interaction with the (nearest image) of all the particles in the simulation box is calculated. As a consequence, the potential is not a constant on the surface of a cube around a given particle. Hence, for the same reasons as mentioned in the previous paragraph, the simple minimum image convention should never be used in Molecular Dynamics simulations.

In the preceding, we described some details on how the energy should be calculated. The implementation of a simple, order-N^2, algorithm to compute the energy will be discussed in section 4.2.2 in the context of a Molecular Dynamics simulation (see Algorithm 5). More advanced schemes to simulate large systems efficiently are described in Appendix F.

Initialization

To start the simulation, we should assign initial positions to all particles in the system. As the equilibrium properties of the system do not (or, at least, should not) depend on the choice of initial conditions, all reasonable initial conditions are in principle acceptable. If we wish to simulate the solid state of a particular model system, it is logical to prepare the system in the crystal structure of interest. In contrast, if we are interested in the fluid phase, we simply prepare the system in any convenient crystal structure. This crystal subsequently melts, because at the temperature and density of a typical liquid-state point, the solid state is not thermodynamically stable. Actually, one should be careful here, because the crystal structure may be metastable, even if it is not absolutely stable. For this reason, it is unwise to use a crystal structure as the starting configuration of a liquid close to the freezing curve. In such cases, it is better to use the final (liquid) configuration of a system at a higher temperature or lower density, where the solid is unstable and has melted spontaneously. In any event, it is usually preferable to use the final (well-equilibrated) configuration of an earlier simulation at a nearby state point as the starting configuration for a new run and adjust the temperature and density to the desired values.

The equilibrium properties of a system should not depend on the initial conditions. If such a dependence nevertheless is observed in a simulation, there are two possibilities. The first is that our results reflect the fact that the system that we simulate really behaves nonergodically. This is the case, for instance, in glassy materials or low-temperature, substitutionally disordered alloys. The second (and much more likely) explanation is the system we simulate is ergodic but our sampling of configuration space is inadequate; in other words, we have not yet reached equilibrium.

Reduced Units

In simulations it is often convenient to express quantities such as temperature, density, pressure, and the like in reduced units. This means that we choose a convenient unit of energy, length and mass and then express all other quantities in terms of these basic units. In the example of a Lennard-Jones system, we use a pair potential that is of the form $u(r) = \epsilon f(r/\sigma)$ (see equation (3.2.3)). A natural (though not unique) choice for our basic units is the following:

- Unit of length, σ
- Unit of energy, ϵ
- Unit of mass, m (the mass of the atoms in the system)

and from these basic units, all other units follow. For instance, our unit of time is

$$\sigma\sqrt{m/\epsilon}$$

and the unit of temperature is

$$\epsilon/k_B.$$

In terms of these reduced units, denoted with superscript *, the reduced pair potential $u^* \equiv u/\epsilon$ is a dimensionless function of the reduced distance $r^* \equiv r/\sigma$. For instance, the reduced form for the Lennard-Jones potential is

$$u^{*lj}(r^*) = 4\left[\left(\frac{1}{r^*}\right)^{12} - \left(\frac{1}{r^*}\right)^{6}\right]. \tag{3.2.10}$$

With these conventions we can define the following reduced units: the potential energy $U^* = U\epsilon^{-1}$, the pressure $P^* = P\sigma^3\epsilon^{-1}$, the density $\rho^* = \rho\sigma^3$, and the temperature $T^* = k_B T\epsilon^{-1}$.

One may wonder why it is convenient to introduce reduced units. The most important reason is that (infinitely) many combinations of ρ, T, ϵ, and σ all correspond to the same state in reduced units. This is the law of corresponding states: the same simulation of a Lennard-Jones model can be used to study Ar at 60 K and a density of 840 kg/m^3 and Xe at 112 K and a density of 1617 kg/m^3. In reduced units, both simulations correspond to the state point $\rho^* = 0.5$, $T^* = 0.5$. If we had not used reduced units, we might have easily missed the equivalence of these two simulations. Another, practical, reason for using reduced units is the following: when we work with real (SI) units, we find that the absolute numerical values of the quantities that we are computing (e.g., the average energy of a particle or its acceleration) are either much less or much larger than 1. If we multiply several such quantities using standard floating-point multiplication, we face a distinct risk that, at some stage, we will obtain a result that creates an overflow or underflow. Conversely, in reduced units, almost all quantities of interest are of order 1 (say, between 10^{-3} and 10^3). Hence, if we suddenly find a very large (or very small) number in our simulations (say, 10^{42}), then there is a good chance that we have made an error somewhere. In other words, reduced

Quantity	Reduced units		Real units
temperature	$T^* = 1$	↔	$T = 119.8$ K
density	$\rho^* = 1.0$	↔	$\rho = 1680$ kg/m^3
time	$\Delta t^* = 0.005$	↔	$\Delta t = 1.09 \times 10^{-14}$ s
pressure	$P^* = 1$	↔	$P = 41.9$ MPa

Table 3.1: Translation of reduced units to real units for Lennard-Jones argon ($\epsilon/k_B = 119.8$ K, $\sigma = 3.405 \times 10^{-10}$m, $M = 0.03994$ kg/mol)

units make it easier to spot errors. Simulation results that are obtained in reduced units can always be translated back into real units. For instance, if we wish to compare the results of a simulation on a Lennard-Jones model at $T^* = 1$ and $P^* = 1$ with experimental data for argon ($\epsilon/k_B = 119.8$ K, $\sigma = 3.405 \times 10^{-10}$ m, $M = 0.03994$ kg/mol), then we can use the translation given in Table 3.1 to convert our simulation parameters to real SI units.[5]

3.2.3 Detailed Balance versus Balance

Throughout this book we use the condition of detailed balance as a test of the validity of a Monte Carlo scheme. However, as stated before, the detailed-balance condition is sufficient, but not necessary. Manousiouthakis and Deem [54] have shown that the weaker "balance condition" is a necessary and sufficient requirement. A consequence of this proof is that one has more freedom in developing Monte Carlo moves. For example, in the simple Monte Carlo scheme shown in Algorithm 2 we select a particle at random and give it a random displacement. During the next trial move, the *a priori* probability to select the *same* particle is the same. Thus the reverse trial move has the same *a priori* probability and detailed balance is satisfied. An alternative scheme is to attempt moving all particles sequentially, i.e., first an attempt to move particle one, followed by an attempt to move particle two, etc. In this sequential scheme, the probability that a single-particle move is followed by its reverse is zero. Hence, this scheme clearly violates detailed balance. However, Manousiouthakis and Deem have shown that such a sequential updating scheme does obey balance and does therefore (usually — see Ref. [54]) result in correct MC sampling.

We stress that the detailed-balance condition remains an important guiding principle in developing novel Monte Carlo schemes. Moreover, most algorithms that do not satisfy detailed balance are simply wrong. This is true in particular for "composite" algorithms that combine different trial moves. Therefore, we suggest that it is good practice to impose detailed balance

[5] In what follows we will always use reduced units, unless explicitly indicated otherwise. We, therefore, omit the superscript * to denote reduced units.