2.4 Large hard-sphere systems

Daily life accustoms us to phase transitions between different forms of matter, for example in water, between ice (solid), liquid, and gas. We usually think of these phase transitions as resulting from the antagonistic interplay between the interactions and the temperature. At low temperature, the interactions win, and the atoms or the ions settle into crystalline order. Materials turn liquid when the atoms' kinetic energy increases with temperature, or when solvents screen the interionic forces. Descriptions of phase transitions which focus on the energy alone are over-simplified for regular materials. They certainly do not explain phase transitions in hard spheres because there simply are no forces; all configurations have the same energy. However, the transition between the disordered phase and the ordered phase still takes place.

Our understanding of these entropic effects will improve in Chapter 6, but we start here by describing the phase transitions of hard disks more quantitatively than by just contemplating snapshots of configurations. To do so, we shall compute the equation of state, the relationship between volume and pressure.

When studying phase transitions, we are naturally led to simulating large systems. Throughout this book, and in particular during the present section, we keep to basic versions of programs. However, we should be aware of engineering tricks which can considerably speed up the execution of programs without changing in any way the output created. We shall discuss these methods in Subsection 2.4.1.

2.4.1 Grid/cell schemes

In this subsection, we discuss grid/cell techniques which allow one to decide in a constant number of operations whether, in a system of N particles, a disk k overlaps any other disk. This task comes up when we must decide whether a configuration is illegal, or whether a move is to be rejected. This can be achieved faster than by our naive checks of the $N-1$ distances from all other disks in the system (see for example Alg. 2.9 (`markov-disks`)). The idea is to assign all disks to an appropriate grid with cells large enough that a disk in one cell can only overlap with particles in the same cell or in the adjacent ones. This reduces the overlap checks to a neighborhood (see Fig. 2.44). Of course, particles may move across cell boundaries, and cell occupancies must be kept consistent (see Fig. 2.45).

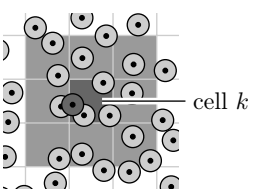

Fig. 2.44 Grid/cell scheme with large cells: a disk in cell k can only overlap with disks in the same cell or in adjacent cells.

There are number of approaches to setting up grid/cell schemes and to handling the bookkeeping involved in them. One may simply keep all disk numbers of cell k in a table. A move between cells k and l has us locate the disk index in the table for cell k, swap it with the last element

of that table, and then reduce the number of elements:

$$\underbrace{\overbrace{\begin{bmatrix}1\\ \boxed{17}\\5\\9\\4\end{bmatrix}}^{\text{locate}} \rightarrow \overbrace{\begin{bmatrix}1\\ \boxed{4}\\5\\9\\ \boxed{17}\end{bmatrix}}^{\text{swap}} \rightarrow \overbrace{\begin{bmatrix}1\\4\\5\\9\end{bmatrix}}^{\text{reduce}}}_{\text{disk 17 leaving a cell containing disks } \{1,17,5,9,4\}}. \tag{2.20}$$

In the case of the target cell l, we simply append the disk's index to the table, and increment its number of elements.

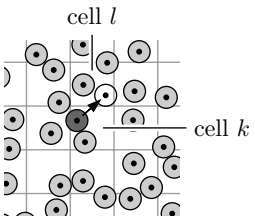

Fig. 2.45 Moving a particle between boxes involves bookkeeping.

There are other solutions for grid/cell schemes. They may involve linked lists rather than tables, that is, a data structure where element 1 in eqn (2.20) points to (is linked to) element 17, which itself points to disk 5, etc. Disk 4 would point to an "end" mark, with a "begin" mark pointing to disk 1. In that case, disk 17 is eliminated by redirecting the pointer of 1 from 17 directly to 5. Besides using linked lists, it is also possible to work with very small cells containing no more than one particle, at the cost of having to check more than just the adjacent cells for overlaps. Any of these approaches can be programmed in several subroutines and requires only a few instructions per bookkeeping operation. The extra memory requirements are no issue with modern computers. Grid/cell schemes reduce running times by a factor of αN, where $\alpha < 1$ because of the bookkeeping overhead. We must also consider the human time that it takes to write and debug the modified code. The computations in this book have been run without improvement schemes on a year 2005 laptop computer, but some of them (in Subsections 2.2.5 and 2.4.2) approach a limit where the extra few hours—more realistically a few days—needed for implementing them were well spent.

Improved codes are easily tested against naive implementations, which we always write first and always keep handy. The output of molecular dynamics runs or Monte Carlo codes should be strictly equivalent between basic and improved versions of a program, even after going through billions of configurations. This frame-to-frame equivalence between two programs is easier to check than statistical equivalence, say, between direct sampling and Markov-chain sampling, where we can only compare average quantities, and only up to statistical uncertainties.

2.4.2 Liquid–solid transitions

In this subsection, we simulate hard disks at constant pressure, in order to obtain the equation of state of a system of hard disks, that is, the relationship between the pressure and the volume of the system. For concreteness, we restrict ourselves to a system of 100 disks in a box with aspect ratio $\sqrt{3}/2$. For this system, we do not need to implement the grid/cell schemes of Subsection 2.4.1. Straight simulation gives the following curve shown in Fig. 2.46 for the mean volume per particle. At

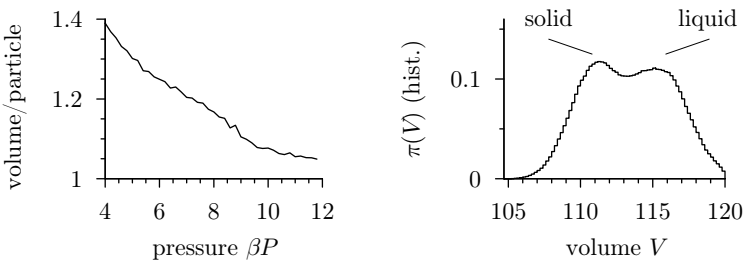

Fig. 2.46 Equation of state for 100 disks (*left*), and histogram of V at $\beta P = 8.4177$ (*right*) ($\sigma = \frac{1}{2}$, $L_x/L_y = \sqrt{3}/2$).

small pressure, the volume is naturally large, and the configurations are liquid-like. At high pressure, the configurations are crystalline, as was already discussed in Subsection 2.2.3. It is of fundamental importance that the volume as a function of pressure behaves differently above and below the transition separating the two regimes, but this cannot be seen very clearly in the equation of state of small systems. It is better to trace the histogram of volumes visited (see Fig. 2.46 again). Both at low pressure and at high pressure, this histogram has a single peak. In the transition region $\beta P \simeq 8.4177$, the histogram has two peaks. The two types of configurations appear (see Fig. 2.47), configurations that are solid-like (at small volume) and configurations that are liquid-like (at large volume). In this same region, the Monte Carlo simulation using the local algorithm slows down enormously. While it mixes up liquid configurations without any problem, it has a very hard time moving from a solid configuration (as the left configuration in Fig. 2.47) to a liquid-like configuration (as the right one in that same figure, see Lee and Strandburg 1992).

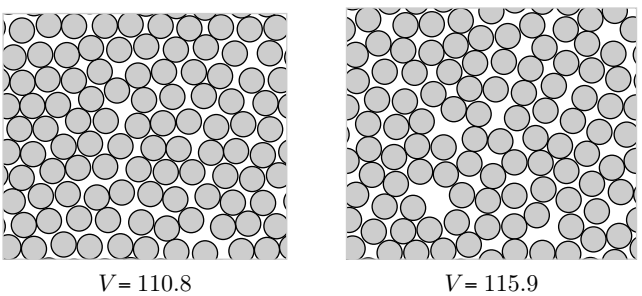

Fig. 2.47 Typical configuration for 100 disks of radius $\frac{1}{2}$ at pressure $\beta P = 8.4177$. *Left*: solid configuration (at small volume); *right*: liquid configuration (at large volume).