spins up. If the spins are updated sequentially from right to left, then if one spin is flipped, all remaining flips would be accepted regardless of the temperature, because the change in energy would be zero. The system would not be ergodic for this implementation of the algorithm, and we would not obtain the correct thermodynamic behavior. A measure of the ergodicity of a system was discussed in Project 8.23.

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

The Metropolis algorithm is implemented in method `doStep` in class `BoltzmannApp`, and the velocity distribution is plotted. One quantity of interest is the probability $P(v)\,\Delta v$ that the particle has a velocity between v and $v + \Delta v$. We will choose the temperature to be large enough such that $\Delta v = 1$ provides a sufficiently small bin size to compute $P(v)$ accurately. As usual, we choose units such that the mass of the particle is unity.

Listing 15.4 The Metropolis algorithm for a single particle.

```
package org.opensourcephysics.sip.ch15;
import org.opensourcephysics.controls.*;
import org.opensourcephysics.frames.HistogramFrame;

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

   public void initialize() {
      velocityDistribution.clearData();
      beta = 1.0/control.getDouble("Temperature");
      velocity = control.getDouble("Initial velocity");
      accepted = 0;
      mcs = 0;
   }

   public void doStep() {
      double delta = control.getDouble("Maximum velocity change");
      mcs++;
      double ke = 0.5*velocity*velocity;
      double vTrial = velocity+delta*(2.0*Math.random()-1.0);
      double keTrial = 0.5*vTrial*vTrial;
```



```
      double dE = keTrial-ke;
      if((dE<0)||(Math.exp(-beta*dE)>Math.random())) {
         accepted++;
         ke = keTrial;
         velocity = vTrial;
      }
      velocityDistribution.append(velocity);
      control.clearMessages();
      control.println("mcs = "+mcs);
      control.println("acceptance probability = "+
           (double)(accepted)/mcs);
   }

   public void reset() {
      control.setValue("Maximum velocity change", 10.0);
      control.setValue("Temperature", 10.0);
      control.setValue("Initial velocity", 0.0);
      enableStepsPerDisplay(true);
   }

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

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

(a) Choose the temperature $T = 10$, the initial velocity equal to zero, and the maximum change in the particle's velocity to be $\delta = 10.0$. Run for a number of Monte Carlo steps until a plot of $\ln P(\mathbf{v})$ versus $\mathbf{v}$ is reasonably smooth. Describe the qualitative form of $P(\mathbf{v})$. (Remember that the velocity $\mathbf{v}$ can be either positive or negative.)

(b) Because the velocity of the particle characterizes the microstate of this single particle system, we need to plot $\ln P(E_s)$ versus $E_s = mv_s^2/2$ to test if the Metropolis algorithm yields the Boltzmann distribution in this case. (The two values of v, one positive and one negative, for each value of E, correspond to different microstates.) Add code to `BoltzmannApp` to compute $P(E_s)$ and determine the slope of $\ln P(E_s)$ versus E_s. The code for extracting information from the `HistogramFrame` class is given on page 207. Is this slope equal to $-\beta = -1/T$, where T is the temperature of the heat bath?

(c) Add code to compute the mean energy and velocity. How do your results for the mean energy compare to the exact value? Explain why the computed mean particle velocity is approximately zero even though the initial particle velocity was not zero. To insure that your results do not depend on the initial conditions, let the initial velocity equal zero and recompute the mean energy and velocity. Do your equilibrium results differ from what you found previously?

(d) Add another `HistogramFrame` object to compute the probability $P(E)\Delta E$ where E is the energy of the configuration. Does $P(E)$ have the form of a Boltzmann distribution? If not, what is the functional form of $P(E)$?

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