Chapter 15

Monte Carlo Simulations of Thermal Systems

©2005 by Harvey Gould, Jan Tobochnik, and Wolfgang Christian
22 March 2005

We discuss how to simulate thermal systems using a variety of Monte Carlo methods including the traditional Metropolis algorithm. Applications to the Ising model and various particle systems are discussed and more efficient Monte Carlo algorithms are introduced.

15.1 Introduction

The Monte Carlo simulation of the particles in the box problem discussed in Chapter 7 and the molecular dynamics simulations discussed in Chapter 8 have exhibited some of the important qualitative features of macroscopic systems such as the irreversible approach to equilibrium and the existence of equilibrium fluctuations in macroscopic quantities. In this chapter we apply various Monte Carlo methods to the simulation of the equilibrium properties of thermal systems. These applications will allow us to explore some of the important concepts of statistical mechanics.

Due in part to the impact of computer simulations, the applications of statistical mechanics have expanded from the traditional areas of dense gases, liquids, crystals, and simple models of magnetism to the study of complex materials, particle physics, and theories of the early universe. For example, the demon algorithm introduced in Section 15.3 was developed by a physicist interested in lattice gauge theories, which are used to describe the interactions of fundamental particles.

15.2 The Microcanonical Ensemble

We first discuss an isolated system for which the number of particles N, the volume V, and the total energy E are fixed and external influences such as gravitational and magnetic fields can be ignored.

The *macrostate* of the system is specified by the values of E, V, and N. At the microscopic level there are many different ways or *configurations* in which the macrostate (E, V, N) can be realized. A particular configuration or *microstate* is accessible if its properties are consistent with the specified macrostate.

All we know about the accessible microstates is that their properties are consistent with the known physical quantities of the system. Because we have no reason to prefer one microstate over another when the system is in equilibrium, it is reasonable to postulate that the system is *equally* likely to be in any one of its accessible microstates. To make this postulate of *equal a priori probabilities* more precise, imagine an isolated system with Ω accessible states. The probability P_s of finding the system in microstate s is

$$P_s = \begin{cases} 1/\Omega, & \text{if } s \text{ is accessible} \\ 0, & \text{otherwise.} \end{cases} \tag{15.1}$$

The sum of P_s over all Ω states is equal to unity. Equation (15.1) is applicable only when the system is in equilibrium.

The averages of physical quantities can be determined in two ways. In the usual laboratory experiment, the physical quantities of interest are measured over a time interval sufficiently long to allow the system to sample a large number of its accessible microstates. We computed such time averages in Chapter 8, where we used the method of molecular dynamics to compute the time-averaged values of quantities such as the temperature and pressure. An interpretation of the probabilities in (15.1) that is consistent with such a time average is that during a sequence of observations, P_s yields the fraction of times that a single system is found in a given microstate.

Although time averages are conceptually simple, it is convenient to imagine a collection or *ensemble* of systems that are identical mental copies characterized by the same macrostate, but, in general, by different microstates. In this interpretation, the probabilities in (15.1) describe an ensemble of identical systems, and P_s is the probability that a system in the ensemble is in microstate s. An ensemble of systems specified by E, N, V is called a *microcanonical* ensemble. We will see that an advantage of ensembles is that statistical averages can be determined by sampling the states according to the desired probability distribution. In fact, much of the power of Monte Carlo methods is that we can devise sampling methods based on a fictitious dynamics that is more efficient than the real dynamics.

Suppose that a physical quantity A has the value A_s when the system is in microstate s. Then the ensemble average of A is given by

$$\langle A \rangle = \sum_{s=1}^{\Omega} A_s P_s, \tag{15.2}$$

where P_s is given by (15.1).

To illustrate these ideas, consider a one-dimensional system of N noninteracting spins on a lattice. The spins can be in one of two possible directions which we take to be up or down. The total energy of the system is $E = -\mu B \sum_i s_i$, where each lattice site has associated with it a number $s_i = \pm 1$, where $s_i = +1$ for an up spin and $s_i = -1$ for a down spin; B is the magnetic field, and μ is the magnetic moment of a spin. A particular microstate of the system of spins

$\downarrow\downarrow\downarrow\downarrow$	$\downarrow\downarrow\downarrow\uparrow$	$\downarrow\downarrow\uparrow\uparrow$	$\downarrow\uparrow\uparrow\uparrow$	$\uparrow\uparrow\uparrow\uparrow$
	$\downarrow\downarrow\uparrow\downarrow$	$\downarrow\uparrow\downarrow\uparrow$	$\uparrow\downarrow\uparrow\uparrow$	
	$\downarrow\uparrow\downarrow\downarrow$	$\downarrow\uparrow\uparrow\downarrow$	$\uparrow\uparrow\downarrow\uparrow$	
	$\uparrow\downarrow\downarrow\downarrow$	$\uparrow\downarrow\downarrow\uparrow$	$\uparrow\uparrow\uparrow\downarrow$	
		$\uparrow\downarrow\uparrow\downarrow$		
		$\uparrow\uparrow\downarrow\downarrow$		
$4\mu B$	$2\mu B$	0	$-2\mu B$	$-4\mu B$

Table 15.1: The sixteen microstates for a one-dimensional system of $N = 4$ noninteracting spins. The total energy E of each microstate also is shown. If the total energy of the system is $E = -2\mu B$, then there are four accessible microstates (see the fourth column). Hence, in this case the ensemble consists of 4 systems, each in a different microstate with equal probability.

is specified by the set of variables $\{s_1, s_2, \ldots, s_N\}$. In this case the macrostate of the system is specified by E and N.

In Table 15.1 we show the 16 microstates with $N = 4$. If the total energy $E = -2\mu B$, we see that there are four accessible microstates. Hence, in this case there are four systems in the ensemble each with an equal probability. The enumeration of the systems in the ensemble and their probability allows us to calculate ensemble averages for the physical quantities of interest.

Exercise 15.1. A simple ensemble average

Consider a one-dimensional system of $N = 4$ noninteracting spins with total energy $E = -2\mu B$. What is the probability P_n that a given spin is up?

15.3 The Demon Algorithm

We found in Chapter 8 that we can do a time average of a system of many particles with E, V, and N fixed by integrating Newton's equations of motion for each particle and computing the time-averaged value of the physical quantities of interest. How can we do an ensemble average at fixed E, V, and N? And what can we do if there is no equation of motion available? One way would be to enumerate all the accessible microstates and calculate the ensemble average of the desired physical quantities as we did in Table 15.1. However, this approach usually is not practical, because the number of microstates for even a small system is much too many to enumerate. In the spirit of Monte Carlo, we wish to develop a practical method of obtaining a representative sample of the total number of microstates. One possible procedure is to fix N, choose each spin to be up or down at random, and retain the configuration if it has the desired total energy. However this procedure is very inefficient, because most configurations would not have the desired total energy and would have to be discarded.

An efficient Monte Carlo procedure for simulating systems at a given energy was developed by Creutz in the context of lattice gauge theory. Suppose that we add an extra degree of freedom to the original macroscopic system of interest. For historical reasons, this extra degree of freedom is called a *demon*. The demon transfers energy as it attempts to change the dynamical variables of the system. If a desired change lowers the energy of the system, the excess energy is given to the

demon. If the desired change raises the energy of the system, the demon gives the required energy to the system if the demon has sufficient energy. The only constraint is that the demon cannot have negative energy.

To be specific, we first apply the demon algorithm to a one-dimensional classical system of N noninteracting particles of mass m (an ideal gas). The total energy of the system is $E = 1/2m \sum_i v_i^2$, where v_i is the velocity of particle i. In general, the demon algorithm is summarized by the following steps:

1. Choose a particle at random and make a trial change in its coordinates.
2. Compute ΔE, the change in the energy of the system due to the change.
3. If $\Delta E \leq 0$, the system gives the amount $|\Delta E|$ to the demon, that is, $E_d = E_d - \Delta E$, and the trial configuration is accepted.
4. If $\Delta E > 0$ and the demon has sufficient energy for this change ($E_d \geq \Delta E$), then the demon gives the necessary energy to the system, that is, $E_d = E_d - \Delta E$, and the trial configuration is accepted. Otherwise, the trial configuration is rejected and the configuration is not changed.

The above steps are repeated until a representative sample of states is obtained. After a sufficient number of steps, the demon and the system will agree on an average energy for each. The total energy of the system plus the demon remains constant, and because the demon is only one degree of freedom in comparison to the many degrees of freedom of the system, the energy fluctuations of the system will be of order $1/N$, which is very small for $N >> 1$.

Note that an ideal gas has a trivial dynamics. That is, because the particles do not interact, their velocities do not change. (The positions of the particles change, but the positions are irrelevant because the energy depends only on the velocity of the particles.) So the introduction of the demon is equivalent to a fictitious dynamics that lets us sample the microstates of the system.

How do we know that this Monte Carlo simulation of the microcanonical ensemble will yield results equivalent to the time-averaged results of molecular dynamics? The assumption that these two types of averages yield equivalent results is called the *quasi-ergodic* hypothesis. Although these two averages have not been proven to be identical in general, they have been found to yield equivalent results in all cases of interest.

`IdealDemon` and `IdealDemonApp` implement a microcanonical Monte Carlo simulation of an ideal classical gas in one dimension. To change a configuration, we choose a particle at random and change its velocity by a random amount. The variable `mcs`, the number of Monte Carlo steps per particle, plays an important role in Monte Carlo simulations. On the average, the demon attempts to change the velocity of each particle once per Monte Carlo step per particle. We frequently will refer to the number of Monte Carlo steps per particle as the "time," even though this time has no obvious direct relation to a physical time.

Of course, we do not need to apply the demon algorithm to an ideal gas because all its properties can be calculated analytically. However, it is a good idea to consider a simple example first.

Listing 15.1: Code for demon algorithm for one-dimensional ideal gas.

```
package org.opensourcephysics.sip.ch15;

public class IdealDemon {
   public double v[];
   public int N;
   public double systemEnergy;
   public double demonEnergy;
   public int mcs = 0;        // number of MC moves per particle
   public double systemEnergyAccumulator = 0;
   public double demonEnergyAccumulator = 0;
   public int acceptedMoves = 0;
   public double delta;

   public void initialize () {
      v = new double[N]; // array to hold particle velocities
      double v0 = Math.sqrt(2.0*systemEnergy/N);
      for (int i = 0; i < N; ++i) {
        v[i] = v0; // give all particles the same velocity
      }
      demonEnergy = 0;
      resetData();
   }

   public void resetData() {
      mcs = 0;
      systemEnergyAccumulator = 0;
      demonEnergyAccumulator = 0;
      acceptedMoves = 0;
   }

   public void doOneMCStep() {
      for (int j = 0; j < N; ++j) {
         int particleIndex = (int)(Math.random()*N); // choose particle at random
         double dv = (2.0*Math.random() - 1.0)*delta; // random change in velocity
         double trialVelocity = v[particleIndex] + dv;
         double dE = 0.5*(trialVelocity*trialVelocity - v[particleIndex]*v[particleIndex ]);
         if (dE <= demonEnergy) {
            v[particleIndex] = trialVelocity ;
             acceptedMoves++;
             systemEnergy += dE;
             demonEnergy -= dE;
         }
         systemEnergyAccumulator += systemEnergy;
         demonEnergyAccumulator += demonEnergy;
      }
      mcs++;
   }
}
```

Listing 15.2: Target application for the simulation of an ideal gas using the demon algorithm.

```
package org.opensourcephysics.sip.ch15;
import org.opensourcephysics.controls.AbstractSimulation;
import org.opensourcephysics.controls.SimulationControl;

public class IdealDemonApp extends AbstractSimulation {
   IdealDemon idealGas = new IdealDemon();

   public void initialize () {
      idealGas.N = control.getInt("N");
      idealGas.systemEnergy = control.getDouble("systemEnergy");
      idealGas.delta = control.getDouble("delta");
      idealGas. initialize ();
   }

   public void doStep() {
      idealGas.doOneMCStep();
   }

   public void stop() {
      double norm = 1.0/(idealGas.mcs*idealGas.N);
      control.println("mcs = " + idealGas.mcs);
      control.println("<Ed> = " + idealGas.demonEnergyAccumulator*norm);
      control.println("<E> = " + idealGas.systemEnergyAccumulator*norm);
      control.println("acceptance ratio = " + idealGas.acceptedMoves*norm);
   }

   public void reset() {
      control.setValue("N", 40);
      control.setValue("systemEnergy", 40);
      control.setValue("delta" , 2.0);
   }

   public void resetData() {
      idealGas.resetData();
      idealGas.delta = control.getDouble("delta");
      control.clearMessages();
   }

   public static void main (String[] args) {
      SimulationControl control = SimulationControl.createApp(new IdealDemonApp());
      control.addButton("resetData","resetData");
   }
}
```

Problem 15.2. Monte Carlo simulation of an ideal gas

a. We will use `IdealDemon` and `IdealDemonApp` to investigate some of the equilibrium properties of an ideal gas. Begin by using the default values given in the listing of `IdealDemonApp`. Note that the mass of the particles has been set equal to unity and the initial demon energy is zero.

Also, for simplicity, the same initial velocity has been assigned to all the particles. What is the mean value of the particle velocities after equilibrium has been reached?

b. The configuration corresponding to all particles having the same velocity is not very likely, and it would be better to choose an initial configuration that is more likely to occur when the system is in equilibrium. In any case, we should let the system evolve for a number of Monte Carlo steps per particle before we accumulate data for the various averages. We call this number the *equilibration time*. We can estimate the equilibration time from a plot of the demon energy or another quantity of interest versus the time. Alternatively, we can reset the data until the computed averages stop changing systematically. Clicking the `resetData` button sets the accumulated sums to zero without changing the configuration. Determine the mean demon energy, $\langle E_d \rangle$, and the mean system energy per particle using the default values for the parameters.

c. Compute the mean energy of the demon and the mean system energy per particle for $E = 10$ and $E = 20$ and $N = 100$. Use your result from part (b) and obtain an approximate relation between the mean demon energy and the mean system energy per particle.

d. In the microcanonical ensemble the total energy is fixed with no reference to the temperature. Define the kinetic temperature by the relation $\frac{1}{2}m\langle v^2 \rangle = \frac{1}{2}kT_{\text{kinetic}}$, where $\frac{1}{2}m\langle v^2 \rangle$ is the mean kinetic energy per particle of the system. Use this relation to obtain T_{kinetic}. How is T_{kinetic} related to the mean demon energy? How do your results compare to the relation given in most introductory physics textbooks that the total energy of an ideal gas of N particles in three dimensions is $E = \frac{3}{2}NkT$, where k is Boltzmann's constant? (In one dimension the analogous relation is $E = \frac{1}{2}NkT$.) We will choose energy units such that Boltzmann's constant k is equal to unity.

e. A limitation of most simulations is the finite number of particles. Is the relation between the mean demon energy and mean kinetic energy per particle the same for $N = 2$ and $N = 10$ as it is for $N = 40$? If there is no statistically significant difference between your results for the three values of N, explain why finite N might not be an important limitation for the ideal gas in this simulation.

Problem 15.3. Demon energy distribution

a. Add a method to `IdealDemon` to compute the probability $P(E_d)\Delta E_d$ that the demon has energy between E_d and $E_d + \Delta E_d$. Because E_d is a continuous variable, it is necessary to place the values of E_d in appropriate bins of width ΔE_d and compute the number of times that the demon's energy is between E_d and $E_d + \Delta E_d$. Choose the same parameters as in Problem 15.2, and be sure to determine $P(E_d)$ only after equilibrium has been obtained.

b. Plot the natural logarithm of $P(E_d)$ and verify that $\ln P(E_d)$ depends linearly on E_d with a negative slope. What is the absolute value of the slope? How does the inverse of this value correspond to the mean energy of the demon and T_{kinetic} as determined in Problem 15.2? As usual we choose units such that $k = 1$.

c. Generalize the `IdealDemon` class and determine the relation between the mean demon energy, the mean energy per particle of the system, and the inverse of the slope of $\ln P(E_d)$ for an ideal gas in two and three dimensions. (It is straightforward to write the class so that it is valid for any spatial dimension.)

15.4 The Demon as a Thermometer

In many respects, the demon acts as a simple thermometer because it is only one degree of freedom in comparison to the many degrees of freedom of the system with which it exchanges energy. We found in Problem 15.3 that the form of $P(E_d)$ is given by

$$P(E_d) = \frac{1}{Z} e^{-E_d/kT}, \tag{15.3}$$

where Z is a normalization constant such that the sum over all the states of the demon is unity. We also found that the parameter T in (15.3) is related to the kinetic temperature of an ideal gas.

In Problem 15.4 we will do some further simulations to determine the generality of the form (15.3).

Problem 15.4. The Boltzmann probability distribution

Modify your simulation of an ideal gas so that the kinetic energy of a particle is proportional to the absolute value of its momentum instead of the square of its momentum. Such a dependence would hold for a relativistic gas where the particles are moving at velocities close to the speed of light. Choose various values of the total energy E and number of particles N. Is the form of $P(E_d)$ the same as in (15.3)? How does the inverse slope of $\ln P(E_d)$ versus E_d compare to the mean energy per particle of the system in this case?

According to the equipartition theorem of statistical mechanics, each degree of freedom contributes $\frac{1}{2}kT$ to the energy per particle. Problem 15.4 shows that the equipartition theorem is not always valid. In fact, it is applicable only to a degree of freedom whose energy-momentum dependence is quadratic.

Although the microcanonical ensemble is conceptually simple, it does not represent the situation usually found in the laboratory. Most laboratory systems are not isolated, but are in thermal contact with their environment. This thermal contact allows energy to be exchanged between the laboratory system and its environment. The laboratory system is usually small relative to its environment. The larger system with many more degrees of freedom is commonly referred to as the *heat reservoir* or *heat bath*. The term heat refers to energy transferred from one body to another due to a difference in temperature. A heat bath is a system for which such energy transfer causes a negligible change in the temperature of the heat bath.

A system that is in equilibrium with a heat bath is characterized by the temperature of the latter. If we are interested in the equilibrium values of such a system, we need to know the probability P_s of finding the system in microstate s with energy E_s. The ensemble that describes the probability distribution of a system in thermal equilibrium with a heat bath is known as the *canonical* ensemble. In general, the canonical ensemble is characterized by the temperature T, the

number of particles N, and the volume V, in contrast to the microcanonical ensemble which is characterized by the energy E, N, and V,

We have already discussed an example of a system in equilibrium with a heat bath, the demon! In Problems 15.2–15.4, the system of interest was an ideal gas and the demon was an auxiliary (special) particle that facilitated the exchange of energy between the particles of the system. If we take the demon to be the system of interest, we see that the demon exchanges energy with a much bigger system which we can take to be the heat bath. Hence, the probability distribution of the microstates of the demon has the same general form as the probability distribution of a system in equilibrium with a heat bath. (Note that the microstate of the demon is characterized by its energy.) Hence, we can conclude that the probability that a system in equilibrium with a heat bath at temperature T is in microstate s with energy E_s has the form given in (15.3):

$$P_s = \frac{1}{Z} e^{-\beta E_s}, \qquad \text{(canonical distribution)} \tag{15.4}$$

where $\beta = 1/kT$ and Z is a normalization constant. Because $\sum P_s = 1$, Z is given by

$$Z = \sum_s e^{-E_s/kT}. \tag{15.5}$$

The summation in (15.5) is over the microstates of the system for a given N and V. The quantity Z is known as the *partition function* of the system. The ensemble defined by (15.4) is known as the *canonical* ensemble, and the probability distribution (15.4) is called the *Boltzmann* or the *canonical distribution*. The derivation of the Boltzmann distribution is given in textbooks on statistical mechanics. We will simulate systems in equilibrium with a heat bath in Section 15.6.

The form (15.4) of $P(E_d)$ provides a simple way of computing the temperature T from the mean demon energy $\langle E_d \rangle$. The latter is given by

$$\langle E_d \rangle = \frac{\int_0^\infty E\, e^{-E/kT}\, dE}{\int_0^\infty e^{-E/kT}\, dE} = kT. \tag{15.6}$$

We see that T is proportional to the mean demon energy. Note that the result $\langle E_d \rangle = kT$ in (15.6) holds only if the energy of the demon can take on a continuum of values and if the upper limit of integration can be taken to be ∞.

The demon is an excellent example of a thermometer. It has a property, namely, its energy, which is proportional to the temperature, and the demon is only one degree of freedom. Thus it can measure the temperature of the system and perturbs the system as little as possible. For example, the demon could be added to a molecular dynamics simulation and provide an independent measure of the temperature.

15.5 The Ising Model

A popular model of a system of interacting variables is the *Ising* model. The model was proposed by Lenz and investigated by Ising, his graduate student, to study the phase transition from a paramagnet to a ferromagnet (cf. Brush). Ising calculated the thermodynamic properties of the

Figure 15.1: The interaction energy between nearest neighbor spins in the absence of an external magnetic field.

model in one dimension and found that the model does not have a phase transition. However, for two and three dimensions the Ising model does exhibit a transition. The nature of the phase transition in two dimensions and the diverse applications of the Ising model are discussed in Section 15.7.

To introduce the Ising model, consider a lattice containing N sites and assume that each lattice site i has associated with it a number s_i, where $s_i = \pm 1$. The s_i are usually called spins, and the model can be used as a model of magnetic systems. The macroscopic properties of a system are determined by the nature of the accessible microstates. Hence, it is necessary to know the dependence of the energy on the configuration of spins. The total energy E of the Ising model is given by

$$E = -J \sum_{i,j=\text{nn}(i)}^{N} s_i s_j - H \sum_{i=1}^{N} s_i, \tag{15.7}$$

where H is proportional to a uniform external magnetic field. (We have absorbed a factor of μ into H.) The first sum in (15.7) represents the energy of interaction of the spins and is over all nearest neighbor pairs. The *exchange constant* J is a measure of the strength of the interaction between nearest neighbor spins (see Fig. 15.1). The second sum in (15.7) represents the energy of interaction between the magnetic moments of the spins and an external magnetic field.

If $J > 0$, then the states $\uparrow\uparrow$ and $\downarrow\downarrow$ are energetically favored in comparison to the states $\uparrow\downarrow$ and $\downarrow\uparrow$. Hence for $J > 0$, we expect that the state of lowest total energy is *ferromagnetic*, that is, the spins all point in the same direction. If $J < 0$, the states $\uparrow\downarrow$ and $\downarrow\uparrow$ are favored and the state of lowest energy is expected to be *antiferromagnetic*, that is, alternate spins are aligned. If we subject the spins to an external magnetic field directed upward, the spins $\uparrow$ and $\downarrow$ possess an additional energy given by $-H$ and $+H$ respectively.

An important virtue of the Ising model is its simplicity. Some of its simplifying features are that the kinetic energy of the atoms associated with the lattice sites has been neglected, only nearest neighbor contributions to the interaction energy are included, and the spins are allowed to have only two discrete values. In spite of the simplicity of the model, we will find that it exhibits very interesting behavior.

For the familiar case of classical particles with continuously varying position and velocity coordinates, the dynamics is given by Newton's laws. For the Ising model the dependence (15.7) of the energy on the spin configuration is not sufficient to determine the time-dependent properties of the system. That is, the relation (15.7) does not tell us how the system changes from one spin

configuration to another, and we have to introduce the dynamics separately. This dynamics will take the form of various Monte Carlo algorithms.

Because we are interested in the properties of an infinite system, we have to choose appropriate boundary conditions. The simplest boundary condition in one dimension is to choose a free surface so that the spins at sites 1 and N each have one nearest neighbor interaction only. Usually, a better choice is periodic boundary conditions. For this choice a one-dimensional lattice becomes a ring and the spins at sites 1 and N interact with one another and hence have the same number of interactions as do the other spins.

What are some of the physical quantities whose averages we wish to compute? An obvious physical quantity is the *magnetization* M given by

$$M = \sum_{i=1}^{N} s_i, \tag{15.8}$$

and the magnetization per spin $m = M/N$. Usually we are interested in the average values $\langle M \rangle$ and the fluctuations $\langle M^2 \rangle - \langle M \rangle^2$.

We will use the demon algorithm to sample configurations of the Ising model with the same energy. The implementation of the demon algorithm is straightforward. We first choose a spin at random. The trial change corresponds to a flip of the spin from $\uparrow$ to $\downarrow$ or $\downarrow$ to $\uparrow$. We then have to compute the change in energy of the system and decide whether to accept or reject the trial change. We can determine the temperature T as a function of the energy of the system in two ways. One way is to measure the probability that the demon has energy E_d. Because we know that this probability is proportional to $\exp(-E_d/kT)$, we can determine T from a plot of the logarithm of the probability as a function of E_d. An easier way to determine T is to measure the mean demon energy. However, because the values of E_d are not continuous for the Ising model, T is not simply proportional to $\langle E_d \rangle$ as it is for the ideal gas. We show in Appendix 15A that for $H = 0$ and the limit of an infinite system, the temperature is related to $\langle E_d \rangle$ by

$$kT/J = \frac{4}{\ln\left(1 + 4J/\langle E_d \rangle\right)}. \tag{15.9}$$

The result (15.9) comes from replacing the integrals in (15.6) by sums over the possible demon energies. Note that in the limit $|J/E_d| \ll 1$, (15.9) reduces to $kT = E_d$ as expected.

The `IsingDemon` class implements the simulation of the Ising model in one dimension using the demon algorithm and periodic boundary conditions. Once the initial configuration is chosen, the demon algorithm is similar to that described in Section 15.3. However, the spins in the one-dimensional Ising model must be chosen randomly.

Listing 15.3: Listing of the implementation of the demon algorithm for one-dimensional Ising model.

```
package org.opensourcephysics.sip.ch15;
import java.awt.Color;
import java.awt.Graphics;
import org.opensourcephysics.display.Drawable;
import org.opensourcephysics.display.DrawingPanel;
```

```
import org.opensourcephysics.display2d.CellLattice;

public class IsingDemon implements Drawable {
   public int [] spin;
   public int [] demonEnergyDistribution;
   int N;   // number of spins
   public int systemEnergy;
   public int demonEnergy = 0;
   public int mcs = 0;        // number of MC moves per particle
   public double systemEnergyAccumulator = 0;
   public double demonEnergyAccumulator = 0;
   public int magnetization = 0;
   public double mAccumulator = 0, m2Accumulator = 0;
   public int acceptedMoves = 0;
   private CellLattice lattice;

   public void initialize(int _N) {
      N = _N;
      lattice = new CellLattice(N,1);    // only used for drawing
      lattice.setIndexedColor(0, Color.red);
      lattice.setIndexedColor(2, Color.green);
      spin = new int[N];
      demonEnergyDistribution = new int[N];
      for (int i = 0; i < N; ++i) {
         spin[i] = 1;     // all spins up
      }
      int tries = 0;
      int E = -N;
      magnetization = N;
      while ( (E < systemEnergy) && (tries < 10*N)) {
         int k = (int) (N*Math.random());
         int dE = 2*spin[k]*( spin[(k + 1) % N] + spin[(k - 1 + N) % N]);
         if (dE > 0) {
            E += dE;
            spin[k] = -spin[k];
            magnetization += 2*spin[k];
         }
         tries++;
      }
      systemEnergy = E;
      resetData();
   }

   public double temperature() {
      return 4.0/Math.log(1.0 + 4.0/(demonEnergyAccumulator/(mcs*N)));
   }

   public void resetData() {
      mcs = 0;
      systemEnergyAccumulator = 0;
```

```
        demonEnergyAccumulator = 0;
        mAccumulator = 0;
        m2Accumulator = 0;
        acceptedMoves = 0;
    }

    public void doOneMCStep() {
        for (int j = 0; j < N; ++j) {
            int i = (int) (N*Math.random());
            int dE = 2*spin[i]*(spin[( i + 1) % N] + spin[(i - 1 + N) % N]);
            if (dE <= demonEnergy) {
                spin[i] = -spin[i];
                acceptedMoves++;
                systemEnergy += dE;
                demonEnergy -= dE;
                magnetization += 2*spin[i];
            }
            systemEnergyAccumulator += systemEnergy;
            demonEnergyAccumulator += demonEnergy;
            mAccumulator += magnetization;
            m2Accumulator += magnetization*magnetization;
            demonEnergyDistribution[demonEnergy]++;
        }
        mcs++;
    }

    public void draw(DrawingPanel panel, Graphics g) {
        if( lattice == null) return;
        for(int i = 0; i < N; i++)
        lattice.setValue(i,0,(byte)(spin[i]+1));
        lattice.draw(panel,g);
    }
}
```

Note that for $H = 0$, the change in energy due to a spin flip is either 0 or $\pm 4J$. Hence the initial energy of the system plus the demon must be an integer multiple of $4J$. Because the spins are interacting, it is difficult to choose an initial configuration of spins with precisely the desired energy. The procedure followed in method `initialize` is to begin with an initial configuration where all spins are up and then randomly flip spins while the energy is less than the desired initial energy.

Problem 15.5. The demon algorithm and the one-dimensional Ising model

a. Use `IsingDemon` and write a class `IsingDemonApp` to simulate the one-dimensional Ising model. Choose $N = 100$ and the desired total energy, $E = -20$. Describe qualitatively how the configuration changes with time. Then let $E = -100$ and describe any qualitative changes in the configurations.

b. Compute the time average of the demon energy and the magnetization M as a function of the

time. As usual, we interpret the time as the number of Monte Carlo steps per spin. What is the approximate time for these quantities to approach their equilibrium values?

c. Compute the equilibrium values of $\langle E_d \rangle$ and $\langle M^2 \rangle$. About 100 mcs is sufficient for testing the program and yields results of approximately 20% accuracy. To obtain better than 5% results, `mcs` should be the order of 1000.

d. Compute T for $N = 100$ and $E = -20$, -40, -60, and -80 from the inverse slope of $P(E_d)$ and the relation (15.9). Compare your results to the exact result for an infinite one-dimensional lattice, $E/N = -\tanh(J/kT)$. How do your computed results for E/N depend on N and on the number of Monte Carlo steps per spin?

e. Use the same runs as in part (d) to compute $\langle M^2 \rangle$ as a function of T. Does $\langle M^2 \rangle$ increase or decrease with T?

f. Modify your application and verify the Boltzmann form (15.3) for the energy distribution of the demon.

Problem 15.6. Additional applications

a. Modify `IsingDemon` so that the antiferromagnetic case, $J = -1$, is treated. Before doing the simulation, describe how you expect the spin configurations to differ from the ferromagnetic case. What is the lowest energy or ground state configuration? Run the simulation with the spins initially in their ground state, and compare your results with your expectations. Compute the mean energy per spin versus temperature and compare your results with the ferromagnetic case.

b.* Modify `IsingDemon` to include a nonzero magnetic field, $H \neq 0$, and compute $\langle E_d \rangle$, $\langle M \rangle$, and $\langle M^2 \rangle$ as a function of H for fixed E. Read the discussion in Appendix 15A and determine the relation of $\langle E_d \rangle$ to T for your choices of H. Or determine T from the inverse slope of $P(E_d)$. Is the equilibrium temperature higher or lower than the $H = 0$ case for the same total energy?

***Problem 15.7.** The demon algorithm and the two-dimensional Ising model

a. Simulate the two-dimensional Ising model on a square lattice using the demon algorithm. The total number of spins $N = L^2$, where L is the length of one side of the lattice. Use periodic boundary conditions as shown in Fig. 15.2 so that spins in the left-hand column interact with spins in the right-hand column, etc. Do not include nonequilibrium configurations in your averages.

b. Compute $\langle E_d \rangle$ and $\langle M^2 \rangle$ as a function of E. Convenient choices of parameters are $L = 20$ and run for at least 500 mcs. Assume $J = 1$ and $H = 0$. Use (15.9) to determine the dependence of T on E and plot E versus T.

c. Repeat the simulations in part (b) for $L = 20$. Run until your averages are accurate to within a few percent. Describe how the energy versus temperature curve changes with lattice size.

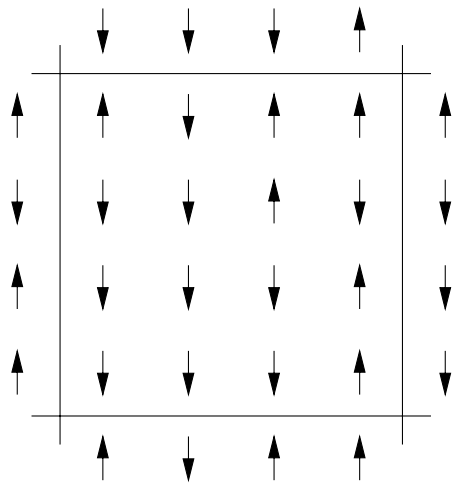

Figure 15.2: One of the 2^N possible configurations of a system of $N = 16$ Ising spins on a square lattice. Also shown are the spins in the four nearest periodic images of the central cell that are used to calculate the energy. An up spin is denoted by $\uparrow$ and a down spin is denoted by $\downarrow$. Note that the number of nearest neighbors on a square lattice is four. The energy of this configuration is $E = -8J + 4H$ with periodic boundary conditions.

d. Modify your program to make "snapshots" of the spin configurations. Describe qualitatively the nature of the configurations at different energies or temperatures. Are they ordered or disordered? Are there domains of up or down spins?

e. Instead of choosing a spin at random to make a trial change, choose the spins sequentially, that is, choose all the x values in ascending order for $y = 0$, then all the x values for $y = 1$, etc. Note, however, that this procedure updates a site and then immediately uses the new spin value when updating the neighbor. Because this process introduces a directional bias, vary the direction of the updates after each sweep. Do you obtain the same results as part (b)?

One advantage of the demon algorithm is that it makes fewer demands on the random number generator than the Metropolis algorithm which we will discuss in Section 15.6. The demon algorithm also does not require computationally expensive calculations of the exponential function. Thus, for some systems the demon algorithm can be much faster than the Metropolis algorithm. In the one-dimensional Ising model we must choose the trial spins at random, but in higher dimensions, the spins can be chosen sequentially (see Problem 15.7e). In this case we can do a Monte Carlo simulation without random numbers! Very fast algorithms have been developed by using one computer bit per spin and multiple demons (see Appendix 15B).

There also are several disadvantages associated with the microcanonical ensemble. One disadvantage is the difficulty of establishing a system at the desired value of the energy. However, the most important disadvantage for us is conceptual. That is, it is more natural to think of the behavior of macroscopic physical quantities as functions of the temperature rather than the total energy.

15.6 The Metropolis Algorithm

As we have mentioned, most physical systems of interest are not isolated, but exchange energy with their environment. If a small system is placed in thermal contact with a heat bath at temperature T, the system reaches thermal equilibrium by exchanging energy with the heat bath until the system reaches the same temperature. Imagine an infinitely large number of copies of a system at fixed volume V and number of particles N in equilibrium at temperature T. Then the probability P_s that the system is in microstate s with energy E_s is given by (15.4)

We can use (15.4) to obtain the ensemble average of the physical quantities of interest. For example, the mean energy is given by

$$\langle E \rangle = \sum_s E_s \, P_s = \frac{1}{Z} \sum_s E_s \, e^{-\beta E_s}. \tag{15.10}$$

Note that the energy fluctuates in the canonical ensemble.

How can we simulate a system of N particles confined in a volume V at a fixed temperature T? Because we can generate only a finite number m of the total number of M microstates, we might expect to obtain an estimate for the mean value of a physical quantity A by writing

$$\langle A \rangle \approx A_m = \frac{\sum\limits_{s=1}^{m} A_s \, e^{-\beta E_s}}{\sum\limits_{s=1}^{m} e^{-\beta E_s}}. \tag{15.11}$$

where A_s is the value of the physical quantity A in microstate s. A crude Monte Carlo procedure is to generate a microstate s at random, calculate E_s, A_s, and $e^{-\beta E_s}$, and evaluate the corresponding contribution of the microstate to the sums in (15.11). However, a microstate generated in this way would likely be very improbable and hence contribute little to the sums. Instead, we use an *importance sampling* method and generate microstates according to a probability distribution function π_s.

We rewrite (15.11) by multiplying and dividing by π_s:

$$A_m = \frac{\sum\limits_{s=1}^{m} (A_s/\pi_s) \, e^{-\beta E_s} \, \pi_s}{\sum\limits_{s=1}^{m} (1/\pi_s) \, e^{-\beta E_s} \, \pi_s}. \qquad \text{(no importance sampling)} \tag{15.12}$$

If we generate microstates with probability π_s, then (15.12) becomes

$$A_m = \frac{\sum\limits_{s=1}^{m} (A_s/\pi_s) \, e^{-\beta E_s}}{\sum\limits_{s=1}^{m} (1/\pi_s) \, e^{-\beta E_s}}. \qquad \text{(importance sampling)} \tag{15.13}$$

That is, if we average over a biased sample, we need to weight each microstate by $1/\pi_s$ to eliminate the bias. Although any form of π_s could be used, the form of (15.13) suggests that a reasonable

choice of π_s is the Boltzmann probability itself, that is,

$$\pi_s = \frac{e^{-\beta E_s}}{\sum\limits_{s=1}^{m} e^{-\beta E_s}}. \tag{15.14}$$

This choice of π_s implies that the estimate A_m of the mean value of A can be written as

$$A_m = \frac{1}{m} \sum_{s=1}^{m} A_s, \tag{15.15}$$

where each state is sampled according to the Boltzmann distribution. The choice (15.14) for π_s is due to Metropolis et al.

Although we discussed the Metropolis sampling method in Section 11.9 in the context of the numerical evaluation of integrals, it is not necessary to read that section to understand the Metropolis algorithm in the present context. The Metropolis algorithm can be summarized in the context of the simulation of a system of spins as follows. The extension to other types of systems is straightforward.

1. Establish an initial microstate.
2. Make a random trial change in the microstate.The easiest change is to choose a spin at random and flip it.
3. Compute $\Delta E \equiv E_{\text{trial}} - E_{\text{old}}$, the change in the energy of the system due to the trial change.
4. If ΔE is less than or equal to zero, accept the new microstate and go to step 8.
5. If ΔE is positive, compute the quantity $w = e^{-\beta \Delta E}$.
6. Generate a random number r in the unit interval $[0, 1]$.
7. If $r \leq w$, accept the new microstate; otherwise retain the previous microstate.
8. Determine the value of the desired physical quantities.
9. Repeat steps (2) through (8) to obtain a sufficient number of microstates.
10. Periodically compute averages over the microstates.

Steps (2) to (7) lead to a transition probability that the system moves from microstate $\{s_i\}$ to $\{s_j\}$ proportional to

$$W(i \to j) = \min\left(1, e^{-\beta \Delta E}\right), \qquad \text{(Metropolis algorithm)} \tag{15.16}$$

where $\Delta E = E_j - E_i$. Because it is necessary to evaluate only the ratio $P_j/P_i = e^{-\beta \Delta E}$, it is not necessary to normalize the probability. Note that because the microstates are generated with a probability proportional to the desired probability, all averages become arithmetic averages as in (15.15). However, because the constant of proportionally is not known, it is not possible to estimate the partition function Z in this way .

Although we choose π_s to be the Boltzmann distribution, other choices of π_s are possible and are useful in some contexts. In addition, the choice (15.16) of the transition probability is not the only one that leads to the Boltzmann distribution. It can be shown that if W satisfies the *detailed balance* condition

$$W(i \to j)\, e^{-\beta E_i} = W(j \to i)\, e^{-\beta E_j}, \qquad \text{(detailed balance)} \tag{15.17}$$

then the corresponding Monte Carlo algorithm generates a sequence of states distributed according to the Boltzmann distribution. The proof that the Metropolis algorithm generates states with a probability proportional to the Boltzmann probability distribution after a sufficient number of steps does not add much to our physical understanding of the algorithm. Instead, in Problems 15.8 and 15.9 we apply the algorithm to the ideal classical gas and to a classical magnet in a magnetic field, respectively, and verify that the Metropolis algorithm yields the Boltzmann distribution after a sufficient number of trial changes have been made.

Note that we have implicitly assumed in our discussion of the demon and Metropolis algorithms that the system is ergodic. That is, we have assumed that the important microstates of the system are being sampled with the desired probability. The existence of ergodicity depends on the way the trial moves are made, and on the nature of the energy barriers between microstates. For example, consider a one-dimensional lattice of Ising spins with all spins up. If the spins are updated sequentially from right to left, then if one spin is flipped, all remaining flips would be accepted regardless of the temperature because the change in energy is zero. Clearly, the system is not ergodic for this implementation of the algorithm, and we would not obtain the correct thermodynamic behavior. A measure of the ergodicity of a system was discussed in Project 8.22.

We first consider the application of the Metropolis algorithm to an ideal classical gas in one dimension and verify that the Metropolis algorithm does indeed sample states according to the Boltzmann algorithm. The energy of an ideal gas depends only on the velocity of the particles, and hence a microstate is completely described by a specification of the velocity (or momentum) of each particle. Because the velocity is a continuous variable, it is necessary to describe the accessible microstates so that they are countable, and hence we place the velocity into bins. Suppose we have $N = 10$ particles and divide the possible values of the velocity into twenty bins. Then the total number of microstates would be 20^{10}. Not only would it be difficult to label these 20^{10} states, it would take a prohibitively long time to obtain an accurate estimate of their relative probabilities, and it would be difficult to verify directly that the Metropolis algorithm yields the Boltzmann distribution. For this reason we consider a single classical particle in one dimension in equilibrium with a heat bath and adopt the less ambitious goal of verifying that the Metropolis algorithm generates the Boltzmann distribution for this system.

The Metropolis algorithm is implemented in method `doStep` in class `BoltzmannApp`, and the velocity distribution is plotted. One quantity of interest is the probability $P(v)\,dv$ that the system has a velocity between v and $v + dv$. To test for the Boltzmann distribution, we need to plot $\ln P(E_s)$ versus $E_s = mv_s^2/2$ and determine if we obtain a straight line with a negative slope equal to $-1/kT$. The subscript s indicates that E_s is the energy of a specific microstate with velocity v_s. The `HistogramFrame` class is used to record $P(v_s)$. The extra code in `stopAnimation` is used to extract the data from the histogram for $P(v_s)$ and save it so that $\ln P(E_s)$ can be plotted. As usual, we choose units such that Boltzmann's constant and the mass of the particle are unity. We also choose the temperature to be large enough such that $dv = 1$ provides a sufficiently small bin size to compute $P(v_s)$ accurately.

Listing 15.4: Code for Boltzmann algorithm for single particle.

```
/*
 * The org.opensourcephysics.sip.ch20 package contains classes for chapter 20
 * of the book Simulations in Physics.
 * Copyright (c) 2002 H. Gould, J. Tobochnik, and W. Christian.
 */
package org.opensourcephysics.sip.ch15;
import org.opensourcephysics.controls.AbstractSimulation;
import org.opensourcephysics.controls.SimulationControl;
import org.opensourcephysics.frames.HistogramFrame;

/**
 * Tests Metropolis algorithm for one particle
 * @author Jan Tobochnik revised 12/20/04
 */

public class BoltzmannApp extends AbstractSimulation {
    double beta;
    int mcs;
    int accepted;
    double velocity;
    HistogramFrame velocityDistribution = new HistogramFrame("v", "P(v)", "Velocity Distribution");


/**
    * Get's parameters and initializes model
    */
    public void initialize () {
        velocityDistribution .clearData();
        beta = 1.0/control.getDouble("Temperature");
        velocity = control.getDouble("Initial velocity");
        accepted = 0;
        mcs = 0;
    }

/**
    * Does one Monte Carlo step and plots distributions
    */
    public void doStep() {
        int stepsBetweenPlots = control.getInt("Steps between plots");
        double delta = control.getDouble("Maximum velocity change");
        for(int i = 0; i < stepsBetweenPlots; i++){
            mcs++;
            double ke = 0.5*velocity*velocity;
            double vTrial = velocity + delta*(2.0*Math.random() - 1.0);
            double keTrial = 0.5*vTrial*vTrial;
            double dE = keTrial - ke;
            if ( ( dE < 0) || (Math.exp(-beta*dE) > Math.random()) ) {
                accepted++;
```

```
                ke = keTrial;
                velocity = vTrial;
            }
            velocityDistribution.append(velocity);
        }
        control.clearMessages();
        control.println("mcs = " + mcs + " acceptance ratio = " + (double)(accepted)/mcs);
    }


  /**
    * Resets to default values
    */
    public void reset() {
        control.setValue("Maximum velocity change", 10.0);
        control.setValue("Temperature", 10.0);
        control.setValue(" Initial velocity" , 0.0);
        control.setValue("Steps between plots", 100);
    }

    /**
    * Starts the Java application.
    * @param args command line parameters
    */

    public static void main (String[] args) {
        SimulationControl.createApp(new BoltzmannApp());
    }
}
```

Problem 15.8. Simulation of a particle in equilibrium with a heat bath

a. Choose the temperature $T = 10$, the initial velocity equal to zero, and the maximum change in the particle's velocity $\delta = 10.0$. Increase the number of Monte Carlo steps until a plot of $\ln P(v)$ versus $E = mv^2/2$ yields a straight line with a slope equal to $-\beta = -1/T$. Describe the qualitative form of $P(v)$. Note that because there are two values of v, one positive and one negative, for each value of E.

b. Add code to `BoltzmannApp` to compute the mean energy and velocity. How do your results for the mean energy and the mean velocity compare with the corresponding exact values? Explain why the computed mean particle velocity is approximately zero even though the initial particle velocity was not zero.

c. To insure that your results do not depend on the initial conditions, let the initial velocity equal zero and compute the mean energy and velocity. How do your results compare with those found in parts (a) and (b)?

d. Add another `HistogramFrame` object to compute the probability $P(E)dE$ where E is the energy of the configuration. Does $P(E)$ have the form of a Boltzmann distribution? If not what is the functional form of $P(E)$? Why is $P(E)$ not the same as $P(E_s)$ computed in step (c)?

e. The *acceptance probability* is the fraction of trial moves that are accepted. What is the effect of changing the value of δ on the acceptance probability?

Problem 15.9. Planar spin in an external magnetic field

a. Consider a classical planar magnet with magnetic moment μ_0. The magnet can be oriented in any direction in the x-y plane, and the energy of interaction of the magnet with an external magnetic field $\mathbf{B}$ is $-\mu_0 B \cos\phi$, where ϕ is the angle between the moment and $\mathbf{B}$. Write a Monte Carlo program to sample the microstates of this system in thermal equilibrium with a heat bath at temperature T. Compute the mean energy as a function of the ratio $\beta\mu_0 B$.

b. Compute the probability density $P(\phi)$ and analyze its dependence on the energy.

In Problem 15.10 we consider the Monte Carlo simulation of a classical ideal gas of N particles in equilibrium with a heat bath. It is convenient to say that one time unit or one Monte Carlo step per particle (mcs) has elapsed after N particles have had one chance each on the average to change their coordinates. That is, if the particles are chosen at random, then during one Monte Carlo step per particle, some particles might not be chosen, but all particles will be chosen equally on the average. The advantage of this definition is that the time is independent of the number of particles. However, this definition of time has no obvious relation to a physical time.

Problem 15.10. Simulation of an ideal gas in one dimension

a. Modify class `BoltzmannApp` to simulate an ideal gas of N particles in one dimension. For simplicity, assume that all particles have the same initial velocity of 10. Let $N = 20$ and $T = 10$ and consider at least 2000 Monte Carlo steps per particle. Choose the value of δ so that the acceptance probability is approximately 40%. What is the mean kinetic energy and mean velocity of the particles?

b. We might expect the total energy of an ideal gas to remain constant because the particles do not interact with one another and hence cannot exchange energy directly. What is the value of the initial total energy of the system in part (a)? Does the total energy remain constant? If not, explain how the energy changes.

c. What is the nature of the time dependence of the total energy starting from the initial condition in (a)? Estimate the number of Monte Carlo steps per particle necessary for the system to reach thermal equilibrium by computing a moving average of the total energy over a fixed time interval. Does this average change with time after a sufficient time has elapsed? What choice of the initial velocities allows the system to reach thermal equilibrium at temperature T as quickly as possible?

d. Compute the probability $P(E)dE$ for the system of N particles to have a total energy between E and $E + dE$. Plot $P(E)$ as a function of E and describe the qualitative behavior of $P(E)$. Does $P(E)$ have the form of the Boltzmann distribution? If not, describe the qualitative features of $P(E)$ and determine its functional form.

e. Compute the mean energy for $T = 10$, 20, 40, 80, and 120 and estimate the heat capacity from its definition $C = \partial E/\partial T$.

f. Compute the mean square energy fluctuations $\langle(\Delta E)^2\rangle = \langle E^2\rangle - \langle E\rangle^2$ for $T = 10$ and $T = 40$. Compare the magnitude of the ratio $\langle(\Delta E)^2\rangle/T^2$ with the heat capacity determined in part (e).

You might have been surprised to find in Problem 15.10d that the form of $P(E)$ is a Gaussian centered about the mean energy of the system. What is the relation of this form of $P(E)$ to the central limit theorem (see Problem 7.15)? That is, the distribution function of a *macroscopic* quantity such as the total energy is sharply peaked about its mean value. If the microstates are distributed according to the Boltzmann probability, why is the total energy distributed according to the Gaussian distribution?

15.7 Simulation of the Ising Model

You are probably familiar with ferromagnetic materials, such as iron and nickel, which exhibit a spontaneous magnetization in the absence of an applied magnetic field. This nonzero magnetization occurs only if the temperature is lower than a well defined temperature known as the Curie or critical temperature T_c. For temperatures $T > T_c$, the magnetization vanishes. Hence T_c separates the disordered phase for $T > T_c$ from the ferromagnetic phase for $T < T_c$.

The origin of magnetism is quantum mechanical in nature and an area of much experimental and theoretical interest. However, the study of simple classical models of magnetism has provided much insight. The two- and three-dimensional Ising model is the most commonly studied classical model and is particularly useful in the neighborhood of the magnetic phase transition. As discussed earlier, the energy of the Ising model is given by (15.7).

The thermal quantities of interest for the Ising model include the mean energy $\langle E\rangle$ and the heat capacity C. One way to determine C at constant external magnetic field is from its definition $C = \partial\langle E\rangle/\partial T$. An alternative way is to relate C to the statistical fluctuations of the total energy in the canonical ensemble (see Appendix 15C):

$$C = \frac{1}{kT^2}\big(\langle E^2\rangle - \langle E\rangle^2\big). \qquad \text{(canonical ensemble)} \tag{15.18}$$

Another quantity of interest is the mean magnetization $\langle M\rangle$ and the corresponding zero field magnetic susceptibility:

$$\chi = \frac{\partial\langle M\rangle}{\partial H}\Big|_{H=0}. \tag{15.19}$$

In the following, we will refer to H as the magnetic field, even though it includes a factor of μ. The zero field magnetic susceptibility χ is an example of a linear response function, because it measures the ability of a spin to respond to a change in the external magnetic field. In analogy to the heat capacity, χ is related to the fluctuations of the magnetization (see Appendix 15C):

$$\chi = \frac{1}{kT}\big(\langle M^2\rangle - \langle M\rangle^2\big), \tag{15.20}$$

where $\langle M\rangle$ and $\langle M^2\rangle$ are evaluated in zero external magnetic field. The relations (15.18) and (15.20) are examples of the general relation between linear response functions and equilibrium fluctuations.

Now that we have specified several equilibrium quantities of interest, we implement the Metropolis algorithm for the Ising model. The simplest possible trial change is the flip of a spin. The Metropolis algorithm was stated in Section 15.6 as a method for generating states with the desired Boltzmann probability, but the flipping of single spins also can be interpreted as a reasonable approximation to the real dynamics of an anisotropic magnet whose spins are coupled to the vibrations of the lattice. The coupling leads to random spin flips, and we expect that one Monte Carlo step per spin is proportional to the average time between single spin flips observed in the laboratory. Hence, we can regard single spin flip dynamics as a time dependent process and observe the relaxation to equilibrium after a sufficiently long time. In the following, we will frequently refer to the application of the Metropolis algorithm to the Ising model as *single spin flip dynamics*.

In Problem 15.11 we use the Metropolis algorithm to simulate the one-dimensional Ising model. Note that the parameters J and kT do not appear separately, but appear together in the dimensionless ratio J/kT. Unless otherwise stated, we measure temperature in units of J/k, and set $H = 0$.

Problem 15.11. One-dimensional Ising model

a. Write a Monte Carlo program to simulate the one-dimensional Ising model in equilibrium with a heat bath. Modify method `doOneMCStep` in `IsingDemon` (see Section 15.5 or see class `Ising`, listed on page 658 for an example of the implementation of the Metropolis algorithm to the two-dimensional Ising model.) Use periodic boundary conditions. For simplicity, assume that the external magnetic field is zero. Draw the microscopic state (configuration) of the system after each Monte Carlo step per spin.

b. First choose $N = 20$ and $T = 1$, and start with all spins up, that is, $s_i = +1$. What is the initial effective "temperature" of the system? Run for at least 1000 mcs. Visually inspect the configuration of the system after each Monte Carlo step and estimate the time it takes for the system to reach equilibrium. Does the sign of the magnetization change during the simulation? Increase N and estimate the time for the system to reach equilibrium and for the magnetization to change sign.

c. Change the initial condition so that the orientation of each spin is chosen at random. What is the initial effective "temperature" of the system in this case? Estimate the time it takes for the system to reach equilibrium in the same way as in part (b).

d. Choose $N = 50$ and determine $\langle E\rangle$, $\langle E^2\rangle$, and $\langle M^2\rangle$ as a function of T in the range $0.1 \leq T \leq 5$. Plot $\langle E\rangle$ as a function of T and discuss its qualitative features. Compare your computed results for $\langle E\rangle$ to the exact result (for $H = 0$):

$$E(T) = -N \tanh \beta J. \tag{15.21}$$

Use the relation (15.18) to determine the T dependence of C.

e. As you probably noticed in part (b), the system can overturn completely during a long run and thus the value of $\langle M\rangle$ can vary widely from run to run. Because $\langle M\rangle = 0$ for $T > 0$ for the one-dimensional Ising model, it is better to assume $\langle M\rangle = 0$ and compute χ from the relation $\chi = \langle M^2\rangle/kT$. Use this relation (15.20) to estimate the T dependence of χ.

f. One of the best laboratory realizations of a one-dimensional Ising ferromagnet is a chain of bichloride-bridged Fe^{2+} ions known as FeTAC (see Greeney et al.). Measurements of χ yield a value of the exchange interaction J given by $J/k = 17.4\,\mathrm{K}$. Note that experimental values of J are typically given in temperature units. Use this value of J to plot your Monte Carlo results for χ versus T with T given in Kelvin. At what temperature is χ a maximum for FeTAC?

g. Is the acceptance probability an increasing or decreasing function of T? Does the Metropolis algorithm become more or less efficient as the temperature is lowered?

h. Compute the probability $P(E)$ for a system of $N = 50$ spins at $T = 1$. Run for at least 1000 `mcs`. Plot $\ln P(E)$ versus $(E - \langle E \rangle)^2$ and discuss its qualitative features.

We next apply the Metropolis algorithm to the two-dimensional Ising model on the square lattice. The `Ising` class is listed in the following.

Listing 15.5: Listing for two-dimensional Ising model.

```
package org.opensourcephysics.sip.ch15;
import java.awt.Color;
import java.awt.Graphics;
import org.opensourcephysics.display.Drawable;
import org.opensourcephysics.display.DrawingPanel;
import org.opensourcephysics.display2d.CellLattice;

public class Ising implements Drawable {
   public static final double criticalTemperature = 2.0/Math.log(1.0 + Math.sqrt(2.0));
   public int [][] spin;
   public int L = 32;
   public int N = L*L;
   public double temperature = criticalTemperature;
   public int mcs = 0; // number of MC moves per spin
   public int energy;
   public double energyAccumulator = 0;
   public double energySquaredAccumulator = 0;
   public int magnetization = 0;
   public double magnetizationAccumulator = 0;
   public double magnetizationSquaredAccumulator = 0;
   public int acceptedMoves = 0;
   private double [] w = new double[9]; // array to hold Boltzmann factors
   private CellLattice lattice;

   public void initialize(int _L) {
      L=_L;
      lattice = new CellLattice(L,L);          // only used for drawing
      lattice.setIndexedColor(0, Color.red);
      lattice.setIndexedColor(2, Color.green);
      spin = new int[L][L];
      N = L*L;
      for (int i = 0; i < L; ++i) {
         for (int j = 0; j < L; ++j) {
```

```
            spin[i][j] = 1;   // all spins up
        }
    }
    magnetization = N;
    energy = −2∗N;
    resetData();
    w[8] = Math.exp(−8.0/temperature); // other array elements never occur for H = 0
    w[4] = Math.exp(−4.0/temperature);
}

public double specificHeat() {
    double energySquaredAverage = energySquaredAccumulator/mcs;
    double energyAverage = energyAccumulator/mcs;
    double heatCapacity = energySquaredAverage − energyAverage∗energyAverage;
    heatCapacity = heatCapacity/(temperature∗temperature);
    return (heatCapacity/N);
}

public double susceptibility() {
    double magnetizationSquaredAverage = magnetizationSquaredAccumulator/mcs;
    double magnetizationAverage = magnetizationAccumulator/mcs;
    return (magnetizationSquaredAverage − Math.pow(magnetizationAverage,2))/(temperature∗N);
}

public void resetData() {
    mcs = 0;
    energyAccumulator = 0;
    energySquaredAccumulator = 0;
    magnetizationAccumulator = 0;
    magnetizationSquaredAccumulator = 0;
    acceptedMoves = 0;
}

public void doOneMCStep() {
    for (int k = 0; k < N; ++k) {
        int i = (int)(Math.random()∗L);
        int j = (int)(Math.random()∗L);
        int dE = 2∗spin[i][j]∗(spin[(i+1)%L][j] + spin[(i−1+L)%L][j] + spin[i][(j+1)%L] + spin[i][(j−1+L)%L]);
        if ((dE <= 0) || (w[dE] > Math.random())) {
            spin[i][j] = −spin[i][j];
            acceptedMoves++;
            energy += dE;
            magnetization += 2∗spin[i][j];
        }
    }
    energyAccumulator += energy;
    energySquaredAccumulator += energy∗energy;
    magnetizationAccumulator += magnetization;
    magnetizationSquaredAccumulator += magnetization∗magnetization;
    mcs++;
```

```
    }

    public void draw (DrawingPanel panel, Graphics g) {
        if( lattice ==null) return;
        for(int i = 0; i < L; i++)
            for(int j = 0; j < L; j++)
                lattice .setValue(i,j,(byte)(spin[i][j]+1));
        lattice .draw(panel,g);
    }
}
```

In method `initialize` we choose the initial directions of the spins and compute the initial values of the energy and magnetization. To compute the total energy, we consider the interaction of a spin with its nearest neighbor spins to the north and the east. In this way we compute the energy of each interaction only once and avoid double counting.

One of the most time consuming parts of the Metropolis algorithm is the calculation of the exponential function $e^{-\beta\Delta E}$. Because there are only a small number of possible values of $\beta\Delta E$ for the Ising model (see Fig. 15.11), we store the small number of different probabilities for the spin flips in the array `w`. The values of this array are computed in method `initialize`.

To implement the Metropolis algorithm, we determine the change in the energy ΔE and then accept the trial flip if $\Delta E \leq 0$. If this condition is not satisfied, we then generate a random number in the unit interval and compare it to $e^{-\beta\Delta E}$. We can use a single if statement for these two conditions, because in Java, the second condition of an `||` (or) statement is evaluated only if the first is false. This feature is very useful because we don't want to waste cpu time by evaluating random numbers when we don't need to if $\Delta E \leq 0$. The same idea is used for an `&` (and) statement for which the second condition is only evaluated if the first is true.

A typical laboratory system has at least 10^{18} spins. In contrast, the number of spins that can be simulated typically ranges from 10^3 to 10^9. As we have discussed in other contexts, the use of periodic boundary conditions minimizes finite size effects. However, periodic boundary conditions reduce the maximum separation between spins to one half the length of the system, and more sophisticated boundary conditions are sometimes convenient. For example, we can give the surface spins extra neighbors, whose direction is related to the mean magnetization of the microstate. We adopt the simpler periodic boundary conditions.

In class `Ising` data for the values of the physical observables are accumulated after each Monte Carlo step per spin. The optimum time for sampling various physical quantities is explored in Problem 15.13. Note that if a flip is rejected and the old configuration is retained, thermal equilibrium is not described properly unless the old configuration is included again in computing the averages.

Achieving thermal equilibrium can account for a substantial fraction of the total run time for very large systems. The most practical choice of initial conditions in these cases is a configuration from a previous run that is at a temperature close to the desired temperature. The code to do this can be found in the appendix of Chapter 8.

Problem 15.12. Equilibration of the two-dimensional Ising model

a. Write a target class that uses `Ising` and plots the magnetization and energy as a function of the

number of Monte Carlo steps. It should also print out the acceptance ratio, mean magnetization, mean energy, specific heat, and susceptibility when the simulation is stopped. Averages such as the mean energy and the mean magnetization should be normalized by the number of spins so that it is easy to compare systems with different values of N. Use a linear dimension for the lattice of $L = 32$ and the heat bath temperature $T = 2$. Determine the time needed to equilibrate the system, if the directions of the spins are initially up. Estimate how many Monte Carlo steps per spin are necessary for the system to reach equilibrium.

b. Visually determine if the spin configurations are "ordered" or "disordered" at $T = 2$ after equilibrium has been established?

c. Repeat part (a) with the initial direction of each spin being random. Does the equilibration time increase or decrease?

d. Repeat parts (a)–(c) for $T = 2.5$.

Problem 15.13. Comparison with exact results

In general, a Monte Carlo simulation yields exact answers only after an infinite number of configurations have been sampled. How then can we be sure that our program works correctly, and our results are statistically meaningful? One way is to ensure that our program can reproduce exact results in known limits. In the following, we test class `Ising` by considering a small system for which the mean energy and magnetization can be calculated analytically.

a. Calculate analytically the T dependence of E, M, C and χ for the two-dimensional Ising model with $L = 2$. (A summary of the calculation is given in Appendix 15D.) (Note that for simplicity, we have omitted the brackets denoting the thermal averages.)

b. Use `Ising` with $L = 2$ and estimate E, M, C, and χ for $T = 0.5$ and 0.25. Use the relations (15.18) to compute C. Compare your estimated values to the exact results found in part (a). Approximately how many Monte Carlo steps per spin are necessary to obtain E and M to within 1%? How many Monte Carlo steps per spin are necessary to obtain C to within 1%?

c. The calculation of χ is more complicated because the sign of M can change during the simulation. Compare your results for χ from using (15.20) and from using (15.20) with $\langle M \rangle$ replaced by $\langle |M| \rangle$.

Now that we have checked our program and obtained typical equilibrium configurations, we consider the calculation of the mean values of the physical quantities of interest. Suppose we wish to compute the mean value of the physical quantity A. In some cases, the calculation of A for a given configuration is time consuming, and we do not want to compute its value more often than necessary. For example, we would not compute A after the flip of only one spin, because the values of A in the two configurations would almost be the same. Ideally, we wish to compute A for configurations that are statistically independent. Because we do not know *a priori* the mean number of spin flips needed to obtain configurations that are statistically independent, it is a good idea to estimate this time in our preliminary calculations.

One way to estimate the time interval over which configurations are correlated is to compute the time displaced *autocorrelation* function $C_A(t)$ which is defined as

$$C_A(t) = \frac{\langle A(t+t_0)A(t_0)\rangle - \langle A\rangle^2}{\langle A^2\rangle - \langle A\rangle^2}, \tag{15.22}$$

where $A(t)$ is the value of the quantity A at time t. The averages in (15.22) are over all possible time origins t_0 for an equilibrium system. Because the choice of the time origin is arbitrary for an equilibrium system, C_A depends only on the time difference t rather than t and t_0 separately. For sufficiently large t, $A(t)$ and $A(0)$ will become uncorrelated, and hence $\langle A(t+t_0)A(t_0)\rangle \to \langle A(t+t_0)\rangle\langle A(t_0)\rangle = \langle A\rangle^2$. Hence $C_A(t) \to 0$ as $t \to \infty$. Note that $C_A(t=0)$ is normalized to unity. In general, $C_A(t)$ will decay exponentially with t with a decay or correlation time τ_A whose magnitude depends on the choice of the physical quantity A as well as the physical parameters of the system, for example, the temperature.

The time dependence of the two most common correlation functions, $C_M(t)$ and $C_E(t)$ is investigated in Problem 15.14. As an example of the calculation of $C_E(t)$, consider the equilibrium time series for E for the $L = 4$ Ising model at $T = 4$: -4, -8, 0, -8, -20, -4, 0, 0, -24, -32, -24, -24, -8, -8, -16, -12. The averages of E and E^2 over these sixteen values are $\langle E\rangle = -12$, $\langle E^2\rangle = 240$, and $\langle E^2\rangle - \langle E\rangle^2 = 96$. We wish to compute $E(t)E(0)$ for all possible choices of the time origin. For example, $E(t=4)E(0)$ is given by

$$\begin{aligned}\langle E(t=4)E(0)\rangle = \frac{1}{12}\big[&(-20\times -4) + (-4\times -8) + (0\times 0)\\ &+ (0\times -8) + (-24\times -20) + (-32\times -4)\\ &+ (-24\times 0) + (-24\times 0) + (-8\times -24)\\ &+ (-8\times -32) + (-16\times -24) + (-12\times -24)\big].\end{aligned} \tag{15.23}$$

We averaged over the twelve possible choices of the origin for the time difference $t = 4$. Verify that $\langle E(t=4)E(0)\rangle = 460/3$ and $C_E(t=4) = 7/72$.

In the above calculation of $\langle E(t)E(0)\rangle$, we included all possible combinations of $E(t)E(0)$ for a given time series. To implement this procedure on a computer, we could store the time series in memory or write it in a data file. An alternative procedure is to save the last `nSave` values of the time series in memory and to average over fewer combinations. The class, `AutoCorrelatorApp`, in Listing 15.6 reads in data from a file produced by saving $M(t)$ and $E(t)$ from `IsingApp`. The method `computeCorrelation` first computes the mean and mean squared of the magnetization and energy needed to compute (15.22). Then it computes the time displaced autocorrelation for all possible choices of t_0 using (15.22).

Listing 15.6: Listing of class for computing autocorrelation function of M and E.

```
package org.opensourcephysics.sip.ch15;
import java.io.BufferedReader;
import java.io.File;
import java.io.FileReader;
import java.util.StringTokenizer;

import javax.swing.JFileChooser;
```

```
import org.opensourcephysics.controls.AbstractCalculation;
import org.opensourcephysics.controls.CalculationControl;
import org.opensourcephysics.frames.PlotFrame;

/**
 * AutoCorrelatorApp computes energy and magnetization
 * time autocorrelation function from file input data
 * @author Jan Tobochnik
 * @version 1.0  JT revised 12/20/04
 */
public class AutoCorrelatorApp extends AbstractCalculation {
    PlotFrame plotFrame = new PlotFrame("tau", "<E(t+tau)E(t)> and <M(t+tau)M(t)>", "Time correlations");
    double [] energy, magnetization;
    int numberOfPoints;


   /**
    * Creates new arrays, reads data in and computes correlation.
    */
    public void calculate() {
        energy = new double[1000];
        magnetization = new double[1000];
        numberOfPoints = 0;
        readData();
        computeCorrelation(control.getInt("Maximum time interval, tau"));
    }

/**
 * Reads data.
 */
    public void readData() {
        JFileChooser chooser = new JFileChooser();
        int returnVal = chooser.showOpenDialog(plotFrame);
        if(returnVal == JFileChooser.APPROVE_OPTION) {
            File f = chooser.getSelectedFile();
            try {
                BufferedReader in = new BufferedReader(new FileReader(f));
                String s = in.readLine();  // read in first line which contains no data
                s = in.readLine();  // read in second line which contains no data
                s = in.readLine();  // read in third line which should contain magnetization data
                while(!s.equals("Dataset 1")) {   // Dataset 1 starts energy data
                    // Create a StringTokenizer with a tab as a delimiter, valid for data from PlottingPanel
                    StringTokenizer st = new StringTokenizer(s, "\t" );
                    // covert string token to a Double and then a double
                    magnetization[(int)Double.valueOf(st.nextToken()).doubleValue()] = Double.valueOf(st.nextToken()).doubleVa
                    s = in.readLine();
                    numberOfPoints++;
                }
                s = in.readLine(); // read first energy data point
```

```
            while ( s != null ) {
                StringTokenizer st = new StringTokenizer(s, "\t" );
                energy[(int)Double.valueOf(st.nextToken()).doubleValue()] = Double.valueOf(st.nextToken()).doubleValue();
                s = in.readLine();
            }
        } //end try
        catch (Exception e) {
            System.out.println ("Exception: " + e);
        }
    }  // end if
  }

/**
 * Computes and plots correlation functions
 * @param tauMax is the maximum time for correlation functions
 */
  public void computeCorrelation(int tauMax) {
    plotFrame.clearData();
    double energyAccumulator = 0, magnetizationAccumulator = 0;
    double energySquaredAccumulator = 0, magnetizationSquaredAccumulator = 0;
    for(int t = 0; t < numberOfPoints; t++) {
        energyAccumulator += energy[t];
        magnetizationAccumulator += magnetization[t];
        energySquaredAccumulator += energy[t]*energy[t];
        magnetizationSquaredAccumulator += magnetization[t]*magnetization[t];
    }
    double averageEnergySquared = Math.pow(energyAccumulator/numberOfPoints, 2);
    double averageMagnetizationSquared = Math.pow(magnetizationAccumulator/numberOfPoints,2);
    // compute normalization factors
    double normE = (energySquaredAccumulator/numberOfPoints) - averageEnergySquared;
    double normM = (magnetizationSquaredAccumulator/numberOfPoints) - averageMagnetizationSquared;
    for(int tau = 1; tau <= tauMax; tau++) {
        double c_MAccumulator = 0;
        double c_EAccumulator = 0;
        int counter = 0;
        for(int t = 0; t < numberOfPoints - tau; t++) {
            c_MAccumulator += magnetization[t]*magnetization[t+tau];
            c_EAccumulator += energy[t]*energy[t+tau];
            counter++;
        }
        // correlation defined so that c(0) = 1 and c( infinity ) -> 0
        plotFrame.append(0,tau,((c_MAccumulator/counter) - averageMagnetizationSquared)/normM);
        plotFrame.append(1,tau,((c_EAccumulator/counter) - averageEnergySquared)/normE);
    }
    plotFrame.setVisible(true);
    control.println("Number of points = " + numberOfPoints);
  }

/**
 * Resets input parameter
```

```
  */
    public void resetCalculation () {
        control.setValue("Maximum time interval, tau", 20);
    }

   /**
    * Starts Java application.
    * @param args command line parameters
    */
    public static void main(String args[]) {
        CalculationControl.createApp(new AutoCorrelatorApp());
    }
}
```

Problem 15.14. Correlation times

a. Choose $L = 4$ and $T = 3$ and equilibrate the system. Then look at the time series of M and E after every Monte Carlo step per spin and estimate how often M changes sign. Does E change sign when M changes sign? How often does M change sign for $L = 8$ (and $T = 3$)? In equilibrium, positive and negative values of M are equally likely in the absence of an external magnetic field. Is your time series consistent with this equilibrium property? Why is it more meaningful to compute the time displaced correlation function of the absolute value of the magnetization rather than the magnetization itself if L is small?

b. Choose $L = 16$ and $T = 1$ and equilibrate the system. Then look at the time series of M. Do you find that positive and negative values of M are equally likely? Explain your results.

c. Use class `AutoCorrelatorApp` to calculate the equilibrium averaged values of $C_M(t)$ and $C_E(t)$. As a check on your program, use the time series for E given in the text to do a hand calculation of $C_E(t)$ in the way that it is computed in the `computeCorrelation` method.

d. Estimate the correlation times from the energy and the magnetization correlation functions for $L = 8$, and $T = 3$, $T = 2.3$, and $T = 2$. Save the last `nsave` $= 100$ values of the magnetization and energy only after the system is equilibrated. Are the correlation times τ_M and τ_E comparable? One way to determine τ is to fit $C(t)$ to an exponential form $C(t) \sim e^{-t/\tau}$. Another way is to define the integrated correlation time as

$$\tau = \sum_{t=1} C(t). \tag{15.24}$$

The sum is cut off at the first negative value of $C(t)$. Are the negative values of $C(t)$ physically meaningful? How does the behavior of $C(t)$ change if you average your results over longer runs? How do your estimates for the correlation times compare with your estimates of the relaxation time found in Problem 15.12? Why would the term "decorrelation time" be more appropriate than "correlation time?"

e. To describe the relaxation toward equilibrium as realistically as possible, we have randomly selected the spins to be flipped. However, if we are interested only in equilibrium properties,

it might be possible to save computer time by selecting the spins sequentially. Determine if the correlation time is greater, smaller, or approximately the same if the spins are chosen sequentially rather than randomly. If the correlation time is greater, does it still save CPU time to choose spins sequentially? Why is it not desirable to choose spins sequentially in the one-dimensional Ising model?

How can we quantify the accuracy of our measurements, for example, the accuracy of the mean energy $\langle E \rangle$? As discussed in Chapter 11, the usual measure of the accuracy is the standard deviation of the mean. If we make n independent measurements of E, then the most probable error in $\langle E \rangle$ is given by

$$\sigma_m = \frac{\sigma}{\sqrt{n-1}}, \tag{15.25}$$

where the standard deviation σ is defined as

$$\sigma^2 = \langle E^2 \rangle - \langle E \rangle^2. \tag{15.26}$$

The difficulty is that, in general, our measurements of the time series E_i are not independent, but are correlated. Hence, σ_m as given by (15.25) is an underestimate of the actual error.

*__Problem 15.15.__ Estimate of errors

How can we determine whether the measurements are independent without computing the correlation time? One way is based on the idea that the magnitude of the error should not depend on how we group the data (see Section 11.5). For example, suppose that we group every two data points to form $n/2$ new data points $E_i^{(2)}$ given by $E_i^{(g=2)} = (1/2)[E_{2i-1} + E_{2i}]$. If we replace n by $n/2$ and E by $E^{(2)}$ in (15.25) and (15.26), we would find the same value of σ_m as before, provided that the original E_i are independent. If the computed σ_m is not the same, we continue this averaging process until σ_m calculated from

$$E_i^{(g)} = \frac{1}{2}[E_{2i-1}^{(g/2)} + E_{2i}^{(g/2)}] \qquad (g = 2, 4, 8, \dots) \tag{15.27}$$

is approximately the same as that calculated from $E^{(g/2)}$.

a. Use the above averaging method to estimate the errors in your measurements of $\langle E \rangle$ and $\langle M \rangle$ for the two-dimensional Ising model. Let $L = 8$, $T = 2.269$, and `mcs` ≥ 16384, and calculate averages after every Monte Carlo step per spin after the system has equilibrated. If necessary, increase the number of Monte Carlo steps for averaging. A rough measure of the correlation time is the number of terms in the time series that need to be averaged for σ_m to be approximately unchanged. What is the qualitative dependence of the correlation time on $T - T_c$?

b. Repeat for $L = 16$. Do you need more Monte Carlo steps than in part (a) to obtain statistically independent data? If so, why?

c. The exact value of E/N for the two-dimensional Ising model on a square lattice with $L = 16$ and $T = T_c = 2/\ln(1+\sqrt{2}) \approx 2.269$ is given by $E/N = -1.45306$ (to five decimal places). (The significance of T_c will be explored in Section 15.8.) The exact result for E/N allows us to determine the actual error in this case. Compute $\langle E \rangle$ by averaging E after each Monte Carlo step per spin for `mcs` $\geq 10^6$. Compare your actual error to the estimated error given by (15.25) and (15.26) and discuss their relative values.

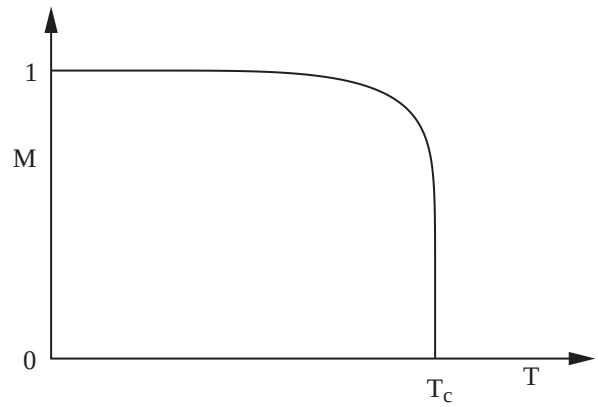

Figure 15.3: The temperature dependence of $m(T)$, the mean magnetization per spin, for the infinite lattice Ising model in two dimensions.

15.8 The Ising Phase Transition

Now that we have tested our program for the two-dimensional Ising model, we are ready to explore its properties.

Problem 15.16. Qualitative behavior of the two-dimensional Ising model

a. Use the Metropolis algorithm and class `Ising` and `IsingApp` to compute the magnetization, the mean energy, the heat capacity, and the susceptibility. Because we will consider the Ising model for different values of L, it will be convenient to convert these quantities to intensive quantities such as the mean energy per spin, the specific heat (per spin), and the susceptibility per spin. For simplicity, we will use the same notation for both the extensive and the corresponding intensive quantities. Choose $L = 4$ and consider T in the range $1.5 \leq T \leq 3.5$ in steps of $\Delta T = 0.2$. Choose the initial condition at $T = 3.5$ so that the orientation of the spins is chosen at random. Because all the spins might overturn and the magnetization change sign during the course of your observation, estimate the mean value of $|M|$ in addition to that of M. The susceptibility should be calculated as

$$\chi = \frac{1}{kT}[\langle M^2\rangle - \langle |M|\rangle^2]. \tag{15.28}$$

Use at least 1000 Monte Carlo steps per spin and estimate the number of equilibrium configurations needed to obtain $\langle M\rangle$ and $\langle E\rangle$ to 5% accuracy. Plot $\langle E\rangle$, m, $|m|$, C, and χ as a function of T and describe their qualitative behavior. Do you see any evidence of a phase transition?

b. Repeat the calculations of part (a) for $L = 8$ and $L = 16$. Plot $\langle E\rangle$, m, $|m|$, C, and χ as a function of T and describe their qualitative behavior. Is the evidence of a phase transition more clear?

c. For a given value of L, for example, $L = 16$, choose a value of T that is well below T_c and choose the directions of the spins at random. Observe the spins evolve in time. Do you see several domains with positive and negative spontaneous magnetization? How does the magnetization evolve with time?

d. The correlation length ξ can be obtained from the r-dependence of the spin correlation function $c(r)$. The latter is defined as:

$$c(r) = \langle s_i s_j \rangle - m^2, \tag{15.29}$$

where r is the distance between sites i and j. We have assumed the system is translationally invariant so that $\langle s_i \rangle = \langle s_j \rangle = m$. The average is over all sites for a given configuration and over many configurations. Because the spins are not correlated for large r, we see that $c(r) \to 0$ in this limit. It is reasonable to assume that $c(r) \sim e^{-r/\xi}$ for r sufficiently large. Use this behavior to estimate ξ as a function of T. How does your estimate of ξ compare with the size of the regions of spins with the same orientation?

One of the limitations of a computer simulation study of a phase transition is the relatively small size of the systems we can study. Nevertheless, we observed in Problem 15.16 that even systems as small as $L = 4$ exhibit behavior that is reminiscent of a phase transition. In Fig. 15.4 we show our Monte Carlo data for the T dependence of the specific heat of the two-dimensional Ising model for $L = 8$ and $L = 16$. We see that C exhibits a broad maximum which becomes sharper for larger L. Does your data for C exhibit similar behavior?

We next summarize some of the qualitative properties of infinite ferromagnetic systems in zero magnetic field. We know that at $T = 0$, the spins are perfectly aligned in either direction, that is, the mean magnetization per spin $m(T) = \langle M(T) \rangle / N$ is given by $m(T = 0) = \pm 1$. As T is increased, the magnitude of $m(T)$ decreases continuously until $T = T_c$ at which $m(T)$ vanishes (see Fig. 15.3). Because $m(T)$ vanishes continuously rather than abruptly, the transition is termed *continuous* rather than discontinuous. (The term *first-order* describes a discontinuous transition.)

How can we characterize a continuous magnetic phase transition? Because a nonzero m implies that a net number of spins are spontaneously aligned, we designate m as the *order parameter* of the system. Near T_c, we can characterize the behavior of many physical quantities by power law behavior just as we characterized the percolation threshold (see Table 12.1). For example, we can write m near T_c as

$$m(T) \sim (T_c - T)^{\beta}, \tag{15.30}$$

where β is a *critical* exponent (not to be confused with the inverse temperature). Various thermodynamic derivatives such as the susceptibility and heat capacity diverge at T_c and are characterized by critical exponents. We write

$$\chi \sim |T - T_c|^{-\gamma}, \tag{15.31}$$

and

$$C \sim |T - T_c|^{-\alpha}, \tag{15.32}$$

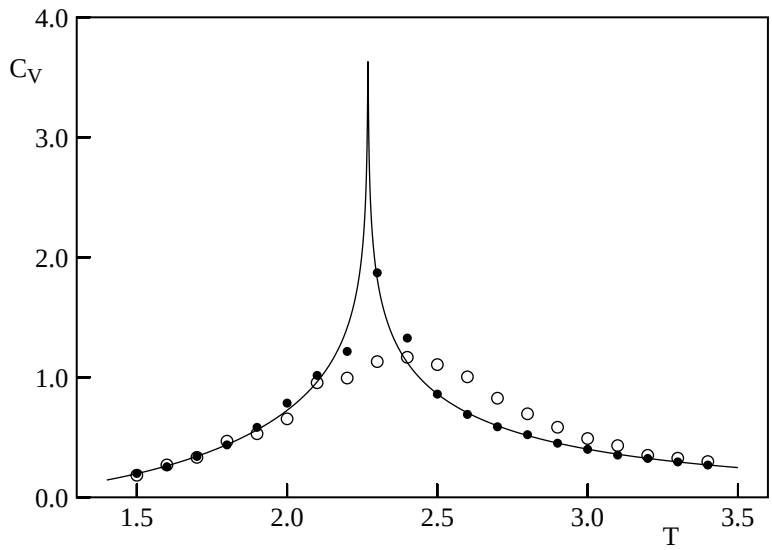

Figure 15.4: The temperature dependence of the specific heat C (per spin) of the Ising model on a $L = 8$ and $L = 16$ square lattice with periodic boundary conditions. One thousand Monte Carlo steps per spin were used for each value of the temperature. The continuous line represents the temperature dependence of C in the limit of an infinite lattice. (Note that C is infinite at $T = T_c$ for an infinite lattice.)

where we have introduced the critical exponents γ and α. We have assumed that χ and C are characterized by the same critical exponents above and below T_c.

Another measure of the magnetic fluctuations is the linear dimension $\xi(T)$ of a typical magnetic domain. We expect the *correlation length* $\xi(T)$ to be the order of a lattice spacing for $T \gg T_c$. Because the alignment of the spins becomes more correlated as T approaches T_c from above, $\xi(T)$ increases as T approaches T_c. We can characterize the divergent behavior of $\xi(T)$ near T_c by the critical exponent ν:

$$\xi(T) \sim |T - T_c|^{-\nu}. \tag{15.33}$$

The calculation of ξ is considered in Problem 15.16d.

As we found in our discussion of percolation in Chapter 12, a finite system cannot exhibit a true phase transition. Nevertheless, we expect that if $\xi(T)$ is less than the linear dimension L of the system, our simulations will yield results comparable to an infinite system. Of course, if T is close to T_c, our simulations will be limited by finite size effects. In the following problem, we obtain preliminary results for the T dependence of m, $\langle E \rangle$, C, and χ in the neighborhood of T_c. These results will help us understand the qualitative nature of the ferromagnetic phase transition in the two-dimensional Ising model.

Because we can simulate only finite lattices, it is difficult to obtain estimates for the critical exponents α, β, and γ by using the definitions (15.30)–(15.32) directly. We learned in Section 12.5

that we can do a *finite size scaling analysis* to extrapolate finite L results to $L \to \infty$. For example, from Fig. 15.4 we see that the temperature at which C exhibits a maximum becomes better defined for larger lattices. This behavior provides a simple definition of the transition temperature $T_c(L)$ for a finite system. According to finite size scaling theory, $T_c(L)$ scales as

$$T_c(L) - T_c(L = \infty) \sim aL^{-1/\nu}, \tag{15.34}$$

where a is a constant and ν is defined in (15.33). The finite size of the lattice is important when the correlation length

$$\xi(T) \sim L \sim |T - T_c|^{-\nu}. \tag{15.35}$$

As in Section 12.5, we can set $T = T_c$ and consider the L-dependence of M, C, and χ:

$$m(T) \sim (T_c - T)^{\beta} \to L^{-\beta/\nu} \tag{15.36}$$
$$C(T) \sim |T - T_c|^{-\alpha} \to L^{\alpha/\nu} \tag{15.37}$$
$$\chi(T) \sim |T - T_c|^{-\gamma} \to L^{\gamma/\nu}. \tag{15.38}$$

In Problem 15.17 we use the relations (15.36)–(15.38) to estimate the critical exponents β, γ, and α.

Problem 15.17. Finite size scaling for the two-dimensional Ising model

a. Use the relation (15.34) together with the exact result $\nu = 1$ to estimate the value of T_c for an infinite square lattice. Because it is difficult to obtain a precise value for T_c with small lattices, we will use the exact result $kT_c/J = 2/\ln(1+\sqrt{2}) \approx 2.269$ for the infinite lattice in the remaining parts of this problem.

b. Determine the specific heat C, $|m|$, and the susceptibility χ at $T = T_c$ for $L = 2$, 4, 8, 16, and 32. Compute χ using (15.20) with $\langle |M| \rangle$ instead of $\langle M \rangle$. Use as many Monte Carlo steps per spin as possible. Plot the logarithm of $|m|$ and χ versus L and use the scaling relations (15.36)–(15.38) to determine the critical exponents β and γ. Assume the exact result $\nu = 1$. Do your log-log plots of $|m|$ and χ yield reasonably straight lines? Compare your estimates for β and γ with the exact values given in Table 12.1.

c. Make a log-log plot of C versus L. If your data for C is sufficiently accurate, you will find that the log-log plot of C versus L is not a straight line but shows curvature. The reason for this curvature is that the exponent α in (15.32) equals zero for the two-dimensional Ising model, and hence (15.37) needs to be interpreted as

$$C \sim C_0 \ln L. \tag{15.39}$$

Is your data for C consistent with (15.39)? The constant C_0 in (15.39) is approximately 0.4995.

So far we have performed our Ising model simulations on a square lattice. How do the critical temperature and the critical exponents depend on the symmetry and the dimension of the lattice? Based on your experience with the percolation transition in Chapter 12, you might have a good idea what the answer is.

Problem 15.18. The effects of symmetry and dimension on the critical properties of the Ising model

a. The nature of the triangular lattice is discussed in Chapter 8 (see Fig. 8.5). The main difference between the triangular lattice and the square lattice is the number of nearest neighbors. Make the necessary modifications in your Ising program, for example, determine the energy changes due to a flip of a single spin and the corresponding values of the transition probabilities. Compute C and χ for different values of T in the interval $[2, 5]$. Assume that $\nu = 1$ and use finite size scaling to estimate T_c in the limit of an infinite triangular lattice. Compare your estimate of T_c to the known value $kT_c/J = 3.641$ (to three decimal places). The simulation of Ising models on the triangular lattice is relevant to the understanding of the experimentally observed phases of materials that can be absorbed on the surface of graphite.

b. No exact results are available for the Ising model in three dimensions. Write a Monte Carlo program to simulate the Ising model on the simple cubic lattice (each site has six nearest neighbors). Compute C and χ for T in the range $3.2 \leq T \leq 5$ in steps of 0.2 for different values of L. Estimate $T_c(L)$ from the maximum of C and χ. How do these estimates of $T_c(L)$ compare? Use the values of $T_c(L)$ that exhibit a stronger L dependence and plot $T_c(L)$ versus $L^{-1/\nu}$ for different values of ν in the range 0.5 to 1 (see (15.34)). Show that the extrapolated value of $T_c(L = \infty)$ does not depend sensitively on the value of ν. Compare your estimate for $T_c(L = \infty)$ to the known value $kT_c/J = 4.5108$ (to four decimal places).

c. Compute $|m|$, C, and χ at $T = T_c \approx 4.5108$ for different values of L on a simple cubic lattice. Do a finite size scaling analysis to estimate β/ν, α/ν, and γ/ν. The best known values of the critical exponents for the three-dimensional Ising model are given in Table 12.1. For comparison, published Monte Carlo results in 1976 for the finite size behavior of the Ising model on the simple cubic Ising lattice are for $L = 6$ to $L = 20$; 2000–5000 Monte Carlo steps per spin were used for calculating the averages after equilibrium had been reached. Can you obtain more accurate results?

Problem 15.19. Critical slowing down

a. Consider the two-dimensional Ising model on a square lattice with $L = 16$. Compute $C_M(t)$ and $C_E(t)$ and determine the correlation times τ_M and τ_E for $T = 2.5, 2.4$, and 2.3. Determine the correlation times as discussed in Problem 15.14d. How do these correlation times compare with one another? Show that τ increases as the critical temperature is approached, a physical effect known as *critical slowing down*.

b. We can define the dynamical critical exponent z by the relation

$$\tau \sim \xi^z. \tag{15.40}$$

On a finite lattice we have the relation $\tau \sim L^z$ at $T = T_c$. Compute τ for different values of L at $T = T_c$ and make a very rough estimate of z. (The value of z for the two-dimensional Ising model with spin flip dynamics is not known definitely, but $z \approx 2.167$.)

The values of τ and z found in Problem 15.19 depend in large part on our choice of dynamics (algorithm). The reason for the large value of z for the Metropolis algorithm is the divergence of

the correlation length and the existence of large domains of parallel spins near the critical point. It is difficult for the Metropolis algorithm to flip a domain because it has to do so one spin at a time and each trial flip has a high probability of being rejected. What is the probability of flipping a spin in the middle of a domain at $T = T_c$? Which spins in a domain are more likely to flip? What is the dominant mechanism for decorrelating a domain of spins?

Although we have generated a trial change by the attempted flip of one spin, it is possible that other types of trial changes, for example, the simultaneous flip of more than one spin, would be more efficient and lead to smaller correlation times and smaller values of z. A problem of much current interest is the development of more efficient algorithms near phase transitions (see Project 15.34).

15.9 Other Applications of the Ising Model

Because the applications of the Ising model are so wide ranging, we can mention only a few of the applications here. In the following, we briefly describe applications of the Ising model to first-order phase transitions, lattice gases, antiferromagnetism, and the order-disorder transition in binary alloys.

So far we have discussed the continuous phase transition in the Ising model and have found that the energy and magnetization vary continuously with the temperature, and thermodynamic derivatives such as the specific heat and the susceptibility diverge near T_c (in the limit of an infinite lattice). In Problem 15.20 we discuss a simple example of a *first-order* phase transition. Such transitions are accompanied by *discontinuous* (finite) changes in thermodynamic quantities such as the energy and the magnetization.

Problem 15.20. The two-dimensional Ising model in an external magnetic field

a. Modify your two-dimensional Ising program so that the energy of interaction with an external magnetic field H is included. It is convenient to measure H in terms of the dimensionless ratio $h = \beta H$. (Remember that H has already absorbed a factor of μ.) Compute m, the mean magnetization per spin, as a function of h for $T < T_c$. Consider a square lattice with $L = 32$ and equilibrate the system at $T = 1.8$ and $h = 0$. Adopt the following procedure to obtain $m(h)$.

   (i) Use an equilibrium configuration at $h = 0$ as the initial configuration for $h_1 = \Delta h = 0.2$.

   (ii) Run the system for 100 Monte Carlo steps per spin before computing averages.

   (iii) Average m over 100 Monte Carlo steps per spin.

   (iv) Use the last configuration for h_n as the initial configuration for $h_{n+1} = h_n + \Delta h$.

   (v) Repeat steps (ii)–(iv) until $m \approx 0.95$. Plot m versus h. Do the measured values of m correspond to equilibrium averages?

b. Decrease h by $\Delta h = -0.2$ in the same way as in part (a) until h passes through zero and until $m \approx -0.95$. Extend your plot of m versus h to negative h values. Does m remain positive for small negative h? Do the measured values of m for negative h correspond to equilibrium averages? Draw the spin configurations for several values of h. Do you see evidence of domains?

c. Increase h by $\Delta h = 0.2$ until the m versus h curve forms an approximately closed loop. What is the value of m at $h = 0$? This value of m is the spontaneous magnetization.

d. A first-order phase transition is characterized by a discontinuity (for an infinite lattice) in the order parameter. In the present case the transition is characterized by the behavior of m as a function of h. What is your measured value of m for $h = 0.2$? If $m(h)$ is double valued, which value of m corresponds to the equilibrium state, an absolute minima in the free energy? Which value of m corresponds to a *metastable* state, a relative minima in the free energy? What are the equilibrium and metastable values of m for $h = -0.2$? The transition from positive m to negative m is first-order because there is a discontinuous jump in the magnetization. First-order transitions exhibit *hysteresis* and the properties of the system depend on the history of the system, for example, whether h is an increasing or decreasing function. Because of the long lifetime of metastable states near a phase transition, a system in such a state can mistakenly be interpreted as being in equilibrium. We also know that near a continuous phase transition, the relaxation to equilibrium becomes very long (see Problem 15.19), and hence a system with a continuous phase transition can behave as if it were effectively in a metastable state. For these reasons it is very difficult to distinguish the nature of a phase transition using computer simulations. This problem is discussed further in Section 15.11.

e. Repeat the above simulation for $T = 3$, a temperature above T_c. Why do your results differ from the simulations in parts (a)–(c) done for $T < T_c$?

The Ising model also describes systems that might appear to have little in common with ferromagnetism. For example, we can interpret the Ising model as a lattice gas, where down represents a lattice site occupied by a molecule and represents an empty site. Each lattice site can be occupied by at most one molecule, and the molecules interact with their nearest neighbors. The lattice gas is a crude model of the behavior of a real gas of molecules and is a simple model of the gas-liquid transition and the critical point. What properties does the lattice gas have in common with a real gas? What properties of real gases does the lattice gas neglect?

If we wish to simulate a lattice gas, we have to decide whether to do the simulation at fixed density or at fixed chemical potential μ. The implementation of the latter is straightforward because the grand canonical ensemble for a lattice gas is equivalent to the canonical ensemble for Ising spins in an external magnetic field, that is, the effect of the magnetic field is to fix the mean number of up spins. Hence, we can simulate a lattice gas in the grand canonical ensemble by doing spin flip dynamics. (The volume of the lattice is an irrelevant parameter.)

Another application of a lattice gas model is to the study of phase separation in a binary or A-B alloy. In this case spin up and spin down correspond to a site occupied by an A atom and B atom, respectively. As an example, the alloy β-brass has a low temperature ordered phase in which the two components (copper and zinc) have equal concentrations and form a cesium chloride structure. As the temperature is increased, some zinc atoms exchange positions with copper atoms, but the system is still ordered. However, above the critical temperature $T_c = 742\,\mathrm{K}$, the zinc and copper atoms become mixed and the system is disordered. This transition is an example of an *order-disorder* transition.

Because the number of A atoms and the number of B atoms is fixed, we cannot use spin flip dynamics to simulate a binary alloy. A dynamics that does conserve the number of down and up spins is known as *spin exchange dynamics*. In this dynamics a trial *interchange* of two nearest

neighbor spins is made and the change in energy ΔE is calculated. The criterion for the acceptance or rejection of the trial change is the same as before.

Problem 15.21. Simulation of a lattice gas

a. Modify your Ising program so that spin exchange dynamics rather than spin flip dynamics is implemented. Determine the possible values of ΔE on the square lattice, determine the possible values of the transition probability, and change the way a trial change is made. If we are interested only in the mean value of quantities such as the total energy, we can reduce the computation time by not interchanging like spins. For example, we can keep a list of bonds between occupied and empty sites and make trial moves by choosing bonds at random from this list. For small lattices such a list is unnecessary and a trial move can be generated by simply choosing a spin and one of its nearest neighbors at random.

b. Consider a square lattice with $L = 32$ and 512 sites initially occupied. (The number of occupied sites is a conserved variable and must be specified initially.) Determine the mean energy for T in the range $1 \leq T \leq 4$. Plot the mean energy as a function of T. Does the energy appear to vary continuously?

c. Repeat the calculations of part (b) with 612 sites initially occupied, and plot the mean energy as a function of T. Does the energy vary continuously? Do you see any evidence of a first-order phase transition?

d. Because the spins correspond to particles, we can compute their single particle diffusion coefficient. Use an array to record the position of each particle as a function of time. After equilibrium has been reached, compute $\langle R(t)^2 \rangle$, the mean square displacement per particle. Is it necessary to "interchange" two like spins? If the particles undergo a random walk, the self-diffusion constant D is defined as

$$D = \lim_{t \to \infty} \frac{1}{2dt} \langle R(t)^2 . \rangle. \tag{15.41}$$

Estimate D for different temperatures and numbers of occupied sites. Note that if a particle starts at x_0 and returns to x_0 by on the average moving in one direction using the periodic boundary conditions, the net displacement in the x direction is L not 0. Thus, when computing $R(t)$ you should accumulate the changes in position at each step.

Although you are probably familiar with ferromagnetism, for example, a magnet on a refrigerator door, nature provides more examples of antiferromagnetism. In the language of the Ising model, antiferromagnetism means that the interaction parameter J is negative and nearest neighbor spins prefer to be aligned in opposite directions. As we will see in Problem 15.22, the properties of the antiferromagnetic Ising model on a square lattice are similar to the ferromagnetic Ising model. For example, the energy and specific heat of the ferromagnetic and antiferromagnetic Ising models are identical at all temperatures in zero magnetic field, and the system exhibits a phase transition at the Néel temperature T_N. On the other hand, the total magnetization and susceptibility of the antiferromagnetic model do not exhibit critical behavior near T_N. Instead, we can define two sublattices for the square lattice corresponding to the red and black squares of a checkerboard and introduce the staggered magnetization M_s equal to the difference of the

magnetization on the two sublattices. We will find in Problem 15.22 that the temperature dependence of M_s and the staggered susceptibility χ_s are identical to the analogous quantities in the ferromagnetic Ising model.

Problem 15.22. Antiferromagnetic Ising model

a. Modify the `Ising` class to simulate the antiferromagnetic Ising model on the square lattice in zero magnetic field. Because J does not appear explicitly in class `Ising`, change the sign of the energy calculations in the appropriate places in the program. To compute the staggered magnetization on a square lattice, define one sublattice to be the sites (x, y) for which the product $\texttt{mod}(x, 2) \times \texttt{mod}(y, 2) = 1$; the other sublattice corresponds to the remaining sites.

b. Choose $L = 32$ and the initial condition to be all spins up. What configuration of spins corresponds to the state of lowest energy? Compute the temperature dependence of the mean energy, specific heat, magnetization, and the susceptibility χ. Does the temperature dependence of any of these quantities show evidence of a phase transition?

c. Compute the temperature dependence of M_s and the staggered susceptibility χ_s defined as (see (15.20))

$$\chi_s = \frac{1}{kT}\big[\langle {M_s}^2\rangle - \langle M_s\rangle^2\big]. \tag{15.42}$$

Verify that the temperature dependence of M_s for the antiferromagnetic Ising model is the same as the temperature dependence of M for the Ising ferromagnet. Could you have predicted this similarity without doing the simulation?

d. In part (b) you might have noticed that χ shows a cusp. Compute χ for different values of L at $T = T_N \approx 2.269$. Do a finite size scaling analysis and verify that χ does not diverge at $T = T_N$.

e. Consider the behavior of the antiferromagnetic Ising model on a triangular lattice. Choose $L \geq 32$ and compute the same quantities as before. Do you see any evidence of a phase transition? Draw several configurations of the system at different temperatures. Do you see evidence of many small domains at low temperatures? Is there a unique ground state? If you cannot find a unique ground state, you share the same frustration as do the individual spins in the antiferromagnetic Ising model on the triangular lattice. We say that this model exhibits *frustration* because there is no spin configuration on the triangular lattice such that all spins are able to minimize their energy (see Fig. 15.5).

The Ising model is one of many models of magnetism. The Heisenberg, Potts, and x-y models are other examples of models of magnetic materials familiar to condensed matter scientists as well as to workers in other areas. Monte Carlo simulations of these models and others have been important in the development of our understanding of phase transitions in both magnetic and nonmagnetic materials. Some of these models are discussed in Section 15.14.

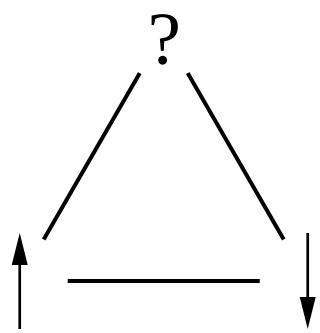

Figure 15.5: An example of frustration on a triangular lattice.

15.10 Simulation of Classical Fluids

The existence of matter as a solid, liquid and gas is well known (see Fig. 15.6). Our goal in this section is to use Monte Carlo methods to gain additional insight into the qualitative differences between these three phases.

Monte Carlo simulations of classical systems are simplified considerably by the fact that the velocity (momentum) variables are decoupled from the position variables. The total energy can be written as $E = K(\{\mathbf{v}_i\}) + U(\{\mathbf{r}_i\})$, where the kinetic energy K is a function of only the particle velocities $\{\mathbf{v}_i\}$, and the potential energy U is a function of only the particle positions $\{\mathbf{r}_i\}$. This separation implies we need to sample only the positions of the molecules, that is, the "configurational" degrees of freedom. Is such a simplification possible for quantum systems? Because the velocity appears quadratically in the kinetic energy, it can be shown using classical statistical mechanics that the contribution of the velocity coordinates to the mean energy is $\frac{1}{2}kT$ per degree of freedom.

The physically relevant quantities of a fluid include its mean energy, specific heat, and equation of state. Another interesting quantity is the *radial distribution function* $g(r)$ which we introduced in Chapter 8. We will find in Problems 15.23–15.25 that $g(r)$ is a probe of the density fluctuations and hence a probe of the local order in the system. If only two-body forces are present, the mean potential energy per particle can be expressed as (see (8.16))

$$\frac{U}{N} = \frac{\rho}{2}\int g(r)V(r)\,d\mathbf{r}, \tag{15.43}$$

and the (virial) equation of state can be written as (see (8.17))

$$\frac{\beta P}{\rho} = 1 - \frac{\beta\rho}{2d}\int g(r)\,r\frac{dV(r)}{dr}\,d\mathbf{r}. \tag{15.44}$$

Hard core interactions. To separate the effects of the short range repulsive interaction from the longer range attractive interaction, we first investigate a model of *hard disks* with the interparticle

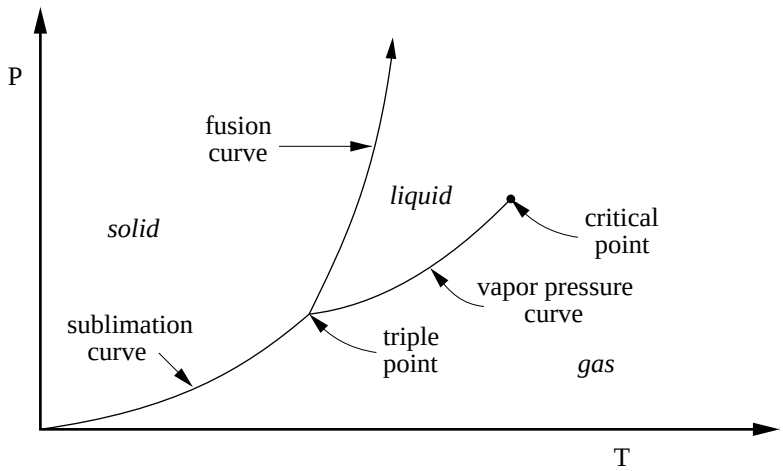

Figure 15.6: A sketch of the phase diagram for a simple material.

interaction

$$V(r) = \begin{cases} +\infty & r < \sigma \\ 0 & r \geq \sigma. \end{cases} \tag{15.45}$$

Such an interaction has been extensively studied in one dimension (hard rods), two dimensions (hard disks), and in three dimensions (hard spheres). Hard sphere systems were the first systems studied by Metropolis and coworkers.

Because there is no attractive interaction present in (15.45), there is no transition from a gas to a liquid. Is there a phase transition between a fluid phase at low densities and a solid at high densities? Can a solid form in the absence of an attractive interaction? What are the physically relevant quantities for a system with a hard core interaction? There are no thermal quantities such as the mean potential energy because the potential quantity is always zero. The major quantity of interest is $g(r)$ which yields information on the correlations of the particles and the equation of state. If the interaction is given by (15.45), it can be shown that (15.44) reduces to

$$\frac{\beta P}{\rho} = 1 + \frac{2\pi}{3}\rho\sigma^3 g(\sigma) \qquad (d = 3) \tag{15.46a}$$

$$= 1 + \frac{\pi}{2}\rho\sigma^2 g(\sigma) \qquad (d = 2) \tag{15.46b}$$

$$= 1 + \rho\sigma g(\sigma). \qquad (d = 1) \tag{15.46c}$$

We will calculate $g(r)$ for different values of r and then extrapolate our results to $r = \sigma$ (see Problem 15.23b).

Because the application of molecular dynamics and Monte Carlo methods to hard disks is similar, we discuss the latter method only briefly and do not include a program here. The idea is to choose a disk at random and move it to a trial position as implemented in the following:

```
int i = (int)(N*Math.random());
xtrial += (2.0*Math.random() - 1.0)*delta;
ytrial += (2.0*Math.random() - 1.0)*delta;
```

If the new position overlaps another disk, the move is rejected and the old configuration is retained; otherwise the move is accepted. A reasonable, although not necessarily optimum, choice for the maximum displacement δ is to choose δ such that approximately one half of all trial states are accepted. We also need to fix the maximum amplitude of the move so that the moves are equally probable in all directions.

The major difficulty in implementing this algorithm is determining the overlap of two particles. If the number of particles is not too large, it is sufficient to compute the distances between the trial particle and all the other particles rather than the smaller number of particles that are in the immediate vicinity of the trial particle. For larger systems this procedure is too time consuming, and it is better to divide the system into cells and to only compute the distances between the trial particle and particles in the same and neighboring cells.

The choice of initial positions for the disks is more complicated than it might first appear. One strategy is to place each successive disk at random in the box. If a disk overlaps one that is already present, generate another pair of random numbers and attempt to place the disk again. If the desired density is low, an acceptable initial configuration can be computed fairly quickly in this way, but if the desired density is high, the probability of adding a disk will be very small (see Problem 15.24a). To reach higher densities, we might imagine beginning with the desired number of particles in a low density configuration and moving the boundaries of the central cell inward until a boundary just touches one of the disks. Then the disks are moved a number of Monte Carlo steps and the boundaries are moved inward again. This procedure also becomes more difficult as the density increases. The most efficient procedure is to start the disks on a lattice at the highest density of interest such that no overlap of disks occurs.

We first consider a one-dimensional system of hard rods for which the equation of state and $g(r)$ can be calculated exactly. The equation of state is given by

$$\frac{P}{NkT} = \frac{1}{L - N\sigma}. \tag{15.47}$$

Because hard rods cannot pass through one another, the excluded volume is $N\sigma$ and the available volume is $L - N\sigma$. Note that this form of the equation of state is the same as the van der Waals equation of state (cf. Reif) with the contribution from the attractive part of the interaction equal to zero.

Problem 15.23. Monte Carlo simulation of hard rods

a. Write a program to do a Monte Carlo simulation of a system of hard rods. Adopt periodic boundary condition and refer to `Program hd` in Chapter 8 for the basic structure of the program. The major difference is the nature of the trial moves. Measure all lengths in terms of the hard rod diameter σ. Choose $L = 36$ and $N = 30$. How does the number density $\rho = N/L$ compare to the maximum possible density? Choose the initial positions to be on a one-dimensional grid and let the maximum displacement be $\delta = 0.1$. Approximately how many Monte Carlo steps per particle are necessary to reach equilibrium? What is the equilibrium acceptance probability ? Compute the pair correlation function $g(x)$.

b. Plot $g(x)$ as a function of the distance x. Why does $g(x) = 0$ for $x < 1$? Why are the values of $g(x)$ for $x > L/2$ not meaningful? What is the physical interpretation of the peaks in $g(x)$? Because the mean pressure can be determined from $g(x)$ at $x = 1^+$ (see (15.46)), you need to determine $g(x)$ at contact by extrapolating your results for $g(x)$ to $x = 1$. An easy way to do so is to fit the three points of $g(x)$ closest to $x = 1$ to a parabola. Use your result for $g(x = 1^+)$ to determine the mean pressure.

c. Compute $g(x)$ at several lower densities by using an equilibrium configuration from a previous run and increasing L. How do the size and the location of the peaks in $g(x)$ change?

Problem 15.24. Monte Carlo simulation of hard disks

a. The maximum packing density can be found by placing the disks on a triangular lattice with the nearest neighbor distance equal to the disk diameter. What is the maximum packing density of hard disks, that is, how many disks can be packed together in a cell of area A? Write a simple program that adds disks at random into a rectangular box of area $A = L_x \times L_y$ with the constraint that no two disks overlap. If a disk overlaps a disk already present, generate another pair of random numbers and try to place the disk again. If the density is low, the probability of adding a disk is high, but if the desired density is high most of the disks will be rejected. For simplicity, do not worry about periodic boundary conditions and accept a disk if its center lies within the box. Choose $L_x = 6$ and $L_y = \sqrt{3}L_x/2$ and determine the maximum density $\rho = N/A$ that you can attain in a reasonable amount of CPU time. How does this density compare to the maximum packing density? What is the qualitative nature of the density dependence of the acceptance probability?

b. Adapt your Monte Carlo program for hard rods to a system of hard disks. Begin at a density ρ slightly lower than the maximum packing density ρ_0. Choose $N = 64$ with $L_x = 8.81$ and $L_y = \sqrt{3}L_x/2$. Compare the density $\rho = N/(L_xL_y)$ to the maximum packing density. Choose the initial positions of the particles to be on a triangular lattice. A reasonable first choice for the maximum displacement δ is $\delta = 0.1$. Compute $g(r)$ for $\rho/\rho_0 = 0.95$, 0.92, 0.88, 0.85, 0.80, 0.70, 0.60, and 0.30. Keep the ratio of L_x/L_y fixed and save a configuration from the previous run to be the initial configuration of the new run at lower ρ. Allow at least 400 Monte Carlo steps per particle for the system to equilibrate and average $g(r)$ for `mcs` ≥ 400.

c. What is the qualitative behavior of $g(r)$ at high and low densities? For example, describe the number and height of the peaks of $g(r)$. If the system is crystalline, then $g(\mathbf{r})$ is not spherically symmetric. How would you compute $g(\mathbf{r})$ in this case and what would you expect to see?

d. Use your results for $g(r = 1^+)$ to compute the mean pressure P as a function of ρ (see (15.46b)). Plot the ratio PV/NkT as a function of ρ, where the volume V is the area of the system. How does the temperature T enter into the Monte Carlo simulation? Is the ratio PV/NkT an increasing or decreasing function of ρ? At low densities we might expect the system to act like an ideal gas with the volume replaced by $(V - N\sigma)$. Compare your low density results with this prediction.

e. Take snapshots of the disks at intervals of ten to twenty Monte Carlo steps per particle. Do you see any evidence of the solid becoming a fluid at lower densities?

f. Compute an effective diffusion coefficient D by determining the mean square displacement $\langle R^2(t)\rangle$ of the particles after equilibrium is reached. Use the relation (15.41) and identify the time t with the number of Monte Carlo steps per particle. Estimate D for the densities considered in part (a), and plot the product ρD as a function of ρ. What is the dependence of D on ρ for a dilute gas? Try to identify a range of ρ where D drops abruptly. Do you observe any evidence of a phase transition?

g. The magnitude of the maximum displacement parameter δ is arbitrary. If δ is large and the density is high, then a high proportion of the trial moves will be rejected. On the other hand, if δ is too small, the acceptance probability will be close to unity, but the successive configurations will be strongly correlated. Hence if δ is too large or is too small, our simulation is inefficient. In practice, δ is usually chosen so that $\approx 25\%$ of the moves are accepted. A better criterion is to choose δ so that the mean square displacement is a maximum for a fixed time interval. The idea is that the mean square displacement is a measure of the exploration of phase space. Fix the density and determine the value of δ that maximizes $\langle R^2(t)\rangle$. What is the corresponding acceptance probability? If this probability is much less than 50%, how can you decide which criterion for δ is more appropriate?

Continuous potentials. Our simulations of hard disks suggest that there is a phase transition from a fluid at low densities to a solid at higher densities. This conclusion is consistent with molecular dynamics and Monte Carlo studies of larger systems. Although the existence of a fluid-solid transition for hard sphere and hard disk systems is well accepted, the relatively small numbers of particles used in any simulation should remind us that results of this type cannot be taken as evidence independently of any theoretical justification.

The existence of a fluid-solid transition for hard spheres implies that the transition is primarily determined by the repulsive part of the potential. We now consider a system with both a repulsive and an attractive contribution. Our primary goal will be to determine the influence of the attractive part of the potential on the structure of a liquid.

We adopt as our model interaction the Lennard-Jones potential:

$$U(r) = 4\epsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right]. \tag{15.48}$$

The nature of the Lennard-Jones potential and the appropriate choice of units for simulations was discussed in Chapter 8 (see Table 8.1). We consider in Problem 15.25 the application of the Metropolis algorithm to a system of N particles in a rectangular cell of fixed volume V interacting via the Lennard-Jones potential. Because the simulation is at fixed T, V, and N, the simulation samples configurations of the system according to the Boltzmann distribution (15.4).

Problem 15.25. Monte Carlo simulation of a Lennard-Jones system

a. The properties of a two-dimensional Lennard-Jones system with the potential energy of interaction (15.48) have been studied by many workers under a variety of conditions. Write a program to compute the total energy of a system of N particles on a triangular lattice of area $L_x \times L_y$ with periodic boundary conditions. Choose $N = 64, L_x = 9.2$, and $L_y = \sqrt{3}L_x/2$. Why does this energy correspond to the energy at temperature $T = 0$? Does the energy per particle change if you consider bigger systems at the same density?

b. Write a program to compute the mean energy, pressure, and the radial distribution function using the Metropolis algorithm. One way of computing the change in the potential energy of the system due to a trial move of one of the particles is to use an array, `pe`, for the potential energy of interaction of each particle. For simplicity, compute the potential energy of particle `i` by considering its interaction with the other $N-1$ particles. The total potential energy of the system is the sum of the array elements `pe(i)` over all N particles divided by two to account for double counting. For simplicity, accumulate data after each Monte Carlo step per particle.

c. Choose the same values of N, L_x, and L_y as in part (a), but give each particle an initial random displacement from its triangular lattice site of magnitude 0.2. Do the Monte Carlo simulation at a very low temperature such as $T = 0.1$. Choose the maximum trial displacement $\delta = 0.15$ and consider `mcs` ≥ 400. Does the system retain its hexagonal symmetry? Does the value of δ affect your results?

d. Use the same initial conditions as in part (a), but take $T = 0.5$. Choose $\delta = 0.15$ and run for a number of Monte Carlo steps per particle that is sufficient to yield a reasonable result for the mean energy. Do a similar simulation at $T = 1$ and $T = 2$. What is the best choice of the initial configuration in each case? The harmonic theory of solids predicts that the total energy of a system is due to a $T = 0$ contribution plus a term due to the harmonic oscillation of the atoms. The contribution of the latter part should be proportional to the temperature. Compare your results for $E(T) - E(0)$ with this prediction. Use the values of σ and ϵ given in Table 8.1 to determine the temperature and energy in SI units for your simulations of solid argon.

e. Describe the qualitative nature of $g(r)$ for a Lennard-Jones solid and compare it with your hard disk results for the same density.

f. Decrease the density by multiplying L_x, L_y, and all the particle coordinates by 1.07. What is the new value of ρ? Estimate the number of Monte Carlo steps per particle needed to compute E and P for $T = 0.5$ to approximately 10% accuracy. Is the total energy positive or negative? How do E and P compare to their ideal gas values? Follow the method discussed in Problem 15.24 and compute an effective diffusion constant. Is the system a liquid or a solid? Plot $g(r)$ versus r and compare $g(r)$ to your results for hard disks at the same density. What is the qualitative behavior of $g(r)$? What is the interpretation of the peaks in $g(r)$ in terms of the structure of the liquid? If time permits, consider a larger system at the same density and temperature and compute $g(r)$ for larger r.

g. Consider the same density system as in part (f) at $T = 0.6$ and $T = 1$. Look at some typical configurations of the particles. Use your results for $E(T)$, $P(T)$, $g(r)$ and the other data you have collected, and discuss whether the system is a gas, liquid, or solid at these temperatures. What criteria can you use to distinguish a gas from a liquid? If time permits, repeat these calculations for $\rho = 0.7$.

h. Compute E, P, and $g(r)$ for $N = 64$, $L_x = L_y = 20$, and $T = 3$. These conditions correspond to a dilute gas. How do your results for P compare with the ideal gas result? How does $g(r)$ compare with the results you obtained for the liquid?

i. The chemical potential can be measured using the Metropolis algorithm and the Widom insertion method. From thermodynamics we know that

$$\mu = \Big(\frac{\partial F}{\partial N}\Big)_{V,T} = -kT \ln \frac{Z_{N+1}}{Z_N} \tag{15.49}$$

in the limit $N \to \infty$, where F is the Helmholtz free energy and Z_N is the N-particle partition function . The ratio Z_{N+1}/Z_N is the average of $e^{-\beta \Delta E}$ over all possible states of the added particle with added energy ΔE. We compute the change in the energy ΔE that would occur if an imaginary particle were added to the N particle system at random for many configurations generated by the Metropolis algorithm. The chemical potential is then given by

$$\mu = -kT \ln \langle e^{-\beta \Delta E} \rangle, \tag{15.50}$$

where the average $\langle \ldots \rangle$ is over many Monte Carlo configurations. Note that in the Widom insertion method, no particle is actually added to the system during the simulation. The chemical potential computed in (15.50) is the excess chemical potential and does not include the part of the chemical potential due to the momentum degrees of freedom, which is equal to the chemical potential of an ideal gas. Compute the chemical potential for a dilute gas, liquid, and solid configuration. Explain your results using the concept that the chemical potential is a measure of how easy it is to add a particle to the system.

15.11 Optimized Monte Carlo Data Analysis

As we have seen, the important physics near a phase transition occurs on long length scales. For this reason, we might expect that simulations, which for practical reasons are restricted to relatively small systems, might not be useful for simulations near a phase transition. Nevertheless, we have found that methods such as finite size scaling can yield information about how systems behave in the thermodynamic limit. We next explore some additional Monte Carlo techniques that are useful near a phase transition.

The Metropolis algorithm yields mean values of various thermodynamic quantities, for example, the energy, at particular values of the temperature T. Near a phase transition many thermodynamic quantities change rapidly, and we need to determine these quantities at many closely spaced values of T. If we were to use standard Monte Carlo methods, we would have to do many simulations to cover the desired range of values of T. To overcome this problem, we introduce the use of *histograms* which allow us to extract more information from a single Monte Carlo simulation. The idea is to use our knowledge of the equilibrium probability distribution at one value of T (and other external parameters) to estimate the desired thermodynamic averages at neighboring values of the external parameters.

The first step of the single histogram method for the Ising model is to simulate the system at an inverse temperature β_0 which is near the values of β of interest and measure the energy of the system after every Monte Carlo step per spin (or other fixed interval). The measured probability that the system has energy E can be expressed as

$$P(E, \beta_0) = \frac{H_0(E)}{\sum_E H_0(E)}. \tag{15.51}$$

The histogram $H_0(E)$ is the number of configurations with energy E, and the denominator is the total number of measurements of E (for example, the number of Monte Carlo steps per spin). Because the probability of a given configuration is given by the Boltzmann distribution, we have

$$P(E,\beta) = \frac{g(E)\, e^{-\beta E}}{\sum_E g(E)\, e^{-\beta E}}, \tag{15.52}$$

where $g(E)$ is the number of microstates with energy E. (The density of states $g(E)$ should not be confused with the radial distribution function $g(r)$.) If the energy is a continuous function, $g(E)$ becomes the number of states per unit energy interval. However, $g(E)$ is usually referred to as the density of states regardless of whether E is a continuous or discrete variable. If we compare (15.51) and (15.52) and note that $g(E)$ is independent of T, we can write

$$g(E) = a_0 H_0(E) e^{\beta_0 E}, \tag{15.53}$$

where a_0 is a proportionality constant that depends on β_0. If we eliminate $g(E)$ from (15.52) by using (15.53), we obtain the desired relation

$$P(E,\beta) = \frac{H_0(E)\, e^{-(\beta-\beta_0)E}}{\sum_E H_0(E)\, e^{-(\beta-\beta_0)E}}. \tag{15.54}$$

Note that we have expressed the probability at the inverse temperature β in terms of $H_0(E)$, the histogram at the inverse temperature β_0.

Because β is a continuous variable, we can estimate the β dependence of the mean value of any function A that depends on E, for example, the mean energy and the specific heat. We write the mean of $A(E)$ as

$$\langle A(\beta)\rangle = \sum_E A(E) P(E,\beta). \tag{15.55}$$

If the quantity A depends on another quantity M, for example, the magnetization, then we can generalize (15.55) to

$$\langle A(\beta)\rangle = \textstyle\sum_{E,M} A(E,M) P(E,M,\beta) \tag{15.56a}$$

$$= \frac{\sum_{E,M} A(E,M) H_0(E,M) e^{-(\beta-\beta_0)E}}{\sum_{E,M} H_0(E,M) e^{-(\beta-\beta_0)E}}. \tag{15.56b}$$

The histogram method is useful only when the configurations relevant to the range of temperatures of interest occur with reasonable probability during the simulation at temperature T_0. For example, if we simulate an Ising model at low temperatures at which only ordered configurations occur (most spins aligned in the same direction), we cannot use the histogram method to obtain meaningful thermodynamic averages at high temperatures for which most configurations are disordered.

Problem 15.26. Application of the histogram method

a. Consider a 4×4 Ising lattice in zero magnetic field and compute using the Metropolis algorithm the mean energy per spin, the mean magnetization per spin, the specific heat, and the susceptibility per spin for $T = 1$ to $T = 3$ in steps of $\Delta T = 0.05$. Average over at least 5000 Monte Carlo steps per spin after equilibrium has been reached for each value of T.

b. What are the minimum and maximum values of the total energy E and magnetization M that might be observed in a simulation of a Ising model on a 4×4 lattice? Use these values to set the size of the two-dimensional array needed to accumulate data for the histogram $H(E, M)$. It is suggested that you modify class `Ising` and save $H(E, M)$ in a file. Accumulate data for $H(E, M)$ at $T = 2.27$, a value of T close to T_c, for at least 5000 Monte Carlo steps per spin after equilibration. Write a separate program to read the histogram file and compute the same thermodynamic quantities as in part (a) using (15.56b). Compare your computed results with the data obtained by simulating the system directly, that is, without using the histogram method, at the same temperatures. At what temperatures does the histogram method break down?

c. Repeat part (b) for a simulation centered about $T = 1.5$ and $T = 2.5$.

d. Repeat part (b) for an 8×8 and a 16×16 lattice at $T = 2.27$.

The histogram method can be used to do a more sophisticated finite size scaling analysis to determine the nature of a transition. Suppose that we perform a Monte Carlo simulation and observe a peak in the specific heat as a function of the temperature. What can this observation tell us about a possible phase transition? In general, we can conclude very little without doing a careful analysis of the behavior of the system at different sizes. For example, a discontinuity in the energy in an infinite system might be manifested in small systems by a peak in the specific heat. However, a phase transition in the infinite system in which the energy is continuous, but its derivative diverges at the transition, might manifest itself in the same way in a small system. Another difficulty is that the peak in the specific heat of a small system occurs at a temperature that differs from the transition temperature in the infinite system (see Project 15.39). Finally, there might be no transition at all, and the peak might simply represent a broad crossover from high to low temperature behavior (see Project 15.40).

We now discuss a method due to Lee and Kosterlitz that uses the histogram data to determine the nature of a phase transition (if it exists). To understand this method, we introduce the (Helmholtz) free energy F of a system. The statistical mechanics definition of F is

$$F = -kT \ln Z. \tag{15.57}$$

At low T, the low energy configurations dominate the contributions to the partition function Z, even though there are relatively few such configurations. At high T, the number of disordered configurations with high E is large, and hence high energy configurations dominate the contribution to Z. These considerations suggest that it is useful to define a restricted free energy $F(E)$ that includes only configurations at a particular energy E. We define

$$F(E) = -kT \ln g(E)\, e^{-\beta E}. \tag{15.58}$$

For systems with a first-order phase transition, a plot of $F(E)$ versus E will show two local minima corresponding to configurations that are characteristic of the high and low temperature phases. At low T the minimum at the lower energy will be the absolute minimum, and at high T the higher energy minimum will be the absolute minimum of F. At the transition, the two minima will have the same value of $F(E)$. For systems with no transition in the thermodynamic limit, there will only be one minimum for all T.

How will $F(E)$ behave for the relatively small lattices considered in simulations? In systems with first-order transitions, the distinction between low and high temperature phases will become more pronounced as the system size is increased. If the transition is continuous, there are domains at all sizes, and we expect that the behavior of $F(E)$ will not change significantly as the system size increases. If there is no transition, there might be a spurious double minima for small systems, but this spurious behavior should disappear for larger systems. Lee and Kosterlitz proposed the following method for categorizing phase transitions.

1. Do a simulation at a temperature close to the suspected transition temperature, and compute $H(E)$. Usually, the temperature at which the peak in the specific heat occurs is chosen as the simulation temperature.
2. Use the histogram method to compute $-\ln H_0(E) + (\beta - \beta_0)E \propto F(E)$ at neighboring values of T. If there are two minima in $F(E)$, vary β until the values of $F(E)$ at the two minima are equal. This temperature is an estimate of the possible transition temperature T_c.
3. Measure the difference ΔF at T_c between $F(E)$ at the minima and $F(E)$ at the maximum between the two minima.
4. Repeat steps (1)–(3) for larger systems. If ΔF increases with size, the transition is first-order. If ΔF remains the same, the transition is continuous. If ΔF decreases with size, there is no thermodynamic transition.

The above procedure is applicable when the phase transition occurs by varying the temperature. Transitions also can occur by varying the pressure or the magnetic field. These *field-driven transitions* can be tested by a similar method. For example, consider the Ising model in a magnetic field at low temperatures below T_c. As we vary the magnetic field from positive to negative, there is a transition from a phase with magnetization $M > 0$ to a phase with $M < 0$. Is this transition first-order or continuous? To answer this question, we can use the Lee-Kosterlitz method with a histogram $H(E, M)$ generated at zero magnetic field, and calculate $F(M)$ instead of $F(E)$. The quantity $F(M)$ is proportional to $-\ln \sum_E H(E, M)\, e^{-(\beta-\beta_0)E}$. Because the states with positive and negative magnetization are equally likely to occur for zero magnetic field, we should see a double minima structure for $F(M)$ with equal minima. As we increase the size of the system, ΔF should increase for a first-order transition and remain the same for a continuous transition.

Problem 15.27. Characterization of a phase transition

a. Use your modified version of class `Ising` from Problem 15.26 to determine $H(E, M)$. Read the $H(E, M)$ data from a file, and compute and plot $F(E)$ for the range of temperatures of interest. First generate data at $T = 2.27$ and use the Lee-Kosterlitz method to verify that the Ising model in two dimensions has a continuous phase transition in zero magnetic field. Consider lattices of sizes $L = 4, 8$, and 16.

b. Perform a Lee-Kosterlitz analysis of the Ising model at $T = 2$ and zero magnetic field by plotting $F(M)$. Determine if the transition from $M > 0$ to $M < 0$ is first-order or continuous. This transition is called field-driven because the transition occurs if we change the magnetic field. Make sure your simulations sample configurations with both positive and negative magnetization by using small values of L such as $L = 4$, 6, and 8.

c. Repeat part (b) at $T = 2.5$ and determine if there is a field-driven transition at $T = 2.5$.

Problem 15.28. The Potts Model

a. In the q-state Potts model, the total energy or Hamiltonian of the lattice is given by

$$H = -J \sum_{i,j=\text{nn}(i)} \delta_{s_i,s_j}, \tag{15.59}$$

where s_i at site i can have the values $1, 2, \cdots, q$; the Kronecker delta function $\delta_{a,b}$ equals unity if $a = b$ and is zero otherwise. As before, we will measure the temperature in energy units. Convince yourself that the $q = 2$ Potts model is equivalent to the Ising model (except for a trivial difference in the energy minimum). One of the many applications of the Potts model is to helium absorbed on the surface of graphite. The graphite-helium interaction gives rise to preferred adsorption sites directly above the centers of the honeycomb graphite surface. As discussed by Plischke and Bergersen, the helium atoms can be described by a three-state Potts model.

b. The transition in the Potts model is continuous for small q and first-order for larger q. Write a Monte Carlo program to simulate the Potts model for a given value of q and store the histogram $H(E)$. Test your program by comparing the output for $q = 2$ with your Ising model program.

c. Use the Lee-Kosterlitz method to analyze the nature of the phase transition in the Potts model for $q = 3$, 4, 5, 6, and 10. First find the location of the specific heat maximum, and then collect data for $H(E)$ at the specific heat maximum. Lattice sizes of order $L \geq 50$ are required to obtain convincing results for some values of q.

Another common method for determining the nature of a phase transition is to use the Binder cumulant method. The cumulant is defined by

$$U_L \equiv 1 - \frac{\langle E^4 \rangle_L}{3\langle E^2 \rangle_L^2}. \tag{15.60}$$

It can be shown that the minimum value of U_L is

$$U_{L,\text{min}} = \frac{2}{3} - \frac{1}{3}\Big(\frac{E_+^2 - E_-^2}{2E_+E_-}\Big)^2 + O(L^{-d}), \tag{15.61}$$

where E_+ and E_- are the energies of the two phases in a first-order transition. These results are derived by considering the distribution of energy values to be a sum of Gaussians about each phase at the transition, which become sharper and sharper as $L \to \infty$. If $U_{L,\text{min}} = 2/3$ in the infinite size limit, then the transition is continuous.

Problem 15.29. The Binder cumulant and the nature of the transition

a. Suppose that the energy in a system is given by a Gaussian distribution. What is the corresponding value of U_L?

b. Consider the two-dimensional Ising model in the absence of a magnetic field and consider the cumulant

$$V_L \equiv 1 - \frac{\langle M^4 \rangle_L}{3\langle M^2 \rangle_L^2}. \tag{15.62}$$

Compute V_L for a temperature much higher than T_c. What is the value of V_L? What is the value of V_L at $T = 0$?

c. Compute V_L for values of T in the range $2.20 \leq T \leq 2.35$ for $L = 10$, 20, and 40, and plot V_L as a function of T for these three values of L. Note that the three curves for V_L cross at a value of T that is approximately T_c. What is the approximate value of V_L at this crossing? Can you conclude that the transition is continuous?

d. Repeat Problem (15.28) using the Binder cumulant method and determine the nature of the transition.

15.12 *Other Ensembles

So far, we have considered the microcanonical ensemble (fixed N, V, and E) and the canonical ensemble (fixed N, V, and T). Monte Carlo methods are very flexible and can be adapted to the calculation of averages in any ensemble. Two other ensembles of particular importance are the constant pressure and the grand canonical ensembles. The main difference in the Monte Carlo method is that there are additional moves corresponding to changing the volume or changing the number of particles. The constant pressure ensemble is particularly important for studying first-order phase transitions because the phase transition occurs at a fixed pressure, unlike a constant volume simulation where the system passes through a two phase coexistence region before changing phase completely as the volume is changed.

In the NPT ensemble, the probability of a microstate is proportional to $e^{-\beta(E+PV)}$. For a classical system, the mean value of a physical quantity A that depends on the coordinates of the particles can be expressed as

$$\langle A \rangle_{\text{NPT}} = \frac{\int_0^\infty dV e^{-\beta PV} \int d\mathbf{r}_1 d\mathbf{r}_2 \ldots d\mathbf{r}_N A(\{\mathbf{r}\})\, e^{-\beta U(\{\mathbf{r}\})}}{\int_0^\infty dV e^{-\beta PV} \int d\mathbf{r}_1 d\mathbf{r}_2 \ldots d\mathbf{r}_N\, e^{-\beta U(\{\mathbf{r}\})}}. \tag{15.63}$$

The potential energy $U(\{\mathbf{r}\})$ depends on the set of particle coordinates $(\{\mathbf{r}\})$. To simulate the NPT ensemble, we need to sample the coordinates $\mathbf{r}_1, \mathbf{r}_2, \cdots, \mathbf{r}_N$ of the particles and the volume V of the system. For simplicity, we assume that the central cell is a square or a cube so that $V = L^d$. It is convenient to use the set of scaled coordinates $\{\mathbf{s}\}$, where $\mathbf{s}_i$ is defined as

$$\mathbf{s}_i = \frac{\mathbf{r}_i}{L}. \tag{15.64}$$

If we substitute (15.64) into (15.63), we can write $\langle A \rangle_{\text{NPT}}$ as

$$\langle A \rangle_{\text{NPT}} = \frac{\int_0^\infty dV e^{-\beta PV} V^N \int d\mathbf{s}_1 d\mathbf{s}_2 \ldots d\mathbf{s}_N A(\{\mathbf{s}\})\, e^{-\beta U(\{\mathbf{s}\})}}{\int_0^\infty dV e^{-\beta PV} V^N \int d\mathbf{s}_1 d\mathbf{s}_2 \ldots d\mathbf{s}_N\, e^{-\beta U(\{\mathbf{s}\})}}, \tag{15.65}$$

where the integral over $\{\mathbf{s}\}$ is over the unit square (cube). The factor of V^N arises from the change of variables $\mathbf{r} \to \mathbf{s}$. If we let $V^N = e^{\ln V^N} = e^{N \ln V}$, we see that the quantity that is analogous to the Boltzmann factor can be written as

$$e^{-W} = e^{-\beta PV - \beta U(\{\mathbf{s}\}) + N \ln V}. \tag{15.66}$$

Because the pressure is fixed, a trial configuration is generated from the current configuration by either randomly displacing a particle or making a random change in the volume, for example, $V \to V + \delta V_{\max}(2r-1)$, where r is a uniform random number in the unit interval. The trial configuration is accepted if the change $\Delta W \leq 0$ and with probability $e^{-\Delta W}$ if $\Delta W > 0$. It is not necessary or efficient to change the volume after every Monte Carlo step per particle.

In the grand canonical or μVT ensemble, the chemical potential μ is fixed and the number of particles fluctuates. The average of any function of the particle coordinates can be written as (in three dimensions)

$$\langle A \rangle_{\mu \mathrm{VT}} = \frac{\sum\limits_{N=0}^{\infty} (1/N!)\, \lambda^{-3N}\, e^{\beta \mu N} \int d\mathbf{r}_1 d\mathbf{r}_2 \ldots d\mathbf{r}_N A(\{\mathbf{r}\})\, e^{-\beta U_N(\{\mathbf{r}\})}}{\sum\limits_{N=0}^{\infty} (1/N!)\, \lambda^{-3N} e^{\beta \mu N} \int d\mathbf{r}_1 d\mathbf{r}_2 \ldots d\mathbf{r}_N\, e^{-\beta U_N(\{\mathbf{r}\})}}, \tag{15.67}$$

where $\lambda = (h^2/2\pi mkT)^{1/2}$. We have made the N-dependence of the potential energy U explicit. If we write $1/N! = e^{-\ln N!}$ and $\lambda^{-3N} = e^{-N \ln \lambda^3}$, we can write the quantity that is analogous to the Boltzmann factor as

$$e^{-W} = e^{\beta \mu N - N \ln \lambda^3 - \ln N! + N \ln V - \beta U_N}. \tag{15.68}$$

If we write the chemical potential as

$$\mu = \mu^* + kT \ln(\lambda^3/V), \tag{15.69}$$

then W can be expressed as

$$e^{-W} = e^{-\beta \mu^* N - \ln N! - \beta U_N}. \tag{15.70}$$

The parameters are μ^*, V, and T. There are two possible ways of obtaining a trial configuration. The first involves the displacement of a selected particle; such a move is accepted or rejected according to the usual criteria, that is, by the change in the potential energy U_N. In the second possible way, we choose with equal probability whether to attempt to add a particle at a randomly chosen position in the central cell or to remove a particle that is already present. In either case, the trial configuration is accepted if W in (15.70) is increased. If W is decreased, the change is accepted with a probability equal to

$$\frac{1}{N+1} e^{\beta\left(\mu^* - (U_{N+1} - U_N)\right)}, \qquad \text{(insertion)} \tag{15.71a}$$

or

$$N e^{-\beta\left(\mu^* + (U_{N-1} - U_N)\right)}. \qquad \text{(removal)} \tag{15.71b}$$

In this approach μ^* is an input parameter and μ is not determined until the end of the calculation when $\langle N \rangle_{\mu \mathrm{VT}}$ is obtained.

As we have discussed, the probability that a system at a temperature T has energy E is given by (see (15.52))

$$P(E, \beta) = \frac{g(E)\, e^{-\beta E}}{Z}, \tag{15.72}$$

where we have included the normalization factor Z defined as

$$Z = \sum_E g(E) e^{-\beta E}. \tag{15.73}$$

The function Z is known as the partition function . If $g(E)$ were known, we could calculate the mean energy (and other thermodynamic quantities) at *any* temperature from the relation

$$\langle E \rangle = \frac{1}{Z} \sum_E E g(E) e^{-\beta E}. \tag{15.74}$$

Hence, the density of states is a quantity of much interest.

Suppose that we were to try to compute $g(E)$ by doing a random walk in energy space by flipping the spins at random and accepting all configurations that are obtained. Then the histogram of the energy, $H(E)$, the number of visits to each possible energy E of the system, would converge to $g(E)$ if the walk visited all possible configurations. However, in practice, it would be impossible to realize such a long random walk given the extremely large number of configurations. For example, the Ising model on a 10×10 square lattice has $2^{100} \approx 1.3 \times 10^{30}$ spin configurations.

The main difficulty of doing a simple random walk to determine $g(E)$ is that the walk would spend most of its time visiting the same energy values over and over again and would not reach the values of E that are less probable. The idea of the *Wang-Landau* algorithm is to do a random walk in energy space by flipping the spins at random, but to accept the changes with a probability that is proportional to the reciprocal of the density of states. That is, energy values that would be visited often using a simple random walk would be visited less often because they have a bigger density of states. There is only one problem – we don't know the density of states. But as we will see the Wang-Landau algorithm estimates the density of states at the same time that it does a random walk in phase space. For simplicity, we discuss the algorithm in the context of the Ising model for which E is a discrete variable.

1. Start with an initial (arbitrary) configuration of spins and a guess for the density of states. The simplest guess is to set $g(E) = 1$ for all possible energies E.

2. Choose a spin at random and make a trial change. Compute the energy before the flip, E_1, and after, E_2, and accept the change with probability

$$p(E_1 \to E_2) = \min\big(\frac{g(E_1)}{g(E_2)}, 1\big), \tag{15.75}$$

   Equation (15.75) implies that if $g(E_2) \leq g(E_1)$, the state with energy E_2 is always accepted; otherwise, it is accepted with probability $g(E_1)/g(E_2)$. That is, the state with energy E_2 is accepted if a random number $r \leq g(E_1)/g(E_2)$.

3. Suppose that after step (2) the energy of the system is E. (E is either E_2 if the change is accepted or remains at E_1 if the change is not accepted.) Then

$$g(E) = fg(E) \tag{15.76}$$
$$H(E) = H(E) + 1. \tag{15.77}$$

That is, we multiply the current value of $g(E)$ by the modification factor $f > 1$, and we update the existing entry for $H(E)$ in the energy histogram. Because $g(E)$ becomes very large, in practice we must work with the logarithm of the density of states, so that $\ln(g(E))$ will fit into double precision numbers. Therefore, each update of the density of states is implemented as $\ln(g(E)) \to \ln(g(E)) + \ln(f)$, and the ratio of the density of states is computed as $\exp[\ln(g(E_1)) - \ln(g(E_2))]$.

4. A reasonable choice of the initial modification factor is $f = f_0 = e^1 \simeq 2.71828\ldots$ If f_0 is too small, the random walk will need a very long time to reach all possible energies; however, too large a choice of f_0 will lead to large statistical errors.

5. Proceed with the random walk in energy space until a "flat" histogram $H(E)$ is obtained, that is, until all the possible energies are visited an equal number of times. If the histogram were truly flat, all the possible energies would have been visited an equal number of times. Of course, it is impossible to obtain a perfectly flat histogram, and we will say that $H(E)$ is flat when $H(E)$ for all possible E is not less than x of the average histogram $\langle H(E)\rangle$; x is chosen according to the size and the complexity of the system and the desired accuracy of the density of states. For the two-dimensional Ising model with nearest-neighbor interactions on small lattices, x can be chosen to be as high as 0.95, but for large systems the criterion for flatness may never be satisfied if x is too large.

6. Once the flatness criterion has been satisfied, reduce the modification factor using a function such as $f_1 = \sqrt{f_0}$, reset the histogram to $H(E) = 0$ for all values of E, and begin the next iteration of the random walk during which the density of states is modified by f_1 at each step. The spin configuration and the density of states are not reset during the simulation. We continue performing the random walk until the histogram $H(E)$ is again flat.

7. Reduce the modification factor, $f_{i+1} = \sqrt{f_i}$, reset the histogram to $H(E) = 0$ for all values of E and continue the random walk. Stop the simulation when f is smaller than a predefined value (such as $f_{\text{final}} = \exp(10^{-8}) \approx 1.00000001$). The modification factor acts as a control parameter for the accuracy of the density of states during the simulation and also determines how many MC sweeps are necessary for the whole simulation.

At the end of the simulation, the algorithm provides only a relative density of states. To determine the normalized density of states $g_n(E)$, we can either use the fact that the total number of states for the Ising model is $\sum_E g(E) = 2^N$, or that the number of ground states (for which $E = -2N$) is 2. The latter normalization guarantees the accuracy of the density of states at low energies which is important in the calculation of thermodynamic quantities at low temperature. If we apply the former condition, we cannot guarantee the accuracy of $g(E)$ for energies at or near the ground state, because the rescaling factor is dominated by the maximum density of states. We can use one of these two normalizations to obtain the absolute density of states, and use the other normalization to check the accuracy of our result.

Problem 15.30. Sampling the density of states

a. Implement the Wang-Landau algorithm for the two-dimensional Ising model for $L = 4$, 8, and 16. For simplicity, choose $x = 0.8$ as your criterion for flatness. Approximately xx MC steps per spin will be needed for each iteration. Determine the density of states and describe its qualitative dependence on E.

b. Compute $P(E) = g(E)e^{-\beta E}/Z$ for different temperatures for the $L = 16$ system. If $T = 0.1$, what range of energies will contribute to the heat capacity? Determine the range of relevant energies for $T = 1.0$, $T = T_c$, and $T = 4.0$.

c. Use the density of states that you computed in part (a) to compute the mean energy, the heat capacity, the free energy, and the entropy as a function of temperature. Compare your results to your results for $\langle E \rangle$ and C that you found using the Metropolis algorithm in Problem 15.16.

d. Use the Wang-Landau algorithm to determine the density of states for the one-dimensional Ising model. In this case you can compare your computed values of $g(E)$ to the exact answer:

$$g(E) = 2\frac{N!}{i!(N-i)!}, \tag{15.78}$$

where $E = 2i - N$, $i = 0, 2, \ldots, N$, and N is even. How does the accuracy of the computed values of $g(E)$ depend on the choice of x for the flatness criterion? (Exact results are available for $g(E)$ for the two-dimensional Ising model as well, but no explicit combinatorial formula exists. See the article by Beale.)

e.* The results that you have obtained so far have probably convinced you that the Wang-Landau algorithm is ideal for simulating a variety of systems with many degrees of freedom. However, what about critical slowing down? Does the Wang-Landau algorithm overcome this limitation of other single spin flip algorithms? To gain some insight, let us ask, given the exact $g(E)$, how efficiently does the Wang-Landau sample the different values of E? Use either the exact density of states in two dimensions computed by Beale or the approximate one that you computed in part (a) and set $f = 1$. Because the system is doing a random walk in energy space, it is reasonable to compute the diffusion constant of the random walker in energy space:

$$D_E(t) = \langle [E(t) - E(0)]^2 \rangle / t, \tag{15.79}$$

where t is the time difference and the choice of the time origin is arbitrary. The idea is to find the dependence of D on the energy E of the system at a particular time origin. How long does it take the system to return to this energy? Run for a sufficiently long time so that D_E is independent of t. Plot D_E as a function of E. Where is D a maximum? If time permits, determine D_E at the energy E_c corresponding to the critical temperature. How does D_{E_c} depend on L?

f.* mean-first passage time.

Problem 15.31. Optimized ensemble algorithm

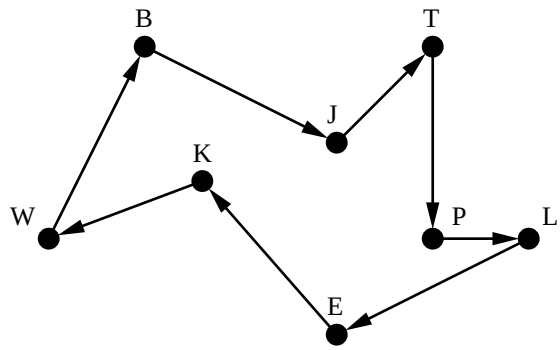

Figure 15.7: What is the optimum route for this random arrangement of $N = 8$ cities? The route begins and ends at city W. A possible route is shown.

15.13 More Applications

You probably do not need to be convinced that Monte Carlo methods are powerful, flexible, and applicable to a wide variety of systems. Extensions to the Monte Carlo methods that we have not discussed include multiparticle moves, biased moves where particles tend to move in the direction of the force on them, manipulation of bits for Ising-like models, and the use of parallel processing to update different parts of a large system simultaneously. We also have not described the simulation of systems with long-range potentials such as Coulombic systems and dipole-dipole interactions. For these potentials, it is necessary to include the interactions of the particles in the center cell with the infinite set of periodic images.

We conclude this chapter with a discussion of Monte Carlo methods in a context that might seem to have little in common with the types of problems we have discussed. This context is called *multivariate* or *combinatorial optimization,* a fancy way of saying, "How do you find the minimum of a function that depends on many parameters?" Problems of this type arise in many areas of scheduling and design as well as in physics, biology, and chemistry. We explain the nature of this type of problem by an example known as the *traveling salesperson problem*, a.k.a., the traveling salesman problem.

Suppose that a salesman wishes to visit N cities and follow a route such that no city is visited more than once and the end of the trip coincides with the beginning. Given these constraints, the traveling salesman problem is to find the optimum route such that the total distance traveled is a minimum. An example of $N = 8$ cities and a possible route is shown in Fig. 15.7. All known exact methods for determining the optimal route require a computing time that increases as e^N, and hence, in practice, an exact solution can be found only for a small number of cities. The traveling salesman problem belongs to a large class of problems known as NP-complete. (The NP refers to non-polynomial, that is, such problems cannot be done in a time proportional to some finite polynomial in N.) What is a reasonable estimate for the maximum number of cities that you can consider without the use of a computer?

To understand the nature of the different approaches to the traveling salesman problem,

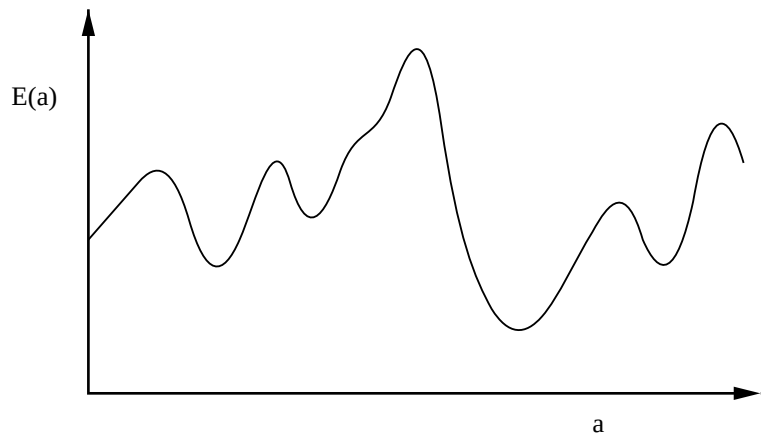

Figure 15.8: Plot of the function $E(a)$ as a function of the parameter a.

consider the plot in Fig. 15.8 of the "energy" or "cost" function $E(a)$. We can associate $E(a)$ with the length of the route and interpret a as a parameter that represents the order in which the cities are visited. If $E(a)$ has several local minima, what is a good strategy for finding the global (absolute) minimum of $E(a)$? One way is to vary a systematically and find the value of E everywhere. This way corresponds to an exact enumeration method and would mean knowing the length of each possible route, an impossible task if the number of cities is too large. Another way is to use a *heuristic method*, that is, an approximate method for finding a route that is close to the absolute minimum. One strategy is to choose a value of a, generate a small random change δa, and accept this change if $E(a+\delta a)$ is less than or equal to $E(a)$. This iterative improvement strategy corresponds to a search for steps that lead downhill. Because this search usually becomes stuck in a local and not a global minimum, it is useful to begin from several initial choices of a and to keep the best result. What would be the application of this type of strategy to the salesman problem?

Because we cannot optimize the path exactly when N becomes large, we have to be satisfied with solving the optimization problem approximately and finding a relatively good local minimum. To understand the motivation for the *simulated annealing* algorithm, consider a seemingly unrelated problem. Suppose we wish to make a perfect single crystal. You might know that we should start with the material at a high temperature at which the material is a liquid melt and then gradually lower the temperature to the desired low temperature. If we lower the temperature too quickly (a rapid "quench"), the resulting crystal would have many defects or not become a crystal at all. The gradual lowering of the temperature is known as *annealing*.

The method of annealing can be used to estimate the minimum of $E(a)$. We choose a value of a, generate a small random change δa, and calculate $E(a+\delta a)$. If $E(a+\delta a)$ is less than or equal to $E(a)$, we accept the change. However, if $\Delta E = E(a+\delta a) - E(a) > 0$, we accept the change with a probability $P = e^{-\Delta E/T}$, where T is an effective temperature. This procedure is the familiar Metropolis algorithm with the temperature playing the role of a control parameter. The *simulated annealing* process consists of first choosing a value for T for which most moves are accepted, and then gradually lowering the temperature. At each temperature, the simulation

should last long enough for the system to reach quasiequilibrium. The annealing schedule, that is, the rate of temperature decrease, determines the quality of the solution.

The idea is to allow moves that result in solutions of worse quality than the current solution (uphill moves) in order to escape from local minima. The probability of doing such a move is decreased during the search. The slower the temperature is lowered, the higher the chance of finding the optimum solution, but the longer the run time. Thus effective use of simulated annealing depends on finding a cooling schedule that yields good enough solutions without taking too much time. It has been proved that if the cooling rate is sufficiently slow, a globally optimal solution will eventually be reached. The bounds for "sufficiently slow" depend on the properties of the search landscape (the nature of $E(a)$) and have to be exceeded for most practical cases. However, simulated annealing is usually superior to conventional heuristic algorithms.

The moral of the simulated annealing method is that sometimes it is necessary to climb a hill to reach a valley. The first application of the method of simulated annealing was to the optimal design of computers. In Problem 15.32 we apply this method to the traveling salesman problem.

Problem 15.32. Simulated annealing and the traveling salesman problem

a. Generate a random arrangement of $N = 8$ cities in a square of linear dimension $L = \sqrt{N}$ and calculate the optimum route by hand. Then write a Monte Carlo program and apply the method of simulated annealing to this problem. For example, use two arrays to store the x and y coordinate of each city and an array to store the distances between them. The state of the system, that is, the route represented by a sequence of cities, can be stored in another array. The length of this route is associated with the energy of an imaginary thermal system. A reasonable choice for the initial temperature is one that is the same order as the initial energy. One way to generate a random rearrangement of the route is to choose two cities at random and to interchange the order of visit. Choose this method or one that you devise and find a reasonable annealing schedule. Compare your annealing results to exact results whenever possible. Extend your results to larger N, for example, $N = 12$, 24, and 48. For a given annealing schedule, determine the probability of finding a route of a given length. More suggestions can be found in the references.

b. The microcanonical Monte Carlo algorithm (demon) discussed in Section 15.3 also can be used to do simulated annealing. The advantages of the demon algorithm are that it is deterministic and allows large temperature fluctuations. One way to implement the analog of simulated annealing is to impose a maximum value on the energy of the demon, $E_{d,\max}$ and then gradually lower the value of $E_{d,\max}$. Guo et al. choose the initial value of $E_{d,\max}$ to equal $\sqrt{N}/4$. Their results are comparable to the canonical simulated annealing method, but require approximately half the CPU time. Apply this method to the same city positions that you considered in part (a) and compare your results.

Problem 15.33. Extremal optimization and the search for ground states

15.14 Projects

Many of the original applications of Monte Carlo methods were done for systems of approximately one hundred particles and lattices of order 32^2 spins. Most of these applications can be done with much better statistics and with larger system sizes. In the following, we discuss some recent developments, but this discussion is not complete and we have omitted other important topics such as Brownian dynamics and umbrella sampling. More ideas for projects can be found in the references.

Project 15.34. Overcoming critical slowing down

The usual limiting factor of most simulations is the speed of the computer. Of course, one way to overcome this problem is to use a faster computer. However, near a phase transition, the most important limiting factor on even the fastest available computers is the existence of critical slowing down (see Problem 15.19). In this project we discuss the nature of critical slowing down and ways of overcoming it in the context of the Ising model.

As mentioned, the existence of critical slowing down is related to the fact that the size of the correlated regions of spins becomes very large near the Ising critical point. The large size of the correlated regions and the corresponding divergent behavior of the correlation length ξ near T_c implies that the time τ required for a region to lose its coherence becomes very long if a *local* dynamics is used. At $T = T_c$, we have $\tau \sim L^z$ for L sufficiently large. For the single spin flip (Metropolis) algorithm, $z \approx 2$, and τ becomes very large for $L >> 1$. On a serial computer, the CPU time needed to obtain n configurations increases as L^2, the time needed to visit L^2 spins. This factor of L^2 is not a problem because a larger system contains proportionally more information. However, the time needed to obtain n approximately *independent* configurations is order $\tau L^2 \sim L^{2+z} \approx L^4$ for the Metropolis algorithm. We see that an increase of L by a factor of 10 requires 10^4 more computing time. Hence, the existence of critical slowing down limits the maximum value of L that can be considered.

If we are interested only in the static properties of the Ising model, the choice of dynamics is irrelevant as long as the transition probability satisfies the detailed balance condition (15.17). As we have mentioned, the Metropolis algorithm becomes inefficient near T_c because only single spins are flipped. Hence, it is reasonable to look for a *global* algorithm for which *clusters* of spins are flipped simultaneously. We already are familiar with cluster properties in the context of percolation (see Chapter 12). A naive definition of a cluster of spins might be a domain or group of parallel nearest neighbor spins. We can make this definition explicit by introducing a bond between any two nearest neighbor spins that point in the same direction. The introduction of a bond between spins of the same direction defines a site-bond percolation problem. More generally, we may assume that such a bond exists with probability p and this bond probability depends on the temperature T.

The dependence of p on T can be determined by requiring that the percolation transition of the clusters occurs at the Ising critical point, and by requiring that the critical exponents associated with the clusters be identical to the analogous thermal exponents. For example, we can define a critical exponent ν_p to characterize the divergence of the connectedness length of the clusters near p_c. The analogous thermal exponent ν quantifies the divergence of the thermal correlation length ξ near T_c. We will argue in the following that these (and other) critical exponents are identical if

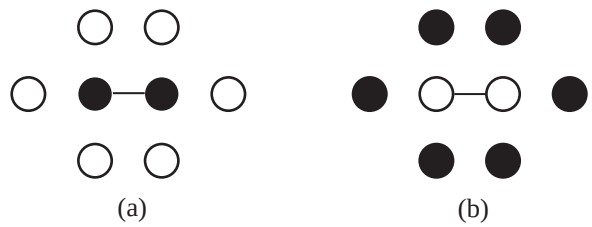

Figure 15.9: (a) A cluster of two up spins. (b) A cluster of two down spins. The solid and open circles represent the up and down spins. Note the bond between the two spins in the cluster. Adapted from Newman and Barkema.

we define the bond probability as

$$p = 1 - e^{-2J/kT}. \qquad \text{(bond probability)} \tag{15.80}$$

The relation (15.80) holds for any spatial dimension. What is the value of p at $T = T_c$ for the two-dimensional Ising model on the square lattice?

A simple argument for the temperature-dependence of p in (15.80) is as follows. Consider the two configurations in Figure 15.9 which differ from one another by the flip of the single cluster of two spins. In Figure 15.9(a) the six nearest neighbor spins of the cluster are in the opposite direction and hence are not part of the cluster. Thus the probability of this configuration with a cluster of two spins is $pe^{-\beta J}e^{6\beta J}$, where p is the probability of a bond between the two up spins, $e^{-\beta J}$ is proportional to the probability that these two spins being parallel, and $e^{6\beta J}$ is proportional to the probability of the six nearest neighbors being down. In Figure 15.9(b) the possible bonds between the cluster spins and its nearest neighbors would have to be "broken" if the cluster is flipped. The probability of this configuration with a cluster of two (down) spins is $p(1-p)^6 e^{-\beta J}e^{-6\beta J}$, where the factor of $1-p)^6$ is the probability that the six nearest neighbor spins are not part of the cluster. Because we want the probability that a cluster is flipped to be unity, we need to have the probability of the two configurations with their corresponding clusters be the same. Hence, we must have

$$pe^{-\beta J}e^{6\beta J} = p(1-p)^6 e^{-\beta J}e^{-6\beta J}, \tag{15.81}$$

or $(1-p)^6 = e^{-12\beta J}$. It is straightforward to solve for p and obtain the relation (15.80).

Now that we know how to generate clusters of spins, we can use these clusters to construct a global dynamics instead of only flipping one spin at a time as in the Metropolis algorithm. The idea is to grow a single (site-bond) percolation cluster in a way that is analogous to the single (site) percolation cluster algorithm discussed in Section 13.1. The algorithm can be implemented by the following steps:

1. Choose a seed spin at random. Its four nearest neighbor sites (on the square lattice) are the perimeter sites. Form an ordered array corresponding to the perimeter spins that are parallel to the seed spin and define a counter for the total number of perimeter spins.

2. Choose the first spin in the ordered perimeter array. Remove it from the array and replace it by the last spin in the array. Generate a random number r. If $r \leq p$, a bond exists and the spin is added to the cluster.

3. If the spin is added to the cluster, inspect its parallel perimeter spins. If any of these spins are not already part of the cluster, add them to the end of the array of perimeter spins.

4. Repeat steps 2 and 3 until no perimeter spins remain.

5. Flip all the spins in the single cluster.

This algorithm is known as single cluster flip or *Wolff* dynamics. Note that bonds rather than sites are tested so that a spin might have more than one chance to join a cluster. In the following, we consider both the static and dynamical properties of the two-dimensional Ising model using the Wolff algorithm to generate the configurations.

a. Modify your program for the two-dimensional Ising model on a square lattice so that single cluster flip dynamics (the Wolff algorithm) is used. Compute the mean energy and magnetization for $L = 16$ as a function of T for $T = 2.0$ to 2.7 in steps of 0.1. Compare your results to those obtained using the Metropolis algorithm. How many cluster flips do you need to obtain comparable accuracy at each temperature? Is the Wolff algorithm more efficient at every temperature near T_c?

b. Fix T at the critical temperature of the infinite lattice ($T_c = 2/\ln(1+\sqrt{2})$) and use finite size scaling to estimate the values of the various static critical exponents, for example, γ and α. Compare your results to those obtained using the Metropolis algorithm.

c. Because we are generating site-bond percolation clusters, we can study their geometrical properties as we did for site percolation. For example, measure the distribution sn_s of cluster sizes at $p = p_c$ (see Problem 13.3a). How does n_s depend on s for large s (see Project 13.18)? What is the fractal dimension of the clusters in the Ising model?

d. The natural unit of time for single cluster flip dynamics is the number of cluster flips $t_{\rm cf}$. Measure $C_M(t_{\rm cf})$ and/or $C_E(t_{\rm cf})$ and estimate the corresponding correlation time $\tau_{\rm cf}$ for $T = 2.5, 2.4$, 2.3, and T_c for $L = 16$. As discussed in Problem 15.19, $\tau_{\rm cf}$ can be found from the relation, $\tau_{\rm cf} = \sum_{t_{\rm cf}=1} C(t_{\rm cf})$. The sum is cut off at the first negative value of $C(t_{\rm cf})$. To compare our results for the Wolff algorithm to our results for the Metropolis algorithm, we should use the same unit of time. Because only a fraction of the spins are updated at each cluster flip, the time $t_{\rm cf}$ is not equal to the usual unit of time, which corresponds to an update of the entire lattice or one Monte Carlo step per spin. We have that τ measured in Monte Carlo steps per spin is related to $\tau_{\rm cf}$ by $\tau = \tau_{\rm cf}\langle c\rangle/L^2$, where $\langle c\rangle$ is the mean number of spins in the single clusters, and L^2 is the number of spins in the entire lattice. Measure $\langle c\rangle$ and compare your values for τ for the Wolff algorithm to the values of τ that you obtained using the Metropolis algorithm. Which values of τ are smaller?

e. Use the finite size scaling relation $\tau_{\rm cf} \sim L^{z_{\rm cf}}$ at $T = T_c$ to estimate $z_{\rm cf}$. Because $z_{\rm cf}$ is small, you will find it very difficult to obtain a good estimate. Verify that the mean cluster size scales as $\langle c\rangle \sim L^{\gamma/\nu}$ with $\gamma = 7/4$ and $\nu = 1$. (Note that these exponents are identical to the

analogous thermal exponents.) To obtain the value of z that is directly comparable to the value found for the Metropolis algorithm, we need to rescale the time as in part (d). We have that $\tau \sim L^z \propto L^{z_{\rm cf}} L^{\gamma/\nu} L^{-d}$. Hence, z is related to the measured value of $z_{\rm cf}$ by $z = z_{\rm cf} - (d - \gamma/\nu)$. It has been estimated that $z_{\rm cf} \approx 0.50$ for the $d = 2$ Ising model, which would imply that $z = 0.25$.

f. One of the limitations of the usual implementation of the Metropolis algorithm is that only one spin is flipped at a time. However, there is no reason why we could not choose f spins at random, compute the change in energy ΔE for flipping all f spins, and accepting or rejecting the trial move in the usual way. Explain why this generalization of the Metropolis algorithm would not be more efficient, especially if $f >> 1$.

Another cluster algorithm is to assign all bonds between parallel spins with probability p. As usual, no bonds are included between sites that have different spin values. From this configuration of bonds, we can form clusters of spins using the Hoshen-Kopelman algorithm (see Section 12.3). The smallest cluster contains a single spin. After the clusters have been identified, all the spins in each cluster are flipped with probability 1/2. This algorithm is known as the *Swendsen-Wang* algorithm and preceded the Wolff algorithm. Because the Wolff algorithm is easier to program and gives a smaller value of z than the Swendsen-Wang algorithm for the $d = 3$ and $d = 4$ Ising models, the Wolff algorithm is more commonly used.

Project 15.35. Invaded cluster algorithm

Project 15.36. Physical test of random number generators

In Section 7.9 we discussed various statistical tests for the quality of random number generators. In this project we will find that the usual statistical tests might not be sufficient for determining the quality of a random number generator for a particular application. The difficulty is that the quality of a random number generator for a specific application depends in part on how the subtle correlations that are intrinsic to all deterministic random number generators couple to the way that the random number sequences are used. In this project we explore the quality of two random number generators when they are used to implement single spin flip dynamics (Metropolis algorithm) and single cluster flip dynamics (Wolff algorithm) for the two-dimensional Ising model.

a. Write methods to generate sequences of random numbers based on the linear congruential algorithm

$$x_n = 16807\, x_{n-1} \bmod (2^{31} - 1) \tag{15.82}$$

and the generalized feedback shift register (GFSR) algorithm

$$x_n = x_{n-103} \oplus x_{n-250}. \tag{15.83}$$

In both cases x_n is the nth random number. Both algorithms require that x_n be divided by the largest possible value of x_n to obtain numbers in the range $0 \leq x_n < 1$. The GFSR algorithm requires bit manipulation. Which random number generator does a better job of passing the various statistical tests discussed in Problem 7.35?

b. Use the Metropolis algorithm and the linear congruential random number generator to determine the mean energy per spin E/N and the specific heat (per spin) C for the $L = 16$ Ising model at $T = T_c = 2/\ln(1+\sqrt{2})$. Make ten independent runs (that is, ten runs that use different random number seeds), and compute the standard deviation of the means σ_m from the ten values of E/N and C, respectively. Published results by Ferrenberg, Landau, and Wong are for 10^6 Monte Carlo steps per spin for each run. Calculate the differences δ_e and δ_c between the average of E/N and C over the ten runs and the exact values (to five decimal places) $E/N = -1.45306$ and $C = 1.49871$. If the ratio δ/σ_m for the two quantities is order unity, then the random number generator does not appear to be biased. Repeat your runs using the GFSR algorithm to generate the random number sequences. Do you find any evidence of statistical bias?

c. Repeat part (b) using Wolff dynamics. Do you find any evidence of statistical bias?

d. Repeat the computations in parts (proj:mc/rantest.b) and (proj:mc/rantest.c) using the random number generator supplied with your programming language.

Project 15.37. Nucleation and the Ising model

a. Equilibrate the two-dimensional Ising model at $T = 4T_c/9$ and $h = 0.3$ for a system with $L \geq 50$. What is the equilibrium value of m? Then flip the magnetic field so that it points down, that is, $h = -0.3$. Use the Metropolis algorithm and plot m as a function of the time t (the number of Monte Carlo steps per spin). What is the qualitative behavior of $m(t)$? Does it fluctuate about a positive value for a time long enough to determine various averages? If so, the system can be considered to have been in a *metastable state*. Watch the spins evolve for a time before m changes sign. Visually determine a place in the lattice where a "droplet" of the stable phase (down spins) first appears and then grows. Change the random number seed and rerun the simulation. Does the droplet appear in the same spot at the same time? Can the magnitude of the field be increased further or is there an upper bound above which a metastable state is not well defined?

b. As discussed in Project 15.34, we can define clusters of spins by placing a bond with probability p given by (15.80) between parallel spins. Use the Hoshen-Kopelman algorithm to find all the clusters of down spins. One way to identify the nucleating droplet is to monitor the number of spins in the largest cluster as a function of time after the quench. At what time does the number of spins in the largest cluster begin to grow quickly? This time is an estimate of the *nucleation time*. Another way of estimating the nucleation time is to follow the evolution of the center of mass of the largest cluster. For early times after the quench, the center of mass position has large fluctuations. However, at a certain time, these fluctuations decrease considerably, which is another possible criterion for the nucleation time.

c. In parts (a) and (b), you found that while the system is in a metastable state, clusters of down spins grow and shrink randomly until eventually one of the clusters becomes large enough to grow, nucleation occurs, and the system decays to its stable macroscopic state. The cluster that initiates this decay is called the *nucleating droplet*. This type of nucleation is due to spontaneous thermal fluctuations and is called *homogeneous nucleation*. Although the criteria for the nucleation time that we used in part (b) are plausible, they are not based on fundamental

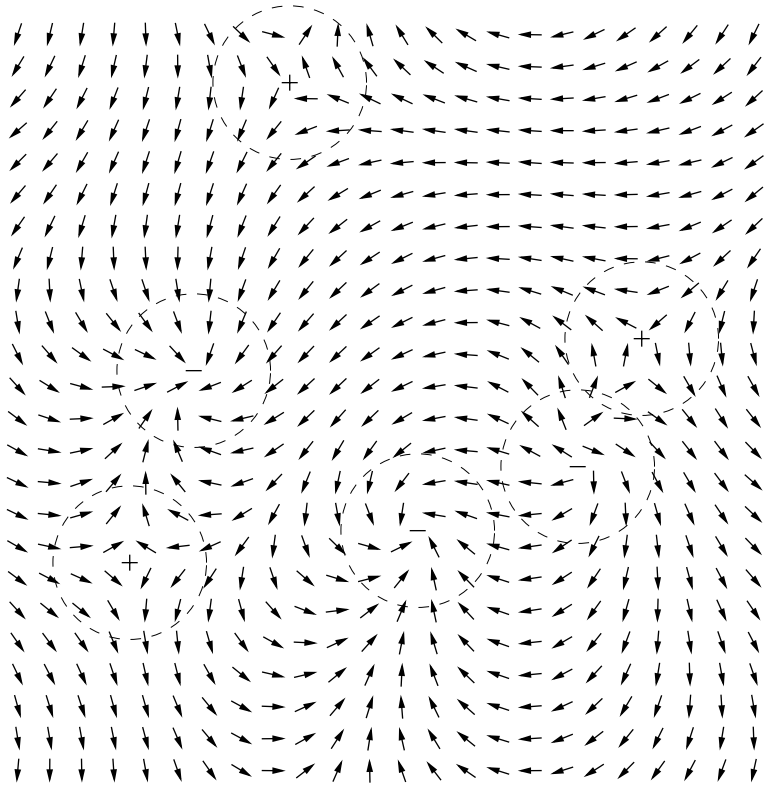

Figure 15.10: A typical configuration of the planar model on a 24×24 square lattice that has been quenched from $T = \infty$ to $T = 0$ and equilibrated for 200 Monte Carlo steps per spin after the quench. Note that there are six vortices. The circle around each vortex is a guide to the eye and is not meant to indicate the size of the vortex.

considerations. From theoretical considerations the nucleating droplet can be thought of as a cluster that just makes it to the top of the saddle point of the free energy that separates the metastable and stable states. We can identify the nucleating droplet by using the fact that a saddle point structure should initiate the decay of the metastable state 50% of the time. The idea is to save the spin configurations at regular intervals about the time that nucleation is thought to have occurred. We then restart the simulation using a saved configuration and a different random number seed each time. If we have intervened at a time such that the largest cluster decays in more than 50% of the trials, then the intervention time is before nucleation. Similarly, if less than 50% of the clusters decay, the intervention is after the nucleation time. The nucleating droplet is the cluster that decays in approximately half of the trial interventions. Because we need to do a number of interventions (usually in the range 20–100) at different times, the intervention method is much more cpu intensive than the other criteria. However, it has the important advantage that it has a sound theoretical basis. Redo some of the simulations that you did in part (b) and compare the different estimates of the nucleating time. What is

the nature of the nucleating droplet? Is it compact? If time permits, determine the probability that the system nucleates at time t for a given quench depth. What is the mean number of spins in the nucleating droplet?

d. *Heterogeneous nucleation* occurs in nature because of the presence of impurities, defects, or walls. One way of simulating heterogeneous nucleation in the Ising model is to fix a certain number of spins in the direction of the stable phase (down). For simplicity, choose the impurity to be five spins in the shape of a + sign. What is the effect of the impurity on the lifetime of the metastable state? What is the probability of droplet growth at and off the impurity as a function of quench depth h?

e. The questions raised in parts (b)–(d) become even more interesting when the interaction between the spins extends beyond nearest neighbors. Assume that a given spin interacts with all spins that are within a distance R with an interaction strength of $4J/q$, where q is the number of spins within the interaction range R. (Note that $q = 4$ for nearest neighbor interactions on the square lattice.) A good choice is $R = 10$, although your preliminary simulations should be for smaller R How does the value of T_c change as R is increased?

Project 15.38. The n-fold way: Simulations at low temperature

At low temperatures Monte Carlo simulations can become very inefficient because almost all new configurations will be rejected. For example, consider an Ising model simulation where all the spins are up, but a small negative magnetic field is applied. The equilibrium state will be a configuration with most spins pointing down. Nevertheless, if the magnetic field is small enough and the temperature low enough, this equilibrium state will take a very long time to occur if it occurs at all on a typical computer.

What we need is a better way of sampling configurations if the acceptance probability is low. The *n-fold way* algorithm is one such method. The basic idea is to include more low probability configurations, but weight them appropriately. If we use the usual Metropolis rule, then the probability of flipping the ith spin is

$$p_i = \min\big[1, e^{-\Delta E/kT}\big]. \tag{15.84}$$

If we sum over all the spins, then we can define the total weight

$$Q = \sum_i p_i \tag{15.85}$$

We can then choose which spin to flip by computing a random number, r_Q, between 0 and Q and choosing the spin to flip by the criteria

$$\sum_{k=0}^{i-1} p_k \leq r_Q < \sum_{k=0}^{i} p_k. \tag{15.86}$$

There are two more ingredients we need to make this algorithm work. First, we need to know how long a spin configuration will remain unchanged before a spin flip, and second the above algorithm is very inefficient because on average the computation of which spin to flip from (15.86) will take

$O(N)$ computations, which is too long. This second problem can be easily overcome by realizing that there are only a few different possible values for p_i. For example, for the two-dimensional Ising model on a square lattice in a magnetic field, there are only ten possible values of p_i. The number "10" is the n in the n-fold way algorithm. Thus, instead of (15.86) we have

$$\sum_{k=0}^{m-1} n_m p_m \leq r_Q < \sum_{k=0}^{m} n_m p_m, \tag{15.87}$$

where m labels one of the n possible values of p_i or classes, and n_m is the number of spins in the mth class. Hence, instead of $O(N)$ calculations, we need only perform $O(n)$ calculations. Once we know which class to use, we can randomly flip one of the spins in that class.

Next we need to determine how to compute the time spent in a configuration. The probability in one Metropolis MC step of picking a spin randomly is $1/N$ and the probability of then actually flipping that spin is p_i, which is given by (15.84). Thus, the probability of flipping any spin is

$$\frac{1}{N}\sum_{i=0}^{N-1} p_i = \frac{1}{N}\sum_{m=0}^{n-1} n_m p_m = \frac{Q}{N}, \tag{15.88}$$

Clearly, the probability of not flipping any spin is $1 - Q/N \equiv q$, and the probability of not flipping after s steps is q^s. Thus, if we generate a random number r between 0 and 1, the time in MC steps s to stay in the current configuration will be determined by solving

$$q^{s-1} \leq r < q^s. \tag{15.89}$$

If $Q/N << 1$, then both sides of (15.89) are approximately equal, and we can approximate s by

$$s \approx \frac{\ln r}{\ln(1 - Q/N)} \approx -\frac{N}{Q}\ln r. \tag{15.90}$$

Thus, the n-fold algorithm is given by the following steps:

1. Start with an initial configuration and determine which class each spin is in. Store all the possible values of p_i in an array. Compute Q. Store in an array the values of the number of spins in class m, n_m.
2. Determine s from (15.90). Accumulate any averages such as energy and magnetization weighted by s. Also, accumulate the total time `sTotal += s`.
3. Choose a class of spin using (15.87) and randomly choose which spin in the chosen class to flip.
4. Update the classes of the chosen spin and its four neighbors.
5. Repeat steps 2–4.

To conveniently carry out step 4 you can set up the following arrays: an array `spinClass[i]` which returns the class of the ith spin, `spinInClass[k][m]` which returns the kth spin in the

mth class, and `spinIndex[i]` which returns the value of k for the ith spin to use in the array `spinInClass[k][m]`. If we define the local field of a spin by the sum of the fields of its four neighbors, then this local field can take on the following values $\{-4, -2, 0, 2, 4\}$. The ten classes correspond to these five local field values and the center spin equal to -1 plus these five local field values and the center spin equal to $+1$. If we label these ten classes from 0 to 9 in order as in the above, then the class of a spin which is flipped changes by $+5 \mod 10$ and the class of a neighbor changes by the new spin value equal to ± 1.

a. Write a program to implement the n-fold way algorithm for the Ising model on a square lattice with an applied magnetic field. Check your program by comparing thermodynamic averages at a few temperatures with the results from a program using the standard Metropolis algorithm.

b. Use the magnetic field $h = -0.5$ at a temperature, $T = 1$. Begin with an initial configuration of all spins up, and use the n-fold way algorithm to estimate how long it takes before the majority of the spins flip. Do the same simulation using the standard Metropolis algorithm. Which algorithm works best?

c. Repeat part (b) for other temperatures and field values. Explain your results by considering the probabilities to flip individual spins at low temperature.

d. Repeat part (b) for different values of the magnetic field and plot the number of MC steps needed to flip the spins as a function of $1/|h|$, for values of h from 0 to about 3.0. Average over at least 10 starting configurations at each field value.

Project 15.39. The Kosterlitz-Thouless transition

The planar model (also called the x-y model) consists of spins of unit magnitude that can point in any direction in the x-y plane. The energy or Hamiltonian function of the planar model in zero magnetic field can be written as

$$E = -J \sum_{i,j=nn(i)} [s_{i,x} s_{j,x} + s_{i,y} s_{j,y}], \tag{15.91}$$

where $s_{i,x}$ represents the x-component of the spin at the ith site, J measures the strength of the interaction, and the sum is over all nearest neighbors. We can rewrite (15.91) in a simpler form by substituting $s_{i,x} = \cos\theta_i$ and $s_{i,y} = \sin\theta_i$. The result is

$$E = -J \sum_{i,j=nn(i)} \cos(\theta_i - \theta_j), \tag{15.92}$$

where θ_i is the angle that the ith spin makes with the x axis. The most studied case is the two-dimensional model on a square lattice. In this case the mean magnetization $\langle \mathbf{M} \rangle = 0$ for all temperatures $T > 0$, but nevertheless, there is a phase transition at a nonzero temperature, T_{KT}, which is known as the Kosterlitz-Thouless (KT) transition. For $T \leq T_{KT}$, the spin-spin correlation function $C(r)$ decreases as a power law for increasing r; for $T > T_{KT}$, $C(r)$ decreases exponentially. The power law decay of $C(r)$ for all $T \leq T_{KT}$ implies that every temperature below T_{KT} acts as if it were a critical point. We say that the planar model has a line of critical points. In the following, we explore some of the properties of the planar model and the mechanism that causes the transition.

a. Write a Monte Carlo program that uses the Metropolis algorithm to simulate the planar model on a square lattice using periodic boundary conditions. Because θ and hence the energy of the system is a continuous variable, it is not possible to store the previously computed values of the Boltzmann factor for each possible value of ΔE. Instead, of computing $e^{-\beta\Delta E}$ for each trial change, it is faster to set up an array `w` such that the array element `w(j)` $= e^{-\beta\Delta E}$, where `j` is the integer part of $1000\Delta E$. This procedure leads to an energy resolution of 0.001, which should be sufficient for most purposes.

b. One way to show that the magnetization $\langle \mathbf{M} \rangle$ vanishes for all T is to compute $\langle \theta^2 \rangle$, where θ is the angle that a spin makes with the magnetization $\mathbf{M}$ at any given instant. (Although the mean magnetization vanishes, $\mathbf{M} \neq 0$ at any given instant.) Compute $\langle \theta^2 \rangle$ as a function of the number of spins N at $T = 0.1$, and show that $\langle \theta^2 \rangle$ diverges as $\ln N$. Begin with a 4×4 lattice and choose the maximum change in θ_i to be $\Delta\theta_{\max} = 1.0$. If necessary, change $\theta_{\max}$ so that the acceptance probability is about 40%. If $\langle \theta^2 \rangle$ diverges, then the spins are not pointing along any preferred direction, and hence there is no mean magnetization.

c. Modify your program so that an arrow is drawn at each site to show the orientation of each spin. We will look at a typical configuration and analyze it visually. Begin with a 32×32 lattice with spins pointing in random directions and do a temperature quench from $T = \infty$ to $T = 0.5$. (Simply change the value of β in the Boltzmann probability.) Such a quench should lock in some long lived, but metastable vortices. A vortex is a region of the lattice where the spins rotate by at least 2π as your eye moves around a closed path (see Fig. 15.10). To determine the center of a vortex, choose a group of four spins that are at the corners of a unit square, and determine whether the spins turn by $\pm 2\pi$ as your eye goes from one spin to the next in a counterclockwise direction around the square. Assume that the difference between the direction of two neighboring spins, $\delta\theta$, is in the range $-\pi < \delta\theta < \pi$. A total rotation of $+2\pi$ indicates the existence of a positive vortex, and a change of -2π indicates a negative vortex. Count the number of positive and negative vortices. Repeat these observations on several configurations. What can you say about the number of vortices of each sign?

d. Write a method to determine the existence of a vortex for each 1×1 square of the lattice. Represent the center of the vortices using a different symbol to distinguish between a positive and a negative vortex. Do a Monte Carlo simulation to compute the mean energy, specific heat, and number of vortices in the range from $T = 0.5$ to $T = 1.5$ in steps of 0.1. Use the last configuration at the previous temperature as the first configuration for the next temperature. Begin at $T = 0.5$ with all $\theta_i = 0$. Draw the vortex locations for the last configuration at each temperature. Use at least 1000 Monte Carlo steps per spin at each temperature to equilibrate and at least 5000 Monte Carlo steps per spin for computing the averages. Use an 8×8 or 16×16 lattice if your computer resources are limited, and larger lattices if you have sufficient resources. Describe the T dependence of the energy, specific heat, and vorticity (equal to the number of vortices per area). Plot the logarithm of the vorticity versus T for $T < 1.1$. What can you conclude about the T-dependence of the vorticity? Explain why this form is reasonable. Describe the vortex configurations. At what temperature can you find a vortex that appears to be free, that is, a vortex that is not obviously paired up with another vortex of opposite sign?

e. The Kosterlitz-Thouless theory predicts that the susceptibility χ diverges above the transition

as

$$\chi \sim A\, e^{b/\epsilon^{\nu}}, \tag{15.93}$$

where ϵ is the reduced temperature $\epsilon = (T - T_{KT})/T_{KT}$, $\nu = 0.5$, and A and b are nonuniversal constants. Compute χ from the relation (15.20) with $\mathbf{M} = 0$ because the mean magnetization vanishes. Assume the exponential form (15.93) for χ in the range $T = 1$ and $T = 1.2$ with $\nu = 0.7$, and find the best values of T_{KT}, A, and b. (Although the analytical theory predicts $\nu = 0.5$, simulations for small systems indicate that $\nu = 0.7$ gives a better fit.) One way to determine T_{KT}, A, and b is to assume a value of T_{KT} and then do a least squares fit of $\ln \chi$ to determine A and b. Choose the set of parameters that minimizes the variance of $\ln \chi$. How does your estimated value of T_{KT} compare with the temperature at which free vortices first appear? At what temperature does the specific heat have a peak? The Kosterlitz-Thouless theory predicts that the specific heat peak does not occur at T_{KT}. This result has been confirmed by simulations (see Tobochnik and Chester). To obtain quantitative results, you will need lattices larger than 32×32.

Project 15.40. Classical Heisenberg model in two dimensions

The energy or Hamiltonian of the classical Heisenberg model is similar to the Ising model and the planar model, except that the spins can point in any direction in three dimensions. The energy in zero external magnetic field is

$$E = -J \sum_{i,j=\mathrm{nn}(i)}^{N} \mathbf{s}_i \cdot \mathbf{s}_j = -J \sum_{i,j=\mathrm{nn}(i)}^{N} [s_{i,x}s_{j,x} + s_{i,y}s_{j,y} + s_{i,z}s_{j,z}], \tag{15.94}$$

where $\mathbf{s}$ is a classical vector of unit length. The spins have three components, in contrast to the spins in the Ising model which only have one component, and the spins in the planar model which have two components. We will consider the two-dimensional Heisenberg model for which the spins are located on a two-dimensional lattice.

Early simulations and approximate theories led researchers to believe that there was a continuous phase transition, similar to that found in the Ising model. The Heisenberg model received more interest after it was related to the confinement for quarks. Lattice models of the interaction between quarks, called lattice gauge theories, predict that the confinement of quarks can be explained if there are no phase transitions in these models. (The lack of a phase transition in these models implies that the attraction between quarks grows with distance.) The Heisenberg model is a two-dimensional analog of the four-dimensional models used to model quark-quark interactions. Shenker and Tobochnik used a combination of Monte Carlo and renormalization group methods to show that this model does not have a phase transition. Subsequent work on lattice gauge theories showed similar behavior.

a. Modify your Ising model program to simulate the Heisenberg model in two dimensions. One way to do so is to define three arrays, one for each of the three components of the unit spin vectors. A trial Monte Carlo move consists of randomly changing the direction of a spin, $\mathbf{s}_i$. First compute a small vector $\Delta\mathbf{s} = \Delta s_{\max}(p_1, p_2, p_3)$, where $-1 \le p_n \le 1$ is a uniform random number, and $\Delta s_{\max}$ is the maximum change of any spin component. If $|\Delta\mathbf{s}| > \Delta s_{\max}$, than compute another

$\Delta\mathbf{s}$. This latter step is necessary to insure that the change in a spin direction is symmetrically distributed around the current spin direction. Next let the trial spin equal $\mathbf{s}_i + \Delta\mathbf{s}$ normalized to a unit vector. The standard Metropolis algorithm can now be used to determine if the trial spin is accepted. Compute the mean energy, specific heat, and susceptibility as a function of T. Choose lattice sizes of $L = 8$, 16, 32 and larger if possible and average over at least 2000 Monte Carlo steps per spin at each temperature. Is there any evidence of a phase transition? Does the susceptibility appear to diverge at a nonzero temperature? Plot the logarithm of the susceptibility versus the inverse temperature, and determine the temperature dependence of the susceptibility in the limit of low temperatures.

b. Use the Lee-Kosterlitz analysis at the specific heat peak to determine if there is a phase transition.

Project 15.41. Domain growth kinetics

When a system is quenched from a high temperature to very low temperatures, domains of the ordered low temperature state typically grow with time as a power law, $R \sim t^{\alpha}$, where R is a measure of the average linear dimension of the domains. A simple measure of the domain size is the perimeter length of a domain which can be computed from the energy per spin, ϵ, and is given by

$$R = \frac{2}{2+\epsilon}. \tag{15.95}$$

Note that (15.95) equals 1 for $\epsilon = 0$ and ∞ for $\epsilon = -2$ as desired. Equation (15.95) comes from the following argument. Imagine a region of N spins made up of a domain of up spins with a perimeter size, R, embedded in a sea of down spins. The total energy of this region is $-2N + 2R$, where for each spin on the perimeter, the energy is increased by 2 because one of the neighbors of a perimeter spin will be of opposite sign. The energy per spin is $\epsilon = -2 + 2R/N$. Because N is of order R^2, we arrive at the result given in (15.95).

a. Modify your Ising model class so that the initial configuration is random, that is, a typical high temperature configuration. Write an application class to simulate a quench of the system. The input parameters should be the lattice size, the quench temperature (use 0.5 initially), the maximum time (measured in Monte Carlo steps per spin) for each trial quench, and to speed up the simulation, the number of MC steps between drawing the lattice. Have your program plot $\ln\langle r\rangle$ versus $\ln t$ after each quench is finished.

b. Run your program for $L = 64$ and a maximum time of 128 mcs. Averages over 10 quenches will give adequate results. What value do you obtain for α? Repeat for other temperatures and system sizes. Does the exponent change? Run for a longer maximum time to check your results.

c. Modify your program to simulate the q-state Potts model. Consider various values of Q. Do your results change? Results for large q and large system sizes are given in the reference by Grest et al.

d.* Modify your program to simulate a three-dimensional system. How must you modify (15.95)? Are your results still the same?

Project 15.42. Ground state energy of the Ising spin glass

A spin glass is a magnetic system with frozen-in disorder. An example of such a system is the Ising model with the exchange constant J_{ij} between nearest neighbor spins randomly chosen to be ± 1. The disorder is said to be "frozen" because the set of interactions $\{J_{ij}\}$ does not change with time. Because the spins cannot arrange themselves so that every pair of spins is in its lowest energy state, the system exhibits frustration similar to the antiferromagnetic Ising model on a triangular lattice (see Problem 15.22). Is there a phase transition in the spin glass model, and if so, what is its nature? The answers to these questions are very difficult to obtain by doing simulations. One of the difficulties is that we need to do not only an average over the possible configurations of spins for a given set of $\{J_{ij}\}$, but we also need to average over different realizations of the interactions. Another difficulty is that there are many local minima in the energy (free energy at finite temperature) as a function of the configurations of spins, and it is very difficult to find the global minimum. As a result, Monte Carlo simulations typically become stuck in these local minima or metastable states. Detailed finite size scaling analyses of simulations indicate that there might be a transition in three dimensions. It is generally accepted that the transition in two dimensions is at zero temperature. In the following, we will look at some of the properties of an Ising spin glass on a square lattice at low temperatures.

a. Write a program to apply simulated annealing to an Ising spin glass using the Metropolis algorithm with the temperature fixed at each stage of the annealing schedule (see Problem 15.32a). Search for the lowest energy configuration for a fixed set of $\{J_{ij}\}$. Use at least one other annealing schedule for the same $\{J_{ij}\}$ and compare your results. Then find the ground state energy for at least ten other sets of $\{J_{ij}\}$. Use lattice sizes of $L = 5$ and $L = 10$. Discuss the nature of the ground states you are able to find. Is there much variation in the ground state energy E_0 from one set of $\{J_{ij}\}$ to another? Theoretical calculations give an average over realizations of $\overline{E_0}/N \approx -1.4$. If you have sufficient computer resources, repeat your computations for the three-dimensional spin glass.

b. Modify your program to do simulated annealing using the demon algorithm (see Problem 15.32b). How do your results compare to those that you found in part (a)?

Project 15.43. Zero temperature dynamics of the Ising model

We have seen that various kinetic growth models (Section 13.3) and reaction-diffusion models (Section 7.8) lead to interesting and nontrivial behavior. Similar behavior can be seen in the zero temperature dynamics of the Ising model. Consider the one-dimensional Ising model with $J > 0$ and periodic boundary conditions. The initial orientation of the spins is chosen at random. We update the configurations by choosing a spin at random and computing the change in energy ΔE. If $\Delta E < 0$, then flip the spin; else if $\Delta E = 0$, flip the spin with 50% probability. The spin is not flipped if $\Delta E > 0$. This type of Monte Carlo update is known as Glauber dynamics. How does this algorithm differ from the Metropolis algorithm at $T = 0$?

a. One quantity of interest is $f(t)$, the fraction of spins that have not flipped at time t. As usual, the time is measured in terms of Monte Carlo steps per spin. Published results (Derrida, Bray, and Godrèche) for $N = 10^5$ indicate that $f(t)$

$$f(t) \sim t^{-\theta} \tag{15.96}$$

for $t \approx 3$ to $t \approx 10,000$. The exact value of θ is 0.375. Verify this result and extend your results to the one-dimensional q-state Potts model. In the latter model each site is initially given a random integer between 1 and q. A site is chosen at random and set equal to either of its two neighbors with equal probability.

b. Another interesting quantity is the probability distribution, $P_n(t)$, that n sites have not flipped as a function of the time t (see Das and Sen). First, plot P_n for two times on the same plot. Discuss the shape of the curve and the difference between the two times. Use at least $L = 100$ and times 50 and 100. Try to fit the curves to a Gaussian distribution. Because the possible values of n are bounded, we can fit each side of the maximum of a curve to a Gaussian with different widths. There are a number of scaling properties that can be investigated. The simplest is to show that $P_{n=0}(t)$ scales approximately as t/L^2. Thus, if you compute $P_{n=0}(t)$ for a number of different times and lengths such that t/L^2 has the same value, you should obtain the same value of $P_{n=0}$.

Project 15.44. The inverse power law potential

Consider the inverse power law potential

$$V(r) = V_0 \, (\frac{\sigma}{r})^n \tag{15.97}$$

with $V_0 > 0$. One reason for interest in potentials of this form is that thermodynamic quantities such as the mean energy E do not depend on V_0 and σ separately, but depend on a single dimensionless parameter. This dimensionless parameter can be defined as

$$\Gamma = \frac{V_0}{kT} \frac{\sigma}{a}, \tag{15.98}$$

where a is defined in three and two dimensions by $4\pi a^3 \rho/3 = 1$ and $\pi a^2 \rho = 1$, respectively. The length a is proportional to the mean distance between particles. A Coulomb interaction corresponds to $n = 1$, and a hard sphere system corresponds to $n \to \infty$. What phases do you expect to occur for arbitrary n?

a. Compare the qualitative features of $g(r)$ for a "soft" potential with $n = 4$ to a system of hard disks at the same density.

b. Let $n = 12$ and compute the mean energy E as a function of T for fixed density for a three-dimensional system. Fix T and consider $N = 16$, 32, 64, and 128. Does E depend on N? Can you extrapolate your results for the N-dependence of E to $N \to \infty$? Fix N and determine E as a function of Γ. Do you see any evidence of a phase transition? If so, estimate the value of Γ at which it occurs. What is the nature of the transition if it exists?

Project 15.45. Rare gas clusters

There has been much recent interest in structures that contain many particles, but that are not macroscopic. An example is the unusual structure of sixty carbon atoms known as a "buckeyball." A less unusual structure is a cluster of argon atoms. Questions of interest include the structure of

the clusters, the existence of "magic" numbers of particles for which the cluster is particularly stable, the temperature dependence of the thermodynamic quantities, and the possibility of different phases. This latter question has been subject to some controversy, because transitions between different kinds of behavior in finite systems are not well defined as they are for infinite systems.

a. Write a Monte Carlo program to simulate a three-dimensional system of particles interacting via the Lennard-Jones potential. Use open boundary conditions, that is, do not enclose the system in a box. The number of particles N and the temperature T should be input parameters.

b. Find the ground state energy E_0 as a function of N. For each value of N begin with a random initial configuration and accept any trial displacement that lowers the energy. Repeat for at least ten different initial configurations. Plot E_0/N versus N for $N = 2$ to 20 and describe the qualitative dependence of E_0/N on N. Is there any evidence of magic numbers, that is, value(s) of N for which E_0/N is a minimum? For each value of N save the final configuration. Plot the positions of the atoms. Does the cluster look like a part of a crystalline solid?

c. Repeat part (b) using simulated annealing. The initial temperature should be sufficiently low so that the particles do not move far away from each other. Slowly lower the temperature according to some annealing schedule. Do your results for E_0/N differ from part (b)?

d. To gain more insight into the structure of the clusters, compute the mean number of neighbors per particle for each value of N. What is a reasonable criteria for two particles to be neighbors? Also compute the mean distance between each pair of particles. Plot both quantities as a function of N, and compare their dependence on N with your plot of E_0/N.

e. Is it possible to find any evidence for a "melting" transition? Begin with the configuration that has the minimum value of E_0/N and slowly increase the temperature T. Compute the energy per particle and the mean square displacement of the particles from their initial positions. Plot your results for these quantities versus T.

Project 15.46. Hard disks

Although we have mentioned (see Section 15.10) that there is reasonable evidence for a transition in a hard disk system, the nature of the transition still is a problem of current research. In this project we follow the work of Lee and Strandburg and apply the constant pressure Monte Carlo method (see Section 15.12) and the Lee-Kosterlitz method (see Section 15.11) to investigate the nature of the transition. Consider $N = L^2$ hard disks of diameter $\sigma = 1$ in a two-dimensional box of volume $V = \sqrt{3}L^2 v/2$ with periodic boundary conditions. The quantity $v \geq 1$ is the reduced volume and is related to the density ρ by $\rho = N/V = 2/(\sqrt{3}v)$; $v = 1$ corresponds to maximum packing. The aspect ratio of $2/\sqrt{3}$ is used to match the perfect triangular lattice. We can perform a constant pressure (actually constant $p^* = P/kT$) Monte Carlo simulation as follows. The trial displacement of each disk is implemented as discussed in Section 15.10. Lee and Strandburg find that a maximum displacement of 0.09 gives a 45% acceptance probability. The other type of move is a random isotropic change of the volume of the system. If the change of the volume leads to an overlap of the disks, the change is rejected. Otherwise, if the trial volume $\tilde{V}$ is less than the current volume V, the change is accepted. A larger trial volume is accepted with probability

$$e^{-p^*(\tilde{V}-V)+N\ln\tilde{V}/V)}. \tag{15.99}$$

Volume changes are attempted 40–200 times for each set of individual disk moves. The quantity of interest is $N(v)$, the distribution of reduced volume v. Because we need to store information about $N(v)$ in an array, it is convenient to discretize the volume in advance and choose the mesh size so that the acceptance probability for changing the volume by one unit is 40–50%. Do a Monte Carlo simulation of the hard disk system for $L = 10$ ($N = 100$) and $p^* = 7.30$. Published results are for 10^7 Monte Carlo steps. To apply the Lee-Kosterlitz method, smooth $\ln N(v)$ by fitting it to an eighth-order polynomial. Then extrapolate $\ln N(v)$ using the histogram method to determine $p_c^*(L = 10)$, the pressure at which the two peaks of $N(v)$ are of equal height. What is the value of the free energy barrier ΔF? If sufficient computer resources are available, compute ΔF for larger L (published results are for $L = 10$, 12, 14, 16, and 20) and determine if ΔF depends on L. Can you reach any conclusions about the nature of the transition?

Project 15.47. Heat flow using the demon algorithm

In our applications of the demon algorithm one demon shared its energy equally with all the spins. As a result the spins all attained the same mean energy of interaction. Many interesting questions arise when the system is not spatially uniform and is in a nonequilibrium but time-independent (steady) state.

Let us consider heat flow in a one-dimensional Ising model. Suppose that instead of all the sites sharing energy with one demon, each site has its own demon. We can study the flow of heat by requiring the demons at the boundary spins to satisfy different conditions than the demons at the other spins. The demon at spin 1 adds energy to the system by flipping this spin so that it is in its highest energy state, that is, in the opposite direction of spin 2. The demon at spin N removes energy from the system by flipping spin N so that it is in its lowest energy state, that is, in the same direction as spin $N - 1$. As a result, energy flows from site 1 to site N via the demons associated with the intermediate sites. In order that energy not build up at the "hot" end of the Ising chain, we require that spin 1 can only add energy to the system if spin N simultaneously removes energy from the system. Because the demons at the two ends of the lattice satisfy different conditions than the other demons, we do not use periodic boundary conditions.

The temperature is determined by the generalization of the relation (15.9), that is, the temperature at site i is related to the mean energy of the demon at site i. To control the temperature gradient, we can update the end spins at a rate different than the other spins. The maximum temperature gradient occurs if we update the end spins after every update of an internal spin. A smaller temperature gradient occurs if we update the end spins less frequently. The temperature gradient between any two spins can be determined from the temperature profile, the spatial dependence of the temperature. The energy flow can be determined by computing the magnitude of the energy per unit time that enters the lattice at site 1.

To implement this procedure we modify `IsingDemon` by converting the variables `demonEnergy` and `demonEnergyAccumulator` to arrays. We do the usual updating procedure for spins 1 through $N - 2$ and visit spins 0 and $N - 1$ at regular intervals denoted by `timeToAddEnergy`.

Listing 15.7: One-dimensional Ising model gas with many demons.

```
/* Simulation of thermal gradient using a demon at each site.
 left end can add energy, right end takes it out. */

package org.opensourcephysics.sip.ch15;
```

```
import java.awt.Color;
import java.awt.Graphics;
import org.opensourcephysics.display.Arrow;
import org.opensourcephysics.display.Drawable;
import org.opensourcephysics.display.DrawingPanel;

public class ManyDemons implements Drawable {
   public int [] s;
   public int N;
   public int systemEnergy;
   public int [] demonEnergy;
   public int mcs = 0; // number of MC moves per particle
   public double systemEnergyAccumulator = 0;
   public double [] demonEnergyAccumulator;
   public int magnetization = 0;
   public double [] mAccumulator;
   public int acceptedMoves = 0;
   public int timeToAddEnergy;

public void initialize () {
   s = new int[N];
   demonEnergy = new int[N];
   demonEnergyAccumulator = new double[N];
   mAccumulator = new double[N];
   for (int i = 0; i < N; ++i)
      s[i] = 1;  // all spins up
   int tries = 0;
   int E = -N;
   magnetization = N;
   while ((E < systemEnergy) && (tries < 10*N)) {
      int k = (int)(1 + Math.random()*(N-2));
      int dE = 2*s[k]*(s[k+1] + s[k-1]);
      if (dE > 0) {
         E += dE;
         s[k] = -s[k];
         magnetization += 2*s[k];
       }
       tries++;
   }
   System.out.println(" Initial energy = " + E);
   systemEnergy = E;
   resetData();
}

   public double temperature(int i) {
      return 4/Math.log(1.0 + 4.0/(demonEnergyAccumulator[i]/(mcs)));
   }

   public void resetData() {
      for (int i = 0; i < N; ++i) {
```

```
            demonEnergyAccumulator[i] = 0;
            mAccumulator[i] = 0;
        }
        mcs = 0;
        systemEnergyAccumulator = 0;
        acceptedMoves = 0;
    }

    public void step() {
        for (int j = 1; j < N-1; ++j) {
          int i = (int)(1 + Math.random()*(N-2));
          int dE = 2*s[i]*(s[i+1] + s[i-1]);
          if (dE <= demonEnergy[i]) {
              s[i] = -s[i];
              acceptedMoves++;
              systemEnergy += dE;
              demonEnergy[i] -= dE;
              magnetization += 2*s[i];
          }
          systemEnergyAccumulator += systemEnergy;
        }
        for (int j = 1; j < N-1; ++j) {
            demonEnergyAccumulator[j] += demonEnergy[j];
            mAccumulator[j] += s[j];
        }
        mcs++;
        if (mcs % timeToAddEnergy == 0) {
            boundarySpins();
        }
    }

    public void boundarySpins() {
        /* attempt to add energy at spin 0 and remove it at spin N
       when spins 0 and 1 are aligned and spin N-2 and N-1 are not aligned */
        if ((s[0]*s[1] == 1) && (s[N-2]*s[N-1] == -1)) {
            demonEnergyAccumulator[0] += 2;
            demonEnergyAccumulator[N-1] -= 2;
            s[0] = -s[0];
            s[N-1] = -s[N-1];
        }
    }

    public void draw (DrawingPanel myWorld, Graphics g) {
        if (s == null) {
            return;
        }
        g.setColor(Color.red);
        double theta = 2*Math.PI/N;
        for (int i = 1; i < N-1; i++) {
            Arrow spin = new Arrow(i,0,0,1.0*s[i]);
```

```
            spin.draw(myWorld,g);
        }
    }
}
```

a. Write a target class that inputs the number of spins, N, and the initial energy of the system, outputs the number of MC steps per spin and the energy added to the system at the hot boundary, and plots the temperature as a function of position.

b. As a check on `ManyDemons`, modify the class so that all the demons are equivalent, that is, impose periodic boundary conditions and do not use method `boundarySpins`. Compute the mean energy of the demon at each site and use (15.9) to define a local site temperature. Use $N \geq 52$ and run for about 10000 `mcs`. Is the local temperature approximately uniform? How do your results compare with the single demon case?

c. In `ManyDemons` the energy is added to the system at site 1 and is removed at site N. Determine the mean demon energy for each site and obtain the corresponding local temperature and the mean energy of the system. Draw the temperature profile by plotting the temperature as a function of site number. The temperature gradient is the difference in temperature from site $N-1$ to site 2 divided by the distance between them. (The distance between neighboring sites is unity.) Because of local temperature fluctuations and edge effects, the temperature gradient should be estimated by fitting the temperature profile in the middle of the lattice to a straight line. Reasonable choices for the parameters are $N = 52$ and `timeToAddEnergy` $= 1$. Run for at least 10000 mcs.

d. The heat flux Q is the energy flow per unit length per unit time. The energy flow is the amount of energy that demon 1 adds to the system at site 1. The time is conveniently measured in terms of Monte Carlo steps per spin. Determine Q for the parameters used in part (c).

e. If the temperature gradient $\partial T/\partial x$ is not too large, the heat flux Q is proportional to $\partial T/\partial x$. We can determine the *thermal conductivity* κ by the relation

$$Q = -\kappa \frac{\partial T}{\partial x}. \tag{15.100}$$

Use your results for $\partial T/\partial x$ and Q to estimate κ.

f. Determine Q, the temperature profile, and the mean temperature for different values of `timeToAddEnergy`. Is the temperature profile linear for all values of `timeToAddEnergy`? If the temperature profile is linear, estimate $\partial T/\partial x$ and determine κ. Does κ depend on the mean temperature?

Note that by using many demons we were able to compute a temperature profile by using an algorithm that manipulates only integer numbers. The conventional approach is to solve a heat equation similar in form to the diffusion equation. Now we use the same idea to compute the magnetization profile when the end spins of the lattice are fixed.

g. Modify `ManyDemons` by not calling `method boundarySpins` and constraining spins 1 and N to be $+1$ and -1 respectively. Estimate the magnetization profile by plotting the mean value of the spin at each site versus the site number. Choose $N = 22$ and `mcs` ≥ 1000. How do your results vary as you increase N?

h. Compute the mean demon energy and hence the local temperature at each site. Does the system have a uniform temperature even though the magnetization is not uniform? Is the system in thermal equilibrium?

i.* The effect of the constraint on the end spins is easier to observe in two and three dimensions than in one dimension. Write a program for a two-dimensional Ising model on a $L \times L$ square lattice. Constrain the spins at site (i, j) to be $+1$ and -1 for $i = 1$ and $i = L$ respectively. Use periodic boundary conditions in the y direction. How do your results compare with the one-dimensional case?

j.* Remove the periodic boundary condition in the y direction and constrain all the boundary spins from $i = 1$ to $L/2$ to be $+1$ and the other boundary spins to be -1. Choose an initial configuration where all the spins on the left half of the system are $+1$ and the others are -1. Do the simulation and draw a configuration of the spins once the system has reached equilibrium. Draw a line between each pair of spins of opposite sign. Describe the curve separating the $+1$ spins from the -1 spins. Begin with $L = 20$ and determine what happens as L is increased.

Appendix 15A: Relation of the Mean Demon Energy to the Temperature

We know that the energy of the demon, E_d, is constrained to be positive, and we found in Problems 15.3 and 15.4 that the probability for the demon to have energy E_d is proportional to $e^{-E_d/kT}$. We then assumed that the same form of the probability of a microstate holds for any macroscopic system in thermodynamic equilibrium. Hence in general, $\langle E_d \rangle$ is given by

$$\langle E_d \rangle = \frac{\sum_{E_d} E_d \, e^{-E_d/kT}}{\sum_{E_d} e^{-E_d/kT}}, \tag{15.101}$$

where the summations in (15.101) are over the possible values of E_d. If an Ising spin is flipped in zero magnetic field, the minimum nonzero decrease in energy of the system is $4J$ (see Fig. 15.11). Hence the possible energies of the demon are 0, $4J$, $8J$, $12J$, ... We write $x = 4J/kT$ and perform the summations in (15.101). The result is

$$\langle E_d/kT \rangle = \frac{0 + xe^{-x} + 2xe^{-2x} + \ldots}{1 + e^{-x} + e^{-2x} + \ldots} = \frac{x}{e^x - 1}. \tag{15.102}$$

The form (15.9) can be obtained by solving (15.102) for T in terms of E_d. Convince yourself that the relation (15.102) is independent of dimension for lattices with an even number of nearest neighbors.

If the magnetic field is nonzero, the possible values of the demon energy are $0, 2H, 4J - 2H, 4J + 2H, \cdots$. If J is a multiple of H, then the result is the same as before with $4J$ replaced by $2H$, because the possible energy values for the demon are multiples of $2H$. If the ratio $4J/2H$ is irrational, then the demon can take on a continuum of values, and thus $\langle E_d \rangle = kT$. The other

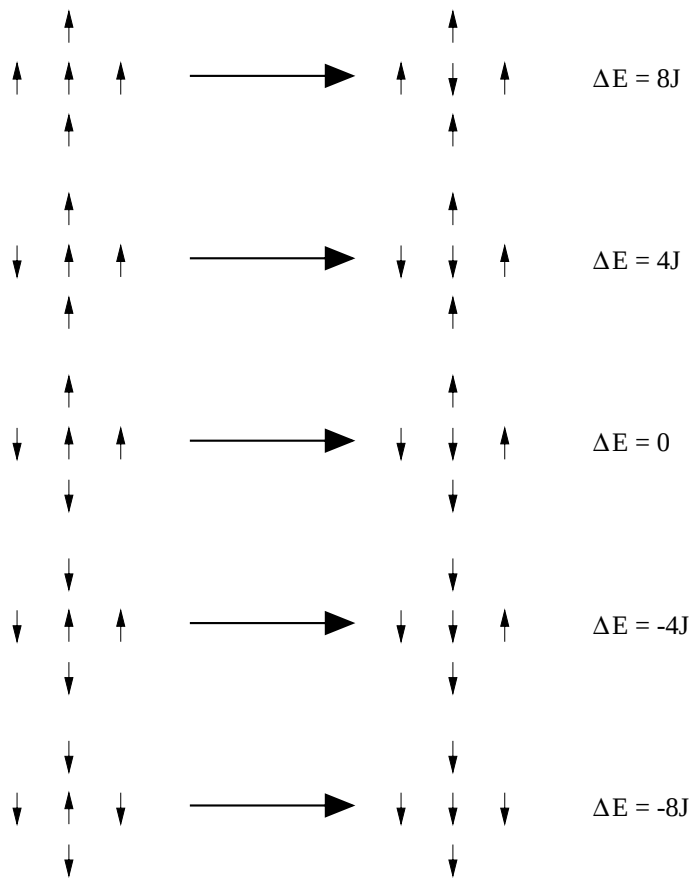

Figure 15.11: The five possible transitions of the Ising model on the square lattice with spin flip dynamics.

possibility is that $4J/2H = m/n$, where m and n are relatively prime positive integers. In this case it can be shown that (see Mak)

$$kT/J = \frac{4/m}{\ln(1 + 4J/m\langle E_d\rangle)}. \tag{15.103}$$

You can test these relations for $H \neq 0$ by choosing values of J and H and computing the sums in (15.101) directly.

Appendix 15B: Fluctuations in the Canonical Ensemble

We first obtain the relation of the constant volume heat capacity C_V to the energy fluctuations in the canonical ensemble. We write C_V as

$$C_V = \frac{\partial \langle E \rangle}{\partial T} = -\frac{1}{kT^2}\frac{\partial \langle E \rangle}{\partial \beta}. \tag{15.104}$$

From (15.10) we have

$$\langle E \rangle = -\tfrac{\partial}{\partial \beta} \ln Z, \tag{15.105}$$

and

$$\frac{\partial \langle E \rangle}{\partial \beta} = -\tfrac{1}{Z^2}\tfrac{\partial Z}{\partial \beta} \textstyle\sum_s E_s\, e^{-\beta E_s} - \tfrac{1}{Z}\textstyle\sum_s E_s^2\, e^{-\beta E_s} \tag{15.106}$$

$$= \langle E \rangle^2 - \langle E^2 \rangle. \tag{15.107}$$

The relation (15.18) follows from (15.104) and (15.107). Note that the heat capacity is at constant volume because the partial derivatives were performed with the energy levels E_s kept constant. The corresponding quantity for a magnetic system is the heat capacity at constant external magnetic field.

The relation of the magnetic susceptibility χ to the fluctuations of the magnetization M can be obtained in a similar way. We assume that the energy can be written as

$$E_s = E_{0,s} - HM_s, \tag{15.108}$$

where $E_{0,s}$ is the energy in the absence of a magnetic field, H is the external applied field, and M_s is the magnetization in the s state. The mean magnetization is given by

$$\langle M \rangle = \frac{1}{Z}\sum M_s\, e^{-\beta E_s}. \tag{15.109}$$

Because $\partial E_s/\partial H = -M_s$, we have

$$\frac{\partial Z}{\partial H} = \sum_s \beta M_s\, e^{-\beta E_s}. \tag{15.110}$$

Hence we obtain

$$\langle M \rangle = \frac{1}{\beta}\frac{\partial}{\partial H}\ln Z. \tag{15.111}$$

If we use (15.109) and (15.111), we find

$$\frac{\partial \langle M \rangle}{\partial H} = -\tfrac{1}{Z^2}\tfrac{\partial Z}{\partial H} \textstyle\sum_s M_s\, e^{-\beta E_s} + \tfrac{1}{Z}\textstyle\sum_s \beta M_s^2\, e^{-\beta E_s} \tag{15.112}$$

$$= -\beta \langle M \rangle^2 + \beta \langle M^2 \rangle. \tag{15.113}$$

The relation (15.20) for the zero field susceptibility follows from (15.113) and the definition (15.19).

# spins up	Degeneracy	Energy	Magnetization
4	1	-8	4
3	4	0	2
2	4	0	0
2	2	8	0
1	4	0	-2
0	1	-8	-4

Table 15.2: The energy and magnetization of the 2^4 states of the zero field Ising model on the 2×2 square lattice. The degeneracy is the number of microstates with the same energy.

Appendix 15C: Exact Enumeration of the 2×2 Ising Model

Because the number of possible states or configurations of the Ising model increases as 2^N, we can enumerate the possible configurations only for small N. As an example, we calculate the various quantities of interest for a 2×2 Ising model on the square lattice with periodic boundary conditions. In Table 15.2 we group the sixteen states according to their total energy and magnetization.

We can compute all the quantities of interest using Table 15.2. The partition function is given by

$$Z = 2\,e^{8\beta J} + 12 + 2\,e^{-8\beta J}. \tag{15.114}$$

If we use (15.105) and (15.114), we find

$$\langle E\rangle = -\frac{\partial}{\partial\beta}\ln Z = -\frac{1}{Z}\big[2(8)e^{8\beta J} + 2(-8)e^{-8\beta J}\big]. \tag{15.115}$$

Because the other quantities of interest can be found in a similar manner, we only give the results:

$$\langle E^2\rangle = \frac{1}{Z}\big[(2\times 64)\,e^{8\beta J} + (2\times 64)\,e^{-8\beta J}\big] \tag{15.116}$$

$$\langle M\rangle = \frac{1}{Z}(0) = 0 \tag{15.117}$$

$$\langle |M|\rangle = \frac{1}{Z}\big[(2\times 4)\,e^{8\beta J} + 8\times 2\big] \tag{15.118}$$

$$\langle M^2\rangle = \frac{1}{Z}\big[(2\times 16)\,e^{8\beta J} + 8\times 4\big]. \tag{15.119}$$

The dependence of C and χ on βJ can be found by using (15.115) and (15.116) and (15.117) and (15.119) respectively.

References and Suggestions for Further Reading

M. P. Allen and D. J. Tildesley, *Computer Simulation of Liquids*, Clarendon Press (1987). See Chapter 4 for a discussion of Monte Carlo methods.

Paul D. Beale, "Exact distribution of energies in the two-dimensional Ising model," Phys. Lev. Lett. **76**, 78–81 (1996). The author discusses a Mathemetica program that can compute the exact density of states for the two-dimensional Ising model.

K. Binder, editor, *Monte Carlo Methods in Statistical Physics*, second edition, Springer-Verlag (1986). Also see K. Binder, editor, *Applications of the Monte Carlo Method in Statistical Physics*, Springer-Verlag (1984) and K. Binder, editor, *The Monte Carlo Method in Condensed Matter Physics*, Springer-Verlag (1992). The latter book discusses the Binder cumulant method in the introductory chapter.

Marvin Bishop and C. Bruin, "The pair correlation function: a probe of molecular order," Am. J. Phys. **52**, 1106 (1984). The authors compute the pair correlation function for a two-dimensional Lennard-Jones model.

A. B. Bortz, M. H. Kalos and J. L. Lebowitz, "A new algorithm for Monte Carlo simulation of Ising spin systems," J. Comput. Phys. **17**, 10 (1975). First discussion of the n-fold way algorithm, which was rediscovered by many workers in the 1970's and 80's.

S. G. Brush, "History of the Lenz-Ising model," Rev. Mod. Phys. **39**, 883 (1967).

James B. Cole, "The statistical mechanics of image recovery and pattern recognition,"Am. J. Phys. **59**, 839 (1991). A discussion of the application of simulated annealing to the recovery of images from noisy data.

Michael Creutz, "Microcanonical Monte Carlo simulation," Phys. Rev. Lett. **50**, 1411–1414 (1983). See also Gyan Bhanot, Michael Creutz, and Herbert Neuberger, "Microcanonical simulation of Ising systems," Nuc. Phys. B **235**, 417 (1984).

Pratap Kumar Das and Parongama Sen, "Probability distributions of persistent spins in an Ising chain," J. Phys. A **37**, 7179–7184 (2004).

B. Derrida, A. J. Bray, and C. Godrèche, "Non-trivial exponents in the zero temperature dynamics of the 1D Ising and Potts models," J. Phys. A **27**, L357 (1994); B. Derrida, V. Hakim, and V. Pasquier, "Exact first passage exponents in 1d domain growth: Relation to a reaction-diffusion model," Phys. Rev. Lett. **75**, 751–754 (1995).

Jerome J. Erpenbeck and Marshall Luban, "Equation of state for the classical hard-disk fluid," Phys. Rev. A **32**, 2920 (1985). These workers use a combined molecular dynamics/Monte Carlo method and consider 1512 and 5822 disks.

Alan M. Ferrenberg, D. P. Landau, and Y. Joanna Wong, "Monte Carlo simulations: hidden errors from "good" random number generators," Phys. Rev. Lett. **69**, 3382 (1992).

Alan M. Ferrenberg and Robert H. Swendsen, "New Monte Carlo technique for studying phase transitions," Phys. Rev. Lett. **61**, 2635 (1988); "Optimized Monte Carlo data analysis," Phys. Rev. Lett. **63**, 1195 (1989); "Optimized Monte Carlo data analysis," Computers in Physics **3** 5, 101 (1989). The second and third papers discuss using the multiple histogram method with data from simulations at more than one temperature.

Daan Frenkel and Berend Smit, *Understanding Molecular Simulation*, second edition, Academic Press (2002).

Harvey Gould and W. Klein, "Spinodal effects in systems with long-range interactions," Physica D **66**, 61–70 (1993). This paper discusses nucleation in the Ising model and Lennard-Jones systems.

Harvey Gould and Jan Tobochnik, "Overcoming critical slowing down," Computers in Physics **3**4, 82 (1989).

James E. Gubernatis, *The Monte Carlo Method in the Physical Sciences*, AIP Press (2004).

Hong Guo, Martin Zuckermann, R. Harris, and Martin Grant, "A fast algorithm for simulated annealing," Physica Scripta **T38**, 40 (1991).

Gary S. Grest, Michael P. Anderson, and David J. Srolovitz, "Domain-growth kinetics for the Q-state Potts model in two and three dimensions," Phys. Rev. B **38**, 4752–4760 (1988).

R. Harris, "Demons at work," Computers in Physics **4** (3), 314 (1990).

J. Kertész, J. Cserti and J. Szép, "Monte Carlo simulation programs for microcomputer," Eur. J. Phys. **6**, 232 (1985).

S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," Science **220**, 671 (1983). See also, S. Kirkpatrick and G. Toulouse, "Configuration space analysis of traveling salesman problems," J. Physique **46**, 1277 (1985).

J. M. Kosterlitz and D. J. Thouless, "Ordering, metastability and phase transitions in two-dimensional systems," J. Phys. C **6**, 1181–1203 (1973); J. M. Kosterlitz and D. J. Thouless, "The critical properties of the two-dimensional xy model," J. Phys. C **7**, 1046 (1974).

D. P. Landau, Shan-Ho Tsai, and M. Exler, "A new approach to Monte Carlo simulations in statistical physics: Wang-Landau sampling, Am. J. Phys. **72**, 1294 (2004).

D. P. Landau, "Finite-size behavior of the Ising square lattice," Phys. Rev. B**13**, 2997 (1976). A clearly written paper on a finite-size scaling analysis of Monte Carlo data. See also D. P. Landau, "Finite-size behavior of the simple-cubic Ising lattice," Phys. Rev. B**14**, 255 (1976).

D. P. Landau and R. Alben, "Monte Carlo calculations as an aid in teaching statistical mechanics," Am. J. Phys. **41**, 394 (1973).

David P. Landau and Kurt Binder, *Monte Carlo Simulations in Statistical Physics*, Cambridge University Press (2000).

D. P. Landau, Shan-Ho Tsai, and M. Exler, "A new approach to Monte Carlo simulations in statistical physics: Wang-Landau sampling," Am. J. Phys., to be published.

Jooyoung Lee and J. M. Kosterlitz, "New numerical method to study phase transitions," Phys. Rev. Lett. **65**, 137 (1990); *ibid.*, "Finite-size scaling and Monte Carlo simulations of first-order phase transitions," Phys. Rev. B **43**, 3265 (1991).

Jooyoung Lee and Katherine J. Strandburg, "First-order melting transition of the hard-disk system," Phys. Rev. B **46**, 11190 (1992).

Jiwen Liu and Erik Luijten, "Rejection-free geometric cluster algorithm for complex fluids," Phys. Rev. Lett. **92** 035504-1–4 (2004).

S. S. Mak, "The analytical demon of the Ising model," Phys. Lett. A **196**, 318 (1995).

J. Marro and R. Toral, "Microscopic observations on a kinetic Ising model," Am. J. Phys. **54**, 1114 (1986).

N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Keller, "Equation of state calculations for fast computing machines," J. Chem. Phys. **6**, 1087 (1953).

M. E. J. Newman and G. T. Barkema, *Monte Carlo Methods in Statistical Physics*, Oxford University Press (1999).

M. A. Novotny, "A new approach to an old algorithm for the simulation of Ising-like systems," Computers in Physics **9** (1), 46 (1995). The n-fold way algorithm is discussed. Also, see M. A. Novotny, "A tutorial on advanced dynamic Monte Carlo methods for systems with discrete state spaces," in *Annual Reviews of Computational Physics IX*, edited by Dietrich Stauffer, World Scientific Publishing Company (2001), pp. 153–210.

Ole G. Mouritsen, *Computer Studies of Phase Transitions and Critical Phenomena*, Springer-Verlag (1984).

E. P. Münger and M. A. Novotny, "Reweighting in Monte Carlo and Monte Carlo renormalization-group studies," Phys. Rev. B **43**, 5773 (1991). The authors discuss the histogram method and combine it with renormalization group calculations.

Michael Plischke and Birger Bergersen, *Equilibrium Statistical Physics*, second edition, Prentice Hall (1994). A graduate level text that discusses some of the more contemporary topics in statistical physics, many of which have been influenced by computer simulations.

William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery, *Numerical Recipes*, second edition, Cambridge University Press (1992). A Fortran program for the traveling salesman problem is given in Section 10.9.

Stephen H. Shenker and Jan Tobochnik, "Monte Carlo renormalization-group analysis of the classical Heisenberg model in two dimensions," Phys. Rev. B **22**, 4462 (1980).

Amihai Silverman and Joan Adler, "Animated simulated annealing," Computers in Physics **6**, 277 (1992). The authors describe a simulation of the annealing process to obtain a defect free single crystal of a model material.

Zoran Slanič, Harvey Gould, and Jan Tobochnik, "Dynamics of the classical Heisenberg chain," Computers in Physics **5**, 630 (1991). Unlike the Ising model, the Heisenberg model has an intrinsic dynamics and can be studied by molecular dynamics as well as Monte Carlo methods.

H. Eugene Stanley, *Introduction to Phase Transitions and Critical Phenomena*, Oxford University Press (1971). See Appendix B for the exact solution of the zero-field Ising model for a two-dimensional lattice.

Jan Tobochnik and G. V. Chester, "Monte Carlo study of the planar model," Phys. Rev. B **20**, 3761 (1979).

I. Vattulainen, T. Ala-Nissila, and K. Kankaala, "Physical tests for random numbers in simulations," Phys. Rev. Lett. **73**, 2513 (1994).

B. Widom, "Some topics in the theory of fluids," J. Chem. Phys. **39**, 2802 (1963). This paper includes the insertion method for calculating the chemical potential.