-fold coordination, and avoiding immediate reversals, they added each new bond to randomly connect to one of the $(q - 1)$ neighbor sites. Thus, for example, on a square lattice the probability that the self-avoiding walk does not 'die' in this step is $(3/q_{\text{eff}})^{-f} \approx 0.880^f$. For large f the probability of growing a star polymer with long arms would be vanishingly small. Thus, the recipe is to attempt to add a bond to each arm not just once but many times, i.e. on average $m = (3/q_{\text{eff}})^f$ times, and keeping track of the survivors. In this way a 'population' of star polymers that is grown in parallel from $\mathcal{N}$ centers neither dies out nor explodes in number as bonds are added consecutively to create arms of length l. Of course there is a price that must be paid: different star polymers that 'survive' in the final 'generation' are not, in general, statistically independent of each other. Nevertheless, this method is useful in a practical sense. At this point, we also draw attention to the fact that the self-avoiding walk problem can also be studied using the importance sampling method (see Section 4.7).

3.8.4 Growing walks and other models

Because of the attrition, the generation of long SAWs is quite difficult. (Of course, the simple sampling of SAWs is mentioned here largely for historical reasons, and other methods, e.g. the pivot algorithm, PERM, etc., yield much

more accurate results.) An alternative strategy is to attempt to develop 'smart walks' which find a way to avoid death. An example is the growing self-avoiding walk (GSAW) (Lyklema and Kremer, 1986). The process proceeds at first as for SAWs, but a walker senses a 'trap' and chooses instead between the remaining 'safe' directions so that it can continue to grow. Other, still 'smarter' walks have been studied numerically (Lyklema, 1985) and a number of sophisticated methods have been devised for the simulation of polymeric models (Baumgärtner, 1992).

To a great extent modeling has concentrated on the 'good solvent' case in which polymers are treated as pure SAWs; however, in θ-solutions the solvent–monomer repulsion leads to a net attraction between the chain monomers. Thus, the SAW can be generalized by introducing an energy that is won if two nearest neighbor sites are visited by the walk. Of course, the weighting of configurations then requires appropriate Boltzmann factors (Kremer and Binder, 1988). Exactly at the θ-point the SAW condition and the attraction cancel and the exponents become those of the simple random walk. The θ-point may then be viewed as a kind of tricritical point, and for $d = 3$ the exponents should thus be mean-field-like. We shall return to the θ-point in Section 4.7.6.

3.9 FINAL REMARKS

In closing this chapter, we wish to emphasize that there are related applications of Monte Carlo 'simple sampling' techniques outside of statistical physics which exist in broad areas of applied mathematics, also including the so-called 'quasi-Monte Carlo methods' (Niederreiter, 1992). These applications deal with mathematical problems (Monte Carlo algorithms for calculating eigenvalues, or for solving integro-differential equations, etc.) and various applications ranging from economy to technology (option pricing, radiosity and illumination problems, computer graphics, road visibility in fog, etc.). One difficulty with quasi-Monte Carlo methods is that error estimation is not straightforward. In fact, Schlier (2004) has shown that predictors of the asymptotic discrepancy function which are often used as measures of the 'quality' of the results actually have little relevance in practical situations. Such problems are completely outside of the scope of our presentation; however, we direct the interested reader to Niederreiter *et al*. (1998) for a series of review articles.

REFERENCES

Babalievski, F. (1998), *J. Mod. Phys. C* **9**, 43.

Baumgärtner, A. (1992), in *The Monte Carlo Method in Condensed Matter Physics*, ed. K. Binder (Springer Verlag, Heidelberg).

Bird, G. A. (1987), *Phys. Fluids* **30**, 364.

Cooper, N. G. (ed.) (1989), *From Cardinals to Chaos* (Cambridge University Press, Cambridge).

Davison, B. (1957), *Neutron Transport Theory* (Oxford University Press, Oxford).
de Gennes, P. G. (1979), *Scaling Concepts in Polymer Physics* (Cornell University Press, Ithaca, New York).
Hammersley, J. H. and Handscomb, D. C. (1964), *Monte Carlo Methods* (Wiley, New York).
Heermann, D. W. and Stauffer, D. (1980), *Z. Phys. B* **40**, 133.
Hoshen, J. and Kopelman, R. (1976), *Phys. Rev. B* **14**, 3438.
Kremer, K. and Binder, K. (1988), *Comput. Phys. Rep*. **7**, 261.
Lyklema, J. W. (1985), *J. Phys. A* **18**, L617.
Lyklema, J. W. and Kremer, K. (1986), *J. Phys. A* **19**, 279.
Mathew, M., Schilling, T., and Oettel, M. (2012), *Phys. Rev. E* **85**, 061407.
Meester, R. and Roy, R. (1996), *Continuum Percolation* (Cambridge University Press, Cambridge).
Niederreiter, H. (1992), *Random Number Generation and Quasi-Monte Carlo Methods* (SIAM, Philadelphia, PA).
Niederreiter, H., Hellekalek, P., Larcher, G., and Zinterhof, P. (1998), *Monte Carlo and Quasi-Monte Carlo Methods* (Springer, New York, Berlin).
Ohno, K. and Binder, K. (1991), *J. Stat. Phys*. **64**, 781.
Sabelfeld, K. K. (1991), *Monte Carlo Methods in Boundary Value Problems* (Springer, Berlin).
Schilling, T., Jungblut, S., and Miller, M. A. (2007), *Phys. Rev. Lett*. **98**, 108303.
Schlier, C. (2004), *Comput. Phys. Commun*. **159**, 93.
Stauffer, D. and Aharony, A. (1994), *Introduction to Percolation Theory* (Taylor and Francis, London).
Watanabe, T., Kaburaki, H., and Yokokawa, M. (1994), *Phys. Rev. E* **49**, 4060.
Wilkinson, D. and Willemsen, J. F. (1983), *J. Phys. A* **16**, 3365.