7
Molecular Simulation Techniques Using Classical Force Fields

Thijs J.H. Vlugt, Kourosh Malek, and Berend Smit

7.1
Introduction

In this chapter, we present an overview of various simulation techniques used to study systems of particles that interact using classical force fields. In particular, we focus on the molecular dynamics (MD) technique (Section 7.2), simulation techniques to study rare events (Section 7.3), and the Monte Carlo (MC) technique (Section 7.4). For a more detailed review of these topics, we refer the reader to Refs. [1–8].

7.2
Molecular Dynamics

7.2.1
Introduction

In classical MD simulations, the system is treated as a set of N interacting atoms (or molecules). The atoms are represented by spherical nuclei that attract and repel each other depending on the distance. Their electronic structure is not considered explicitly. After assigning point charges to each particle, the forces acting on the particles are usually obtained from a combination of bonded and nonbonded interactions. The motions of the atoms are calculated using the laws of classical mechanics. Before starting a simulation, a model system is built consisting all chemical components interacting in a simulation box. Just like any real experiment, this system needs to be carefully prepared. It should be a realistic representation of the system that is to be studied. The result of an MD simulation is a trajectory of the positions and velocities of all N atoms in the system. If simulated with an appropriate time step and for a sufficiently long time, thermodynamic properties, time-dependent correlation functions, and transport properties can be reliably calculated. The required length of the trajectory depends on the length scale

Computational Methods in Catalysis and Materials Science. Edited by Rutger A. van Santen and Philippe Sautet

ISBN: 978-3-527-32032-5

of the system under study and the time scale needed for calculation of physical parameters. Nowadays, the typical trajectory time in atomistic MD simulations varies between few nanoseconds (ns) up to hundreds of nanoseconds.

The trajectory (positions and velocities as a function of time) is obtained from solving a system of second-order differential equations that follow from Newton's second law, and can be written as

$$m_i \frac{d^2\mathbf{r}_i}{dt^2} = \sum_j \mathbf{F}_{ij} + \sum \mathbf{F}_{\mathrm{ex,i}} \quad i = 1, 2, \ldots, N. \tag{7.1}$$

Here i denotes the current particle, and m_i and $\mathbf{r}_i$ the mass and the position of this particle, respectively. The forces $\mathbf{F}_{ij}$ represent two-body interactions between atom i and atom j. The forces $\mathbf{F}_{\mathrm{ex,i}}$ due to external fields, for example, electric fields, are added as extra terms. After determining these forces as functions of atomic and molecular degrees of freedom, the equations of motion can be integrated. The combination of all forces in the model, including van der Waals interactions and electrostatic interactions, is called the force field. The choice of the force field is the key to accurate results that appropriately reproduce the true physical phenomena in the system. Popular force fields are, for example, TraPPE [9,10], CHARMM [11], and AMBER [12]. Some force fields use a so-called united atom approach, in which carbon atoms with attached hydrogen atoms are treated as a single interaction site.

The forces acting on the atoms are derived from the gradients of the potential energy function,

$$\mathbf{F}_i = -\nabla U(\mathbf{r}^N), \tag{7.2}$$

where U is the total potential energy of the system and the vector $\mathbf{r}^N$ represents the positions of all N particles in the system. The force fields can be split up into two contributions: *nonbonded interactions* between all nuclei and *bonded interactions* between nuclei that are part of the same molecule. The nonbonded interactions consist of the following terms: electrostatic interactions, van der Waals interactions, and polarization effects. Polarization effects are the result of varying electron densities; they cannot be described explicitly using force-field methods that ignore electron dynamics. It is common practice to include them implicitly in the van der Waals interactions as an overall effect. This leaves two terms for the nonbonded interactions. The first type of nonbonded interactions corresponds to dispersion or van der Waals forces. These are the interactions between atoms that arise from (quantum) fluctuations of the electronic charge densities. Van der Waals interactions are usually modeled using the Lennard–Jones potential,

$$u(r_{ij}) = 4\epsilon \left[\left(\frac{\sigma}{r_{ij}} \right)^{12} - \left(\frac{\sigma}{r_{ij}} \right)^{6} \right]. \tag{7.3}$$

In this equation, ϵ represents the depth of the potential at the minimum ($r_{\mathrm{min}} = 2^{1/6}\sigma$) and $r = \sigma$ is the point at which $u(r) = 0$ (see also Figure 7.1). The term $4\epsilon\sigma^{12}/r^{12}$ is due to the strong repulsion of atoms at short distances. The number of interactions that needs to be computed can be reduced by neglecting all interactions

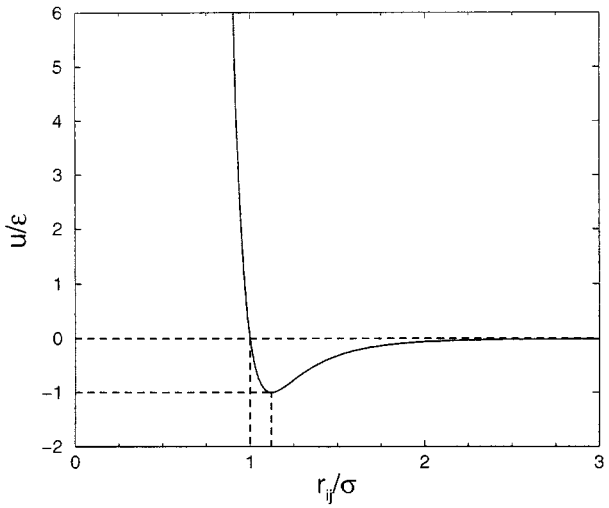

Figure 7.1 The Lennard–Jones potential. Note that the axis labels u/ϵ and r_{ij}/σ are dimensionless.

beyond a certain cut-off radius r_{cut}, which should not be too small (in practice, $r_{\text{cut}} = 2.5\sigma$ is often used). This results in the so-called truncated and shifted Lennard–Jones potential,

$$u\left(r_{ij}\right) = \begin{cases} 4\epsilon\left[\left(\frac{\sigma}{r_{ij}}\right)^{12} - \left(\frac{\sigma}{r_{ij}}\right)^{6}\right] - u_{\text{cut}} & r \leq r_{\text{cut}} \\ 0 & r > r_{\text{cut}} \end{cases}, \tag{7.4}$$

where

$$u_{\text{cut}} = 4\epsilon\left[\left(\frac{\sigma}{r_{\text{cut}}}\right)^{12} - \left(\frac{\sigma}{r_{\text{cut}}}\right)^{6}\right]. \tag{7.5}$$

It is important to note that the result of a computer simulation may depend on r_{cut} and whether a truncated or truncated and shifted potential is used. Note that other truncation methods that remove discontinuities in higher order derivatives of $u(r_{ij})$ are also often used in simulations. The second type of nonbonded interactions are the Coulomb interactions

$$u(r_{ij}) = \frac{q_i q_j}{4\pi\epsilon_0 r_{ij}} \tag{7.6}$$

between two charged spheres at a distance r_{ij} from each other. In this equation, q_i represents the charge of particle i and ϵ_0 is the dielectric constant of vacuum. Particles can be assigned partial charges or integer values in the case of ions. Unlike the Lennard–Jones interactions, simple truncation of Coulombic interactions can lead to serious artifacts (e.g., an unphysical dipole moment at the cut-off radius). The generally accepted method to handle Coulombic interactions is the Ewald summation or equivalent method [4, 13, 14]. Recently, pairwise alternatives for the Ewald summation have been proposed [15, 16].

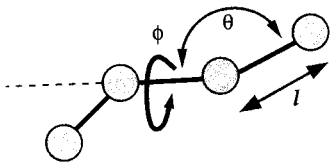

Figure 7.2 Schematic representation of the bond-stretching (bond length l), bond-bending (angle θ), and torsion interaction (torsion angle ϕ).

For atoms that are part of molecules, the bonded interactions also have to be calculated. These bonded interactions are usually one of the following: bond-stretching (2-body), bond-bending (3-body), and dihedral angle (4-body) interactions. For clarity, the interactions are depicted in Figure 7.2. The first two interactions can be described using a harmonic potential. The same mechanism can also be used to describe the motion of a vibrating spring or a pendulum:

$$u_{\text{stretch}}(l) = \frac{k_l}{2}(l - l_0)^2, \tag{7.7}$$

$$u_{\text{bend}}(\theta) = \frac{k_\theta}{2}(\theta - \theta_0)^2, \tag{7.8}$$

where l_0 and θ_0 represent the bond length and bond angle equilibrium values, and k_l and k_θ are the force constants. The dihedral interaction cannot be described using a harmonic potential but rather a periodic function is used, because of the rotational symmetry,

$$u_{\text{torsion}}(\phi) = \sum_{i=0}^{n} c_i(\cos\phi)^i, \tag{7.9}$$

where $c_0, \ldots, c_n$ are constants. Together the bonded interactions provide flexibility to the molecular structure. They play a large role in, for example, simulations of a hydrated nafion membrane, the transport of proton and water through the membrane, simulations on protein folding, or the permeation event of an ion through an ion channel.

7.2.2 Integrating the Equations of Motion

To integrate the equations of motion (Eq. (7.1)), special integration algorithms are used. The simplest algorithm is the so-called Verlet algorithm, which is based on a Taylor expansion of the coordinate of a particle at time $t + \Delta t$ and $t - \Delta t$ about time t:

$$r(t + \Delta t) = r(t) + v(t)\Delta t + \frac{f(t)}{2m}\Delta t^2 + \frac{\partial^3 r}{\partial t^3}\frac{\Delta t^3}{3!} + \mathcal{O}(\Delta t^4), \tag{7.10}$$

and

$$r(t-\Delta t) = r(t) - v(t)\Delta t + \frac{f(t)}{2m}\Delta t^2 - \frac{\partial^3 r}{\partial t^3}\frac{\Delta t^3}{3!} + \mathcal{O}(\Delta t^4). \tag{7.11}$$

Adding these two equations and subtracting $r(t-\Delta t)$ on both sides gives

$$r(t+\Delta t) = 2r(t) - r(t-\Delta t) + \frac{f(t)}{m}\Delta t^2 + \mathcal{O}(\Delta t^4). \tag{7.12}$$

Note that the new position is accurate to order Δt^4. The Verlet algorithm does not use the velocity to compute the new position, but the velocity can be derived from the trajectory, which is only accurate to order Δt^2:

$$v(t) = \frac{r(t+\Delta t) - r(t-\Delta t)}{2\Delta t} + \mathcal{O}(\Delta t^2). \tag{7.13}$$

The instantaneous temperature T follows directly from the kinetic energy

$$E_{\text{kin}} = \sum_{i=1}^{N} \frac{m v_i^2}{2} = \frac{3Nk_{\text{B}}T}{2}, \tag{7.14}$$

where $\mathbf{v}_i^2 = v_{x,i}^2 + v_{y,i}^2 + v_{z,i}^2$ and k_{B} is the Boltzmann constant. As $T \propto \sum_{i=1}^{N} \mathbf{v}_i^2$, the instantaneous temperature $T(t)$ can be adjusted to match the desired temperature T by scaling all velocities with a factor $\sqrt{T/T(t)}$. It is important to note that the Verlet algorithm conserves:

- the total linear momentum of the system, i.e.,

$$\frac{d}{dt}\sum_{i=1}^{N} m_i \mathbf{v}_i(t) = 0 \tag{7.15}$$

- the *total* energy of the system, which is the sum of the potential energy E_{pot} and the kinetic energy E_{kin} (Eq. (7.14)).

If these quantities are not conserved in an actual simulation, there must be an error in the simulation program. Other well-known integration schemes are the leap-frog algorithm, the velocity–Verlet algorithm, and predictor–corrector algorithms [1,4]. The choice of the algorithm depends on several things. One consideration would be the accuracy, because this controls the size of the time step. The larger the time steps, the fewer evaluations of the forces are necessary, thus saving on CPU time. It is important to note that the goal of MD simulations is to compute time averages for a *representative* trajectory, and *not* to integrate the equations of motion as accurately as possible. In fact, no algorithm even exists that can predict the "real" trajectory over a large time scale. This is due to the fact that small integration errors and round-off errors will accumulate over time and the calculated trajectory will diverge from the "real" trajectory. In general, the choice for the algorithm will depend on how well the total energy of the system is conserved, since this property is more important for statistical predictions. It turns out that *time-reversible*

integration algorithms like the Verlet algorithm conserve the total energy even on long time scales and that algorithms that are *not* time reversible show an energy drift for long time scales, even though they conserve the total energy very well for short time scales [4].

A program that performs an MD simulation consists of the following steps (see also Table 7.1):

Table 7.1 Pseudocomputer code for a molecular dynamics simulation of npart particles that interact with a Lennard–Jones potential.[a]

```
program md                          Molecular Dynamics algorithm
call init                           generate initial positions and velocities
t=0.0                               start at time t = 0
do while (t.lt.tmax)                for t < t_max
   call force                       calculate the forces on all particles
   call integrate                   integrate equations of motion
   t=t+deltat                       update the time with Δt
   call sample                      sample time averages
enddo
end

subroutine force                    calculation of the forces
en=0                                set total energy to zero
do i=1,npart
   f(i)=0                           set forces to zero
enddo
do i=1,npart-1                      consider all particle pairs
   do j=i+1,npart
      xr=x(i)-x(j)                  distance between i and j
      xr=xr-box*nint(xr/box)        apply periodic boundary conditions
      r2=xr**2                      distance, note that x**2 = x^2
      if(r2.lt.rc2) then            rc^2 = rc * rc, where rc is the cut-off distance
        r2i=1/r2
        r6i=r2i**3
        ff=48*r2i*r6i(r6i-0.5)      Lennard–Jones forces
        f(i)=f(i)+ff*xr             update force for particle i
        f(j)=f(j)-ff*xr             update force for particle j
        en=en+4*r6i*(r6i-1)-ecut    update total potential energy
      endif
   enddo
enddo
return
end

subroutine integrate                integration of equations of motion
sumv=0
sumv2=0
do i=1,npart
```

Table 7.1 (*Continued*).

```
      xx=2*x(i)-xm(i)+delt**2*f(i)      Verlet algorithm (Eq. (7.12))
      vi=(xx-xm(i))/(2*delt)            estimate for the velocity
      sumv2=sumv2+vi**2                 compute ∑ v²
      xm(i)=x(i)                        update positions
      x(i)=xx
    enddo
    temp=sumv2/(3*npart)                compute temperature
    etot=(en+0.5*sumv2)/npart           total energy per particle
    return
    end
```

[a]The subroutine `init` generates initial positions and velocities. Counters to compute ensemble averages are updated in the subroutine `sample`. The function `nint` rounds off to the nearest integer.

1. Read the initial conditions: temperature, number of particles, integration time step, etc.
2. Initialize the positions and velocities of all atoms in the system.
3. Compute the forces on all particles. This is the most time-consuming step.
4. Integrate the equations of motion. Steps 3 and 4 form the core of the MD simulation. They are repeated until the desired number of time steps is reached.
5. Compute the average of the measured quantities and stop.

When the simulation has finished, the results can be analyzed. First the simulation results need to be validated. Do the pressure, temperature, and energy components remain constant during the simulation? Do large molecules such as proteins drift over time? These inspections can provide valuable clues to whether a simulation has been successful or not. After validation, macroscopic properties of the system can be estimated. Examples of such properties are the viscosity of liquids or the self-diffusion coefficient. The estimation usually takes place by averaging one or more quantities over the simulation time and over a large amount of molecules.

7.2.3 Practical Issues

While evaluation of the molecular interactions and the integration of the equations of motion make up the largest part of the simulation, some practical problems also have to be accounted for. The simulated system is always finite, which means that a part of the molecules is located at the boundaries of the system. These molecules are not completely surrounded by other molecules. Such a situation would lead to serious boundary effects in systems with a relatively small number of atoms. Imagine a cubic system containing n^3 atoms arranged on a cubic lattice. A large fraction of them $(n^3 - (n-2)^3/n^3 \approx 6/n)$ is located at one of the surfaces of the box. This means that for 1000 particles ($n = 10$), 60% of all particles belong to the

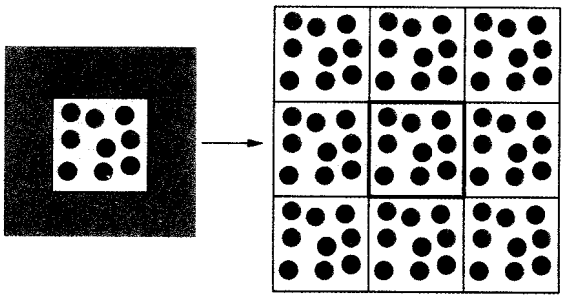

Figure 7.3 Periodic boundary conditions. A central box is surrounded by copies of itself.

surface. Even for a system of 10^6 particles ($n = 100$), still 6% of all particles belong to the surface. To minimize the influence of boundary effects, multiple images of the simulation box in all directions are used, in order to obtain a continuous system. This approach is known as periodic boundary conditions (see Figure 7.3). It is similar to treating the simulation system as a unit cell of a crystal. However, it is necessary to keep the amount of calculations manageable due to the limits in CPU time. Instead of calculating the interactions between all (original and periodic) atoms, only the interaction between one (original) atom and the closest image of the other atoms in one of the surrounding boxes could be calculated. This is called the nearest-image convention and it is satisfied automatically when the box length is at least twice the value for r_{cut}.

While molecular dynamics simulations have their obvious advantage in the amount of detail they provide, one should also be aware of the limitations:

1. The time span of current simulations is in the order of tens of nanoseconds for large systems. This is enough if the permeation of water molecules through channels is under investigation, but it will not be long enough to study the permeation of ions. It remains difficult to get information on ion-channel conduction and phenomena such as channel gating. Special techniques to study infrequent but important events will be discussed in Section 7.3.
2. The absence of atomic polarizability in the most force fields may present a flaw. Since the atoms are treated as point (partial) charges, a dynamic redistribution of electronic charge is not allowed. Instead, the polarizability is an average effect that is included by fitting the parameters of the Lennard–Jones potential to experimental data. This approach is sufficient to model bulk-like behavior of solvent molecules but it might not correctly represent individual water molecules inside a narrow channel pore.
3. The last limitation is caused by the classical treatment of the atoms. Using classical mechanics it is only possible to calculate the positions, accelerations, and forces between atoms. Due to the fact that protons and electrons are not explicitly modeled, proton conduction or reactions cannot be simulated. The blocking of protons would be worthwhile to investigate in this case, since it also occurs in the water channels.

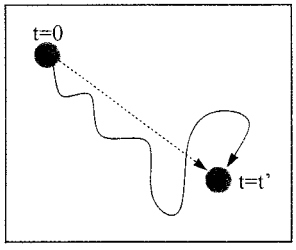

Figure 7.4 The self-diffusion coefficient can be computed from the displacements of individual particles (Eq. (7.16)).

These issues are still under development. Force fields for atomistic simulations are constantly updated and improved. Addressing the third point, for large systems the use of quantum mechanical methods is still too detailed and is not expected to provide solutions in the next few years on the scale of molecular dynamics. However, with computational power doubling roughly every other year we might be able to apply quantum mechanics to these large systems in the near future.

7.2.4 Diffusion

Transport properties like the diffusion coefficient can be computed directly from an MD simulation. The self-diffusion coefficient follows from the mean-squared displacements of individual particles:

$$D_{\alpha,\text{self}} = \frac{1}{2N} \lim_{t\to\infty} \frac{d}{dt} \left\langle \sum_{i=1}^{N} \left(r_{i\alpha}(t) - r_{i\alpha}(0)\right)^2 \right\rangle, \tag{7.16}$$

where $r_{i\alpha}(t)$ is the position of particle i at time t ($\alpha = x, y, z$) (see also Figure 7.4).

To describe the *transport* of particles due to a gradient, the Maxwell–Stefan approach is often more useful than the traditional Fick formulation, especially for multicomponent systems [17]. In the Maxwell–Stefan approach, the driving force for transport is a gradient in chemical potential, which is balanced by frictional forces between compounds (and/or the host structure). The Maxwell–Stefan diffusion coefficients can be computed directly from MD simulations *at equilibrium* (i.e., without gradients in chemical potential or concentration) [18], or, alternatively, from nonequilibrium MD (NEMD) simulations [19,20]. For a review on diffusion in porous materials and liquids, we refer the reader to Refs. [17,21,22] and references therein.

7.3 Rare Events

Diffusion coefficients of adsorbed molecules can vary as much as 10 orders of magnitude. If the diffusion coefficient is sufficiently high, straightforward MD can

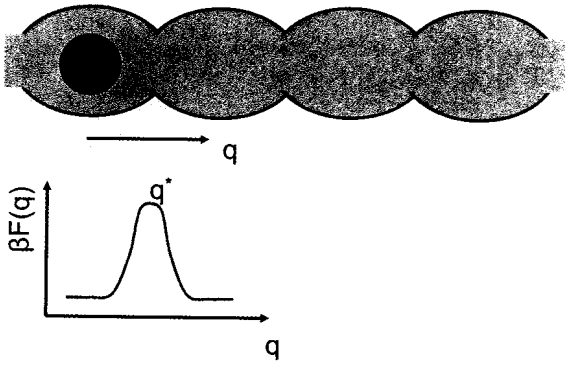

Figure 7.5 Schematic representation of a single molecule diffusing in a one-dimensional pore (a) and the free energy of this molecule as a function of the position q (b).

be used to simulate the system. If, however, the diffusion coefficient is very low, one often observes that molecules are trapped in low (free) energy sites and once in a while the molecule hops to another adsorption site. To compute a diffusion coefficient reliably, one has to observe a sufficient number of hops. Most of the CPU time is spent on molecules that "wait" at an adsorption site until a fluctuation gives them sufficient kinetic energy to take the barrier between adsorption site. The higher the barrier the longer the molecules remain trapped and – on the time scale of an MD simulation – such a hopping becomes a very rare event.

Special techniques have been developed to simulate such rare events [4]. The basic idea is to compute the hopping rate in two steps [23–25]. First, we compute the probability that a molecule can be found on top of the free-energy barrier followed by a separate simulation in which the average time is computed it takes for a molecule on top of the barrier to actually cross it.

Let us consider, as an example, the system shown in Figure 7.5. The adsorption sites are in the cavities and the windows are the diffusion barriers. In a rare-event simulation it is important to define an appropriate reaction coordinate which characterizes the progress of the "reaction." In case of a single atom, an obvious choice is the position along the tube and a typical free energy as a function of the reaction coordinate q as shown in Figure 7.6. The probability to find a molecule on top of the barrier can be computed directly from the free-energy profile

$$P(q^*)\mathrm{d}q = \frac{\exp[-\beta F(q^*)]\mathrm{d}q}{\int_{-\infty}^{q^*} \mathrm{d}q \exp[-\beta F(q)]}, \tag{7.17}$$

where q^* is defined as the top of the barrier. $F(q)$ is the free energy as a function of the order parameter. This free energy can be computed using the techniques described in Refs. [4, 26, 27].

The second step involves the average time taken by a molecule to cross the barrier. The simplest approach is to assume that transition state theory (TST) holds. A molecule that arrives at the top of the barrier is assumed to be in equilibrium

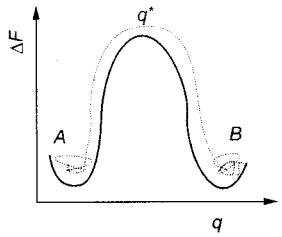

Figure 7.6 Free energy as a function of the reaction coordinate q.

with its surrounding. As a consequence, the velocity distribution $\mathcal{P}(v)$ is given by the Maxwell distribution corresponding to the temperature of the system:

$$\mathcal{P}(v) \propto \exp[-\beta m v^2/2]. \tag{7.18}$$

TST assumes that half of the molecules that reach the barrier also cross the barrier, i.e, those with a positive velocity of the order parameter. The TST approximation of the hopping rate is

$$k^{\mathrm{TST}} = \frac{1}{2}|\dot{q}|P(q^*) = \sqrt{\frac{k_{\mathrm{B}}T}{2\pi m}}\frac{\exp[-\beta F(q^*)]}{\int_{-\infty}^{q^*} \mathrm{d}q \exp[-\beta F(q)]}. \tag{7.19}$$

The advantage of TST is that one has to compute only the free energy as a function of the reaction coordinate to compute the hopping rate. The disadvantage is that one does not know in advance whether the assumptions underlying TST hold. In addition, TST also assumes that the transition state is known exactly, i.e., the top of the free energy, q^*, exactly corresponds to the true transition state. In practice, we do not know the free energy exactly and we, therefore, can only approximate the transition state.

More importantly, in the system we consider in Figure 7.5, the choice of the reaction coordinate is straightforward. However, in practice one has to be very careful. Consider, for example, the zeolites shown in Figure 7.7. Both the zeolites are one-dimension channels of cages connected via narrow windows. In analogy with the system of Figure 7.5, one would take as order parameter the position of the atom projected on the axis of the channel (red-short dashed line), but we could have also taken a projection on a line through the window that has an angle with the channel axes (blue-long dashed line). Depending on the particular choice, the free energy of the transition state will be different. If we use these free energies and compute the hopping rate using Eq. (7.19), we would find different values of this hopping rate. In fact, TST gives an upper limit; the true hopping rate is always lower compared to the TST result. It is, therefore, important to select the reaction coordinate that gives the highest free energy of the transition state. Since the free energy appears in the exponential in Eq. (7.19), TST can give a large error in the hopping rate if the choice of reaction coordinate is far from optimal. It may look strange to use a reaction coordinate that does not correspond to the direction

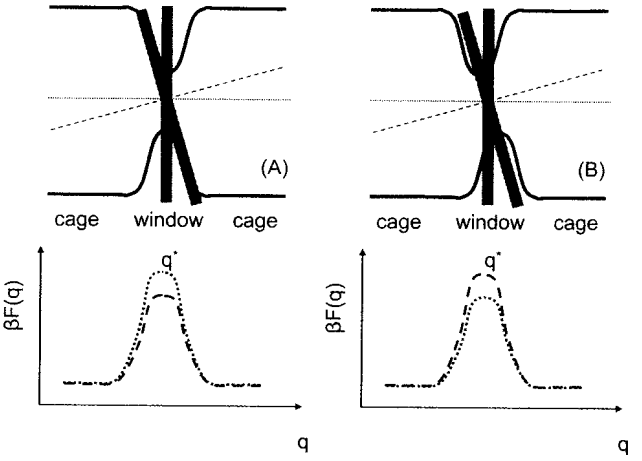

Figure 7.7 Comparison of two different choices of the reaction coordinate q. The red (short dashed) choice is parallel to the axis of the zeolite, while the blue choice (long dashed) makes an angle with the horizontal axis. In the left figure the window is perpendicular to the axis of the zeolite while in the right the window takes an angle. The shaded areas show the part of the zeolite of which the free energy is projected on the transition state $q*$. The bottom figures show that depending on the choice of reaction coordinate, the free energy of the transition state, $F(q^*)$, has a different value.

of diffusion. Figure 7.7 shows that for the zeolite in which the window is not perpendicular to the channel axis this choice results in a higher free energy of the transition state. This example illustrates that even for diffusion of an atom the choice of reaction coordinate can be nontrivial. For molecules, the number of possible reaction coordinates increases dramatically and it will be impossible to compute the free energies for all possible reaction coordinates. Finally, even if one is able to select the optimal reaction coordinate, TST may not give the correct hopping rate since in Eq. (7.19) it is assumed that all particles that start in one cage and arrive on top of the barrier with a positive velocity of the order parameter arrive in the *product* cage. TST ignores the possibility that such a particle recrosses the barrier and returns in the cage it originates due to, for example, collisions with the zeolite atoms.

To compute the "true" hopping rate, one has to correct the TST to take into account the recrossing of the barrier. These recrossings can be intrinsic to the system or due to the nonoptimum choice of the reaction coordinate. This correction is obtained using, for example, the Bennet–Chandler [23–25] approach in which MD simulations are performed to compute the transmission coefficient κ. The computation of the transmission coefficient κ involves many MD simulations, which all start on top of the barrier (q^*) and of which we determine the fraction that ends up in the product cage. The time-dependent transmission coefficient is defined as

$$\kappa(t) = \frac{k(t)}{k^{\mathrm{TST}}} = \frac{\langle \dot{q}(0)\delta(q(0) - q^*)\theta(q(t) - q^*)\rangle}{0.5\langle|\dot{q}(0)|\rangle}, \tag{7.20}$$

where $\theta(q)$ is the Heaviside function ($\theta(q) = 1$ if $q > 0$ and $\theta(q) = 0$ otherwise) and $\delta(q)$ the Dirac delta function. The delta function in Eq. (7.20) indicates that the trajectories are initiated on top of the barrier and the Heaviside function takes a value if the particle is on the product side of the barrier. Equation (7.20) shows that if all particles with a positive velocity of the order parameter stay in the product cage the transmission coefficient is one and TST gives the correct result. For those systems in which barrier recrossings are important, Eq. (7.20) gives a plateau value for intermediate times that can be used to correct the TST hopping rate. The important aspect of these MD simulations is that they are initiated on top of the barrier, which is a very unfavorable configuration for which the relaxation to equilibrium, one of the cages, is relatively fast. These simulations, therefore, do not require much CPU time for most systems.

In our discussion, we focus on one-dimensional order parameters and for some systems it can be desirable to use a multidimensional order parameter. TST can be generalized to higher dimensions and one has to locate the saddle point in such a multidimensional space. Special techniques have been developed for this [28]. For some systems, however, the dynamics on top of the barrier can be diffusive; because of collisions with the atoms of zeolite a particle may spend a relatively long time on top of the barrier before it falls in one of the cages. For such systems, it can be advantageous to compute the hopping rate using the approach of Ruiz–Montero et al. [29]. In the methods we have discussed so far we assume that a good estimate of the transition state can be obtained. Although Eq. (7.20) can be used to correct an unfortunate choice of reaction coordinate, if the transmission coefficient is very small it is expensive to compute it accurately. Special techniques like transition path sampling [7, 30–34] have been developed to compute hopping rates without prior knowledge of the reaction coordinate. This method can also be used to check whether the assumed transition state resembles to true transition state.

7.4 Monte Carlo

7.4.1 Introduction

In molecular dynamics, one integrates the equation of motion for a system of particles (atoms, molecules) that interact according to a certain force field, and from this trajectory one can calculate *time averages*. According to the ergodicity hypothesis [4], these *time averages* should, in principle, be equal to averages over *all possible configurations* of the particles in the system.

Consider a system where the positions of all N particles in the system are denoted by the vector $\mathbf{r}^N$. From elementary statistical thermodynamics, it is well known that the statistical weight of each configuration $\mathbf{r}^N$ is not equal. At a fixed number of particles N, fixed volume V and absolute temperature T, the statistical weight of a system with configuration $\mathbf{r}^N$ is proportional to $\exp[-\beta U(\mathbf{r}^N)]$, where $\beta = 1/(k_B T)$,

k_B is the Boltzmann factor ($\approx 1.38066 \times 10^{-23}$ J K^{-1}), and $U(\mathbf{r}^N)$ is the potential energy that depends on the positions of all the particles in the system. Therefore, ensemble averages that only depend on the positions of the particles in the system (e.g., the average energy or average pressure) can be calculated in the following way:

$$\langle X \rangle = \frac{\int d\mathbf{r}^N X(\mathbf{r}^N) \exp[-\beta U(\mathbf{r}^N)]}{\int d\mathbf{r}^N \exp[-\beta U(\mathbf{r}^N)]}. \tag{7.21}$$

Naively, one might think that the ensemble average $\langle X \rangle$ can be evaluated by the conventional numerical integration techniques such as numerical quadrature. However, evaluating the integrand on a grid in the high-dimensional phase space is impossible as the number of gridpoints becomes more than astronomically large. For instance, $N = 100$ particles in $D = 3$ dimensions using a very rough grid of only $m = 5$ gridpoints already leads to $m^{DN} = 5^{300}$ gridpoints at which this integrand has to be evaluated. This is impossible within the lifetime of our universe. In addition, suppose that we would somehow be able to perform the integration of Eq. (7.21). Our claim is that the statistical error will then be so large that the result is not meaningful anyway. The reason is that when two particles overlap, the potential energy is extremely large and therefore the Boltzmann factor equals zero. In fact, it turns out that for a typical liquid this is the case for almost all configurations $\mathbf{r}^N$ and only an extremely small part of the phase space will have a nonzero contribution to the integrals of Eq. (7.21) (see also Figure 7.8).

We therefore have to resort to more suitable numerical techniques. One such a technique is the Monte Carlo method. The basic idea in the Metropolis Monte Carlo method is that phase points are generated in phase space according to the desired probability. In this way, one avoids the numerical evaluation of the integrand on a high-dimensional grid where most of the gridpoints result in configurations with an extremely low Boltzmann weight. Consider, for example, a sequence of configurations $\mathbf{r}_1^N, \mathbf{r}_2^N, \mathbf{r}_3^N, \ldots, \mathbf{r}_n^N$. If this sequence consists of *random* configurations, the ensemble average $\langle X \rangle$ is simply an unweighted average

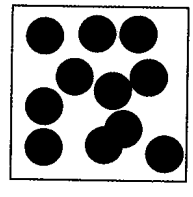

Figure 7.8 If particles are inserted *randomly* in a simulation box, already at a moderate density it is very likely that there is at least a single particle overlap in the system, resulting in a Boltzmann weight $\exp[-\beta U]$ that is extremely low.

$$\langle X\rangle = \lim_{n\to\infty} \frac{\sum_{i=1}^{n} X(\mathbf{r}_i^N)\exp[-\beta U(\mathbf{r}_i^N)]}{\sum_{i=1}^{n}\exp[-\beta U(\mathbf{r}_i^N)]}. \tag{7.22}$$

Again, computing $\langle X\rangle$ in this way is often not meaningful as $\exp[-\beta U(\mathbf{r}_i^N)]$ is nearly always zero for random configurations. However, suppose that we are able to generate $\mathbf{r}_i^N$ in such a way that the probability of generating $\mathbf{r}_i^N$ is proportional to $\exp[-\beta U(\mathbf{r}_i^N)]$, then the ensemble average is simply

$$\langle X\rangle = \lim_{n\to\infty} \frac{\sum_{i=1}^{n} X(\mathbf{r}_i^N)}{n}. \tag{7.23}$$

This sampling is called Metropolis sampling [35]. As we cannot generate an infinitely long sequence of configurations on the computer, we estimate the ensemble average by taking a large value of n.

We will now present a method to generate $\mathbf{r}_i^N$ with a probability proportional to $\exp\left[-\beta U(\mathbf{r}_i^N)\right]$ (Metropolis sampling). For a system of N particles in volume V, this works as follows: We first generate a configuration of N particles at positions $o = \mathbf{r}_o^N$ with a nonvanishing Boltzmann weight $\exp[-\beta U(o)]$. Next we generate a random new trial configuration $n = \mathbf{r}_n^N$, for example, by picking randomly a particle and by displacing it randomly. The Boltzmann weight of this new trial configuration is $\exp[-\beta U(n)]$. We must now decide whether we accept or reject this trial configuration satisfying the constraint that on average the probability of finding the system in a configuration $\mathbf{r}^N$ is proportional to the probability distribution $f_c(\mathbf{r}^N) = \exp[-\beta U(\mathbf{r}^N)]$. If we reject this trial move, the next element in our sequence of configurations (Eq. (7.23)) will be $\mathbf{r}_o^N$, otherwise it will be $\mathbf{r}_n^N$. The rule to accept or reject $\mathbf{r}_n^N$ must satisfy three conditions: (1) the number of points in any configuration o should be proportional to $f_c(o)$; (2) all configurations can, in principle, be visited, (3) in equilibrium, the average number of accepted trial moves leaving state o should be equal to the average number of accepted trial moves from all other states to state o. This is called the balance condition [36]. It is however, convenient to impose a much stronger condition (detailed-balance condition); namely that in equilibrium the average number of accepted trial moves leaving state o to state n is equal to the average number of accepted trial moves leaving state n to state o, which essentially means that all "fluxes of configurations" are equal:

$$f_c(o)\pi(o\to n) = f_c(n)\pi(n\to o), \tag{7.24}$$

where $\pi(o\to n)$ denotes the transition probability that can be split into two terms: (1) the probability $\alpha(o\to n)$ to perform a trial move from $o\to n$, and (2) the probability $\mathrm{acc}(o\to n)$ of accepting a trial move from $o\to n$:

$$\pi(o\to n) = \alpha(o\to n)\mathrm{acc}(o\to n). \tag{7.25}$$

In the original Metropolis scheme [35], $\alpha(o \rightarrow n)$ is chosen to be symmetric, i. $\alpha(o \rightarrow n) = \alpha(n \rightarrow o)$ and we arrive at

$$\frac{\text{acc}(o \rightarrow n)}{\text{acc}(n \rightarrow o)} = \frac{f_c(n)}{f_c(o)} = \exp[-\beta(U(n) - U(o))] = \exp[-\beta \Delta U]. \quad (7.2$$

An example of such a symmetric transition is the random displacement of randomly selected particle. Many choices for $\text{acc}(o \rightarrow n)$ satisfy this condition, b the commonly used choice is of Metropolis:

$$\begin{aligned} \text{acc}(o \rightarrow n) &= f_c(n)/f_c(o) \text{ if } f_c(n) < f_c(o) \\ &= 1 \quad \text{if } f_c(n) \geq f_c(o). \end{aligned} \quad (7.2'$$

More explicitly, to decide whether a trial move will be accepted or rejected w generate a random number, denoted by *ranf()*, from a uniform distribution in th interval [0,1]. If $ranf() < \text{acc}(o \rightarrow n)$, we accept the trial move and reject it otherwis This rule satisfies the Metropolis condition and can be written as

$$\text{acc}(o \rightarrow n) = \min(1, \exp[-\beta(U(n) - U(o))]) = \min(1, \exp[-\beta \Delta U]), \quad (7.2$$

where $\min(a, b) = a$ when $a < b$, and b otherwise. This means that new confi urations that lower the energy are always accepted, and new configurations tha increase the total energy are accepted with a certain probability that depend on their energy difference and the temperature. In summary, our Monte Carl approach to compute ensemble averages of a system of N particles in volume V i as follows (see also Table 7.2):

1. Generate an initial configuration.
2. Start with a configuration o, and calculate its energy $U(o)$.
3. Select a particle at random.
4. Give the selected particle a random displacement $x(n) = x(o) + \Delta$, where Δ is a uniformly distributed random number from $[-\Delta x, \Delta x]$
5. Calculate the energy $U(n)$ of the new configuration n.
6. Accept the trial move with a probability

$$\text{acc}(o \rightarrow n) = \min(1, \exp[-\beta(U(n) - U(o))]) = \min(1, \exp[-\beta \Delta U]) \quad (7.29)$$

7. Update the calculation of ensemble averages, also after rejected trial moves.

A pseudocomputer code of this algorithm is listed in Table 7.2.

7.4.2 The Grand-Canonical Ensemble

In the previous section, we have discussed the Monte Carlo technique for a system with a constant number of particles N and a constant volume V at temperature T.

Table 7.2 Pseudocomputer code of the Monte Carlo algorithm described in the text.[a]

```
program mc                                          basic Monte Carlo algorithm
do icycle=1,ncycle                                  number of MC cycles
   call move                                        displace a randomly selected particle
   if(mod(icycle,nsample).eq.0)                     collect ensemble averages
+     call sample                                   each nsample MC cycles
enddo
end


subroutine move                                     perform a trial move
i=int(ranf()*npart)+1                               select particle i at random
call energy(x(i),eold,i)                            calculate energy of old configuration
xnew=x(i)+(2*ranf()-1)*Δx                           random displacement of particle i
call energy(xnew,enew,i)                            calculate energy of new configuration
if(ranf().lt.exp(-beta*(enew-eold))                 acceptance rule
+   x(i)=xnew                                       accept new position of particle i
return
end


subroutine energy(xi,e,i)                           subroutine to calculate the energy
e=0.0
do j=1,npart                                        loop over all particles
   if (j.ne.i) then                                 if particle j is not i
      dx=x(j)-xi                                    calculate distance between particles i and j
      dx=dx-box*nint(dx/box)                        apply periodic boundary conditions
      eij=4*(1.0/(dx**12)-1.0/(dx**6))              calculate Lennard–Jones potential energy
      e=e+eij
   endif
enddo
return
end
```

[a]The program consists of three parts. The main program `mc` controls the simulation. The subroutine `move` displaces a randomly selected particle, and the subroutine `energy` computes the energy of a particle. The function `ranf()` generates a uniformly distributed random number between 0 and 1. The function `int` truncates a real number to its inter value, while the function `nint` converts a real number to its nearest integer value.

However, the choice of a constant number of particles N is not always a convenient one.[1)] Consider, for example, the case where one is interested in studying the adsorption of guest molecules inside a microporous host such as a zeolite or carbon nanotube. In this system, the gas phase (with pressure P) is in equilibrium with the host material in such a way that the chemical potential μ of the guest molecules in the gas phase and in the porous host is identical. A direct MD or

1) For a description of other ensembles that are useful in molecular simulation, we refer the reader to Ref. [4].

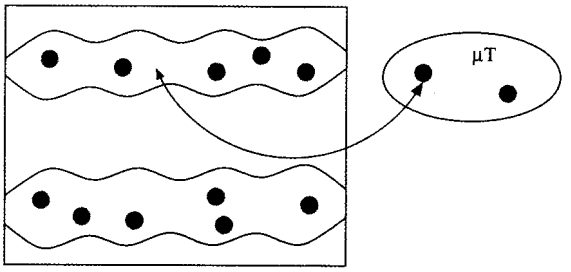

Figure 7.9 Illustration of the grand-canonical ensemble. A porous host structure is coupled to an infinitely large reservoir of guest molecules at temperature T and chemical potential μ. There is no direct interaction between the molecules in the reservoir and in the host. Monte Carlo trial moves are used to insert/remove guest molecules in/from the host structure. In this way, one can compute the adsorption isotherm (average number of guest molecules $\langle N \rangle$ in the host structure as a function of the chemical potential μ).

MC simulation of the gas phase and the porous solid in a single simulation box is not very convenient as this will create a large surface area and the zeolite cannot be periodic in all directions. In this case, the grand-canonical (μVT) ensemble is more useful. In this ensemble, the volume V and chemical potential μ of the guest molecules are fixed, while the number of adsorbed guest molecules (N) is fluctuating. The host structure is coupled to an infinitely large particle reservoir at a certain chemical potential μ and temperature T (see Figure 7.9). The reservoir and the host structure can exchange particles. For a given chemical potential μ and temperature T one can compute average of the number of particles $\langle N \rangle$ adsorbed in the host structure.

From statistical mechanics, it is known that the statistical weight of the host structure with N guest molecules each consisting of a single atom with coordinates $\mathbf{r}^N$ is proportional to

$$\mathcal{W}(N, \mathbf{r}^N) \propto \frac{V^N \exp[\beta\mu N - \beta U(\mathbf{r}^N)]}{\Lambda^{3N} N!}. \tag{7.30}$$

In this equation, V is the volume of the host system, μ is the chemical potential of the guest molecules in the reservoir, and Λ is the thermal wave vector

$$\Lambda = \frac{h}{\sqrt{2\pi m k_B T}} = \frac{h}{\sqrt{2\pi m/\beta}}, \tag{7.31}$$

where m is the mass of a guest molecule. The acceptance rules for the Monte Carlo method in the grand-canonical ensemble follow directly from Eq. (7.30), see also Refs. [4, 8]. The following trial moves are used:

1. *Particle displacement.* A randomly selected particle in the host is given a random displacement. This move is accepted according to Eq. (7.28)

2. *Particle insertion.* A guest molecule is inserted at a random position in the host structure. This move is accepted with a probability

$$\mathrm{acc}(N \to N+1) = \min\left(1, \frac{V}{\Lambda^3(N+1)} \exp[\beta(\mu - U(\mathbf{r}^{N+1}) + U(\mathbf{r}^N))]\right), \tag{7.32}$$

3. *Particle deletion.* A randomly selected guest molecule in the host structure is deleted. This move is accepted with a probability

$$\mathrm{acc}(N \to N-1) = \min\left(1, \frac{N\Lambda^3}{V} \exp[-\beta(\mu + U(\mathbf{r}^{N-1}) - U(\mathbf{r}^N))]\right). \tag{7.33}$$

It can be shown that Eqs. (7.32) and (7.33) obey detailed balance. The trial moves are schematically illustrated in Figure 7.10. It is selected at random which move is performed. For the simulation of mixtures, it can be very advantageous to include

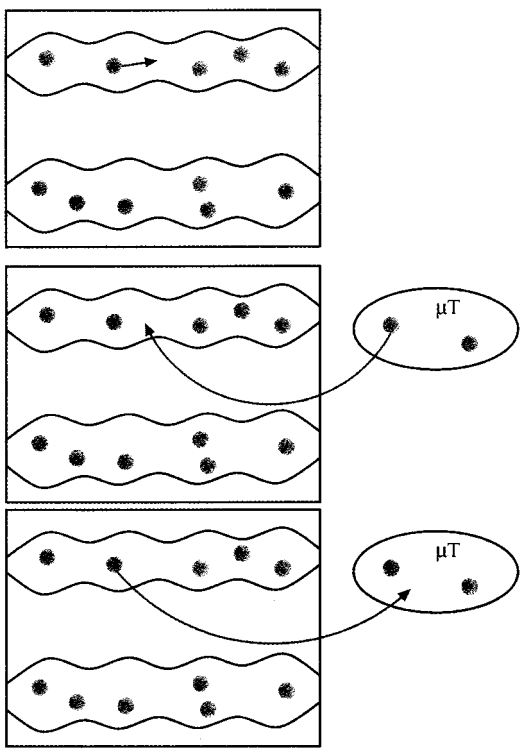

Figure 7.10 Trial moves in the grand-canonical ensemble: particle displacement, particle insertion, and particle deletion. Here, it is convenient to select the trial moves at random [5].

trial moves that change the identity of a guest molecule in the host (semigrand ensemble). For details, we refer the reader to Refs. [37–40].

The chemical potential of the molecules in the reservoir can be converted to the pressure P or fugacity f of the reservoir. The fugacity of a system is defined as the pressure that the system would have if it would be an ideal gas, at exactly the same chemical potential. As for an ideal gas $\mu = k_B T \ln \rho \Lambda^3$ it follows directly that

$$f = \frac{\exp[\beta\mu]}{\beta\Lambda^3}. \tag{7.34}$$

It can be shown that the pressure P and the fugacity f are related according to [41]

$$\ln \frac{f}{P} = \int_0^P dP' \frac{Z(P') - 1}{P'}, \tag{7.35}$$

where Z is the compressibility factor $Z = P\overline{V}/k_B T$ and $\overline{V} = V/N$ is the volume per molecule. The compressibility Z can be taken from experiments or it can be computed in a separate simulation.

7.4.3 Chain Molecules

In the previous section, we have shown that the grand-canonical Monte Carlo technique critically relies on the insertion and deletion of guest molecules. For small guest molecules like methane or ethane, random insertion of guest molecules in the host structure does not pose any problem. However, for longer chain molecules, random insertion of a guest molecule becomes increasingly more difficult as nearly always there is an overlap with one of the zeolite atoms (see Figure 7.11).

The low acceptance probability can be overcome by using the configurational-bias Monte Carlo (CBMC) algorithm [4, 42–45]. This algorithm uses the Rosenbluth

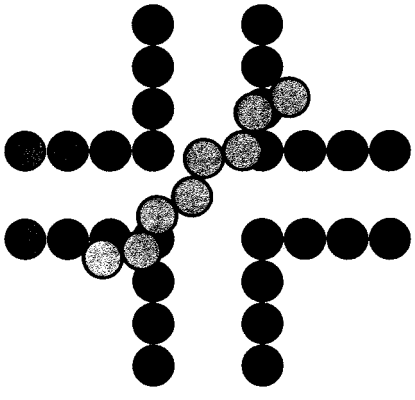

Figure 7.11 Random insertion of long-chain molecules is difficult as nearly always there will be an overlap with the host structure.

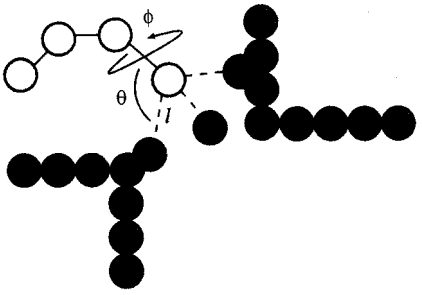

Figure 7.12 Schematic representation of the CBMC technique. k trial directions (each represented by values for l, θ, ϕ) are generated according to Eq. (7.36), and one of the trial directions is selected according to Eq. (7.37). Here, $k = 3$.

scheme [46] to generate chain configurations. This introduces a bias, which can be removed exactly in the acceptance rules. The CBMC algorithm significantly improves the acceptance probability of insertions and deletions in the grand-canonical ensemble, as well as the internal rearrangements of adsorbed chains.

The algorithm works as follows. Consider a linear chain consisting of M monomers. The interactions of this chain can be split into *bonded* interactions (e.g., bond-stretching, bond-bending, torsion) and *nonbonded* interactions (Lennard–Jones, Coulombic interactions). The nonbonded interactions can be intramolecular (within the same molecule) or intermolecular (with other molecules or the host structure). For the insertion of a chain molecule, the following steps are executed (see also Figure 7.12):

1. The first monomer of a chain is placed at a random position in the system and its energy u_1 is recorded.
2. For the next monomer, k trial positions are generated. The position of each trial direction is generated with a probability proportional to the Boltzmann factor of the *bonded* interactions. In general, the position of the next monomer can be characterized by the bond length l, bond-bending angle θ, and a torsion angle ϕ with corresponding energies $u_{\text{stretch}}(l)$, $u_{\text{bend}}(\theta)$, and u_{torsion}. For each trial direction, the values of l, θ, and ϕ are generated according to

$$\begin{aligned} P(l) &\propto dl\, l^2 \exp[-\beta u_{\text{stretch}}(l)] \\ P(\theta) &\propto d\theta\, \sin(\theta) \exp[-\beta u_{\text{bend}}(\theta)] \\ P(\phi) &\propto d\phi\, \exp[-\beta u_{\text{torsion}}(\phi)]. \end{aligned} \tag{7.36}$$

A method to draw random numbers from arbitrary distributions is presented in Ref. [47].

3. One of the trial directions (a) is selected with a probability proportional to the Boltzmann factor of the *nonbonded* energy of that monomer:

$$P_a = \frac{\exp[-\beta u_{ia}]}{\sum_{j=1}^{k} \exp[-\beta u_{ij}]}, \tag{7.37}$$

where u_{ij} is the nonbonded energy of the jth trial direction for monomer i. In is way, it is very unlikely that unfavorable trial directions (with a low Boltzmann weight due to a large u_{ij}) are chosen and in this way overlaps with other particles are avoided.

4. This process is continued until the chain has been grown. The Rosenbluth weight of the generated chain equals

$$W = \frac{\exp[-\beta u_1] \prod_{i=2}^{M} \left[\sum_{i=1}^{k} \exp[-\beta u_{ij}] \right]}{k^{M-1}}. \tag{7.38}$$

The insertion of a chain is accepted with a probability

$$\text{acc}(N \rightarrow N+1) = \min\left(1, \frac{\beta V f W}{(N+1)\langle W_{IG}\rangle}\right), \tag{7.39}$$

where f is the fugacity, V the volume of the simulation box, W is the Rosenbluth weight of the generated chain, and $\langle W_{IG}\rangle$ is the Rosenbluth weight of an isolated chain (ideal gas phase). In Ref. [4], it is shown that this scheme obeys detailed balance. Note that for the insertion of a chain consisting of a single monomer, this equation reduces to Eq. (7.32).

For the removal of a molecule, the same procedure is followed, but with the difference that the first trial direction is always the old chain, and this trial direction is always selected. One can show that the acceptance rule equals

$$\text{acc}(N \rightarrow N-1) = \min\left(1, \frac{N\langle W_{IG}\rangle}{\beta V f W}\right). \tag{7.40}$$

The CBMC technique can also be applied to branched molecules [40,48–50] and to various ensembles [4]. Over the years, various improvements of the algorithm have been proposed, we refer the reader to Refs. [48–57].

7.4.4
Calculating Adsorption Properties

In the previous sections, we have shown how to compute the adsorption isotherm (number of adsorbed guest molecules $\langle N\rangle$ as a function of the pressure, fugacity,

or chemical potential). From the adsorption isotherm, two other properties that are directly accessible by experiments can be computed: the Henry coefficient and the heat of adsorption. For an overview on how to compute the adsorption entropy or free energy we refer the reader to Ref. [58].

7.4.5 Henry Coefficient

In the limit of a very low loading of guest molecules, the adsorption isotherm is a linear function:

$$\langle N \rangle = K_H V P, \tag{7.41}$$

where V is the volume of the host, P the pressure of the gas phase, and K_H the Henry coefficient (in units of molecules per unit of volume per unit of pressure). In principle, the Henry coefficient follows from the result of grand-canonical simulations. However, the Henry coefficient can also be computed directly. In molecular simulations, the most convenient way to calculate the Henry coefficient is using Widom's test particle method (see Figure 7.13) [1, 59]:

$$K_H = \beta \times \exp[-\beta \mu_{ex}] = \beta \times \frac{\langle \exp[-\beta u^+] \rangle}{\langle \exp[-\beta u^+_{IG}] \rangle}, \tag{7.42}$$

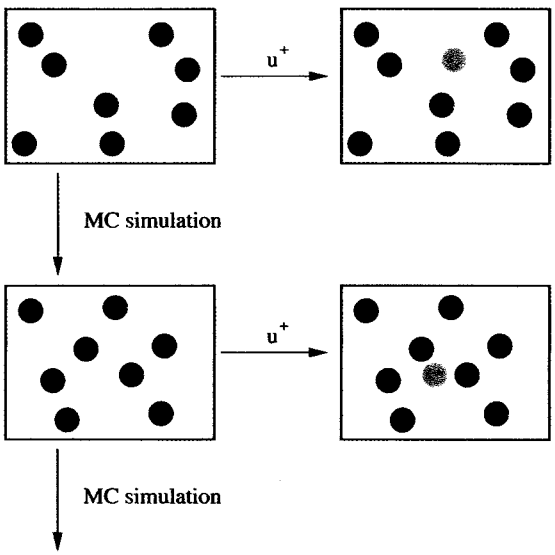

Figure 7.13 Schematic illustration of Widom's test particle method. During a simulation in the *NVT* ensemble, a test particle is positioned at a random position and the energy change of the system due to this (u^+) is computed. Note that the insertion is actually never accepted. The excess chemical potential follows from $\mu_{ex} = -k_B T \ln \langle \exp[-\beta u^+] \rangle$, where the brackets $\langle \cdots \rangle$ denote an average over all positions of the test particle and over all configurations of the system. When CBMC is used, the excess chemical potential equals $\mu_{ex} = -k_B T \ln [\langle W \rangle / \langle W_{IG} \rangle]$.

where μ_{ex} is the excess chemical potential of the guest molecules, $\langle \exp[-\beta u^+] \rangle$ is the average Boltzmann factor of a test (guest) molecule inserted at a random position in the host, and $\langle \exp[-\beta u_{IG}^+] \rangle$ is the average Boltzmann factor of a test (guest) molecule inserted at a random position in an empty box without the presence of the host (often referred to as an isolated chain). For chain molecules like alkanes, it is well known that insertion of a test chain at a random position in the zeolite nearly always results in overlaps with zeolite atoms, and therefore the sampling statistics of the average $\langle \exp[-\beta u^+] \rangle$ will be extremely poor [60]. For chains that are not too long (< 50 monomers), it is convenient to use the CBMC method to insert test chains. In this case, the Henry coefficient is computed from [60]

$$K_H = \beta \times \frac{\langle W \rangle}{\langle W_{IG} \rangle}, \tag{7.43}$$

where $\langle W \rangle$ is the average Rosenbluth weight of a test chain in the host and $\langle W_{IG} \rangle$ is the average Rosenbluth weight of an isolated test chain in an empty box.

7.4.6 Heat of Adsorption

The heat of adsorption describes the change in enthalpy when a molecule is transferred from the gas phase into the pores of a zeolite. In experiments, the heat of adsorption is usually computed using the Clausius–Clapeyron equation [61] or direct calorimetry experiments [62]. In molecular simulations, there are several routes to compute this quantity [63]:

1. Directly from the Clausius–Clapeyron equation

$$-q = \Delta H = k_B \left(\frac{\partial \ln [P/P_0]}{\partial T^{-1}} \right)_{\langle N \rangle = \text{constant}} = \left(\frac{\partial \ln [P/P_0]}{\partial \beta} \right)_{\langle N \rangle = \text{constant}}, \tag{7.44}$$

where P_0 is an arbitrary reference pressure and $R = k_B/N_{av}$ is the gas constant. Note that the differentiation is carried out at a constant number of adsorbed guest molecules ($\langle N \rangle$). At low loading, this equation becomes

$$-q = \Delta H = -\frac{\partial \ln [K_H/K_{H0}]}{\partial \beta}, \tag{7.45}$$

where K_{H0} is an arbitrary constant (that has the same units as the Henry coefficient K_H). Note that this method requires more than one simulation as one needs to compute temperature derivatives.

2. From energy differences computed in the canonical (NVT) ensemble [60, 61]

$$-q = \Delta H = \langle U_1 \rangle_1 - \langle U_0 \rangle_0 - \langle U_g \rangle - \frac{1}{\beta}, \tag{7.46}$$

where U_N is the total energy of a host with N guest molecules present, $\langle \cdots \rangle_X$ refers to an ensemble average at constant V,T, and X guest molecules, and $\langle U_g \rangle$ is the average energy of an isolated guest molecule (without the host present). The average $\langle U_g \rangle$ for a certain guest molecule only depends on temperature and needs to be calculated only once. Note that Eq. (7.46) only applies at low loading. It turns out that this method has severe difficulties when applied to zeolites with strongly interaction nonframework cations [63].

3. From energy/particle fluctuations in the grand-canonical (μVT) ensemble [64]. We can approximate the change in potential energy upon adsorption of a single guest molecule:

$$\langle U_{N+1} \rangle_{N+1} - \langle U_N \rangle_N \approx \left(\frac{\partial \langle U \rangle_\mu}{\partial \langle N \rangle_\mu} \right)_\beta = \frac{\left(\frac{\partial \langle U \rangle_\mu}{\partial \mu} \right)_\beta}{\left(\frac{\partial \langle N \rangle_\mu}{\partial \mu} \right)_\beta}$$
$$= \frac{\langle U \times N \rangle_\mu - \langle U \rangle_\mu \langle N \rangle_\mu}{\langle N^2 \rangle_\mu - \langle N \rangle_\mu \langle N \rangle_\mu}, \tag{7.47}$$

where the brackets $\langle \cdots \rangle_\mu$ denote an average in the grand-canonical ensemble, N is the number of guest molecules, and μ is the chemical potential of the guest molecules. This leads to [64]

$$-q = \Delta H = \frac{\langle U \times N \rangle_\mu - \langle U \rangle_\mu \langle N \rangle_\mu}{\langle N^2 \rangle_\mu - \langle N \rangle_\mu \langle N \rangle_\mu} - \langle U_g \rangle - \frac{1}{\beta}, \tag{7.48}$$

where ΔH in Eq. (7.48) is usually defined as the *isosteric heat of adsorption* and it is often applied at nonzero loading. It is assumed here that the gas phase is ideal.

4. From a direct application Widom's test particle method using the Rosenbluth method [63]

$$-q = \Delta H = \frac{\langle (U_N + u^+) \times W \rangle_N}{\langle W \rangle_N} - \langle U_N \rangle_N - \langle U_g \rangle - \frac{1}{\beta}, \tag{7.49}$$

where W is the Rosenbluth weight of a test chain and u^+ its energy. $\langle \cdots \rangle_X$ refers to an ensemble average at constant V,T, and X guest molecules. This method is equivalent to Eq. (7.48) and it requires only a single simulation of the host structure.

Acknowledgments

TJHV acknowledges financial support from the Netherlands Organization for Scientific Research (NWO-CW) through a VIDI grant.

References

1 Allen, M.P., Tildesley, D.J. *Computer Simulation of Liquids*, Clarendon Press; Oxford, **1987**.

2 Landau, D.P., Binder, K. *A Guide to Monte Carlo Simulations in Statistical Physics*, Cambridge University Press; Cambridge, **2000**.

3 Bolhuis, P.G., Chandler, D., Dellago, C., Geissler, P.G. *Annu. Rev. Phys. Chem.* **2002**, 53, 291–318.

4 Frenkel, D., Smit, B. *Understanding Molecular Simulation: from Algorithms to Applications*, 2nd edn., Academic Press; San Diego, **2002**.

5 Frenkel, D. *Pnas* **2004**, 101, 17571–17575.

6 Rapaport, D.C. *The Art of Molecular Dynamics Simulation*, 2nd edn., Cambridge University Press; Cambridge, **2004**.

7 van Erp, T.S., Bolhuis, P.G. *J. Comput. Phys.* **2005**, 205, 157–181.

8 Vlugt, T.J.H., Van der Eerden, J.P.J.M., Dijkstra, M., Smit, B., Frenkel, D., *Introduction to Molecular Simulation and Statistical Thermodynamics*, http://www.phys.uu.nl/~vlugt, **2008**.

9 Martin, M.G., Siepmann, J.I. *J. Phys. Chem. B* **1998**, 102, 2569–2577.

10 http://siepmann6.chem.umn.edu.

11 MacKerell, A.D., Banavali, N., Foloppe, N. *Biopolymers* **2001**, 56, 257–265.

12 Case, D.A., Cheatham, T.E., Darden, T., Gohlke, H., Luo, R., Merz, K.M.M., Onufriev, A., Simmerling, C., Wang, B., Woods, R.J. *J. Comput. Chem.* **2005**, 26, 1667–1803.

13 Ewald, P.P. *Ann. Phys.* **1921**, 64, 253–287.

14 Fincham, D. *Mol. Sim.* **1994**, 13, 1–9.

15 Wolf, D., Keblinski, P., Phillpot, S.R., Eggebrecht, J. *J. Chem. Phys.* **1999**, 110, 8254–8282.

16 Fennell, C.J., Gezelter, J.D. *J. Chem. Phys.* **2006**, 124, 234104.

17 Krishna, R., Wesselingh, J.A. *Chem. Eng. Sci.* **1997**, 52, 861–911.

18 Wheeler, D.R., Newman, J. *J. Phys. Chem. B* **2004**, 108, 18353–18361.

19 Chempath, S., Krishna, R., Snurr, R.Q. *J. Phys. Chem. B* **2004**, 108, 13481–13491.

20 Wheeler, D.R., Newman, J. *J. Phys. Chem. B* **2004**, 108, 18362–18367.

21 Arya, G., Chang, H.C., Maginn, E.J. *J. Chem. Phys.* **2001**, 115, 8112–8124.

22 Dubbeldam, D., Snurr, R.Q. *Mol. Sim.* **2007**, 33, 305–325.

23 Bennett, C.H. *J. Comp. Phys.* **1976**, 22, 245–268.

24 Bennett, C.H., in *Algorithms for Chemical Computations*, ACS Symposium Series, edited by Christoffersen, R.E. American Chemical Society; Washington, D.C., **1977**, pp. 63–97.

25 Chandler, D. *J. Chem. Phys.* **1978**, 68, 2959–2970.

26 Beerdsen, E., Dubbeldam, D., Smit, B. *Phys. Rev. Lett.* **2004**, 93, 0248301.

27 Dubbeldam, D., Beerdsen, E., Vlugt, T.J.H., Smit, B. *J. Chem. Phys.* **2004**, 122, 224712.

28 Sevick, E.M., Bell, A.T., Theodorou, D.N. *J. Chem. Phys.* **1992**, 98, 3196–3212.

29 Ruiz-Montero, M.J., Frenkel, D., Brey, J.J. *Mol. Phys.* **1997**, 90, 925–941.

30 Bolhuis, P.G., Dellago, C., Chandler, D. *Faraday Discuss.* **1998**, 110, 421–436.

31 Dellago, C., Bolhuis, P.G., Csajka, F.S., Chandler, D. *J. Chem. Phys.* **1998**, 108, 1964–1977.

32 Dellago, C., Bolhuis, P.G., Chandler, D. *J. Chem. Phys.* **1999**, 110, 6617–6625.

33 Allen, R.J., Warren, P.B., ten Wolde, P.R. *Phys. Rev. Lett.* **2005**, 94, 018104.

34 van Erp, T.S. *Phys. Rev. Lett.* **2007**, 98, 268301.

35 Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.N., Teller, E. *J. Chem. Phys.* **1953**, 21, 1087–1092.

36 Manousiouthakis, V.I., Deem, M.W. *J. Chem. Phys.* **1999**, 110, 2753–2756.

37 Kofke, D.A., Glandt, E.D. *Mol. Phys.* **1988**, 64, 1105–1131.

38 Panagiotopoulos, A.Z. *Int. J. Thermophys.* **1989**, 10, 447.

39 Martin, M.G., Siepmann, J.I. *J. Am. Chem. Soc.* **1997**, 119, 8921–8924.

40 Vlugt, T.J.H., Krishna, R., Smit, B. *J. Phys. Chem. B* **1999**, 103, 1102–1118.

41 McQuarrie, D.A., Simon, J.D., *Physical Chemistry: A Molecular Approach*, 1st edn., University Science Books; Sausalito, **1997**.

42 Siepmann, J.I., Frenkel, D. *Mol. Phys.* **1992**, 75, 59–70.

43 Frenkel, D., Mooij, G.C.A.M., Smit, B. *J. Phys.: Condens. Matter* **1992**, 4, 3053–3076.

44 De Pablo, J.J., Laso, M., Suter, U.W. *J. Chem. Phys.* **1992**, 96, 6157–6162.

45 Siepmann, J.I., in *Computer Simulation of Biomolecular Systems: Theoretical and Experimental Applications*, edited by van Gunsteren, W.F., Weiner, P.K., Wilkinson, A.J. Escom Science Publisher, Leiden, **1993**, pp. 249–264.

46 Rosenbluth, M.N., Rosenbluth, A.W. *J. Chem. Phys.* **1955**, 23, 356–359.

47 Press, W.H., Flannery, B.P., Teukolsky, S.A., Vetterling, W.T. *Numerical Recipes: The Art of Scientific Computing*, Cambridge University Press; Cambridge, **1986**.

48 Martin, M.G., Siepmann, J.I. *J. Phys. Chem. B* **1999**, 103, 4508–4517.

49 Macedonia, M.D., Maginn, E.J. *Mol. Phys.* **1999**, 96, 1375–1390.

50 Martin, M.G., Frischknecht, A.L. *Mol. Phys.* **2006**, 104, 2439–2456.

51 Esselink, K., Loyens, L.D.J.C., Smit, B. *Phys. Rev. E* **1995**, 51, 1560–1568.

52 Vlugt, T.J.H., Martin, M.G., Smit, B., Siepmann, J.I., Krishna, R. *Mol. Phys.* **1998**, 94, 727–733.

53 Vlugt, T.J.H. *Mol. Sim.* **1999**, 23, 63–78.

54 Consta, S., Wilding, N.B., Frenkel, D., Alexandrowicz, Z. *J. Chem. Phys.* **1999**, 110, 3220–3228.

55 Consta, S., Vlugt, T.J.H., Wichers Hoeth, J., Smit, B., Frenkel, D. *Mol. Phys.* **1999**, 97, 1243–1254.

56 Houdayer, J. *J. Chem. Phys.* **2002**, 116, 1783–1787.

57 Combe, N., Vlugt, T.J.H., ten Wolde, P.R., Frenkel, D. *Mol. Phys.* **2003**, 101, 1675–1682.

58 Dubbeldam, D., Calero, S., Vlugt, T.J.H., Krishna, R., Maesen, T.L.M., Smit, B. *J. Phys. Chem. B* **2004**, 108, 12301–12313.

59 Widom, B. *J. Chem. Phys.* **1963**, 39, 2802–2812.

60 Smit, B., Siepmann, J.I. *J. Phys. Chem.* **1994**, 98, 8442–8452.

61 Wood, G.B., Panagiotopoulos, A.Z., Rowlinson, J.S. *Mol. Phys.* **1988**, 63, 49–63.

62 Dunne, J. A., Mariwala, R., Rao, M., Sircar, S., Gorte, R. J., Myers, A. L. *Langmuir* **1996**, 12, 5888–5895.

63 Vlugt, T.J.H., García-Pérez, E., Dubbeldam, D., Ban, S., Calero, S. *J. Chem. Theory Comp.* **2008**, 4, 1107–1118.

64 Karavias, F., Myers, A.L. *Langmuir* **1991**, 7, 3118–3126.