Projects for Computational Physics (2019)

1. Projects-Program Requirements:

A. Objective: You should emphasize on the algorithm(s) implementation and capabilities of your program.

B. Presentation/Poster: state clearly what your program is doing and shows us how. Discuss algorithm(s) and problem(s) to be solved.

C. Program: readable, clear input/output, executable.

D. Results: demonstrate your achievements, such as the advantage(s) of your program over other approach. They must be repeatable.

2. Important Dates:

April 30, Tuesday, by 11:00, select projects done

June 14, Friday, anytime
Draft of Project Report Hand In (report draft, program(s), etc.)

June 17, Monday, 15:30-16:15, 16:25-17:10, 18:00-18:45, 18:55-19:40, Presentation
Each presentation is 5 minutes; send ppt copy to TA before **13:30**;
Everyone should be there except approved leave.

June 18, Tuesday, 13:30-16:30, Poster
At least one of authors should present.
June 19, Wednesday, same as June 18

June 20, Thursday, anytime
Final Report sends to TA and Lin HQ.

June 30, final grade.

Report Outline:

1. Introduction
2. Problems to solve
3. Numerical algorithm(s) to be used
4. Results and Discussions
5. Conclusions and Outlook
6. References

Appendix: program with user notes

Professor Notes

Project Title: _____

_____

Students: _____

(A) Project Performance and Report (evaluated by the supervisor)

1. Program (30%):
   (a) Algorithm(s) implementation (25%)
   (b) Readable, clear input/output files (5%)

2. Results (20%):
   (a) (Demonstrate) achievements and capabilities (10%)
   (b) Analysis (10%)

(B) Presentation (50%, evaluated by the panel):

(a) State clearly what the program is doing and shows us how, discuss algorithm(s) and problem(s) to be solved. (30%)
(b) Enlightening and stimulating presentation (20%): Intelligent choice of suitable materials, clear description of the background, key contents, helpful examples, major advancements, and historical/current impacts, performance in questioning and answering.

Total Marks = Difficulty Level * (A) + (B)

General Remark: Outstanding/Excellent/Very Good/Good/Poor

Projects: from “An Introduction to Computer Simulation Methods, Applications to Physical Systems”, 3rd edition, Harvey Gould, Jan Tobochnik, and Wolfgang Christian

Chapter 4 Oscillatory Systems: (p104)
Chapter 5 Few-Body Problems: (p137)
Chapter 6 Chaotic Motion: (p181)
Chapter 7 Random Process: (p245)
Chapter 8 Molecular Dynamics: (p300)
Chapter 10 E&M: (p407)
Chapter 15 Monte Carlo Simulations of Thermal Systems: (p646)
Chapter 16 Quantum Systems: (p715)

Difficulty Level 1 (11 topics, 1 topic for 1 group only)
4.18, 5.18, 6.22, 6.23, 6.27, 6.29, 16.32, 16.33, 16.34, 16.35, 16.36

Difficulty Level 2 (10 topics, 1 topic for no more than 2 groups)
5.19, 6.28, 7.40, 8.22, 8.23, 15.34, 15.38, 15.39, 15.40, 15.41

Difficulty Level 3 (12 topics, 1 topic for no more than 2 groups))
6.24, 6.26, 7.41, 8.24, 8.25, 15.32, 15.35, 15.36, 15.37, 15.42, 15.43, 15.44

Difficulty Level 4 (4 topics, 1 topic for no more than 2 groups))
8.26, 10.26, 15.45, 15.46

[Ma & Zhu: Too Easy (save for someone really need)
4.17(0.5), 5.17(0.5), 6.25(0.5)]

1-2 students form a group

(*pdf pages*)

Difficulty Level 1 (11 topics)
Project-4.18 Page: 62-63
Project-5.18 Page: 78-79
Project-6.22 Page: 100-101
Project-6.23 Page: 101
Project-6.27 Page: 103
Project-6.29 Page: 104

Difficulty Level 2 (10 topics)

Difficulty Level 3 (12 topics)

Difficulty Level 4 (4 topics)

Too Easy (save for someone really need)

Project-4.17(0.5) Page: 62

Project-5.17(0.5) Page: 78

Project-6.25(0.5) Page: 102

V. Choose $R = 2000\,\Omega$, $C = 10^{-6}$ farads, and $V = 10$ volts. Do you expect $Q(t)$ to increase with t? Does $Q(t)$ increase indefinitely, or does it reach a steady-state value? Use a program to solve (4.23) numerically using the Euler algorithm. What value of Δt is necessary to obtain three decimal accuracy at $t = 0.005$?

(b) What is the nature of your numerical solution to (4.23) at $t = 0.05$ for $\Delta t = 0.005$, 0.0025, and 0.001? Does a small change in Δt lead to a large change in the computed value of Q? Is the Euler algorithm stable for reasonable values of Δt? ■

4.7 ■ PROJECTS

Project 4.17 Chemical oscillations

The kinetics of chemical reactions can be modeled by a system of coupled first-order differential equations. As an example, consider the following reaction:

$$A + 2B \rightarrow 3B + C, \tag{4.26}$$

where A, B, and C represent the concentrations of three different types of molecules. The corresponding rate equations for this reaction are

$$\frac{dA}{dt} = -kAB^2 \tag{4.27a}$$

$$\frac{dB}{dt} = kAB^2 \tag{4.27b}$$

$$\frac{dC}{dt} = kAB^2. \tag{4.27c}$$

The rate at which the reaction proceeds is determined by the reaction constant k. The terms on the right-hand side of (4.27) are positive if the concentration of the molecule increases in (4.26) as it does for B and C, and negative if the concentration decreases as it does for A. Note that the term $2B$ in the reaction (4.26) appears as B^2 in the rate equation (4.27). In (4.27) we have assumed that the reactants are well stirred so that there are no spatial inhomogeneities. In Section 7.8 we will discuss the effects of spatial inhomogeneities due to molecular diffusion.

Most chemical reactions proceed to equilibrium, where the mean concentrations of all molecules are constant. However, if the concentrations of some molecules are replenished, it is possible to observe oscillations and chaotic behavior (see Chapter 6). To obtain oscillations, it is essential to have a series of chemical reactions such that the products of some reactions are the reactants of others. In the following, we consider a simple set of reactions that can lead to oscillations under certain conditions (see Lefever and Nicolis):

$$A \rightarrow X \tag{4.28a}$$

$$B + X \rightarrow Y + D \tag{4.28b}$$

$$2X + Y \rightarrow 3X \tag{4.28c}$$

$$X \rightarrow C. \tag{4.28d}$$

If we assume that the reverse reactions are negligible and A and B are held constant by an external source, the corresponding rate equations are

$$\frac{dX}{dt} = A - (B + 1)X + X^2Y \tag{4.29a}$$

$$\frac{dY}{dt} = BX - X^2Y. \tag{4.29b}$$

For simplicity, we have chosen the rate constants to be unity.

(a) The steady state solution of (4.29) can be found by setting dX/dt and dY/dt equal to zero. Show that the steady state values for (X, Y) are $(A, B/A)$.

(b) Write a program to solve numerically the rate equations given by (4.29). Your program should input the initial values of X and Y and the fixed concentrations A and B, and plot X versus Y as the reactions evolve.

(c) Systematically vary the initial values of X and Y for given values of A and B. Are their steady state behaviors independent of the initial conditions?

(d) Let the initial value of (X, Y) equal $(A + 0.001, B/A)$ for several different values of A and B, that is, choose initial values close to the steady state values. Classify which initial values result in steady state behavior (stable) and which ones show periodic behavior (unstable). Find the relation between A and B that separates the two types of behavior. ■

Project 4.18 Nerve impulses

In 1952 Hodgkin and Huxley developed a model of nerve impulses to understand the nerve membrane potential of a giant squid nerve cell. The equations they developed are known as the Hodgkin–Huxley equations. The idea is that a membrane can be treated as a capacitor where $CV = q$, and thus the time rate of change of the membrane potential V is proportional to the current dq/dt flowing through the membrane. This current is due to the pumping of sodium and potassium ions through the membrane, a leakage current, and an external current stimulus. The model is capable of producing single nerve impulses, trains of nerve impulses, and other effects. The model is described by the following first-order differential equations:

$$C\frac{dV}{dt} = -g_K n^4(V - V_K) - g_{Na}m^3h(V - V_{Na}) - g_L(V - V_L) + I_{\text{ext}}(t) \tag{4.30a}$$

$$\frac{dn}{dt} = \alpha_n(1 - n) - \beta_n n \tag{4.30b}$$

$$\frac{dm}{dt} = \alpha_m(1 - m) - \beta_m m \tag{4.30c}$$

$$\frac{dh}{dt} = \alpha_h(1 - h) - \beta_h h, \tag{4.30d}$$

where V is the membrane potential in millivolts (mV), n, m, and h are time dependent functions that describe the gates that pump ions into or out of the cell, C is the membrane capacitance per unit area, the g_i are the conductances per unit area for potassium, sodium,

and the leakage current, V_i are the equilibrium potentials for each of the currents, and α_j and β_j are nonlinear functions of V. We use the notation n, m, and h for the gate functions because the notation is universally used in the literature. These gate functions are empirical attempts to describe how the membrane controls the flow of ions into and out of the nerve cell. Hodgkin and Huxley found the following empirical forms for α_j and β_j:

$$\alpha_n = 0.01(V + 10)/[e^{(1+V/10)} - 1] \tag{4.31a}$$

$$\beta_n = 0.125\, e^{V/80} \tag{4.31b}$$

$$\alpha_m = 0.01(V + 25)/[e^{(2.5+V/10)} - 1] \tag{4.31c}$$

$$\beta_m = 4\, e^{V/18} \tag{4.31d}$$

$$\alpha_h = 0.07\, e^{V/20} \tag{4.31e}$$

$$\beta_m = 1/[e^{(3+V/10)} + 1]. \tag{4.31f}$$

The parameter values are $C = 1.0\,\mu\text{F/cm}^2$, $g_K = 36\,\text{mmho/cm}^2$, $g_{Na} = 120\,\text{mmho/cm}^2$, $g_L = 0.3\,\text{mmho/cm}^2$, $V_K = 12\,\text{mV}$, $V_{Na} = -115\,\text{mV}$, and $V_L = 10.6\,\text{mV}$. The unit mho represents ohm^{-1}, and the unit of time is milliseconds (ms). These parameters assume that the resting potential of the nerve cell is zero; however, we now know that the resting potential is about $-70\,\text{mV}$.

We can use the ODE solver to solve (4.30) with the state vector $\{V, n, m, h, t\}$; the rates are given by the right-hand side of (4.30). The following questions ask you to explore the properties of the model.

(a) Write a program to plot n, m, and h as a function of V in the steady state (for which $\dot{n} = \dot{m} = \dot{h} = 0$). Describe how these gates are operating.

(b) Write a program to simulate the nerve cell membrane potential and plot $V(t)$. You can use a simple Euler algorithm with a time step of 0.01 ms. Describe the behavior of the potential when the external current is 0.

(c) Consider a current that is zero except for a one millisecond interval. Try a current spike amplitude of 7 μA (that is, the external current equals 7 in our units). Describe the resulting nerve impulse $V(t)$. Is there a threshold value for the current below which there is no large spike but only a broad peak?

(d) A constant current should produce a train of spikes. Try different amplitudes for the current and determine if there is a threshold current and how the spacing between spikes depends on the amplitude of the external current.

(e) Consider a situation where there is a steady external current I_1 for 20 ms and then the current increases to $I_2 = I_1 + \Delta I$. There are three types of behavior depending on I_2 and ΔI. Describe the behavior for the following four situations: (1) $I_1 = 2.0\,\mu$A, $\Delta I = 1.5\,\mu$A; (2) $I_1 = 2.0\,\mu$A, $\Delta I = 5.0\,\mu$A; (3) $I_1 = 7.0\,\mu$A, $\Delta I = 1.0\,\mu$A; and (4) $I_1 = 7.0\,\mu$A, $\Delta I = 4.0\,\mu$A. Try other values of I_1 and ΔI as well. In which cases do you obtain a steady spike train? Which cases produce a single spike? What other behavior do you find?

(f) Once a spike is triggered, it is frequently difficult to trigger another spike. Consider a current pulse at $t = 20$ ms of 7 μA that lasts for one millisecond. Then give a second current pulse of the same amplitude and duration at $t = 25$ ms. What happens? What happens if you add a third pulse at 30 ms? ■

REFERENCES AND SUGGESTIONS FOR FURTHER READING

F. S. Acton, *Numerical Methods That Work* (The Mathematical Association of America, 1999), Chapter 5.

G. L. Baker and J. P. Gollub, *Chaotic Dynamics: An Introduction*, 2nd ed. (Cambridge University Press, 1996). A good introduction to the notion of phase space.

Eugene I. Butikov, "Square-wave excitation of a linear oscillator," Am. J. Phys. **72**, 469–476 (2004).

A. Douglas Davis, *Classical Mechanics* (Saunders College Publishing, 1986). The author gives simple numerical solutions of Newton's equations of motion. Much emphasis is given to the harmonic oscillator problem.

S. Eubank, W. Miner, T. Tajima, and J. Wiley, "Interactive computer simulation and analysis of Newtonian dynamics," Am. J. Phys. **57**, 457–463 (1989).

Richard P. Feynman, Robert B. Leighton, and Matthew Sands, *The Feynman Lectures on Physics*, Vol. 1 (Addison–Wesley, 1963). Chapters 21 and 23–25 are devoted to various aspects of harmonic motion.

A. P. French, *Newtonian Mechanics* (W. W. Norton & Company, 1971). An introductory level text with a good discussion of oscillatory motion.

M. Gitterman, "Classical harmonic oscillator with multiplicative noise," Physica A **352**, 309–334 (2005). The analysis is analytical and at the graduate level. However, it would be straightforward to reproduce most of the results after you learn about random processes in Chapter 7.

A. L. Hodgkin and A. F. Huxley, "A quantitative description of ion currents and its applications to conduction and excitation in nerve membranes," J. Physiol. (Lond.) **117**, 500–544 (1952).

Charles Kittel, Walter D. Knight, and Malvin A. Ruderman, *Mechanics*, 2nd ed., revised by A. Carl Helmholz and Burton J. Moyer (McGraw–Hill, 1973).

R. Lefever and G. Nicolis, "Chemical instabilities and sustained oscillations," J. Theor. Biol. **30**, 267 (1971).

Jerry B. Marion and Stephen T. Thornton, *Classical Dynamics*, 5th ed. (Harcourt, 2004). Excellent discussion of linear and nonlinear oscillators.

M. F. McInerney, "Computer-aided experiments with the damped harmonic oscillator," Am. J. Phys. **53**, 991–996 (1985).

William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery, *Numerical Recipes*, 2nd ed. (Cambridge University Press, 1992). Chapter 16 discusses the integration of ordinary differential equations.

Scott Hamilton, *An Analog Electronics Companion* (Cambridge University Press, 2003). A good discussion of the physics and mathematics of basic circuit design including an extensive introduction to circuit simulation using the PSpice simulation program.

S. C. Zilio, "Measurement and analysis of large-angle pendulum motion," Am. J. Phys. **50**, 450–452 (1982).

The `ThreeBodyApp` class in Listing 5.10 is the target class for the three-body program. The `doStep` method merely increments the model's differential equations solver and repaints the view.

Listing 5.10 A program that displays the trajectories of three bodies interacting via gravitational forces.

```
package org.opensourcephysics.sip.ch05;
import org.opensourcephysics.controls.*;
import org.opensourcephysics.frames.*;

public class ThreeBodyApp extends AbstractSimulation {
   PlotFrame frame = new PlotFrame("x", "y", "Three-Body Orbits");
   ThreeBody trajectory = new ThreeBody();

   public ThreeBodyApp() {
      frame.addDrawable(trajectory);
      frame.setSquareAspect(true);
      frame.setSize(450, 450);
   }

   public void initialize() {
      trajectory.odeSolver.setStepSize(control.getDouble("dt"));
      trajectory.initialize(ThreeBodyInitialConditions.MONTGOMERY);
      frame.setPreferredMinMax(-1.5, 1.5, -1.5, 1.5);
   }

   public void reset() {
      control.setValue("dt", 0.1);
      enableStepsPerDisplay(true);
      initialize();
   }

   protected void doStep() {
      trajectory.doStep();
      frame.setMessage("t="+decimalFormat.format(trajectory.state[4]));
   }

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

Problem 5.16 Stability of solutions to the three-body problem

Examine the stability of the three solutions to the three-body problem by slightly varying the initial velocity of one of the masses. Before passing your new initial state to `trajectory.initialize`, calculate the center of mass velocity and subtract this velocity from every object. Show that any instability is due to the physics and not to the numerical differential equation solver. Which of the three analytic solutions is stable? Check conservation of the total energy and angular momentum. ■

5.12 ■ PROJECTS

Project 5.17 Effect of a "solar wind"

(a) Assume that a satellite is affected not only by the Earth's gravitational force, but also by a weak uniform "solar wind" of magnitude W acting in the horizontal direction. The equations of motion can be written as

$$\frac{d^2x}{dt^2} = -\frac{GMx}{r^3} + W \tag{5.32a}$$

$$\frac{d^2y}{dt^2} = -\frac{GMy}{r^3}. \tag{5.32b}$$

Choose initial conditions so that a circular orbit would be obtained for $W = 0$. Then choose a value of W whose magnitude is about 3% of the acceleration due to the gravitational field and compute the orbit. How does the orbit change?

(b) Determine the change in the velocity space orbit when the solar wind (5.32) is applied. How does the total angular momentum and energy change? Explain in simple terms the previously observed change in the position space orbit. See Luehrmann for further discussion of this problem. ■

Project 5.18 Resonances and the asteroid belt

(a) A histogram of the number of asteroids versus their distance from the Sun shows some distinct gaps. These gaps, called the *Kirkwood gaps*, are due to resonance effects. That is, if asteroids were in these gaps, their periods would be simple fractions of the period of Jupiter. Modify class `Planet2` so that planet two has the mass of Jupiter by setting `GM1 = 0.001*GM`. Because the asteroid masses are very small compared to that of Jupiter, the gravitational force on Jupiter due to the asteroids can be neglected. The initial conditions listed in `Planet2` are approximately correct for Jupiter. The initial conditions for the asteroid (planet one in `Planet2`) correspond to the 1/3 resonance (the period of the asteroid is one third that of Jupiter). Run the program with these changes and describe the orbit of the asteroid.

(b) Use Kepler's third law, $T^2/a^3 =$ constant, to determine the values of a, the asteroid's semimajor axis, such that the ratio of its period of revolution about the Sun to that of Jupiter is 1/2, 3/7, 2/5, and 2/3. Set the initial value of `x(1)` equal to a for each of these ratios and choose the initial value of `vy(1)` so that the asteroid would have a circular orbit if Jupiter was not present. Describe the orbits that you obtain.

(c) It is instructive to plot a as a function of time. However, because it is not straightforward to measure a directly in the simulation, it is more convenient to plot the quantity $-2GMm/E$, where E is the total energy of the asteroid and m is the mass of the asteroid. Because E is proportional to m, the quantity $-2GMm/E$ is independent of m. If the interaction of the asteroid with Jupiter is ignored, it can be shown that $a = -2GMm/E$, where E is the asteroid kinetic energy plus the asteroid-Sun potential energy. Derive this result for circular orbits. Plot the quantity $-2GMm/E$ versus time for about thirty revolutions for the initial conditions in Problem 5.18b.

(d) Compute the time dependence of $-2GMm/E$ for asteroid orbits whose initial position `x(1)` ranges from 2.0 to 5.0 in steps of 0.2. Choose the initial values of `vy(1)` so that circular orbits would be obtained in the absence of Jupiter. Are there any values of `x(1)` for which the time dependence of a is unusual?

(e) Make a histogram of the number of asteroids versus the value of $-2GMm/E$ at $t = 2000$. (You can use the `HistogramFrame` class described on page 206 if you wish.) Assume that the initial value of `x(1)` ranges from 2.0 to 5.0 in steps of 0.02 and choose the initial values of `vy(1)` as before. Use a histogram bin width of 0.1. If you have time, repeat for $t = 5000$ and compare the histogram with your previous results. Is there any evidence for Kirkwood gaps? A resonance occurs when the periods of the asteroid and Jupiter are related by simple fractions. We expect the number of asteroids with values of a corresponding to resonances to be small.

(f) Repeat part (e) with initial velocities that vary from their values for a circular orbit by 1, 3, and 5%. ■

Project 5.19 The classical helium atom

The classical helium atom is a relatively simple example of a three-body problem and is similar to the gravitational three-body problem of a heavy sun and two light planets. The important difference is that the two electrons repel one another, unlike the planetary case where the intraplanetary interaction is attractive. If we ignore the small motion of the heavy nucleus, the equations of motion for the two electrons can be written as

$$\mathbf{a}_1 = -2\frac{\mathbf{r}_1}{r_1^3} + \frac{\mathbf{r}_1 - \mathbf{r}_2}{r_{12}^3} \tag{5.33a}$$

$$\mathbf{a}_2 = -2\frac{\mathbf{r}_2}{r_2^3} + \frac{\mathbf{r}_2 - \mathbf{r}_1}{r_{12}^3}, \tag{5.33b}$$

where $\mathbf{r}_1$ and $\mathbf{r}_2$ are measured from the fixed nucleus at the origin, and r_{12} is the distance between the two electrons. We have chosen units such that the mass and charge of the electron are both unity. The charge of the helium nucleus is two in these units. Because the electrons are sometimes very close to the nucleus, their acceleration can become very large, and a very small time step Δt is required. It is not efficient to use the same small time step throughout the simulation, and instead a variable time step or an *adaptive* step size algorithm is suggested. An adaptive step size algorithm can be used with any standard numerical algorithm for solving differential equations. The RK45 algorithm described in Appendix 3A is adaptive and is a good all-around choice for these types of problems.

(a) For simplicity, we restrict our atom to two dimensions. Modify `Planet2` to simulate the classical helium atom. Choose units such that the electron mass is one and the other constants are absorbed into the unit of charge so that the force between two electrons is

$$|F| = \frac{1}{r^2}. \tag{5.34}$$

Choose the initial value of the time step to be $\Delta t = 0.001$. Some of the possible orbits are similar to those we have seen in our mini-solar system. For example, try the initial condition $\mathbf{r}_1 = (2, 0)$, $\mathbf{r}_2 = (-1, 0)$, $\mathbf{v}_1 = (0, 0.95)$, and $\mathbf{v}_2 = (0, -1)$.

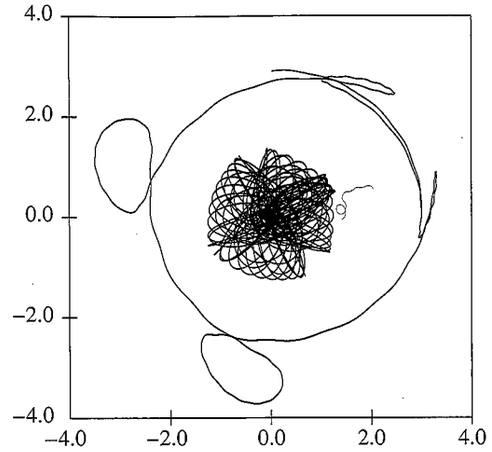


Figure 5.8 Orbits of the two electrons in the classical helium atom with the initial condition $\mathbf{r}_1 = (3, 0)$, $\mathbf{r}_2 = (1, 0)$, $\mathbf{v}_1 = (0, 0.4)$, and $\mathbf{v}_2 = (0, -1)$ (see Project 5.19c).

(b) Most initial conditions result in unstable orbits in which one electron eventually leaves the atom (autoionization). The initial condition $\mathbf{r}_1 = (1.4, 0)$, $\mathbf{r}_2 = (-1, 0)$, $\mathbf{v}_1 = (0, 0.86)$, and $\mathbf{v}_2 = (0, -1)$ gives "braiding" orbits. Make small changes in this initial condition to observe autoionization.

(c) The classical helium atom is capable of very complex orbits (see Figure 5.8). Investigate the motion for the initial condition $\mathbf{r}_1 = (3, 0)$, $\mathbf{r}_2 = (1, 0)$, $\mathbf{v}_1 = (0, 0.4)$, and $\mathbf{v}_2 = (0, -1)$. Does the motion conserve the total angular momentum? Also try $\mathbf{r}_1 = (2.5, 0)$, $\mathbf{r}_2 = (1, 0)$, $\mathbf{v}_1 = (0, 0.4)$, and $\mathbf{v}_2 = (0, -1)$.

(d) Choose the initial condition $\mathbf{r}_1 = (2, 0)$, $\mathbf{r}_2 = (-1, 0)$, and $\mathbf{v}_2 = (0, -1)$. Then vary the initial value of $\mathbf{v}_1$ from $(0.6, 0)$ to $(1.3, 0)$ in steps of $\Delta v = 0.02$. For each set of initial conditions, calculate the time it takes for autoionization. Assume that ionization occurs when either electron exceeds a distance of six from the nucleus. Run each simulation for a maximum time of 2000. Plot the ionization time versus v_{1x}. Repeat for a smaller interval of Δv centered about one of the longer ionization times. These calculations require much computer resources. Do the two plots look similar? If so, such behavior is called "self-similar" and is characteristic of chaotic systems and the geometry of fractals (see Chapters 6 and 13). More discussion on the nature of the orbits can be found in Yamamoto and Kaneko. ■

REFERENCES AND SUGGESTIONS FOR FURTHER READING

Harold Abelson, Andrea diSessa, and Lee Rudolph, "Velocity space and the geometry of planetary orbits," Am. J. Phys. **43**, 579–589 (1975). See also Andrea diSessa, "Orbit: a mini-environment for exploring orbital mechanics," O. Lecarme and R. Lewis, editors, *Computers in Education*, 359, North-Holland (1975). Detailed geometrical rather than calculus-based arguments on the origin of closed orbits for inverse-square forces are presented. Sections 5.7 and 5.8 are based on these papers.

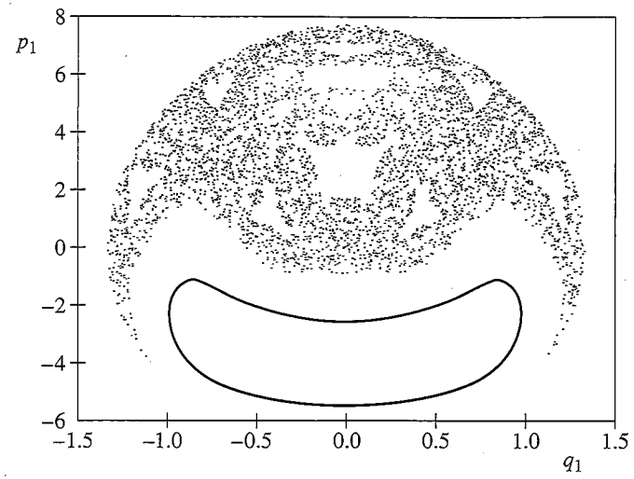


Figure 6.13 Poincaré plot for the double pendulum with p_1 plotted versus q_1 for $q_2 = 0$ and $p_2 > 0$. Two sets of initial conditions, $(q_1, q_2, p_1) = (0, 0, 0)$ and $(1.1, 0, 0)$, respectively, were used to create the plot. The initial value of the coordinate p_2 is found from (6.52) by requiring that $E = 15$.

E. The initial values of q_1 and q_2 can be chosen either randomly within the interval $|q_i| < \pi$ or by the user. Then set the initial $p_1 = 0$ and solve for p_2 using (6.52) with $H = E$. First explore the pendulum's behavior by plotting the generalized coordinates and momenta as a function of time in four windows. Consider the energies $E = 1$, 5, 10, 15, and 40. Try a few initial conditions for each value of E. Visually determine whether the steady state behavior is regular or appears to be chaotic. Are there some values of E for which all the trajectories appear regular? Are there values of E for which all trajectories appear chaotic? Are there values of E for which both types of trajectories occur?

(b) Repeat part (a) but plot the phase space diagrams p_1 versus q_1 and p_2 versus q_2. Are these plots more useful for determining the nature of the trajectories than those drawn in part (a)?

(c) Draw the Poincaré plot with p_1 plotted versus q_1 only when $q_2 = 0$ and $p_2 > 0$. Overlay trajectories from different initial conditions but with the same total energy on the same plot. Duplicate the plot shown in Figure 6.13. Then produce Poincaré plots for the values of E given in part (a) with at least five different initial conditions for each energy. Describe the different types of behavior.

(d) Is there a critical value of the total energy at which some chaotic trajectories first occur?

(e) Animate the double pendulum, showing the two masses moving back and forth. Describe how the motion of the pendulum is related to the behavior of the Poincaré plot. ■

Hamiltonian chaos has important applications in physical systems such as the solar system, the motion of the galaxies, and plasmas. It also has helped us understand the foundation for statistical mechanics. One of the most fascinating applications has been

to quantum mechanics, which has its roots in the Hamiltonian formulation of classical mechanics. A current area of interest is the quantum analogue of classical Hamiltonian chaos. The meaning of this analogue is not obvious because well-defined trajectories do not exist in quantum mechanics. Moreover, Schrödinger's equation is linear and can be shown to have only periodic and quasiperiodic solutions.

6.10 ■ PERSPECTIVE

As the many books and review articles on chaos can attest, it is impossible to discuss all aspects of chaos in a single chapter. We will revisit chaotic systems in Chapter 13 where we introduce the concept of fractals. We will find that one of the characteristics of chaotic dynamics is that the resulting attractors often have an intricate geometrical structure.

The most general ideas that we have discussed in this chapter are that *simple systems can exhibit complex behavior* and that chaotic systems exhibit *extreme sensitivity to initial conditions*. We have also learned that computers allow us to explore the behavior of dynamical systems and visualize the numerical output. However, the simulation of a system does not automatically lead to understanding. If you are interested in learning more about the phenomena of chaos and the associated theory, the suggested readings at the end of the chapter are a good place to start. We also invite you to explore chaotic phenomenon in more detail in the following projects.

6.11 ■ PROJECTS

The first several projects are on various aspects of the logistic map. These projects do not exhaust the possible investigations of the properties of the logistic map.

Project 6.22 A more accurate determination of δ and α

We have seen that it is difficult to determine δ accurately by finding the sequence of values of b_k at which the trajectory bifurcates for the kth time. A better way to determine δ is to compute it from the sequence s_m of superstable trajectories of period 2^{m-1}. We already have found that $s_1 = 1/2$, $s_2 \approx 0.80902$, and $s_3 \approx 0.87464$. The parameters $s_1, s_2, \ldots$ can be computed directly from the equation

$$f^{(2^{m-1})}\left(x = \frac{1}{2}\right) = \frac{1}{2}. \tag{6.53}$$

For example, s_2 satisfies the relation $f^{(2)}(x = 1/2) = 1/2$. This relation, together with the analytical form for $f^{(2)}(x)$ given in (6.7), yields

$$8r^2(1 - r) - 1 = 0. \tag{6.54}$$

If we wish to solve (6.54) numerically for $r = s_2$, we need to be careful not to find the irrelevant solutions corresponding to a lower period. In this case we can factor out the solution $r = 1/2$ and solve the resultant quadratic equation analytically to find $s_2 = (1 + \sqrt{5})/4$. Clearly $r = s_1 = 1/2$ solves (6.54) with period 1, because from (6.53), $f^{(1)}(x = 1/2) = 4r\frac{1}{2}(1 - \frac{1}{2}) = r = 1/2$ only for $r = 1/2$.

Table 6.2 Values of the control parameter s_m for the superstable trajectories of period 2^{m-1}. Nine decimal places are shown.

m	Period	s_m
1	1	0.500 000 000
2	2	0.809 016 994
3	4	0.874 640 425
4	8	0.888 660 970
5	16	0.891 666 899
6	32	0.892 310 883
7	64	0.892 448 823
8	128	0.892 478 091

(a) It is straightforward to adapt the bisection method discussed in Section 6.6. Adapt the class `RecursiveFixedPointApp` to find the numerical solutions of (6.53). Good starting values for the left-most and right-most values of r are easy to obtain. The left-most value is $r = r_\infty \approx 0.8925$. If we already know the sequence $s_1, s_2, \ldots, s_m$, then we can determine δ by

$$\delta_m = \frac{s_{m-1} - s_{m-2}}{s_m - s_{m-1}}. \tag{6.55}$$

We use this determination for δ_m to find the right-most value of r:

$$r_{\text{right}}^{(m+1)} = \frac{s_m - s_{m-1}}{\delta_m}. \tag{6.56}$$

We choose the desired precision to be 10^{-16}. A summary of our results is given in Table 6.2. Verify these results and determine δ.

(b) Use your values of s_m to obtain a more accurate determination of α and δ. ■

Project 6.23 From chaos to order

The bifurcation diagram of the logistic map (see Figure 6.2) has many interesting features that we have not explored. For example, you might have noticed that there are several smooth dark bands in the chaotic region for $r > r_\infty$. Use `BifurcateApp` to generate the bifurcation diagram for $r_\infty \le r \le 1$. If we start at $r = 1.0$ and decrease r, we see that there is a band that narrows and eventually splits into two parts at $r \approx 0.9196$. If you look closely, you will see that the band splits into four parts at $r \approx 0.899$. If you look even more closely, you will see many more bands. What type of change occurs near the splitting (merging) of these bands? Use `IterateMap` to look at the time series of (6.5) for $r = 0.9175$. You will notice that although the trajectory looks random, it oscillates back and forth between two bands. This behavior can be seen more clearly if you look at the time series of $x_{n+1} = f^{(2)}(x_n)$. A detailed discussion of the splitting of the bands can be found in Peitgen et al. ■

Project 6.24 Calculation of the Lyapunov spectrum

In Section 6.5 we discussed the calculation of the Lyapunov exponent for the logistic map. If a dynamical system has a multidimensional phase space, for example, the Hénon map

and the Lorenz model, there is a set of Lyapunov exponents called the Lyapunov spectrum that characterize the divergence of the trajectory. As an example, consider a set of initial conditions that forms a filled sphere in phase space for the (three-dimensional) Lorenz model. If we iterate the Lorenz equations, then the set of phase space points will deform into another shape. If the system has a fixed point, this shape contracts to a single point. If the system is chaotic, then typically the sphere will diverge in one direction but become smaller in the other two directions. In this case we can define three Lyapunov exponents to measure the deformation in three mutually perpendicular directions. These three directions generally will not correspond to the axes of the original variables. Instead, we must use a Gram–Schmidt orthogonalization procedure.

The algorithm for finding the Lyapunov spectrum is as follows:

(i) Linearize the dynamical equations. If $\mathbf{r}$ is the f-component vector containing the dynamical variables, then define $\Delta\mathbf{r}$ as the linearized difference vector. For example, the linearized Lorenz equations are

$$\frac{d\Delta x}{dt} = -\sigma\Delta x + \sigma\Delta y \tag{6.57a}$$

$$\frac{d\Delta y}{dt} = -x\Delta z - z\Delta x + r\Delta x - \Delta y \tag{6.57b}$$

$$\frac{d\Delta z}{dt} = x\Delta y + y\Delta x - b\Delta z. \tag{6.57c}$$

(ii) Define f orthonormal initial values for $\Delta\mathbf{r}$. For example, $\Delta\mathbf{r}_1(0) = (1, 0, 0)$, $\Delta\mathbf{r}_2(0) = (0, 1, 0)$, and $\Delta\mathbf{r}_3(0) = (0, 0, 1)$. Because these vectors appear in a linearized equation, they do not have to be small in magnitude.

(iii) Iterate the original and linearized equations of motion. One iteration yields a new vector from the original equation of motion and f new vectors $\Delta\mathbf{r}_\alpha$ from the linearized equations.

(iv) Find the orthonormal vectors $\Delta\mathbf{r}'_\alpha$ from the $\Delta\mathbf{r}_\alpha$ using the Gram–Schmidt procedure. That is,

$$\Delta\mathbf{r}'_1 = \frac{\Delta\mathbf{r}_1}{|\Delta\mathbf{r}_1|} \tag{6.58a}$$

$$\Delta\mathbf{r}'_2 = \frac{\Delta\mathbf{r}_2 - (\Delta\mathbf{r}'_1 \cdot \Delta\mathbf{r}_2)\Delta\mathbf{r}'_1}{|\Delta\mathbf{r}_2 - (\Delta\mathbf{r}'_1 \cdot \Delta\mathbf{r}_2)\Delta\mathbf{r}'_1|} \tag{6.58b}$$

$$\Delta\mathbf{r}'_3 = \frac{\Delta\mathbf{r}_3 - (\Delta\mathbf{r}'_1 \cdot \Delta\mathbf{r}_3)\Delta\mathbf{r}'_1 - (\Delta\mathbf{r}'_2 \cdot \Delta\mathbf{r}_3)\Delta\mathbf{r}'_2}{|\Delta\mathbf{r}_3 - (\Delta\mathbf{r}'_1 \cdot \Delta\mathbf{r}_3)\Delta\mathbf{r}'_1 - (\Delta\mathbf{r}'_2 \cdot \Delta\mathbf{r}_3)\Delta\mathbf{r}'_2|}. \tag{6.58c}$$

It is straightforward to generalize the method to higher-dimensional models.

(v) Set the $\Delta\mathbf{r}_\alpha(t)$ equal to the orthonormal vectors $\Delta\mathbf{r}'_\alpha(t)$.

(vi) Accumulate the running sum, S_α as $S_\alpha \to S_\alpha + \log|\Delta\mathbf{r}_\alpha(t)|$.

(vii) Repeat steps (iii)–(vi) and periodically output the approximate Lyapunov exponents $\lambda_\alpha = (1/n)S_\alpha$, where n is the number of iterations.

To obtain a result for the Lyapunov spectrum that represents the steady state attractor, include only data after the transient behavior has ended.

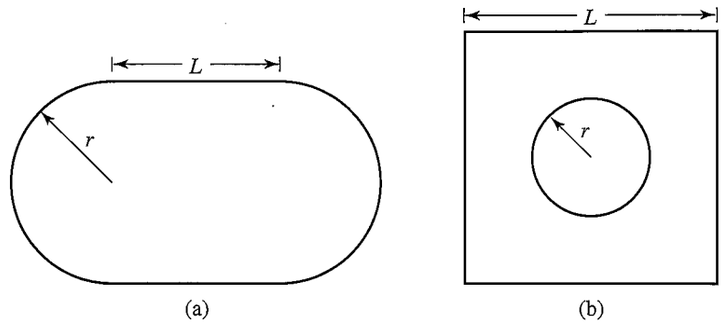


Figure 6.14 (a) Geometry of the stadium billiard model. (b) Geometry of the Sinai billiard model.

(a) Compute the Lyapunov spectrum for the Lorenz model for $\sigma = 16$, $b = 4$, and $r = 45.92$. Try other values of the parameters and compare your results.

(b) Linearize the equations for the Hénon map and find the Lyapunov spectrum for $a = 1.4$ and $b = 0.3$ in (6.32). ■

Project 6.25 A spinning magnet

Consider a compass needle that is free to rotate in a periodically reversing magnetic field which is perpendicular to the axis of the needle. The equation of motion of the needle is given by

$$\frac{d^2\phi}{dt^2} = -\frac{\mu}{I} B_0 \cos\omega t \sin\phi, \tag{6.59}$$

where ϕ is the angle of the needle with respect to a fixed axis along the field, μ is the magnetic moment of the needle, I its moment of inertia, and B_0 and ω are the amplitude and the angular frequency of the magnetic field, respectively. Choose an appropriate numerical method for solving (6.59) and plot the Poincaré map at time $t = 2\pi n/\omega$. Verify that if the parameter $\lambda = \sqrt{2B_0\mu/I/\omega^2} > 1$, then the motion of the needle exhibits chaotic motion. Briggs (see references) discusses how to construct the corresponding laboratory system and other nonlinear physical systems. ■

Project 6.26 Billiard models

Consider a two-dimensional planar geometry in which a particle moves with constant velocity along straight line orbits until it elastically reflects off the boundary. This straight line motion occurs in various "billiard" systems. A simple example of such a system is a particle moving with fixed speed within a circle. For this geometry the angle between the particle's momentum and the tangent to the boundary at a reflection is the same for all points.

Suppose that we divide the circle into two equal parts and connect them by straight lines of length L as shown in Figure 6.14a. This geometry is called a *stadium billiard*. How does the motion of a particle in the stadium compare to the motion in the circle? In both cases we can find the trajectory of the particle by geometrical considerations. The stadium billiard model and a similar geometry known as the Sinai billiard model (see Figure 6.14b) have been used as model systems for exploring the foundations of statistical mechanics. There is also much interest in relating the behavior of a classical particle in various billiard models to the solution of Schrödinger's equation for the same geometries.

(a) Write a program to simulate the stadium billiard model. Use the radius r of the semicircles as the unit of length. The algorithm for determining the path of the particle is as follows:

(i) Begin with an initial position (x_0, y_0) and momentum (p_{x0}, p_{y0}) of the particle such that $|\mathbf{p}_0| = 1$.

(ii) Determine which of the four sides the particle will hit. The possibilities are the top and bottom line segments and the right and left semicircles.

(iii) Calculate the next position of the particle from the intersection of the straight line defined by the current position and momentum, and also determine the equation for the segment where the next reflection occurs.

(iv) Determine the new momentum, (p'_x, p'_y), of the particle after reflection such that the angle of incidence equals the angle of reflection. For reflection off the line segments we have $(p'_x, p'_y) = (p_x, -p_y)$. For reflection off a circle we have

$$p'_x = [y^2 - (x - x_c)^2]p_x - 2(x - x_c)yp_y \tag{6.60a}$$

$$p'_y = -2(x - x_c)yp_x + [(x - x_c)^2 - y^2]p_y, \tag{6.60b}$$

where $(x_c, 0)$ is the center of the circle. (Note that the momentum p_x rather than p'_x is on the right-hand side of (6.60b). Remember that all lengths are scaled by the radius of the circle.)

(v) Repeat steps (ii)–(iv).

(b) Determine if the particle dynamics is chaotic by estimating the largest Lyapunov exponent. One way to do so is to start two particles with almost identical positions and/or momenta (varying by say 10^{-5}). Compute the difference Δs of the two phase space trajectories as a function of the number of reflections n, where Δs is defined by

$$\Delta s = \sqrt{|\mathbf{r}_1 - \mathbf{r}_2|^2 + |\mathbf{p}_1 - \mathbf{p}_2|^2}. \tag{6.61}$$

Choose $L = 1$ and $r = 1$. The Lyapunov exponent can be found from a semilog plot of Δs versus n. Repeat your calculation for different initial conditions and average your values of Δs before plotting. Repeat the calculation for $L = 0.5$ and 2.0 and determine if your results depend on L.

(c) Another test for the existence of chaos is the reversibility of the motion. Reverse the momentum after the particle has made n reflections, and let the drawing color equal the background color so that the path can be erased. What limitation does roundoff error place on your results? Repeat this simulation for $L = 1$ and $L = 0$.

(d) Place a small hole of diameter d in one of the circular sections of the stadium so that the particle can escape. Choose $L = 1$ and set $d = 0.02$. Give the particle a random position and momentum, and record the time when the particle escapes through the hole. Repeat for at least 10^4 particles and compute the fraction of particles $S(n)$ remaining after a given number of reflections n. The function $S(n)$ will decay with n. Determine the functional dependence of S on n and calculate the characteristic decay time if $S(n)$ decays exponentially. Repeat for $L = 0.1$, 0.5, and 2.0. Is the decay time a function of L? Does $S(n)$ decay exponentially for the circular billiard model ($L = 0$) (see Bauer and Bertsch)?

(e) Choose an arbitrary initial position for the particle in a stadium with $L = 1$ and a small hole as in part (d). Choose at least 5000 values of the initial value p_{x0} uniformly distributed between 0 and 1. Choose p_{y0} so that $|\mathbf{p}| = 1$. Plot the escape time versus p_{x0} and describe the visual pattern of the trajectories. Then choose 5000 values of p_{x0} in a smaller interval centered about the value of p_{x0} for which the escape time was greatest. Plot these values of the escape time versus p_{x0}. Do you see any evidence of self-similarity?

(f) Repeat steps (a)–(e) for the Sinai billiard geometry. ■

Project 6.27 The circle map and mode locking

The driven damped pendulum can be approximated by a one-dimensional difference equation for a range of amplitudes and frequencies of the driving force. This difference equation is known as the *circle map* and is given by

$$\theta_{n+1} = \left(\theta_n + \Omega - \frac{K}{2\pi}\sin 2\pi\theta_n\right) \quad \text{(modulo 1)}. \tag{6.62}$$

The variable θ represents an angle, and Ω represents a frequency ratio, the ratio of the natural frequency of the pendulum to the frequency of the periodic driving force. The parameter K is a measure of the strength of the nonlinear coupling of the pendulum to the external force. An important quantity is the winding number which is defined as

$$W = \lim_{m\to\infty}\frac{1}{m}\sum_{n=0}^{m-1}\Delta\theta_n, \tag{6.63}$$

where $\Delta\theta_n = \Omega - (K/2\pi)\sin 2\pi\theta_n$.

(a) Consider the linear case $K = 0$. Choose $\Omega = 0.4$ and $\theta_0 = 0.2$ and determine W. Verify that if Ω is a ratio of two integers, then $W = \Omega$ and the trajectory is periodic. What is the value of W if $\Omega = \sqrt{2}/2$, an irrational number? Verify that $W = \Omega$ and that the trajectory comes arbitrarily close to any particular value of θ. Does θ_n ever return exactly to its initial value? This type of behavior of the trajectory is termed *quasiperiodic*.

(b) For $K > 0$, we will find that $W \neq \Omega$ and "locks" into rational frequency ratios for a range of values of K and Ω. This type of behavior is called *mode locking*. For $K < 1$, the trajectory is either periodic or quasiperiodic. Determine the value of W for $K = 1/2$ and values of Ω in the range $0 < \Omega \leq 1$. The widths in Ω of the various mode-locked regions where W is fixed increase with K. Consider other values of K, and draw a diagram in the $K\Omega$-plane ($0 \leq K, \Omega \leq 1$) so that those areas corresponding to frequency locking are shaded. These shaded regions are called *Arnold tongues*.

(c) For $K = 1$, all trajectories are frequency-locked periodic trajectories. Fix K at $K = 1$ and determine the dependence of W on Ω. The plot of W versus Ω for $K = 1$ is called the *Devil's staircase*. ■

Project 6.28 Chaotic scattering

In Chapter 5 we discussed the classical scattering of particles off a fixed target, and found that the differential cross section for a variety of interactions is a smoothly varying function

of the scattering angle. That is, a small change in the impact parameter b leads to a small change in the scattering angle θ. Here we consider examples where small changes in b lead to large changes in θ. Such a phenomenon is called *chaotic scattering* because of the sensitivity to initial conditions that is characteristic of chaos. The study of chaotic scattering is relevant to the design of electronic nanostructures, because many experimental structures exhibit this type of scattering.

A typical scattering model consists of a target composed of a group of fixed hard disks and a scatterer consisting of a point particle. The goal is to compute the path of the scatterer as it bounces off the disks and measure θ and the time of flight as a function of the impact parameter b. If a particle bounces inside the target region before leaving, the time of flight can be very long. There are even some trajectories for which the particle never leaves the target region.

Because it is difficult to monitor a trajectory that bounces back and forth between the hard disks, we consider instead a two-dimensional map that contains the key features of chaotic scattering (see Yalcinkaya and Lai for further discussion). The map is given by

$$x_{n+1} = a\left[x_n - \frac{1}{4}(x_n + y_n)^2\right] \tag{6.64a}$$

$$y_{n+1} = \frac{1}{a}\left[y_n + \frac{1}{4}(x_n + y_n)^2\right], \tag{6.64b}$$

where a is a parameter. The target region is centered at the origin. In an actual scattering experiment, the relation between (x_{n+1}, y_{n+1}) and (x_n, y_n) would be much more complicated, but the map (6.64) captures most of the important features of realistic chaotic scattering experiments. The iteration number n is analogous to the number of collisions of the scattered particle off the disks. When x_n or y_n is significantly different from zero, the scatterer has left the target region.

(a) Write a program to iterate the map (6.64). Let $a = 8.0$ and $y_0 = -0.3$. Choose 10^4 initial values of x_0 uniformly distributed in the interval $0 < x_0 < 0.1$. Determine the time $T(x_0)$, the number of iterations for which $x_n \leq -5.0$. After this time, x_n rapidly moves to $-\infty$. Plot $T(x_0)$ versus x_0. Then choose 10^4 initial values in a smaller interval centered about a value of x_0 for which $T(x_0) > 7$. Plot these values of $T(x_0)$ versus x_0. Do you see any evidence of self-similarity?

(b) A trajectory is said to be *uncertain* if a small change ϵ in x_0 leads to a change in $T(x_0)$. We expect that the number of uncertain trajectories N will depend on a power of ϵ, that is, $N \sim \epsilon^{\alpha}$. Determine $N(\epsilon)$ for $\epsilon = 10^{-p}$ with $p = 2$ to 7 using the values of x_0 in part (a). Then determine the uncertainty dimension $1 - \alpha$ from a log-log plot of N versus ϵ. Repeat these measurements for other values of a. Does α depend on a?

(c) Choose 4×10^4 initial conditions in the same interval as in part (a) and determine the number of trajectories $S(n)$ that have not yet reached $x_n = -5$ as a function of the number of iterations n. Plot $\ln S(n)$ versus n and determine if the decay is exponential. It is possible to obtain algebraic decay for values of a less than approximately 6.5.

(d) Let $a = 4.1$ and choose 100 initial conditions uniformly distributed in the region $1.0 < x_0 < 1.05$ and $0.60 < y_0 < 0.65$. Are there any trajectories that are periodic and hence have infinite escape times? Due to the accumulation of roundoff error, it

is possible to find only finite, but very long escape times. These periodic trajectories form closed curves, and the regions enclosed by them are called KAM surfaces. ■

Project 6.29 Chemical reactions

In Project 4.17 we discussed how chemical oscillations can occur when the reactants are continuously replenished. In this project we introduce a set of chemical reactions that exhibits the period doubling route to chaos. Consider the following reactions (see Peng et al.):

$$P \to A \tag{6.65a}$$

$$P + C \to A + C \tag{6.65b}$$

$$A \to B \tag{6.65c}$$

$$A + 2B \to 3B \tag{6.65d}$$

$$B \to C \tag{6.65e}$$

$$C \to D. \tag{6.65f}$$

Each of the above reactions has an associated rate constant. The time dependence of the concentrations of A, B, and C is given by

$$\frac{dA}{dt} = k_1 P + k_2 PC - k_3 A - k_4 A B^2 \tag{6.66a}$$

$$\frac{dB}{dt} = k_3 A + k_4 A B^2 - k_5 B \tag{6.66b}$$

$$\frac{dC}{dt} = k_4 B - k_5 C. \tag{6.66c}$$

We assume that P is held constant by replenishment from an external source. We also assume the chemicals are well mixed so that there is no spatial dependence. In Section 7.8 we discuss the effects of spatial inhomogeneities due to molecular diffusion. Equations (6.65) can be written in a dimensionless form as

$$\frac{dX}{d\tau} = c_1 + c_2 Z - X - XY^2 \tag{6.67a}$$

$$c_3 \frac{dY}{d\tau} = X + XY^2 - Y \tag{6.67b}$$

$$c_4 \frac{dZ}{d\tau} = Y - Z, \tag{6.67c}$$

where the c_i are constants, $\tau = k_3 t$, and X, Y, and Z are proportional to A, B, and C, respectively.

(a) Write a program to solve the coupled differential equations in (6.67). Use a fourth-order Runge–Kutta algorithm with an adaptive step size. Plot $\ln Y$ versus the time τ.

(b) Set $c_1 = 10$, $c_3 = 0.005$, and $c_4 = 0.02$. The constant c_2 is the control parameter. Consider $c_2 = 0.10$ to 0.16 in steps of 0.005. What is the period of $\ln Y$ for each value of c_2?

(c) Determine the values of c_2 at which the period doublings occur for as many period doublings as you can determine. Compute the constant δ (see (6.9)) and compare its value to the value of δ for the logistic map.

(d) Make a bifurcation diagram by taking the values of $\ln Y$ from the Poincaré plot at $X = Z$ and plotting them versus the control parameter c_2. Do you see a sequence of period doublings?

(e) Use three-dimensional graphics to plot the trajectory of (6.67) with $\ln X$, $\ln Y$, and $\ln Z$ as the three axes. Describe the attractors for some of the cases considered in part (a). ■

APPENDIX 6A: STABILITY OF THE FIXED POINTS OF THE LOGISTIC MAP

In the following, we derive analytical expressions for the fixed points of the logistic map. The fixed-point condition is given by

$$x^* = f(x^*). \tag{6.68}$$

From (6.5) this condition yields the two fixed points

$$x^* = 0 \quad \text{and} \quad x^* = 1 - \frac{1}{4r}. \tag{6.69}$$

Because x is restricted to be positive, the only fixed point for $r < 1/4$ is $x = 0$. To determine the stability of x^*, we let

$$x_n = x^* + \epsilon_n, \tag{6.70a}$$

and

$$x_{n+1} = x^* + \epsilon_{n+1}. \tag{6.70b}$$

Because $|\epsilon_n| \ll 1$, we have

$$x_{n+1} = f(x^* + \epsilon_n) \approx f(x^*) + \epsilon_n f'(x^*) = x^* + \epsilon_n f'(x^*). \tag{6.71}$$

If we compare (6.70b) and (6.71), we obtain

$$\epsilon_{n+1}/\epsilon_n = f'(x^*). \tag{6.72}$$

If $|f'(x^*)| > 1$, the trajectory will diverge from x^* because $|\epsilon_{n+1}| > |\epsilon_n|$. The opposite is true for $|f'(x^*)| < 1$. Hence, the local stability criteria for a fixed point x^* are

1. $|f'(x^*)| < 1$, x^* is stable;
2. $|f'(x^*)| = 1$, x^* is marginally stable;
3. $|f'(x^*)| > 1$, x^* is unstable.

a function of functions) that is proportional to ϵ^2 or a higher power of ϵ. An important extremum principle in classical mechanics is based on the action S:

$$S = \int_{t_0}^{t_{\text{final}}} L\, dt, \tag{7.64}$$

where t_0 and t_{final} are the initial and final times, respectively. The Lagrangian L in (7.64) is the kinetic energy minus the potential energy. The extremum principle for the action is known as *the principle of least action* or Hamilton's action principle. The path where (7.64) is stationary (either a minimum or a saddle point) satisfies Newton's second law (for conservative forces). One reason for the importance of the principle of least action is that quantum mechanics can be formulated in terms of an integral over the action (see Section 16.10).

To use (7.64) to find the motion of a single particle in one dimension, we fix the position at the chosen initial and final times, $x(t_0)$ and $x(t_{\text{final}})$, and then choose the velocities and positions for the intermediate times $t_0 < t < t_{\text{final}}$ to minimize the action. One way to implement this procedure numerically is to convert the integral in (7.64) to a sum:

$$S \approx \sum_{i=1}^{N-1} L(t_i)\, \Delta t, \tag{7.65}$$

where $t_i = t_0 + i\Delta t$. (The approximation used to obtain (7.65) is known as the rectangular approximation and is discussed in Chapter 11.) For a single particle in one dimension moving in an external potential $u(x)$, we can write

$$L_i \approx \frac{m}{2(\Delta t)^2}(x_{i+1} - x_i)^2 - u(x_i), \tag{7.66}$$

where m is the mass of the particle, and $u(x_i)$ is the potential energy of the particle at x_i. The velocity has been approximated as the difference in position divided by the change in time Δt.

Problem 7.39 Principle of least action

(a) Write a program to minimize the action S given in (7.64) for the motion of a single particle in one dimension. Use the approximate form of the Lagrangian given in (7.66). One way to write the program is to modify class `Fermat` so that the vertical coordinate for the light ray becomes the position of the particle, and the horizontal region number `i` becomes the discrete time interval of duration Δt.

(b) Verify your program for the case of free fall for which the potential energy is $u(y) = mgy$. Choose $y(t = 0) = 2\,\text{m}$ and $y(t = 10\,\text{s}) = 8\,\text{m}$ and begin with $N = 20$. Allow the maximum change in the position to be 5 m.

(c) Consider the harmonic potential $u(x) = \frac{1}{2}kx^2$. What shape do you expect the path $x(t)$ to be? Increase N to approximately 50 and estimate the path by minimizing the action. ■

It is possible to extend the principle of least action to more dimensions or particles; however, it is necessary to begin with a path close to the optimum one to obtain a good approximation to the optimum path in a reasonable time.

In Problems 7.37–7.39, a simple Monte Carlo algorithm that always accepts paths that reduce the time or action is sufficient. However, for more complicated index of refraction distributions or potentials, it is possible that such a simple algorithm will find only a local minimum, and the global minimum will be missed. The problem of finding the global minimum is very general and is shared by all optimization algorithms if the system has many relative minima. Optimization is a very active area of research in many fields of science and engineering. Ideas from physics, biology, and computer science have led to many improved algorithms. We will discuss some of these algorithms in Chapter 15. In most of these algorithms, paths that are worse than the current path are sometimes accepted in an attempt to climb out of a local minimum. Other algorithms involve ways of sampling over a wider range of possible paths. Another approach is to convert the Monte Carlo algorithm into a deterministic algorithm. We have already mentioned that an analytical variational calculation leads to Newton's second law. Passerone and Parrinello discuss an algorithm for looking for extrema in the action by maintaining the discrete structure in (7.66) and then finding the extremum by taking the derivative with respect to each coordinate x_i and setting the resulting equations equal to zero. This procedure leads to a set of deterministic equations that need to be solved numerically. The performance can be improved by enforcing energy conservation and using some other tricks.

7.11 ■ PROJECTS

Almost all of the problems in this chapter can be done using more efficient programs, greater number of trials, and larger systems. More applications of random walks and random number sequences are discussed in subsequent chapters. Many more ideas for projects can be gained from the references.

Project 7.40 Competition between diffusion and fragmentation

As we have discussed, random walks are useful for understanding diffusion in contexts more general than the movement of a particle. Consider a particle in solution whose mass can grow either by the absorption of particles or shrink by the loss of small particles, including fragmentation. We can model this process as a random walk by replacing the position of the particle by its mass. One difference between this case and the random walks we have studied so far is that the random variable, the mass, must be positive. The model of Ferkinghoff–Berg et al. can be summarized as follows:

(i) Begin with N objects with some distribution of lengths. Let the integer L_i represent the length of the ith object.

(ii) All the objects change their length by ± 1. This step is analogous to a random walk. If the length of an object becomes equal to 0, it is removed from the system. An easy way to eliminate the ith object is to set its length equal to the length of the last object and reduce N by unity.

(iii) Choose one object at random with a probability that is proportional to the length of the object. Fragment this object into two objects, where the fraction of the mass going to each object is random.

(iv) Repeat steps (ii) and (iii).

(a) Write a program to implement this algorithm in one dimension. One way to implement step (iii) is given in the following code, where `totalMass` is the sum of the lengths of all the objects.

```
int i = 0;   // label of object
// length of first object, all lengths are integers
int sum = length[0];
// choose object to fragment so that choice is proportional to
// length
int x = (int)(Math.random()*totalMass);
while(sum < x) {
   i++;
   sum += length[i];
}
// if object big enough to fragment, choose random fraction for
// each part
if(length[i] > 1) {
   int partA = 1 + (int)(Math.random()*(length[i]-1));
   int partB = length[i] - partA;
   length[i] = partA;
   length[numberOfObjects] = partB;  // new object
   numberOfObjects++;
}
```

The main quantity of interest is the distribution of lengths $P(L)$. Explore a variety of initial length distributions with a total mass of 5000 for which the distribution is peaked at about 20 mass units. Is the long time behavior of $P(L)$ similar in shape for any initial distribution? Compute the total mass (sum of the lengths) and output this value periodically. Although the total mass will fluctuate, it should remain approximately constant. Why?

(b) Collect data for three different initial distributions with the same number of objects N, and scale $P(L)$ and L so that the three distributions roughly fall on the same curve. For example, you can scale $P(L)$ so that the maximum of the three distributions has the same value. Then multiply each value of L by a factor so that the distributions overlap.

(c) The analytical results suggest that the universal behavior can be obtained by scaling L by the total mass raised to the $1/3$ power. Is this prediction consistent with your results? Test this hypothesis by adjusting the initial distributions so that they all have the same total mass. Your results for the long time behavior of $P(L)$ should fall on a universal curve. Why is this universality interesting? How can this result be used to analyze different systems? Would you need to do a new simulation for each value of L?

(d) What happens if step (iii) is done more or less often than each random change of length. Does the scaling change? ■

Project 7.41 Application of the pivot algorithm to self-avoiding walks

The algorithms that we have discussed for generating self-avoiding random walks are all based on making *local* deformations of the walk (polymer chain) for a given value of N, the number of bonds. As discussed in Problem 7.31, the time τ between statistically independent configurations is nonzero. The problem is that τ increases with N as some power, for example, $\tau \sim N^3$. This power law dependence of τ on N is called *critical*

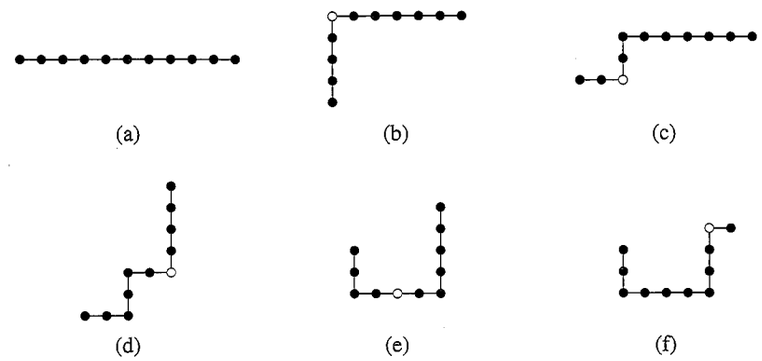


Figure 7.14 Examples of the first several changes generated by the pivot algorithm for a self-avoiding walk of $N = 10$ bonds (11 sites). The open circle denotes the pivot point. This figure is adopted from the article by MacDonald et al.

slowing down and implies that it becomes increasingly more time consuming to generate long walks. We now discuss an example of a *global* algorithm that reduces the dependence of τ on N. Another example of a global algorithm that reduces critical slowing down is discussed in Project 15.32.

(a) Consider the walk shown in Figure 7.14a. Select a site at random and one of the four possible directions. The shorter portion of the walk is rotated (pivoted) to this new direction by treating the walk as a rigid structure. The new walk is accepted only if the new walk is self-avoiding; otherwise, the old walk is retained. (The shorter portion of the walk is chosen to save computer time.) Some typical moves are shown in Figure 7.14. Note that if an end point is chosen, the previous walk is retained. Write a program to implement this algorithm and compute the dependence of the mean square end-to-end distance R^2 on N. Consider values of N in the range $10 \leq N \leq 80$. A discussion of the results and the implementation of the algorithm can be found in MacDonald et al. and Madras and Sokal, respectively.

(b) Compute the correlation time τ for different values of N using the approach discussed in Problem 7.31b. ■

Project 7.42 Pattern formation

In Problem 7.34 we saw that simple patterns can develop as a result of random behavior. The phenomenon of pattern formation is of much interest in a variety of contexts ranging from the large scale structure of the universe to the roll patterns seen in convection (for example, smoke rings). In the following, we explore the patterns that can develop in a simple reaction diffusion model based on the reactions, $A + 2B \rightarrow 3B$ and $B \rightarrow C$, where C is inert. Such a reaction is called *autocatalytic*.

In Problem 7.34 we considered chemical reactions in a closed system where the reactions can proceed to equilibrium. In contrast, open systems allow a continuous supply of fresh reactants and a removal of products. These two processes allow steady states to be realized and oscillatory conditions to be maintained indefinitely. In this problem we assume that A is added at a constant rate, and that both A and B are removed by the feed pro-

distances. Although this particular problem can be overcome by using a faster computer, there are many problems for which no imaginable supercomputer would be sufficient. One of the biggest current challenges is the *protein folding problem*. The biological function of a protein is determined by its three-dimensional structure which is encoded by the sequence of amino acids in the protein. At present, we know little about how the protein forms its three-dimensional structure. Such formidable computational challenges remind us that we cannot simply put a problem on a computer and let the computer tell us the answer. In particular, for many problems, molecular dynamics methods need to be complemented by other simulation methods, especially Monte Carlo methods (see Chapter 15).

The emphasis in current applications of molecular dynamics is shifting from studies of simple equilibrium fluids to studies of more complex fluids and nonequilibrium systems. For example, how does a solid form when the temperature of a liquid is lowered quickly? How does a crack propagate in a brittle solid? What is the nature of the glass transition? Molecular dynamics and related methods will play an important role in aiding our understanding of these and many other problems.

8.12 ■ PROJECTS

Many of the pioneering applications of molecular dynamics were done on relatively small systems. It is interesting to peruse the research literature of the past three decades to see how much physical insight was obtained from these simulations. Many research-level problems can be generated by first reproducing previously published work and then extending the work to larger systems or longer run times to obtain better statistics. Many related projects are discussed in Chapter 15.

Project 8.22 The classical Heisenberg model of magnetism

Magnetism is intrinsically a quantum phenomenon. One common model of magnetism is the Heisenberg model which is defined by the Hamiltonian or energy function:

$$H = -J \sum_{<ij>} \mathbf{S}_i \cdot \mathbf{S}_j, \tag{8.43}$$

where $\mathbf{S}_i$ is the spin operator at the ith lattice site. The sum is over nearest neighbor sites of the lattice, and the (positive) coupling constant J is a measure of the strength of the interaction between spins. The negative sign indicates that the lowest energy state is ferromagnetic. The magnetic moment of a particle on a site is proportional to the particle's spin, and the proportionality constant is absorbed into the constant J.

For many models of magnetism, such as the Ising model (see Section 15.5), there is no obvious dynamics. However, for the Heisenberg model we can motivate a dynamics using the standard rule for the time evolution of an operator given in quantum mechanics texts. For simplicity, we will consider a one-dimensional lattice. The equation for the time development becomes (see Slanič et al.)

$$\frac{d\mathbf{S}_i}{dt} = J\mathbf{S}_i \times (\mathbf{S}_{i-1} + \mathbf{S}_{i+1}). \tag{8.44}$$

In general, $\mathbf{S}$ in (8.44) is an operator. However, if the magnitude of the spin is sufficiently large, the system can be treated classically, and $\mathbf{S}$ can be interpreted as a three-dimensional unit vector. The dynamics in (8.44) conserves the total energy given in (8.43) and the total magnetization, $\mathbf{M} = \sum_i \mathbf{S}_i$.

We can simulate the classical Heisenberg magnet using an ODE solver to solve the first-order differential equation (8.44).

(a) Explain why there is no obvious way to determine the mean temperature of this system.

(b) Write a program to simulate the Heisenberg model on a one-dimensional lattice using periodic boundary conditions. Choose $J = 1$ and $N \geq 100$. Use the RK4 ODE solver, and plot the energy and magnetization as a function of time. These two quantities should be constant within the accuracy of the ODE solver. Also, plot each component of the spin versus position or draw a three-dimensional representation of the spin at each site, so that you can visualize the state of the system.

(c) Begin with all spins in the positive z direction, except for one spin pointing in the negative z direction. Define the energy of spin i as $\epsilon_i = -\mathbf{S}_i \cdot (\mathbf{S}_{i-1} + \mathbf{S}_{i+1})/2$. Use $N \geq 1000$. Plot the local energy ϵ_i as a function of i. Describe how the local energy diffuses. What patterns do you observe? Do the locations of the peaks in the local energy move with a constant speed?

(d) One of the interesting dynamical phenomena we can explore is that of *spin waves*. Begin with all $S_{z,i} = 1$ except for a group of 20 spins, where $S_{x,i} = A \cos ki$, $S_{y,i} = A \sin ki$, and $S_{z,i} = \sqrt{1 - S_{x,i}^2 + S_{y,i}^2}$. Choose $A = 0.2$ and $k = 1$. Describe the motion of the spins. Compute the mean position of this spin wave defined by $x = \sum_i i(1 - S_{z,i})$. Show that x changes linearly with time indicating a constant spin wave velocity. Vary k and A to determine what effect their values have on the speed of the spin wave.

(e) Read about sympletic algorithms in the article by Tsai, Lee, and Landau and write your own ODE solver for one of them. Compare your results to the results you found for the RK4 algorithm. Is the total energy better conserved for the same value of the time step? ■

Project 8.23 Single particle metrics and ergodic behavior

As mentioned in Section 8.7, the quasi-ergodic hypothesis assumes that time averages and ensemble averages are identical for a system in thermodynamic equilibrium. The assumption is that if we run a molecular dynamics simulation for a sufficiently long time, then the dynamical trajectory will fill the accessible phase space.

One way to confirm the quasi-ergodic hypothesis is to compute an ensemble average by simulating many independent copies of the system of interest using different initial configurations. Another way is to simulate a very large system and compare the behavior of different parts. A more direct measure of ergodicity (see Thirumalai and Mountain) is based on a comparison of the time averaged quantity $\overline{f_i(t)}$ of f_i for particle i to its average for all other particles. If the system is ergodic, then all particles see the same average environment, and the time average $\overline{f_i(t)}$ for each particle will be the same if t is sufficiently long. Note that $\overline{f_i(t)}$ is the average of the quantity f_i over the time interval t and not the value of f_i at time t. The time average of f_i is defined as

$$\overline{f_i(t)} = \frac{1}{t} \int_0^t f(t')\, dt', \tag{8.45}$$

and the average of $\overline{f_i(t)}$ over all particles is written as

$$\langle f(t)\rangle = \frac{1}{N}\sum_{i=1}^{N}\overline{f_i(t)}. \tag{8.46}$$

One of the physical quantities of interest is the energy of a particle e_i defined as

$$e_i = \frac{p_i^2}{2m_i} + \frac{1}{2}\sum_{i\neq j} u(r_{ij}). \tag{8.47}$$

The factor of $1/2$ is included in the potential energy term in (8.47) because the interaction energy is shared between pairs of particles. The above considerations lead us to define the energy *metric*, $\Omega_e(t)$, as

$$\Omega_e(t) = \frac{1}{N}\sum_{i=1}^{N}[\overline{e_i(t)} - \langle e(t)\rangle]^2. \tag{8.48}$$

(a) Compute $\Omega_e(t)$ for a system of Lennard–Jones particles at a relatively high temperature. Determine $e_i(t)$ at time intervals of 0.5 or less and average Ω_e over as many time origins as possible. If the system is ergodic over the time interval t, then it can be shown that $\Omega_e(t)$ decreases as $1/t$. Plot $1/\Omega_e(t)$ versus t. Do you find that $1/\Omega_e(t)$ eventually behaves linearly with t? Nonergodic behavior might be found by rapidly reducing the kinetic energy (a temperature quench) and obtaining an amorphous solid or glass rather than a crystalline solid. However, it would be necessary to consider three-dimensional rather than two-dimensional systems because the latter system forms a crystalline solid very quickly.

(b) Another quantity of interest is the velocity metric Ω_v:

$$\Omega_v(t) = \frac{1}{dN}\sum_{i=1}^{N}[\overline{\mathbf{v}_i(t)} - \langle \mathbf{v}(t)\rangle]^2. \tag{8.49}$$

The factor of $1/d$ in (8.49) is included because the velocity is a vector with d components. If we choose the total momentum of the system to be zero, then $\langle \mathbf{v}(t)\rangle = 0$, and we can write (8.49) as

$$\Omega_v(t) = \frac{1}{dN}\sum_{i=1}^{N}\overline{\mathbf{v}_i(t)}\cdot\overline{\mathbf{v}_i(t)}. \tag{8.50}$$

As we will see, the time dependence of $\Omega_v(t)$ is not a good indicator of ergodicity, but can be used to determine the diffusion coefficient D. We write

$$\overline{\mathbf{v}_i(t)} = \frac{1}{t}\int_0^t \mathbf{v}_i(t')\,dt' = \frac{1}{t}[\mathbf{r}_i(t) - \mathbf{r}_i(0)]. \tag{8.51}$$

If we substitute (8.51) into (8.50), we can express the velocity metric in terms of the mean square displacement:

$$\Omega_v(t) = \frac{1}{dNt^2}\sum_{i=1}^{N}[\mathbf{r}_i(t) - \mathbf{r}_i(0)]^2 = \frac{\overline{R^2(t)}}{dt^2}. \tag{8.52}$$

The average in (8.52) is over all particles. If the particles are diffusing during the time interval t, then $\overline{R^2(t)} = 2dDt$ and

$$\Omega_v(t) = 2D/t. \tag{8.53}$$

From (8.53) we see that $\Omega_v(t)$ goes to zero as $1/t$ as claimed in part (a). However, if the particles are localized (as in a crystalline solid and a glass), then $\overline{R^2}$ is bounded for all t and $\Omega_v(t) \sim 1/t^2$. Because a crystalline solid is ergodic and a glass is not, the velocity metric is not a good measure of the lack of ergodicity. Use the t-dependence of $\Omega_v(t)$ in (8.53) to determine D for the same configurations as in Problem 8.19. ■

Project 8.24 Constant temperature molecular dynamics

In the molecular dynamics simulations we have discussed so far, the energy is constant up to truncation, discretization, and floating point errors, and the temperature fluctuates. However, sometimes it is more convenient to do simulations at constant temperature. In Chapter 15 we will see how to simulate systems at constant T, V, and N (the canonical ensemble) by using Monte Carlo methods. However, we can also do constant temperature simulations by modifying the dynamics.

A crude way of maintaining a constant temperature is to rescale the velocities after every time step to keep the mean kinetic energy per particle constant. This approach is equivalent to a constant temperature simulation when $N \to \infty$. However, the fluctuations of the kinetic energy can be non-negligible in small systems. For such systems keeping the total kinetic energy constant in this way is not equivalent to a constant temperature simulation.

A better way of maintaining a constant temperature is based on imagining that every particle in the system is connected to a much larger system called a *heat bath*. The heat bath is sufficiently large so that it has a constant temperature even if it loses or gains energy. The particles in the system of interest occasionally collide with particles in this heat bath. The effect of these collisions is to give the particles random velocities with the desired probability distribution (see Problem 8.6). We first list the algorithm and give its rationale later. Add the following statements to method `step` after all the particles have been moved.

Listing 8.22 Andersen thermostat.

```
for (int i = 0; i < N; i++) {
   if (Math.random() < collisionProbability) {
      double r1 = Math.random();
      double r2 = Math.random()*2.0*Math.PI;
      state[4*i+1] =
           Math.sqrt(-2.0*temperature*Math.log(r1))*Math.cos(r2); // vx
      state[4*i+3] =
           Math.sqrt(-2.0*temperature*Math.log(r1))*Math.sin(r2); // vy
   }
}
```

The parameter `collisionProbability` is much less than unity and determines how often there is a collision with the heat bath. This way of maintaining constant temperature is known as the *Andersen thermostat*.

(a) Do a constant energy simulation as before, using an initial configuration for which the desired temperature is equal to 1.0. Make sure the total momentum is zero. Choose

$N = 64$ and place the particles initially on a triangular lattice with $L_x = 10$ and $L_y = \sqrt{3}L_x/2$. Plot the instantaneous temperature defined as in (8.5) and compute the average temperature. Estimate the magnitude of the temperature fluctuations. Repeat your simulation for some other initial configurations.

(b) Modify your program to use the Andersen thermostat at a constant temperature set equal to 1.0. Set `collisionProbability` $= 0.0001$. Repeat the calculations of part (a) and compare them. Discuss the differences. Do the results change significantly?

(c) Modify your program to do a simple constant kinetic energy ensemble where the velocities are rescaled after every time step so that the total kinetic energy does not change. What is the final temperature now? How do your results compare with parts (a) and (b)? Are the differences in the computed thermodynamic averages statistically significant?

(d) Compute the velocity probability distribution for each case. How do they compare? Consider `collisionProbability` $= 0.001$ and 0.00001.

(e) A deterministic algorithm for constant temperature molecular dynamics is the Nosé–Hoover thermostat. The idea is to introduce an additional degree of freedom s that plays the role of the heat bath. The derivation of the appropriate equations of motion is an excellent example of the Lagrangian formulation of mechanics. The equations of motion of Nosé–Hoover dynamics are

$$\frac{d\mathbf{p}_i}{dt} = \mathbf{F}_i(t) - s\mathbf{p}_i \tag{8.54}$$

$$\frac{ds}{dt} = \frac{1}{M}\left[\sum_i \frac{p_i^2}{m_i} - dNkT\right], \tag{8.55}$$

where T is the desired temperature, and M is a parameter that can be interpreted as the mass associated with the extra degree of freedom. Equation (8.54) is similar to Newton's equations of motion with an additional friction term. However, the coefficient s can be positive or negative. Equation (8.55) defines the way s is changed to control the temperature. Apply the Nosé–Hoover algorithm to simulate a simple harmonic oscillator at constant temperature. Plot the phase space trajectory. If the energy was constant, the trajectory would be an ellipse. How does the shape of the trajectory depend on M? Choose M so that the period of any oscillations due to the finite value of M is much longer than the period of the system. ■

Project 8.25 Simulations on the surface of a sphere

Because of the long-range nature of the Coulomb potential, we have to sum all the periodic images of the particles to compute the force on a given particle. Although there are special methods to do these sums so that they converge quickly (Ewald sums), the simulation of systems of charged particles is more difficult than systems with short-range potentials. An alternative approach that avoids periodic boundary conditions is to not have any boundaries at all. For example, if we wish to simulate a two-dimensional system, we can consider the motion of the particles on the surface of a sphere. If the radius of the sphere is sufficiently large, the curvature of the surface can be neglected. Of course, there is a price—the coordinate system is no longer Cartesian.

Although this approach can also be applied to systems with short-range interactions, it is more interesting to apply it to charged particles. The simplest system of interest is a model of charged particles moving in a uniform background of opposite charge to ensure overall charge neutrality, the one-component plasma (OCP). In two dimensions this system is a simplified model of electrons on the surface of liquid Helium. The properties of the OCP are determined by the dimensionless parameter Γ given by the ratio of the potential energy between nearest neighbor particles to the mean kinetic energy of a particle, $\Gamma = (e^2/a)/kT$, where $\rho\pi a^2 = 1$ and ρ is the number density. Systems with $\Gamma \gg 1$ are called strongly coupled. For $\Gamma \sim 100$ in two dimensions, the system forms a solid. Strongly coupled one-component plasmas in three dimensions are models of dense astrophysical matter.

Assume that the origin of the coordinate system is at the center of the sphere and that $\mathbf{u}_i$ is a unit vector from the origin to the position of particle i on the sphere. Then $R\theta_{ij}$ is the length of the chord joining particle i and j, where $\cos\theta_{ij} = \mathbf{u}_i \cdot \mathbf{u}_j$. Newton's equation of motion for the ith electron has the form

$$m\ddot{\mathbf{u}}_i = -\frac{e^2}{R^2}\sum_{j\neq i}\frac{1}{\theta_{ij}^2 \sin\theta_{ij}}[\mathbf{u}_j - \cos\theta_{ij}\mathbf{u}_i]. \tag{8.56}$$

Note that the unit vector $\mathbf{w}_{ij} = [\mathbf{u}_j - \cos\theta_{ij}\mathbf{u}_i]/\sin\theta_{ij}$ is orthogonal to $\mathbf{u}_i$. In addition, we must take into account that the particles must stay on the surface of the sphere, so there is an additional force on particle i toward the center of magnitude $m|\dot{\mathbf{u}}_i|^2/R$.

(a) What are the appropriate units for length, time, and the self-diffusion constant?

(b) Write a program to compute the velocity correlation function given by

$$C(t) = \frac{1}{v_0^2}\overline{\dot{\mathbf{u}}(t)\cdot\dot{\mathbf{u}}(0)}, \tag{8.57}$$

where $v_0^2 = \overline{\dot{\mathbf{u}}(0)\cdot\dot{\mathbf{u}}(0)}$. To compute the self-diffusion constant D, we let $\cos\theta(t) = \mathbf{u}(t)\cdot\mathbf{u}(0)$, so that $R\theta$ is the circular arc from the initial position of a particle to its position on the sphere at time t. We then define

$$D(t) = \frac{1}{a^2}\frac{\overline{\theta^2(t)}}{4t}, \tag{8.58}$$

where D and t are dimensionless variables. The self-diffusion constant D corresponds to the limit $t \to \infty$. Choose $N = 104$ and a radius R corresponding to $\Gamma \approx 36$ as in the original simulations by Hansen et al. and then consider bigger systems. Can you conclude that the self-diffusion exists for the two-dimensional OCP?

*(c) Use a similar procedure to compute the velocity autocorrelation function and the self-difusion constant D for a two-dimensional system of Lennard–Jones particles. Can you conclude that the self-diffusion exists for this two-dimensional system? ■

Project 8.26 Granular matter

Recently, physicists have become very interested in granular matter such as sand. The key difference between molecular systems and granular systems is that the interparticle

interactions in the latter are inelastic. The lost energy goes into the internal degrees of freedom of a grain and ultimately is dissipated. From the point of view of the motion of the granular particles, the energy is lost. Experimentalists have studied models of granular material composed of small steel balls or glass beads using sophisticated imaging techniques that can track the motion of individual particles. There have also been many complementary computer simulation studies.

What are some of the interesting properties of granular matter? Because the interactions are inelastic, granular particles will ultimately come to rest unless there is an external source of energy, usually a vibrating wall or gravity (for example, the fall of particles through a funnel). When granular particles come to rest, they can form a granular solid that is different than molecular solids. One difference is that there frequently exists a complex network of force lines within the solid. In addition, unlike ordinary liquids, the pressure does not increase with depth because the walls of the container help support the grains. As a consequence, sand flowing out of an aperture flows at a constant rate independent of the height of the sand above the aperture. For this reason sand is used in hour glasses. Another interesting property is that under some conditions, the large grains in a mixture of large and small grains can move to the top while the container is being vibrated—the "Brazil nut" effect. Under other conditions, the large grains might move to the bottom. What happens depends on the size and density of the large grains compared to the small grains (see Sanders et al.).

It is also known that there is a critical angle for the slope of a sand pile, above which the sand pile is unstable. This slope is called the angle of repose. These and many other effects have been studied using theoretical, computational, and experimental techniques.

The first step in simulating granular matter is to determine the effective force law between particles. For granular gases the details of the force do not influence the qualitative results, as long as the force is purely repulsive and short range, and there is some mechanism for dissipating energy. Common examples of force laws are spring-like forces with stiff spring constants and hard disks with inelastic collisions. For simplicity, we will consider the Lennard–Jones potential with a cut off at $r_c = 2^{1/6}$ so that the force is always repulsive. To remove energy during a collision, we will introduce a viscous damping force given by

$$f_{ij} = -\gamma(\mathbf{v}_{ij} \cdot \mathbf{r}_{ij})\frac{\mathbf{r}_{ij}}{r_{ii}^2}, \tag{8.59}$$

where the viscous damping coefficient γ equals 100 in reduced units. A more realistic force model necessary for granular flow problems is given in Hirchfeld et al.

(a) Modify class `LJParticles` so that the cutoff is at $2^{1/6}$. Is the total energy conserved? Include a viscous damping force as in (8.59) and plot the kinetic energy per particle versus time. We will define the kinetic temperature to be the mean kinetic energy per particle. Why does this definition of temperature not have the same significance as the temperature in molecular systems in which the energy is conserved? Choose $N = 64$, $L = 20$, and $\Delta t = 0.001$. Begin with a random configuration and initial kinetic temperature equal to 10. How long does it take for the kinetic temperature to decrease to 10% of its initial value? Describe the spatial distribution of the particles at this time.

(b) Compute the mean kinetic temperature versus time averaged over three runs. What functional form describes your results for the mean kinetic temperature at long times?

(c) To prevent "granular collapse" where the particles ultimately come to rest, we need to add energy to the system. The simplest way of doing so is to give random kicks to randomly selected particles. You can use the same algorithm we used to set the initial velocities in `LJParticles`:

```
int i = (int)(N*math.random();  // selects random particle
// use to generate Gaussian distribution
double r = Math.random();
double a = -Math.log(r);
double theta = 2.0*Math.PI*Math.random();
// assign velocities according to Maxwell-Boltzmann distribution
// using Box-Muller method
state[4*i+1] = Math.sqrt(2.0*desiredKE*a)*Math.cos(theta);    // vx
state[4*i+3] = Math.sqrt(2.0*desiredKE*a)*Math.sin(theta);    // vy
```

(The Box–Mueller method is described in Section 11.5.) Assume that at each time step one particle is chosen at random and receives a random kick. Adjust `desiredKE` so that the mean kinetic energy per particle remains roughly constant at about 5.0. Compute the velocity distribution function for each component of the velocity. Compare this distribution on the same plot to the Gaussian distribution:

$$p(v_x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(v_x - \langle v_x \rangle)^2/2\sigma^2}, \tag{8.60}$$

where $\sigma^2 = \langle v_x^2 \rangle - \langle v_x \rangle^2$. Is the velocity distribution function of the system a Gaussian? If not, give a physical explanation for the difference. ■

APPENDIX 8A: READING AND SAVING CONFIGURATIONS

For most of the problems in this chapter, qualitative results can be obtained fairly quickly. However, in research applications, the time for running a simulation is likely to be much longer than a few minutes, and runs that require days or even months are not uncommon. In such cases it is important to be able to save the intermediate configurations to prevent the potential loss of data in the case of a computer crash or power failure.

Also, in many cases it is easier to save the configurations periodically and then use a separate program to analyze the configurations and compute the quantities of interest. In addition, if we wish to compute averages as a function of a parameter, such as the temperature, it is convenient to make small changes in the temperature and use the last configuration from the previous run as the initial configuration for the simulation at the new temperature.

The standard Java API has methods for reading and writing files. The usual way of saving a configuration is to use these methods to simply write all the positions and velocities as numbers into a file. Additional simulation parameters and information about the configuration would be saved using a custom format. Although this approach is the traditional one for data storage, the use of a custom format means that you might not remember the format later, and sharing data between programs and other users becomes more difficult.

An alternative is to use a more structured and widely shared format for storing data. The Open Source Physics library has support for the Extensive Markup Language (XML). The XML format offers a number of advantages for computational physics: clear markup of

(e) How much must you change the factor `dampingCoef` in the damping method before you can visually see a difference in the simulation? What problems occur when the damping is removed?

*(f) The amplitude of the fields far from the current loop should be characteristic of radiation fields for which the amplitude falls off as $1/r$, where r is the distance from the current loop to the observation point. Do a simulation to detect this dependence if you have sufficient computer resources. ■

Problem 10.24 Microwave cavity resonators

(a) Cavity resonators are a practical way of storing energy in the form of oscillating electric and magnetic fields without losing as much energy as would be dissipated in a resonant LC circuit. Consider a cubical resonator of linear dimension L whose walls are made of a perfectly conducting material. The tangential components of $\mathbf{E}$ and the normal component of $\mathbf{B}$ vanish at the walls. Standing microwaves can be set up in the box of the form (cf. Reitz et al.)

$$E_x = E_{x0} \cos k_x x \sin k_y y \sin k_z z \, e^{i\omega t} \tag{10.59a}$$

$$E_y = E_{y0} \cos k_y y \sin k_x x \sin k_z z \, e^{i\omega t} \tag{10.59b}$$

$$E_z = E_{z0} \cos k_z z \sin k_x x \sin k_y y \, e^{i\omega t}. \tag{10.59c}$$

The wave vector $\mathbf{k} = (k_x, k_y, k_z) = (m_x\pi/L, m_y\pi/L, m_z\pi/L)$, where m_x, m_y, and m_z are integers. A particular mode is labeled by the integers (m_x, m_y, m_z). The initial electric field is perpendicular to $\mathbf{k}$, and $\omega = ck$. Implement the boundary conditions at $(x = 0, y = 0, z = 0)$ and $(x = L, y = L, z = L)$. Set $\Delta t = 0.05$, $\Delta l = 0.1$, and $L = 1$. At $t = 0$, set $\mathbf{B} = 0$, $\mathbf{j} = 0$ (there are no currents within the cavity), and use (10.59) with $(m_x, m_y, m_z) = (0, 1, 1)$ and $E_{x0} = 1$. Plot the field components at specific positions as a function of t and find the resonant frequency ω. Compare your computed value of ω with the analytic result. Do the magnetic fields change with time? Are they perpendicular to $\mathbf{k}$ and $\mathbf{E}$?

(b) Repeat part (a) for two other modes.

(c) Repeat part (a) with a uniform random noise added to the initial field at all positions. Assume the amplitude of the noise is δ and describe the resulting fields for $\delta = 0.1$. Are they similar to those without noise? What happens for $\delta = 0.5$? More quantitative results can be found by computing the power spectrum $|E(\omega)|^2$ for the electric field at a few positions. What is the order of magnitude of δ for which the maximum of $|E(\omega)|^2$ at the standing wave frequency is swamped by the noise?

(d) Change the shape of the container slightly by removing a 0.1×0.1 cubical box from each of the corners of the original resonator. Do the standing wave frequencies change? Determine the standing wave frequency by adding noise to the initial fields and looking at the power spectrum. How do the standing wave patterns change?

(e) Change the shape of the container slightly by adding a 0.1×0.1 cubical box at the center of one of the faces of the original resonator. Do the standing wave frequencies change? How do the standing wave patterns change?

(f) Cut a 0.2×0.2 square hole in a face in the yz-plane and double the computational region in the x direction. Begin with a (0, 1, 1) standing wave and observe how the fields "leak" out of the hole. ■

Problem 10.25 Billiard microwave cavity resonators

(a) Repeat Problem 10.24a for $L_x = L_y = 2$, $L_z = 0.2$, $\Delta l = 0.1$, and $\Delta t = 0.05$. Indicate the magnitude of the electric field in the $L_z = 0.1$ plane by a color code. Choose an initial normal mode field distribution and describe the pattern that you obtain. Then repeat your calculation for a random initial field distribution.

(b) Place an approximately circular conductor in the middle of the cavity of radius $r = 0.4$. Describe the patterns that you see. Such a geometry leads to chaotic trajectories for particles moving within such a cavity (see Project 6.26). Is there any evidence of chaotic behavior in the field pattern?

(c) Repeat part (b) with the circular conductor placed off center. ■

10.9 ■ PROJECTS

Part of the difficulty in understanding electromagnetic phenomena is visualizing its three-dimensional nature. Many interesting problems can be posed based on the simple, but nontrivial question of how three-dimensional electromagnetic fields can be best represented visually in various contexts (cf. Belcher and Olbert). However, we have not suggested projects in this area because of their difficulty.

Many of the techniques used in this chapter, for example, the random walk method and the relaxation method for solving Laplace's equation, have applications in other fields, especially problems in fluid flow and transport. Similarly, the multigrid method, discussed in Project 10.26, has far reaching applications.

Project 10.26 Multigrid method

In general, the relaxation method for solving Laplace's equation is very slow even when using overrelaxation. The reason is that the local updates of the relaxation method cannot quickly take into account effects at very large length scales. The *multigrid method* greatly improves performance by using relaxation at many length scales. The important idea is to use a relaxation method to find the values of the potential on coarser and coarser grids, and then to use the coarse grid values to determine the fine grid values. The fine grid relaxation updates take into account effects at short length scales. If we define the initial grid by a lattice spacing $b = 1$, then the coarser grids are characterized by $b = 2^n$, where n determines the coarseness of the grid and is known as the grid level. We need to decide how to use the fine grid values of the potential to assign values to a coarser grid, and then how to use a coarse grid to assign values to a finer grid. The first step is called *prolongation* and the second step is called *restriction*. There is some flexibility on how to do these two operations. We discuss one approach.

We define the centers of the sites of the coarse grid to be located at the centers of every other site of the fine grid. That is, if the set $\{i, j\}$ represents the positions of the sites of the

fine grid, then $\{2i, 2j\}$ represents the positions of the coarse grid sites. The fine grid sites that are at the same position as a coarse grid point are assigned the value of the potential of the corresponding coarse grid point. The fine grid sites that have two coarse grid points as nearest neighbors are assigned the average value of these two coarse grid sites. The other fine grid sites have four coarse grid sites as next nearest neighbors and are assigned the average value of these four coarse grid sites. This prescription specifies how values on the fine grid are computed using the values on the coarse grid.

In the full weighting prolongation method, each coarse grid site receives one fourth of the potential of the fine grid site at the same position, one eighth of the potential for the four nearest neighbor sites of the fine grid, and one sixteenth of the potential for the four next nearest neighbor points of the fine grid. The sum of these fractions, $1/4 + 4(1/8) + 4(1/16)$, adds up to unity. An alternative procedure, known as half weighting, ignores the next nearest neighbors and uses one half of the potential of the fine grid site at the same position as the coarse grid site.

(a) Write a program that implements the multigrid method using Gauss–Seidel relaxation on a checkerboard lattice (see Problem 10.11b). In its simplest form, the program should allow the user to intervene and decide whether to go to a finer or coarser grid or to remain at the same level for the next relaxation step. Have the program print the potential at each site of the current level after each relaxation step. Test your program on a 4×4 grid whose boundary sites are all equal to unity and whose initial internal sites are set to zero. Make sure that the boundary sites of the coarser grids are also set to unity.

(b) The exact solution for part (a) gives a potential of unity at each point. How many relaxation steps does it take to reach unity within 0.1% at every site by simply using the 4×4 grid? How many steps does it take if you use one coarse grid and continue until the coarse grid values are within 0.1% of unity? Is it necessary to carry out any fine grid relaxation steps to reach the desired accuracy on the fine grid? Next start with the coarsest scale, which is just one site. How many relaxation steps does it take now?

(c) Repeat part (b) but change the boundary so that one side of the boundary is held at a potential of 0.5. Experiment with different sequences of prolongation, restriction, and relaxation.

(d) Assume that the boundary points alternate between zero and unity and repeat part (b). Does the multigrid method work? Should one go up and down in levels many times instead of staying at the coarsest level and then going down to the finest level? ■

APPENDIX 10A: VECTOR FIELDS

The frames package contains the `Vector2DFrame` class for displaying two-dimensional vector fields. To use this class we instantiate a multidimensional array to store components of the vector. The first array index indicates the component, the second index indicates the column or x-position, and the third index indicates the row or y-position. The vectors in the visualization are set by passing the data array to the frame using the `setAll` method. The program in Listing 10.10 demonstrates how this is done by displaying the electric field of a unit charge located at the origin.

Listing 10.10 A vector field test program.

```
package org.opensourcephysics.sip.ch10;
import javax.swing.JFrame;
import org.opensourcephysics.frames.Vector2DFrame;

public class VectorPlotApp {
   public static void main(String[] args) {
      Vector2DFrame frame =
           new Vector2DFrame("x", "y", "Vector field");
      double a = 2; // half width of frame in world coordinates
      frame.setPreferredMinMax(-a, a, -a, a);
      int nx = 15, ny = 15; // grid sizes in x and y direction
      // generate sample data
      double[][][] vectorField = new double[2][nx][ny];
      frame.setAll(vectorField); // vector field displays zero data
      for(int i = 0;i<nx;i++) {
         double x = frame.indexToX(i);
         for(int j = 0;j<ny;j++) {
            double y = frame.indexToY(j);
            double r2 = x*x+y*y;            // distance squared
            double r3 = Math.sqrt(r2)*r2; // distance cubed
            vectorField[0][i][j] = (r2==0) ? 0 : x/r3; // x component
            vectorField[1][i][j] = (r2==0) ? 0 : y/r3; // y component
         }
      }
      frame.setAll(vectorField); // vector field displays new data
      frame.setVisible(true);
      frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
   }
}
```

The arrows in the visualization have a fixed length that is chosen to fill the viewing area. The arrow's color represents the field's magnitude. We have found that using an arrow's color rather than its length to represent field strength produces a more effective representation of vector fields over a wider dynamic range. The frame's Legend menu item under Tools shows this mapping. The appropriate representation of vector fields is an active area of interest.

Problem 10.27 Gradient of a scalar field

The gradient of a scalar field, $A(x, y)$, defines a vector field. In a two-dimensional Cartesian coordinate system, the components of the gradient are equal to the derivative of the scalar field along the x- and y-axes, respectively:

$$\nabla A = \frac{\partial A}{\partial x}\hat{\mathbf{x}} + \frac{\partial A}{\partial y}\hat{\mathbf{y}}. \tag{10.60}$$

Write a short program that displays both a scalar field and its gradient. (Hint: Define a function and use numerical derivatives along the rows and columns.) Create separate frames for the scalar and vector field visualizations. *Open Source Physics: A User's Guide with Examples* describes how a vector field visualization can be superimposed on a scalar field visualization. ■

to redo many of these applications with much better statistics and with larger system sizes. In the following, we discuss some additional recent developments, but we have omitted other important topics such as Brownian dynamics and umbrella sampling. More ideas for projects can be found in the references.

Project 15.32 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. Near a continuous 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 we have 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 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 $T = T_c$, $\tau \sim L^z$ for $L \gg 1$. For single spin flip algorithms, $z \approx 2$ and τ becomes very large for $L \gg 1$. 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 expected and not a problem because a larger system contains proportionally more information. However, the time needed to obtain n approximately *independent* configurations is of order $\tau L^2 \sim L^{2+z} \approx L^4$ for the Metropolis algorithm. We conclude 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 (15.18). It is reasonable to look for a *global* algorithm for which groups or *clusters* of spins are flipped simultaneously. We are already familiar with cluster properties in the context of percolation (see Chapter 12). A naive definition of a cluster of spins might be a domain of parallel nearest neighbor spins. We can make this definition explicit by introducing a bond between any two nearest neighbor spins that are parallel. The introduction of a bond between parallel spins defines a site-bond percolation problem. More generally, we may assume that such a bond exists with probability p and that 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 we define the bond probability as

$$p = 1 - e^{-2J/kT} \quad \text{(bond probability)}. \tag{15.79}$$

The relation (15.79) 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 (15.79) is as follows. Consider the two configurations in Figure 15.9 which differ from one another by the flip of the cluster of two spins. In Figure 15.9(a) the six nearest neighbor spins of the cluster are in the opposite

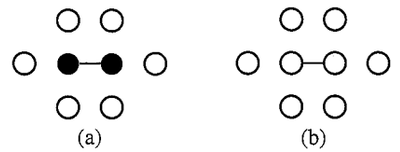


Figure 15.9 (a) A cluster of two up spins. (b) A cluster of two down spins. The filled and open circles represent the up and down spins, respectively. Note the bond between the two spins in the cluster. Adapted from Newman and Barkema.

direction and, hence, are not part of the cluster. Thus, the probability of this configuration with a cluster of two spins is $p\, e^{-\beta J} e^{6\beta J}$, 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 are parallel, and $e^{6\beta J}$ is proportional to the probability that the six nearest neighbors are antiparallel. In Figure 15.9(b) the cluster spins have been flipped, and the possible bonds between the cluster spins and its nearest neighbors have to be "broken." 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 and their corresponding clusters be the same. Hence, we must have

$$p e^{-\beta J} e^{6\beta J} = p(1-p)^6 e^{-\beta J} e^{-6\beta J}, \tag{15.80}$$

or $(1-p)^6 = e^{-12\beta J}$. It is straightforward to solve for p and obtain the relation (15.79).

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:

(i) 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.

(ii) 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 $r \leq p$, the bond exists between the two spins, and the perimeter spin is added to the cluster.

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

(iv) Repeat steps (ii) and (iii) until no perimeter spins remain.

(v) 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 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 13.3). How does n_s depend on s for large s (see Project 13.15)? What is the fractal dimension of the clusters in the Ising model at $T = T_c$?

(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 $C(t_{\rm cf})$. Estimate the value of $z_{\rm cf}$ from the relation $\tau_{\rm cf} = L^{z_{\rm cf}}$.

(e) 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. Verify that the mean cluster size scales as $\langle c\rangle \sim L^{\gamma/\nu}$ with $\gamma = 7/4$ and $\nu = 1$. (The quantity $\langle c\rangle$ is the same quantity as the mean cluster size S defined in Chapter 12. The exponents characterizing the divergence of the various properties of the clusters are identical to the analogous thermal exponents.)

(f) 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 (e). 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 $z = z_{\rm cf} - (d - \gamma/\nu)$. What is your estimated value of z? (It has been estimated that $z_{\rm cf} \approx 0.50$ for the $d = 2$ Ising model, which would imply that $z \approx 0.25$.)

(g) 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 these f spins, and accepting or rejecting the trial move in the usual way according to the Boltzmann probability. Explain why this generalization of the Metropolis algorithm would be very inefficient, especially if $f \gg 1$. We conclude that the groups of spins to be flipped must be chosen with the physics of the system in mind and not simply at random. ■

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 orientations. From this configuration of bonds, we can form clusters of spins using one of the cluster identification algorithms we discussed in Chapter 12. 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 15.33 Invaded cluster algorithm

In Problem 13.7 we found that invasion percolation is an example of a self-organized critical phenomenon. In this cluster growth algorithm, random numbers are independently assigned to the bonds of a lattice. The growth starts from the seed sites of the left-most column. At each step the cluster grows by the occupation of the perimeter bond with the smallest random number. The growth continues until the cluster satisfies a stopping condition. We found that if we stop adding sites when the cluster is comparable in extent to the linear dimension L, then the fraction of bonds that are occupied approaches the percolation threshold p_c as $L \to \infty$. The invaded percolation algorithm automatically finds the percolation threshold!

Machta and co-workers have used this idea to find the critical temperature of a spin system without knowing its value in advance. For simplicity, we will discuss their algorithm in the context of the Ising model, although it can be easily generalized to the q-state Potts model (see the references). Consider a lattice on which there is a spin configuration $\{s_i\}$. The bonds of the lattice are assigned a random order. Bonds (i, j) are tested in this assigned order to see if s_i is parallel to s_j. If so, the bond is occupied and spins i and j are a part of the same cluster. Otherwise, the bond is not occupied and is not considered for the remainder of the current Monte Carlo step. The set of occupied bonds partitions the lattice into clusters of connected sites. The clusters can be found using the Newman–Ziff algorithm (see Section 12.3). The cluster structure evolves until a stopping condition is satisfied. Then a new spin configuration is obtained by flipping each cluster with probability 1/2, thus completing one Monte Carlo step. The fraction f of bonds that were occupied during the growth process and the energy of the system are measured. The bonds are then randomly reordered and the process begins again. Note that the temperature is not an input parameter.

If open boundary conditions are used, the appropriate stopping rule is that a cluster spans the lattice (see Chapter 12, page 457). For periodic boundary conditions, the spanning rule discussed in Project 12.17 is appropriate.

Write a program to simulate the invaded cluster algorithm for the Ising model on the square lattice. Start with all spins up and determine how many Monte Carlo steps are needed for equilibration. How does this number compare to that required by the Metropolis algorithm at the critical temperature for the same value of L? An estimate for the critical temperature can be found from the relation (15.79) with f corresponding to p.

After you are satisfied that your program is working properly, determine the dependence of the critical temperature on the concentration c of nonmagnetic impurities. That is, randomly place nonmagnetic impurities on a fraction c of the sites. ■

Project 15.34 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 (the 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

$$x_n = 16{,}807 x_{n-1} \bmod (2^{31} - 1), \tag{15.81}$$

and the generalized feedback shift register (GFSR) algorithm

$$x_n = x_{n-103} \oplus x_{n-250}. \tag{15.82}$$

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 (b) and (c) using the random number generator supplied with your programming language. ■

Project 15.35 Nucleation and the Ising model

(a) Equilibrate the two-dimensional Ising model at $T = 4T_c/9$ and $B = 0.3$ 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, $B = -0.3$. 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 15.32, we can define clusters of spins by placing a bond with probability p between parallel spins. In this case there is an external field and the proper definition of the clusters is more difficult. For simplicity, assume that there is a bond between all nearest neighbor down spins and find all the clusters of down spins. One way to identify the droplet that initiates the decay of the metastable state 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 criterion for the nucleation time. What is the order of magnitude of the nucleation time?

(c) 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 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 at about the time that nucleation is thought to have occurred. We then restart the simulation using a saved configuration at a certain time and use a different random number sequence to flip the spins. If we have intervened at a time such that the largest cluster decays in more than 50% of the trials, then the intervention time (the time at which we changed the random number seed) 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 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 nucleation time. What is the nature and size of the nucleating droplet? If time permits, determine the probability that the system nucleates at time t for a given quench depth. (Measure the time t after the flip of the field.)

(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 on and off the impurity as a function of quench depth B?

(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 15.36 The n-fold way: Simulations at low temperature

Monte Carlo simulations become very inefficient at low temperatures because almost all trial configurations will be rejected. For example, consider an Ising model for which all spins are up, but a small magnetic field is applied in the negative direction. The equilibrium state will have most spins pointing down. Nevertheless, if the magnetic field is small and the temperature is low enough, equilibrium will take a very long time to occur.

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

$$p_i = \min[1, e^{-\Delta E/kT}]. \tag{15.83}$$

One limitation of the Metropolis algorithm is that it becomes very inefficeint if the probabilities p_i are very small. If we sum over all the spins, then we can define the total weight

$$Q = \sum_i p_i. \tag{15.84}$$

The idea is to choose a spin to flip (with probability one) by computing a random number r_Q between 0 and Q and finding spin i that satisfies the condition

$$\sum_{k=0}^{i-1} p_k \le r_Q < \sum_{k=0}^{i} p_k. \tag{15.85}$$

There are two more ingredients we need to make this algorithm practical. We need to determine how long a configuration would remain unchanged if we had used the Metropolis algorithm. Also, the algorithm would be very inefficient because on average the computation of which spin to flip from (15.85) would take $O(N)$ computations. This second problem can be easily overcome by realizing that there are only a few possible values of p_i. For example, for the Ising model on a square lattice in a magnetic field, there are only $n = 10$ possible values of p_i. Thus, instead of (15.85), we have

$$\sum_{\alpha=0}^{i-1} n_\alpha p_\alpha \le r_Q < \sum_{\alpha=0}^{i} n_\alpha p_\alpha, \tag{15.86}$$

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

Next we need to determine the time spent in a configuration. The probability in one Metropolis Monte Carlo step of choosing a spin at random is $1/N$, and the probability of actually flipping that spin is p_i, which is given by (15.83). Thus, the probability of flipping any spin is

$$\frac{1}{N}\sum_{i=0}^{N-1} p_i = \frac{1}{N}\sum_{\alpha=0}^{n-1} n_\alpha p_\alpha = \frac{Q}{N}. \tag{15.87}$$

The probability of not flipping any spin is $q \equiv 1 - Q/N$, 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 s in Monte Carlo steps per spin to remain in the current configuration will be determined by solving

$$q^{s-1} \le r < q^s. \tag{15.88}$$

If $Q/N \ll 1$, then both sides of (15.88) are approximately equal, and we can approximate s by

$$s \approx \frac{\ln r}{\ln q} = \frac{\ln r}{\ln(1 - Q/N)} \approx -\frac{N}{Q}\ln r. \tag{15.89}$$

That is, we would have to wait s Monte Carlo steps per spin on the average before we would flip a spin using the Metropolis algorithm. Note that the random number r in (15.88) and (15.89) should not be confused with the random number r_Q in (15.86).

The n-fold algorithm can be summarized by the following steps:

(i) Start with an initial configuration and determine the class to which each spin belongs. Store all the possible values of p_i in an array. Compute Q. Store in an array the number of spins in class α, n_α.

(ii) Determine s from (15.89). Accumulate any averages, such as the energy and magnetization weighted by s. Also, accumulate the total time `tTotal += s`.

(iii) Choose a class of spin using (15.86) and randomly choose which spin in the chosen class to flip.

(iv) Update the classes of the chosen spin and its four neighbors.

(v) Repeat steps (ii)–(iv).

To conveniently carry out step (iv), set up the following arrays: `spinClass[i]` returns the class of the ith spin, `spinInClass[k][alpha]` returns the kth spin in class α, and `spinIndex[i]` returns the value of k for the ith spin to use in the array `spinInClass[k][alpha]`. 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 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 order these ten classes from

0 to 9, then the class of a spin that 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 various averages at a few temperatures with the results from your program using the Metropolis algorithm.

(b) Choose the magnetic field $B = -0.5$ at the temperature $T = 1$. Begin with an initial configuration of all spins up and use the n-fold way to estimate how long it takes before the majority of the spins flip. Do the same simulation using the Metropolis algorithm. Which algorithm is more efficient?

(c) Repeat part (b) for other temperature and field values. For what conditions is the n-fold way algorithm more efficient than the standard Metropolis algorithm?

(d) Repeat part (b) for different values of the magnetic field and plot the number of Monte Carlo steps needed to flip the spins as a function of $1/|B|$ for values of B from 0 to ≈ 3. Average over at least 10 starting configurations for each field value. ■

Project 15.37 The Kosterlitz–Thouless transition

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

$$E = -J \sum_{i,j=nn(i)} [s_{i,x}s_{j,x} + s_{i,y}s_{j,y}], \tag{15.90}$$

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 (15.90) in a simpler form by substituting $s_{i,x} = \cos\theta_i$ and $s_{i,y} = \sin\theta_i$. The result is

$$E = -J \sum_{i,j=nn(i)} \cos(\theta_i - \theta_j), \tag{15.91}$$

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 $T \leq T_{\text{KT}}$, the spin-spin correlation function $C(r)$ decreases as a power law; for $T > T_{\text{KT}}$, $C(r)$ decreases exponentially. The power law decay of $C(r)$ for $T \leq T_{\text{KT}}$ implies that every temperature below T_{KT} acts as if it was 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 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 $\mathbf{M}$ for any given configuration. (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 fluctuations in the direction of the spins diverges, which implies that there is no preferred direction for the spins, and hence the mean magnetization vanishes.

(c) Modify your program so that an arrow is drawn at each site to show the orientation of each spin. You can use the `Vector2DFrame` to draw a lattice of arrows. 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 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 Figure 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 rotate 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 for 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, the 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, the specific heat, and the vorticity (equal to the number of vortices per unit 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 do you find a vortex that appears to be free, that is, a vortex that is not obviously paired with another vortex of opposite sign?

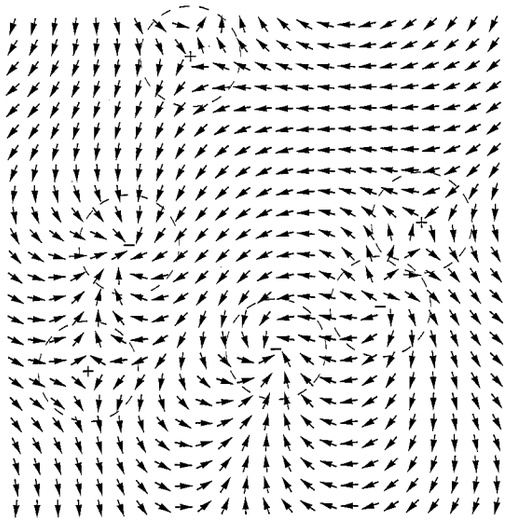


Figure 15.10 A typical configuration of the planar model on a 24×24 square lattice that has been quenched from $T = \infty$ to $T = 0$ and equilibrated for 200 Monte Carlo steps per spin after the quench. Note that there are six vortices. The circle around each vortex is a guide to the eye and is not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

(e) The Kosterlitz–Thouless theory predicts that the susceptibility χ diverges above the transition as

$$\chi \sim A\, e^{b/\epsilon^{\nu}}, \tag{15.92}$$

where ϵ is the reduced temperature $\epsilon = (T - T_{\rm KT})/T_{\rm KT}$, $\nu = 0.5$, and A and b are nonuniversal constants. Compute χ from the relation (15.21) with $\mathbf{M} = 0$. Assume the exponential form (15.92) for χ in the range $T = 1$ and $T = 1.2$ with $\nu = 0.7$ and find the best values of $T_{\rm KT}$, A, and b. (Although theory predicts $\nu = 0.5$, simulations for small systems indicate that $\nu = 0.7$ gives a better fit.) One way to determine $T_{\rm KT}$, A, and b is to assume a value of $T_{\rm KT}$ 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 $T_{\rm KT}$ 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 $T_{\rm KT}$. This prediction has been confirmed by simulations (see Tobochnik and Chester). To obtain quantitative results, you will need lattices larger than 32×32. ■

Project 15.38 The 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

$$E = -J \sum_{i,j=nn(i)}^{N} \mathbf{s}_i \cdot \mathbf{s}_j = -J \sum_{i,j=nn(i)}^{N} [s_{i,x}s_{j,x} + s_{i,y}s_{j,y} + s_{i,z}s_{j,z}], \tag{15.93}$$

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 quark confinement. Lattice models of the interaction between quarks, called lattice gauge theories, predict that the confinement of quarks could 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 two-dimensional Heisenberg model is an 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 $\Delta\mathbf{s} = \Delta s_{\rm max}(q_1, q_2, q_3)$, where $-1 \le q_n \le 1$ is a uniform random number, and $\Delta s_{\rm max}$ is the maximum change of any spin component. If $|\Delta\mathbf{s}| > \Delta s_{\rm max}$, then 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. Then 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, the specific heat, and the 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 15.39 Domain growth kinetics

When the Ising model is quenched from a high temperature to very low temperatures, domains of the ordered low temperature phase 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

$$R = \frac{2}{2+\epsilon}. \tag{15.94}$$

Equation (15.94) can be motivated by 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 (15.94).

(a) Modify your Ising model program so that the initial configuration is random, that is, a typical high temperature configuration. Write a target 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 quench, and the number of Monte Carlo steps between drawing the lattice. Plot $\ln\langle R\rangle$ versus $\ln t$ after each quench is finished, where t is measured from the time of the quench.

(b) Choose $L = 64$ and a maximum time of 128 mcs. Averages over 10 quenches will give acceptable 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 Grest et al.

*(d) Modify your program to simulate a three-dimensional system. How should you modify (15.94)? Are your results similar? ■

Project 15.40 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-in" 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 also an 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 15.31a). 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 15.31b). How do your results compare to those that you found in part (a)? ■

Project 15.41 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) A quantity of interest is $f(t)$, the fraction of spins that have not yet flipped at time t. As usual, the time is measured in terms of Monte Carlo steps per spin. Published results (Derrida et al.) for $N = 10^5$ indicate that $f(t)$ behaves as

$$f(t) \sim t^{-\theta}, \tag{15.95}$$

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 yet flipped as a function of the time t (see Das and Sen). Plot P_n versus n for two times on the same graph. Discuss the shape of the curves and their differences. Choose $L \geq 100$ and $t = 50$ and 100. Try to fit the curves to a Gaussian distribution. Because the possible values of n are bounded, fit each side of the maximum of P_n to a Gaussian with different widths. There are a number of scaling properties that can be investigated. 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 15.42 The inverse power law potential

Consider the inverse power law potential

$$V(r) = V_0 \left(\frac{\sigma}{r}\right)^n, \tag{15.96}$$

with $V_0 > 0$. One reason for the 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, which is defined as (see Project 8.25)

$$\Gamma = \frac{V_0}{kT}\frac{\sigma}{a}, \tag{15.97}$$

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 Γ for a three-dimensional system with $N = 16$, 32, 64, and 128. Does E depend on N? Can you extrapolate your results for the N-dependence of E to $N \to \infty$? Do you see any evidence of a fluid-solid phase transition? If so, estimate the value of Γ at which it occurs. What is the nature of the transition if it exists? What is the symmetry of the ground state?

(c) Let $n = 4$ and determine the symmetry of the ground state. For this value of n, there is a solid-to-solid phase transition at which the solid changes symmetry. To determine the value of Γ at which this phase transition exists and the symmetry of the smaller Γ solid phase (see Dubin and Dewitt), it is necessary to use a Monte Carlo method in which the shape of the simulation cell changes to accomodate the different symmetry (the Rahman–Parrinello method), an interesting project. An alternative is to prepare a bcc lattice at $\Gamma \approx 105$ (for example, $T = 0.06$ and $\rho = 0.95$). Then instantaneously change the potential from $n = 4$ to $n = 12$; the new value of Γ is ≈ 4180, and the new stable phase is fcc. The transition can be observed by watching the evolution of $g(r)$. ■

Project 15.43 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 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. Are your results for E_0/N lower than those you obtained in 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) Do you 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 15.44 The hard disk fluid-solid transition

Although we have mentioned (see Section 15.10) that there is much evidence for a fluid-solid 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 $V = \sqrt{3}L^2v/2$ 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. Do a constant pressure (actually constant $p^* = P/kT$) Monte Carlo simulation. 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

$$e^{-p^*(\tilde{V}-V)+N\ln(\tilde{V}/V)}. \tag{15.98}$$

Volume changes are attempted 40–200 times for each set of individual disk moves. The quantity of interest is $N(v)$, the distribution of the 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 15.45 Vacancy mediated dynamics in binary alloys

When a binary alloy is rapidly quenched from a high temperature to a low temperature unstable state, a pattern of domain formation called *spinodal decomposition* takes place as the two metals in the alloy separate. This process is of much interest experimentally. Lifshitz and Slyozov have predicted that at long times, the linear domain size increases with time as $R \sim t^{1/3}$. This result is independent of the dimension for $d \geq 2$, and has been verified experimentally and in computer simulations. The behavior is modified for binary fluids due to hydrodynamic effects.

Most of the computer simulations of this growth process have been based on the Ising model with spin exchange dynamics. In this model there is an A or B atom (spin up or spin down) at each site, where A and B represent different metals. The energy of interaction between atoms on two neighboring sites is $-J$ if the two atoms are the same type and $+J$ if they are different. Monte Carlo moves are made by exchanging unlike atoms. (The number of A and B atoms must be conserved.) A typical simulation begins with an equilibrated system at high temperatures. Then the temperature is changed instantaneously to a low temperature below the critical temperature T_c. If there are equal numbers of A and B atoms on the lattice, then spinodal decomposition occurs. If you watch a visualization of the evolution of the system, you will see wavy-like domains of each type of atom thickening with time.

The growth of the domains is very slow if we use spin exchange dynamics. We will see that if simulations are performed with vacancy mediated dynamics, the scaling behavior begins at much earlier times. Because of the large energy barriers that prevent real metallic atoms from exchanging position, it is likely that spinodal decomposition in real alloys also occurs with vacancy mediated dynamics. We can do a realistic simulation by including just one vacancy because the number of vacancies in a real alloy is also very small. In this case the only possible Monte Carlo move on a square lattice is to exchange the vacancy with one of its four neighboring atoms. To implement this algorithm, you will need an array to keep track of which type of atom is at each lattice site and variables to keep track of the location of the single vacancy. The simulation will run very fast because there is little bookkeeping and all the possible trial moves are potentially good ones. In contrast, in standard spin exchange dynamics, it is necessary to either waste computer time checking for unlike nearest neighbor atoms or keep track of where they are.

The major quantity of interest is the growth of the domain size R. One way to determine R is to measure the pair correlation function $C(r) = \langle s_i s_j \rangle$, where $r = |\mathbf{r}_i - \mathbf{r}_j|$, and $s_i = 1$ for an A atom and $s_i = -1$ for a B atom. The first zero in $C(r)$ is a measure of the domain size. An alternative measure of the domain size is the quantity $R = 2/(\langle E \rangle/N + 2)$, where $\langle E \rangle/N$ is the average energy per site and N is the number of sites (see Project 15.39). The quantity R is a rough measure of the length of the perimeter of a domain and is proportional to the domain size.

(a) Write a program to simulate vacancy mediated dynamics. The initial state consists of the random placement of A and B atoms (half of the sites have A and half B atoms); one vacancy replaces one of the atoms. Explain why this configuration corresponds to infinite temperature. Choose a square lattice with $L \geq 50$.

(b) Instantaneously quench the system by running the Metropolis algorithm at a temperature of $T = T_c/2 \approx 1.13$. You should first look at the lattice after every attempted move of the vacancy to see the effect of vacancy dynamics. After you are satisfied that your program is working correctly and that you understand the algorithm, speed up the simulation by only collecting data and showing the lattice at times equal to $t = 2^n$ where $n = 1, 2, 3 \ldots$. Measure the domain size using either the energy or $C(r)$ as a function of time averaged over many different initial configuration quenches.

(c) At what time does the log R versus log t plot become linear? Do both measures of the domain size give the same results? Does the behavior change for different quench temperatures? Try $0.2T_c$ and $0.7T_c$. A log-log plot of the domain size versus time should give the exponent $1/3$.

(d) Repeat the measurements in three dimensions. Do you obtain the same exponent? ■

Project 15.46 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 0 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 1. The demon at spin $N - 1$ removes energy from the system by flipping spin $N - 1$ so that it is in its lowest energy state, that is, in the same direction as spin $N - 2$. As a result, energy flows from site 0 to site $N - 1$ 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 0 can only add energy to the system if spin $N - 1$ 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 (15.10); 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 different rate 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 0.

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`. The class `ManyDemons` can be downloaded from the ch15 directory.

(a) Write a target class that inputs the number of spins N and the initial energy of the system, outputs the number of Monte Carlo steps per spin and the energy added to the system at the high temperature 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 (15.10) to define a local site temperature. Use $N \geq 52$ and run for about 10,000 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 0 and is removed at site $N-1$. 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 $N-2$ to site 1 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 10,000 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 0 adds to the system at site 0. 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

$$Q = -\kappa \frac{\partial T}{\partial x}. \tag{15.99}$$

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 T vs. x 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`. Also, constrain spins 0 and $N-1$ 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 $i = 0$ and $i = L-1$, 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 $i = 0$ to $(L/2) - 1$ 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 that the probability for the demon to have energy E_d is proportional to $e^{-E_d/kT}$. Hence, in general, $\langle E_d \rangle$ is given by

$$\langle E_d \rangle = \frac{\sum_{E_d} E_d\, e^{-E_d/kT}}{\sum_{E_d} e^{-E_d/kT}}, \tag{15.100}$$

where the summations in (15.100) 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 Figure 15.11). Hence, the possible energies of the demon are $0, 4J, 8J, 12J, \ldots$. We write $x = 4J/kT$ and perform the summations in (15.100). The result is

$$\langle E_d/kT \rangle = \frac{0 + xe^{-x} + 2xe^{-2x} + \cdots}{1 + e^{-x} + e^{-2x} + \cdots} = \frac{x}{e^x - 1}. \tag{15.101}$$

The form (15.10) can be obtained by solving (15.101) for T in terms of E_d. Convince yourself that the relation (15.101) 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 $0, 2H, 4J - 2H, 4J + 2H, \ldots$. 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

which is discussed in many texts (see Griffiths for example). It is also possible to compute T from averages over $(x_j - x_{j-1})^2$, but the virial theorem yields a smaller variance. The ground state wave function $\phi(x)$ is obtained from the normalized probability $P(x)\Delta x$ by dividing by Δx and taking the square root.

We can also find the thermodynamic properties of a particle that is connected to a heat bath at temperature $T = 1/\beta$ by not taking the $\beta = N\Delta\tau \to \infty$ limit. To obtain the ground state, which corresponds to the zero temperature limit ($\beta \gg 1$), we have to make $N\Delta\tau$ as large as possible. However, we need $\Delta\tau$ to be as small as possible to approximate the continuum time limit. Hence, to obtain the ground state we need a large number of time intervals N. For the finite temperature simulation, we can use smaller values of N for the same level of accuracy as the zero temperature simulation.

The path integral method is very flexible and can be generalized to higher dimensions and many mutually interacting particles. For three dimensions, x_j is replaced by the three-dimensional displacement $\mathbf{r}_j$. Each real particle is represented by a ring of N "atoms" with a spring-like potential connecting each atom within a ring. Each atom in each ring also interacts with the atoms in the other rings through an interparticle potential. If the quantum system is a fluid where indistinguishability is important, then we must consider the effect of exchange by treating the quantum system as a classical polymer system where the "atoms" represent the monomers of a polymer, and where polymers can split up and reform. Chandler and Wolynes discuss how the quantum mechanical effects due to exchanging identical particles can be associated with the chemical equilibrium of the polymers. They also discuss Bose condensation using path integral techniques.

Problem 16.31 Path integral calculation

(a) Write a program to implement the path integral algorithm for the one-dimensional harmonic oscillator potential with $V(x) = x^2/2$. Use the structure of your Monte Carlo Lennard–Jones program from Chapter 15 as a guide.

(b) Let $N\Delta\tau = 15$ and consider $N = 10$, 20, 40, and 80. Equilibrate for at least 2000 Monte Carlo steps per atom and average over at least 5000 mcs. Compare your results with the exact result for the ground state energy given by $E_0 = 0.5$. Estimate the equilibration time for your calculation. What is a good initial configuration? Improve your results by using larger values of $N\Delta\tau$.

(c) Find the mean energy $\langle E\rangle$ of the harmonic oscillator at the temperature T determined by $\beta = N\Delta\tau$. Find $\langle E\rangle$ for $\beta = 1$, 2, and 3 and compare it with the exact result $\langle E\rangle = \frac{1}{2}\coth(\beta/2)$.

(d) Repeat the above calculations for the Morse potential $V(x) = 2(1 - e^{-x})^2$. ■

16.11 ■ PROJECTS

Many of the techniques described in this chapter can be extended to two-dimensional quantum systems. The `Complex2DFrame` tool in the frames package is designed to show two-dimensional complex scalar fields such as quantum wave functions. Listing 16.13 in Appendix 16A shows how this class is used to show a two-dimensional Gaussian wave packet with a momentum boost.

Project 16.32 Separable systems in two dimensions

The shooting method is inappropriate for the calculation of eigenstates and eigenvalues in two or more dimensions with arbitrary potential energy functions $V(\mathbf{r})$. However, the special case of separable potentials can be reduced to several one-dimensional problems that can be solved using the numerical methods described in this chapter. Many molecular modeling programs use the Hartree–Fock self-consistent field approximation to model nonseparable systems as a set of one-dimensional problems. Recently, there has been significant progress motivated by a molecular dynamics algorithm developed by Car and Parrinello.

Write a two-dimensional eigenstate class `Eigenstate2d` that calculates eigenstates and eigenvalues for a separable potential of the form

$$V(x, y) = V_1(x) + V_2(y). \tag{16.106}$$

Test this class using the known analytic solutions for the two-dimensional rectangular box and two-dimensional harmonic oscillator. Use this class to model the evolution of superposition states. Under what conditions are there wave function revivals? ■

Project 16.33 Excited state wave functions using quantum Monte Carlo

Quantum Monte Carlo methods can be extended to compute the excited state wave functions using a Gram–Schmidt procedure to insure that each excited state is orthogonal to all lower lying states (see Roy et al.). A quantum Monte Carlo method is used to compute the ground state wave function. A trial wave function for the first exited state is then selected and the ground state component is subtracted from the trial wave function. This subtraction is repeated after every iteration of the Monte Carlo algorithm. Because excited states decay with a time constant $e^{-(E_j - E_0)}$, the lowest remaining excited state dominates the remaining wave function. After the first excited state is obtained, the second excited state is computed by subtracting both known states from the trial wave function. This process is repeated to obtain additional wave functions.

Implement this procedure to find the first few excited state wave functions for the one-dimensional harmonic oscillator. Then consider the one-dimensional double-well oscillator

$$V(x) = -\frac{1}{2}kx^2 + a_3x^3 + a_4x^4, \tag{16.107}$$

with $k = 40$, $a_3 = 1$, and $a_4 = 1$. ■

Project 16.34 Quantum Monte Carlo in two dimensions

The procedure described in Project 16.33 can be used to compute two-dimensional wave functions (see Roy et al.).

(a) Test your program using a separable two-dimensional double-well potential.

(b) Find the first few excited states for the two-dimensional double-well potential

$$V(x, y) = -\frac{1}{2}k_xx^2 - \frac{1}{2}k_yy^2 + \frac{1}{2}(a_{xx}x^4 + 2a_{xy}x^2y^2 + a_{yy}y^4), \tag{16.108}$$

with $k_x = k_y = 20$ and $a_{xx} = a_{yy} = a_{xy} = 5$. Repeat with $k_x = k_y = 20$ and $a_{xx} = a_{yy} = a_{xy} = 1$. ■

Project 16.35 Evolution of a wave packet in two dimensions

Both the half-step and split-operator algorithms can be extended to model the evolution of two-dimensional systems with arbitrary potentials $V(x, y)$. (See *Numerical Recipes* for how the FFT algorithm is extended to more dimensions.) Implement either algorithm and model a wave packet scattering from a central barrier and a wave packet passing through a double slit.

A clever way to insure stability in the half-step algorithm is to use a boolean array to tag grid locations where the solution becomes unstable and to set the wave function to zero at these grid points.

```
double minV = -2/dt;
double maxVx = 2/dt-2/(dx*dx);
double maxVy = 2/dt-2/(dy*dy);
double maxV = Math.min(maxVx,maxVy);
for(int i = 0, n = potential.length; i <= n; i++) {
   for(int j = 0, m = potential[0].length; j <= m; j++) {
      if (potential[i][j] >= minV && potential[i][j] <= maxV)
         stable[i][j] = true;  // stable
      else
         stable[i][j] = false; // unstable, set wave function to zero
   }
}
```

■

Project 16.36 Two-particle system

Rubin Landau has studied the time dependence of two particles interacting in one dimension with a potential that depends on their relative separation:

$$V(x_1, x_2) = V_0 e^{-(x_1-x_2)^2/2\alpha^2}. \tag{16.109}$$

Model a scattering experiment for particles having momentum p_1 and p_2 by assuming the following (unnormalized) initial wave function:

$$\Psi(x_1, x_2) = e^{ip_1x_1}\, e^{-(x_1-a)^2/4w^2}\, e^{ip_2x_2}\, e^{-(x_2-a)^2/4w^2}, \tag{16.110}$$

where $2a$ is the separation and w is the variance in each particle's position. Do the particles bounce off each other when the interaction is repulsive? What happens when the interaction is attractive?

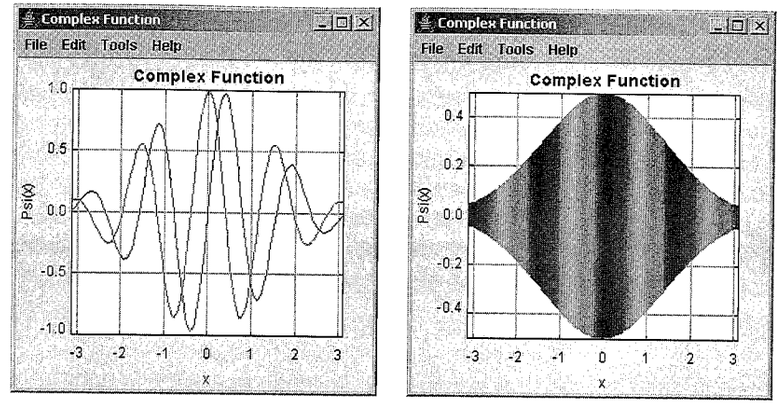
■

APPENDIX 16A: VISUALIZING COMPLEX FUNCTIONS

Complex functions are essential in quantum mechanics and the frames package contains classes for displaying and analyzing these functions. Listing 16.12 uses `ComplexPlotFrame` to display a one-dimensional wave function.

Listing 16.12 The `ComplexPlotFrameApp` class displays a one-dimensional Gaussian wave packet with a momentum boost.

```
package org.opensourcephysics.sip.ch16;
import org.opensourcephysics.frames.ComplexPlotFrame;
```

(a) Real and imaginary. (b) Amplitude and phase.

Figure 16.2 Two representations of complex wave functions. (The actual output is in color.)

```
public class ComplexPlotFrameApp {
   public static void main(String[] args) {
      ComplexPlotFrame frame = new ComplexPlotFrame("x", "Psi(x)",
            "Complex function");
      int n = 128;
      double xmin = -Math.PI, xmax = Math.PI;
      double x = xmin, dx = (xmax-xmin)/n;
      double[] xdata = new double[n];
      // real and imaginary values alternate
      double[] zdata = new double[2*n];
      // test function is e^(-x*x/4)e^(i*mode*x) for x=[-pi,pi)
      int mode = 4;
      for(int i = 0;i<n;i++) {
         double a = Math.exp(-x*x/4);
         zdata[2*i] = a*Math.cos(mode*x);
         zdata[2*i+1] = a*Math.sin(mode*x);
         xdata[i] = x;
         x += dx;
      }
      frame.append(xdata, zdata);
      frame.setVisible(true);
      frame.setDefaultCloseOperation(javax.swing.JFrame.EXIT_ON_CLOSE);
   }
}
```

Figure 16.2 shows two representations of a quantum wave function. The real and imaginary representation displays the real and imaginary parts of the wave function $\Psi(x)$ by drawing two curves. In the amplitude and phase representation the vertical height represents the wave function magnitude and the color indicates phase. Note that the complex phase is oscillating, indicating that the wave function has a nonzero momentum expectation value, which is known as a *momentum boost*.