Chapter 13

Biased Monte Carlo Schemes

Up to this point, we have not addressed a fairly obvious question: what is the point of using the Monte Carlo technique in simulations? After all, Molecular Dynamics simulations can be used to study the static properties of many-body systems and, in addition, MD provides information about their dynamical behavior. Moreover, a standard MD simulation is computationally no more expensive than the corresponding MC simulation. Hence, it would seem tempting to conclude that the MC method is an elegant but outdated scheme.

As the reader may have guessed, we believe that there are good reasons to use MC rather than MD in certain cases. But we stress the phrase *in certain cases*. All other things being equal, MD is clearly the method of choice. Hence, if we use the Monte Carlo technique, we should always be prepared to justify our choice. Of course, the reasons may differ from case to case. Sometimes it is simply a matter of ease of programming: in MC simulations there is no need to compute forces. This is irrelevant if we work with pair potentials, but for many-body potentials, the evaluation of the forces may be nontrivial. Another possible reason is that we are dealing with a system that has no natural dynamics. For instance, this is the case in models with discrete degrees of freedom (e.g., Ising spins). And, indeed, for simulations of lattice models, MC is almost always the technique of choice. But even in off-lattice models with continuous degrees of freedom, it is sometimes better, or even essential, to use Monte Carlo sampling. Usually, the reason to choose the MC technique is that it allows us to perform *unphysical* trial moves, that is, moves that cannot occur in nature (and, therefore, have no counterpart in Molecular Dynamics) but are essential for the equilibration of the system.

This introduction is meant to place our discussion of Monte Carlo techniques for simulating complex fluids in a proper perspective: in most published simulations of complex (often macromolecular) fluids, Molecular Dynamics is used, and rightly so. The Monte Carlo techniques that we discuss here have been developed for situations where either MD cannot be used at all or the natural dynamics of the system are too slow to allow the system to equilibrate on the time scale of a simulation.

Examples of such simulations are Gibbs ensemble and grand-canonical Monte Carlo simulations. Both techniques require the exchange of particles, either between a reservoir and the simulation box or between the two boxes. Such particle exchanges are not related to any real dynamics and therefore require the use of Monte Carlo techniques. But, in the case of complex fluids, in particular fluids consisting of chain molecules, the conventional Monte Carlo techniques for grand-canonical or Gibbs ensemble simulations fail. The reason is that, in the case of large molecules, the probability of acceptance of a random trial insertion in the simulation box is extremely small and hence the number of insertion attempts has to be made prohibitively large. For this reason, the conventional grand-canonical and Gibbs ensemble simulations were limited to the study of adsorption and liquid-vapor phase equilibria of small molecules.

13.1 Biased Sampling Techniques

In this chapter,[1] we discuss extensions of the standard Monte Carlo algorithm that allow us to overcome some of these limitations. The main feature of these more sophisticated Monte Carlo trial moves is that they are no longer completely random: the moves are biased in such a way that the molecule to be inserted has an enhanced probability to "fit" into the existing configuration. In contrast, no information about the present configuration of the system is used in the generation of normal (unbiased) MC trial moves: that information is used only to accept or reject the move (see Chapters 3 and 5). Biasing a Monte Carlo trial move means that we are no longer working with a symmetric *a priori* transition matrix. To satisfy detailed balance, we therefore also should change the acceptance rules. This point is discussed in some detail. Clearly, the price we pay for using configurationally biased MC trial moves is a greater complexity of our program. However, the reward is that, with the help of these techniques, we can sometimes speed up a calculation by many orders of magnitude. To illustrate this, we shall discuss examples of simulations that were made possible only through the use of bias sampling.

[1] Readers who are not familiar with the Rosenbluth scheme are advised to read section 11.2 first.

13.1.1 Beyond Metropolis

The general idea of biased sampling is best explained by considering a simple example. Let us assume that we have developed a Monte Carlo scheme that allows us to generate trial configurations with a probability that depends on the potential energy of that configuration:

$$\alpha(o \to n) = f[\mathcal{U}(n)].$$

For the reverse move, we have

$$\alpha(n \to o) = f[\mathcal{U}(o)].$$

Suppose we want to sample the N,V,T ensemble, which implies that we have to generate configurations with a Boltzmann distribution (5.2.2). Imposing detailed balance (see section 5.1) yields, as a condition for the acceptance rule,

$$\frac{\text{acc}(o \to n)}{\text{acc}(n \to o)} = \frac{f[\mathcal{U}(o)]}{f[\mathcal{U}(n)]} \exp\{-\beta[\mathcal{U}(n) - \mathcal{U}(o)]\}.$$

A possible acceptance rule that obeys this condition is

$$\text{acc}(o \to n) = \min\left(1, \frac{f[\mathcal{U}(o)]}{f[\mathcal{U}(n)]} \exp\{-\beta[\mathcal{U}(n) - \mathcal{U}(o)]\}\right). \qquad (13.1.1)$$

This derivation shows that we can introduce an arbitrary biasing function $f(\mathcal{U})$ in the sampling scheme and generate a Boltzmann distribution of configurations, provided that the acceptance rule is modified in such a way that the bias is removed from the sampling scheme. Ideally, by biasing the probability to generate a trial conformation in the right way, we could make the term on the right-hand side of equation (13.1.1) always equal to unity. In that case, every trial move will be accepted. In Chapter 14.3, we have seen that it is sometimes possible to achieve this ideal situation. However, in general, biased generation of trial moves is simply a technique for enhancing the acceptance of such moves without violating detailed balance.

We now give some examples of the use of non-Metropolis sampling techniques to demonstrate how they can be used to enhance the efficiency of a simulation.

13.1.2 Orientational Bias

To perform a Monte Carlo simulation of molecules with an intermolecular potential that depends strongly on the relative molecular orientation (e.g., polar molecules, hydrogen-bond formers, liquid-crystal forming molecules), it is important to find a position that not only does not overlap with the other molecule but also has an acceptable orientation. If the probability of finding a suitable orientation by chance is very low, we can use biased trial moves to enhance the acceptance.

Algorithm

Let us consider a Monte Carlo trial move in which a randomly selected particle has to be moved and reoriented. We denote the old configuration by o and the trial configuration by n. We use standard random displacement for the translational parts of the move, but we bias the generation of trial orientations, as follows:

1. Move the center of mass of the molecule over a (small) random distance and determine all those interactions that do not depend on the orientations. These interactions are denoted by $u^{pos}(n)$. In practice, there may be several ways to separate the potential into orientation-dependent and orientation-independent parts.
2. Generate k trial orientations $\{\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_k\}$ and for each of these trial orientations, calculate the energy $u^{or}(\mathbf{b}_i)$.
3. We define the Rosenbluth[2] factor

$$W(n) = \sum_{j=1}^{k} \exp[-\beta u^{or}(\mathbf{b}_j)]. \qquad (13.1.2)$$

   Out of these k orientations, we select one, say, n, with a probability

$$p(\mathbf{b}_n) = \frac{\exp[-\beta u^{or}(\mathbf{b}_n)]}{\sum_{j=1}^{k} \exp[-\beta u^{or}(\mathbf{b}_j)]}. \qquad (13.1.3)$$

4. For the old configuration, o, the part of the energy that does not depend on the orientation of the molecules is denoted by $u^{pos}(o)$. The orientation of the molecule in the old position is denoted by $\mathbf{b}_o$, and we generate $k-1$ trial orientations denoted by $\mathbf{b}_2, \cdots, \mathbf{b}_k$. Using these k orientations, we determine

$$W(o) = \exp[-\beta u^{or}(\mathbf{b}_o)] + \sum_{j=2}^{k} \exp[-\beta u^{or}(\mathbf{b}_j)]. \qquad (13.1.4)$$

5. The move is accepted with a probability

$$\text{acc}(o \to n) = \min\left(1, \frac{W(n)}{W(o)} \exp\{-\beta[u^{pos}(n) - u^{pos}(o)]\}\right). \qquad (13.1.5)$$

It is clear that equation (13.1.3) ensures that energetically favorable configurations are more likely to be generated. An example implementation of this scheme is shown in Algorithm 22. Next, we should demonstrate that the sampling scheme is correct.

[2] Since this algorithm for biasing the orientation of the molecules is very similar to an algorithm developed by Rosenbluth and Rosenbluth in 1955 [295] for sampling configurations of polymers (see section 11.2), we refer to the factor W as the Rosenbluth factor.

Algorithm 22 (Orientational Bias)

```
  PROGRAM orien_bias            move a particle to a random
                                position using an orient. bias
  o=int(ranf()*npart)+1         select a particle at random
  xt=ranf()*box                 start: generate new configuration
  call ener(xt,en)              calculate u^pos
  wn=exp(-beta*en)
  sumw=0
  do j=1,k                      generate k trial orientations
    call ranor(b(j))            random vector on a sphere
    call enero(xt,b(j),eno)     calculate trial orientation j u^or(j)
    w(j)= exp(-beta*eno)        calculate Rosenbluth factor (13.1.2)
    sumw=sumw+w(j)
  enddo
  call select(w,sum,n)          select one of the orientations
  bn=b(n)                       n is the selected conformation
  wn=wn*sumw                    Rosenbluth factor new configuration
                                consider the old conformation
  call ener(x(o),en)            calculate u^pos
  wo=exp(-beta*en)
  sumw=0
  do j=1,k                      consider k trial orientations
   if (j.eq.1) then
     b(j)=u(o)                  use actual orientation of particle o
   else
     call ranor(b(j))           generate a random orientation
   endif
   call enero(x(o),b(j),eno)    calculate energy of trial orientation j
   sumw=sumw+exp(-beta*eno)     calculate Rosenbluth factor (13.1.4)
  enddo
  wo=wo*sumw                    Rosenbluth factor old configuration
  if (ranf().lt.wn/wo)          acceptance test (13.1.5)
+    call accept                accepted: do bookkeeping
  end
```

Comments to this algorithm:

1. *The subroutine* `ener` *calculates the energy associated with the position, the subroutine* `enero` *the energy associated with the orientations.*
2. *The subroutine* `ranor` *generates a random vector on a unit sphere (Algorithm 42), subroutine* `accept` *does the bookkeeping associated with the acceptance of a new configuration, and the subroutine* `select` *selects one of the orientations with probability* $p(i) = w(i)/\sum_j w(j)$ *(see, Algorithm 41).*

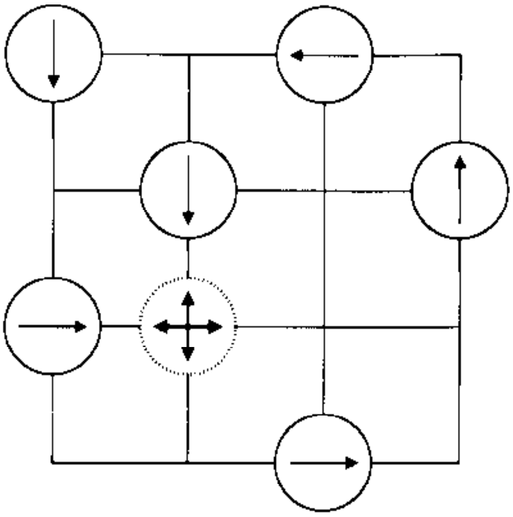

Figure 13.1: Lattice model in which the molecules can take four orientations (indicated by arrows, $k = 4$). The dotted circle indicates the trial position of the particle that we attempt to move.

Justification of Algorithm

To show that the orientational-bias Monte Carlo scheme just described is correct, that is, generates configurations according to the desired distribution, it is convenient to consider lattice models and continuum models separately. For both cases we assume that we work in the canonical ensemble, for which the distribution of configurations is given by equation (5.2.2)

$$\mathcal{N}(\mathbf{q}^N) \propto \exp[-\beta\mathcal{U}(\mathbf{q}^N)],$$

where $\mathcal{U}(\mathbf{q}^N)$ is the sum of orientational and nonorientational part of the energy:

$$\mathcal{U} = u^{or} + u^{pos}.$$

We first consider a lattice model.

Lattice Models

We assume that the molecules in our lattice model can have k discrete orientations (see Figure 13.1). We impose the condition of detailed balance (5.1.1):

$$K(o \to n) = K(n \to o).$$

The flow of configurations o to n is (equation (5.1.2))

$$K(o \to n) = \mathcal{N}(o) \times \alpha(o \to n) \times acc(o \to n). \qquad (13.1.6)$$

In the orientational-bias scheme, the probability of selecting conformation n is (see equation (13.1.3))

$$\alpha(o \to n) = \frac{\exp[-\beta u^{or}(n)]}{W(n)}.$$

Imposing detailed balance and substitution of the desired distribution for $\mathcal{N}(n)$ and $\mathcal{N}(o)$ imposes the following condition on the acceptance rules:

$$\begin{aligned}\frac{\mathrm{acc}(o \to n)}{\mathrm{acc}(n \to o)} &= \frac{\exp[-\beta \mathcal{U}(n)]}{\exp[-\beta \mathcal{U}(o)]} \times \frac{\exp[-\beta u^{or}(o)]}{W(o)} \times \frac{W(n)}{\exp[-\beta u^{or}(n)]} \\ &= \frac{W(n)}{W(o)} \exp\{-\beta[u^{pos}(n) - u^{pos}(o)]\}. \qquad (13.1.7)\end{aligned}$$

Acceptance rule (13.1.5) satisfies this condition. This demonstrates that for a lattice model detailed balance is fulfilled.

Continuum Model

If the orientation of a molecule is described by a continuous variable, then there is an essential difference with the previous case. In the lattice model all the possible orientations can be considered explicitly, and the corresponding Rosenbluth factor can be calculated exactly. For the continuum case, we can never hope to sample *all* possible orientations. It is impossible to determine the exact Rosenbluth factor since an infinite number of orientations are possible.[3] Hence, the scheme for lattice models, in which the Rosenbluth factor for all orientations is calculated, cannot be used for a continuum model. A possible solution would be to use a large but finite number of trial directions. Surprisingly, this is not necessary. It is possible to devise a *rigorous* algorithm using an *arbitrary subset* of all possible trial directions. The answer we get does *not* depend on the number of trial directions we choose but the statistical accuracy does.

Let us consider the case in which we use a set of k trial orientations; this set is denoted by

$$\{\mathbf{b}\}_k = \{\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_k\}.$$

Conformation $\mathbf{b}_n$ can be selected only if it belongs to the set $\{\mathbf{b}\}_k$. The set of all sets $\{\mathbf{b}\}_k$ that includes conformation n is denoted by

$$\mathcal{B}_n = \{\{\mathbf{b}\}_k | \mathbf{b}_n \in \{\mathbf{b}\}_k\}.$$

Every element of $\mathcal{B}_n$ can be written as $(\mathbf{b}_n, b^*)$, where b^* is the set of $k-1$ additional trial orientations. In the flow of configuration o to n, we have to

[3] In Example 17 we discuss a special case for which the Rosenbluth factor *can* be calculated exactly.

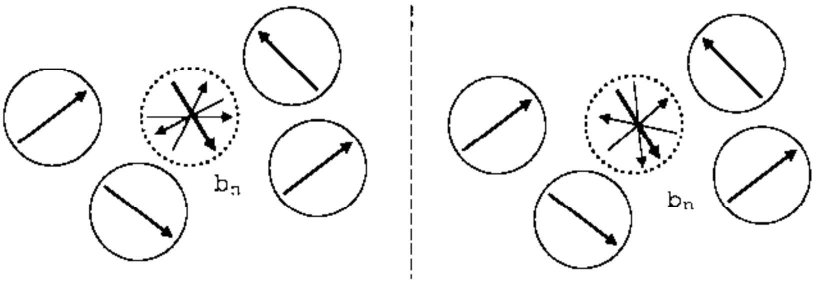

Figure 13.2: Continuum model in which the molecule can have an arbitrary orientation (indicated by arrows). The figure shows two different sets of four trial orientations that both include orientation $\mathbf{b}_n$.

consider the sum over all sets in $\mathcal{B}_n$

$$K(o \to n) = \mathcal{N}(o) \sum_{i \in \mathcal{B}_n} \alpha(o \to n, i) \times \text{acc}(o \to n, i), \tag{13.1.8}$$

in which the probability of generating configuration n and the acceptance depend on the particular set of trial orientations i.

Similarly, for the reverse move, we define the set $\mathcal{B}_o$

$$\mathcal{B}_0 = \{\{\mathbf{b}\}_k | \mathbf{b}_o \in \{\mathbf{b}\}_k\},$$

for which each element can be written as $(\mathbf{b}_o, b'^*)$. The expression for the reverse flow then becomes

$$K(n \to o) = \mathcal{N}(n) \sum_{j \in \mathcal{B}_o} \alpha(n \to o, j) \times \text{acc}(n \to o, j). \tag{13.1.9}$$

It should be stressed that infinitely many different sets of orientations include $\mathbf{b}_n$, and the same holds for sets that include $\mathbf{b}_o$. Moreover, the probability of selecting $\mathbf{b}_n$ from such a set depends on the remainder of the set b^* (see Figure 13.2). Hence, the acceptance probability must also depend on the sets b^* and b'^*.

Detailed balance is certainly obeyed if we impose a much stronger condition, "super-detailed balance," which states that for every particular choice of the sets b^* and b'^*, detailed balance should be obeyed,

$$\begin{aligned} K(o \to n, b^*, b'^*) &= K(n \to o, b'^*, b^*), \\ \mathcal{N}(o)\, \alpha(o \to n, b^*, b'^*)\, \text{acc}(o \to n, b^*, b'^*) & \\ &= \mathcal{N}(n)\, \alpha(n \to o, b'^*, b^*)\, \text{acc}(n \to o, b'^*, b^*), \end{aligned} \tag{13.1.10}$$

in which b^* and b'^* are two sets of $k-1$ arbitrary additional trial orientations. It may seem strange that the sets b^* and b'^* show up on *both* sides of the equations. However, bear in mind that, to decide on the acceptance of the forward move, one should generate both the set b^* that includes the new orientation *and* the set b'^* around the old orientation. Hence, the construction of a trial move includes both sets of trial orientations. As the probabilities of generating b^* and b'^* appear on both sides of the equations, they cancel each other. Moreover, the *a priori* probability of generating a random orientation $\mathbf{b}_n$ in the forward move is equal to the *a priori* probability of generating $\mathbf{b}_o$ in the reverse move. So these generation probabilities also cancel each other. This leads to a great simplification of the acceptance criterion. For the canonical ensemble, substitution of equations (13.1.2) and (13.1.3) yields

$$\begin{aligned}\frac{\mathrm{acc}(o \to n, b^*, b'^*)}{\mathrm{acc}(n \to o, b'^*, b^*)} &= \frac{\exp[-\beta\mathcal{U}(n)]}{\exp[-\beta\mathcal{U}(o)]}\,\frac{\exp[-\beta u^{\mathrm{or}}(o)]}{W(\mathbf{b}_o, b'^*)}\,\frac{W(\mathbf{b}_n, b^*)}{\exp[-\beta u^{\mathrm{or}}(n)]}\\ &= \frac{W(\mathbf{b}_n, b^*)}{W(\mathbf{b}_o, b'^*)}\exp\{-\beta[u^{\mathrm{pos}}(n) - u^{\mathrm{pos}}(o)]\}.\end{aligned} \tag{13.1.11}$$

As acceptance rule (13.1.5) satisfies this condition, detailed balance is indeed obeyed.

Note that, in this demonstration, we did not have to assume that the number of trial orientations k had to be large. In fact, the result is *independent* of the number of trial orientations.

Example 16 (Orientational Bias of Water)

Cracknell *et al.* [353] used an orientational-bias scheme to simulate liquid water. At ambient temperature, water has a relatively open structure, in which the water molecules form a network due to the hydrogen bonds. To insert a water molecule successfully, one has not only to place the molecule in an empty spot but also find a good orientation. The method used by Cracknell *et al.* to find this optimum orientation is similar to the one introduced in this section, in the sense that a bias in the orientation is introduced and is subsequently removed by adjusting the acceptance rules. Yet, the philosophy behind the approach of Cracknell *et al.* is fundamentally different.

In the scheme of Cracknell *et al.*, a random position of a water molecule r is generated and one trial orientation ω is drawn from a distribution $f(r, \omega)$. The problem is that the optimum distribution $f(r, \omega)$ is not known *a priori* and depends on the conformations of the other water molecules. However, as we have shown, any distribution can be used (as long as detailed balance and microscopic reversibility are obeyed). Since the construction of the true orientational distribution requires too much computer time, Cracknell *et al.* constructed a distribution that was meant to mimic the true distribution. To this end, one axis of the water molecule was given a random orientation and, for the other axis a biasing scheme was used. For this axis, n equidistant angles ψ_i were generated

$$\psi_i = 2\pi p/n, \quad p \in \{1, \cdots, n\}.$$