```
   for(int i = 0; i<N; ++i) {
      // sets unknown collision times to a big number
      collisionTime[i] = bigTime;
   }
   // find initial collision times for all particles
   for(int i = 0; i<N-1; i++) {
      for(int j = i+1; j<N; j++) {
         checkCollision(i, j);
      }
   }
}

public void resetAverages() {
   t = 0;
   virialSum = 0;
}
```

Method `checkCollision` uses the relations (8.33) and (8.35) to determine whether particles `i` and `j` will collide and if so, the time `tij` until their collision. We check for collisions with particle `j` in the central cell as well as with particle `j` in the eight image cells surrounding the central cell as shown in Figure 8.8. For very dilute systems, we might need to check further periodic images. For the densities we will consider, such a check should not be necessary.

Listing 8.18 Method for checking the collision time and collision partners of each particle.

```
public void checkCollision(int i, int j) {
   // consider collisions between i and j and periodic images of j
   double dvx = vx[i]-vx[j];
   double dvy = vy[i]-vy[j];
   double v2 = dvx*dvx+dvy*dvy;
   for(int xCell = -1; xCell<=1; xCell++) {
      for(int yCell = -1; yCell<=1; yCell++) {
         double dx = x[i]-x[j]+xCell*Lx;
         double dy = y[i]-y[j]+yCell*Ly;
         double bij = dx*dvx+dy*dvy;
         if(bij<0) {
            double r2 = dx*dx+dy*dy;
            double discriminant = bij*bij-v2*(r2-1);
            if(discriminant>0) {
               double tij = (-bij-Math.sqrt(discriminant))/v2;
               if(tij<collisionTime[i]) {
                  collisionTime[i] = tij;
                  partner[i] = j;
               }
               if(tij<collisionTime[j]) {
                  collisionTime[j] = tij;
                  partner[j] = i;
               }
            }
         }
      }
   }
}
```


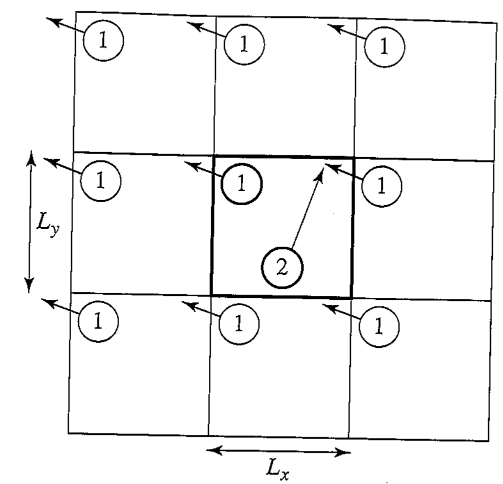

Figure 8.8 The positions and velocities of disks 1 and 2 in the figure are such that disk 1 collides with an image of disk 2 that is not the image closest to disk 1. The periodic images of disk 2 are not shown.

The main thermodynamic quantity of interest for hard disks is the mean pressure P. Because the forces act only when two disks are in contact, we have to modify the form of (8.9). We write $\mathbf{F}_{ij}(t) = \mathbf{I}_{ij}\,\delta(t - t_c)$, where t_c is the time at which the collision occurs. This form of $\mathbf{F}_{ij}$ implies that the force is nonzero only when there is a collision between i and j. The delta function $\delta(t)$ is infinite for $t = 0$ and is zero otherwise; $\delta(t)$ is defined by its use in an integral as shown in (8.36). This form of the force yields

$$\int_0^t \mathbf{I}_{ij}\,\delta(t' - t_c)\,dt' = \mathbf{I}_{ij} = m\Delta\mathbf{v}_{ij}, \tag{8.36}$$

where we have used Newton's second law and assumed that a single collision has occurred during the time interval t. The quantity $\Delta\mathbf{v}_{ij}$ is given by $\Delta\mathbf{v}_{ij} = \mathbf{v}_i' - \mathbf{v}_i - (\mathbf{v}_j' - \mathbf{v}_j)$. If we explicitly include the time average to account for all collisions during the time interval t, we can write (8.9) as

$$\begin{aligned}\frac{PV}{NkT} - 1 &= \frac{1}{dNkT}\frac{1}{t}\sum_{ij}\int_0^t \mathbf{r}_{ij}\cdot\mathbf{I}_{ij}\,\delta(t' - t_c)\,dt' \\ &= \frac{1}{dNkT}\frac{1}{t}\sum_{c_{ij}} m\Delta\mathbf{v}_{ij}\cdot\mathbf{r}_{ij}.\end{aligned} \tag{8.37}$$

The sum in (8.37) is over all collisions c_{ij} between disks i and j in the time interval t; $\mathbf{r}_{ij}$ is the vector between the centers of the disks at the time of a collision; the magnitude of $\mathbf{r}_{ij}$ in (8.37) is σ.

Listing 8.19 Method for computing the pressure.

```
public double pressure() {
   double area = Lx*Ly;
   return 1+virialSum/(2*t*area*N*temperature);
}
```