As discussed in Problem 8.16, an important check on the calculated trajectories of a hard disk system is that no two disks overlap. The following method tests for this condition.

```
public void checkOverlap() {
   for (int i = 0; i < N-1; ++i) {
      for(int j = i+1; j < N; ++j) {
         double dx = PBC.separation(x[i] - x[j],Lx);
         double dy = PBC.separation(y[i] - y[j],Ly);
         if (dx*dx+dy*dy < 1.0) {
            System.out.println("Particles " + i + " and" + j +
                  " overlap");
         }
      }
   }
}
```

To complete class `HardDisks`, we need to add the class declarations, which we show in Listing 8.20, and the `draw` method, which is the same as in class `LJParticles`. You can use a slightly modified version of class `LJParticlesApp` as the target class for this application, but note that you will need to modify the `LJParticlesLoader` class to store different arrays. The number of collisions, the time, and a plot of the pressure versus time should be displayed. We will leave the task of writing the target class as an exercise.

Listing 8.20 Class declarations for `HardDisks`.

```
package org.opensourcephysics.sip.ch08.hd;
import java.awt.*;
import org.opensourcephysics.display.*;
import org.opensourcephysics.numerics.*;

public class HardDisks implements Drawable {
   public double x[], y[], vx[], vy[];
   public double collisionTime[];
   public int partner[];
   public int N;
   public double Lx;
   public double Ly;
   public double keSum = 0, virialSum = 0;
   public int nextCollider, nextPartner;
   public double timeToCollision;
   public double t = 0;
   public double bigTime = 1.0E10;
   public double temperature;
   public int numberOfCollisions = 0;
```

Problem 8.16 Initial test of class `HardDisks`

(a) Because even a small error in computing the trajectories of the disks will eventually lead to their overlap and hence to a fatal error, it is necessary to test class `HardDisks` carefully. For simplicity, start from a lattice configuration. The most important test of the program is to monitor the computed positions of the hard disks for overlaps. If the distance between the centers of any two hard disks is less then unity (distances are measured in units of σ), there must be a serious error in the program. To check for the overlap of hard disks, include method `checkOverlap` in method `step` while you are testing the program.

(b) The temperature for a system of hard disks is constant and can be defined as in (8.6). Why does the temperature not fluctuate as it does for a system of particles interacting with a continuous potential? The constancy of the temperature can be used as another check on your program. What is the effect of increasing all the velocities by a factor of two? What is the natural unit of time? Explain why the state of the system is determined by the density only and not by the temperature.

(c) Generate equilibrium configurations of a system of $N = 64$ disks in a square cell of linear dimension $L = 12$. Suppose that at $t = 0$, the constraint that $0 \le x \le 12$ is removed, and the disks are allowed to move in a rectangular cell with $L_x = 24$ and $L_y = 12$. Does the system become more or less random? What is the qualitative nature of the time dependence of $n(t)$, the number of disks on the left half of the cell?

(d) Modify your program so that averages are not computed until the system is in equilibrium. Compute the virial (8.37) and make a rough estimate of the error in your determination of the mean pressure due to statistical fluctuations.

(e) Modify your program so that you can compute the velocity and speed distributions and verify that the computed distributions have the expected forms. ■

Problem 8.17 Static properties of hard disks

(a) As we have seen in Section 8.7, a very time consuming part of the simulation is equilibrating a system from an arbitrary initial configuration. One way to obtain a set of initial positions is to add the hard disks sequentially with random positions and reject an additional hard disk if it overlaps any disks already present. Although this method is very inefficient at high densities, try it so that you will have a better idea of how difficult it is to obtain a high density configuration in this way. A much better method is to place the disks on the sites of a lattice.

(b) The largest number of hard disks that can be placed into a fixed volume defines the maximum density. What is the maximum density if the disks are placed on a square lattice? What is the maximum density if the disks are placed on a triangular lattice? Suppose that the initial condition is chosen to be a square lattice with $N = 100$ and $L = 11$ so that each particle has four nearest neighbors. What is the qualitative nature of the system after several hundred collisions have occurred? Do most particles still have four nearest neighbors, or are there regions where most particles have six neighbors?

(c) The dependence of the mean pressure P on the density ρ is of interest, as it is for a system with a continuous potential. Is P a monotonically increasing function of ρ? Is a system of hard disks always a fluid, or is there a fluid to solid transition at higher densities? You will not be able to find definitive answers to these questions for $N = 64$. Most simulations in the 1960s and 70s were done for systems of $N = 108$ hard disks. The largest simulations were for several hundred particles, and new insight into the properties of liquids was found. Find the dependence of the pressure on the density, beginning at low densities and slowly increasing the density starting from a configuration from a lower density. At any given time, into the maximum density increase is given by the minimum distance between any two disks. To increase