10. MOLECULAR SIMULATIONS

10.1 Introduction

Statistical mechanics provides a general, rigorous formalism for the calculation of macroscopic properties given the Hamiltonian of a material system. As we have already pointed out in Chapter 3, however, an analytical solution to this formalism is impossible for most material systems of practical interest. The analytical treatments we discussed in chapters 5 and 9 (*e.g.*, Bragg- Williams approximation for the lattice model of a binary mixture, integral equation theories and perturbation theories for liquids) are designed to provide closed-form solutions to the statistical mechanical formulation, given a model for the molecular geometry and energetics. They can only do so, however, at the expense of introducing simplifying approximations in the mathematical treatment of the problems at hand.

An alternative approach is to undertake a *numerical solution of the full statistical mechanics* given a model of the molecular geometry and energetics; this is the task of *molecular simulations.* Simulations can provide *exact* results (subject only to testable numerical error). When applied without any simplification in the statistical mechanical formalism, they constitute valuable tools for testing the approximations that underlie analytical theories, as well as the models invoked to represent real-life materials (see Figure 3.1). In practice, simulations may be combined with judicious approximations to attack complex problems that cannot yield to entirely analytical treatments. Hierarchical approaches, resting on a range of theoretical concepts and simulation techniques, are valuable for elucidating the molecular mechanisms underlying macroscopic behavior and for establishing pathways to the prediction of properties from molecular-level information.

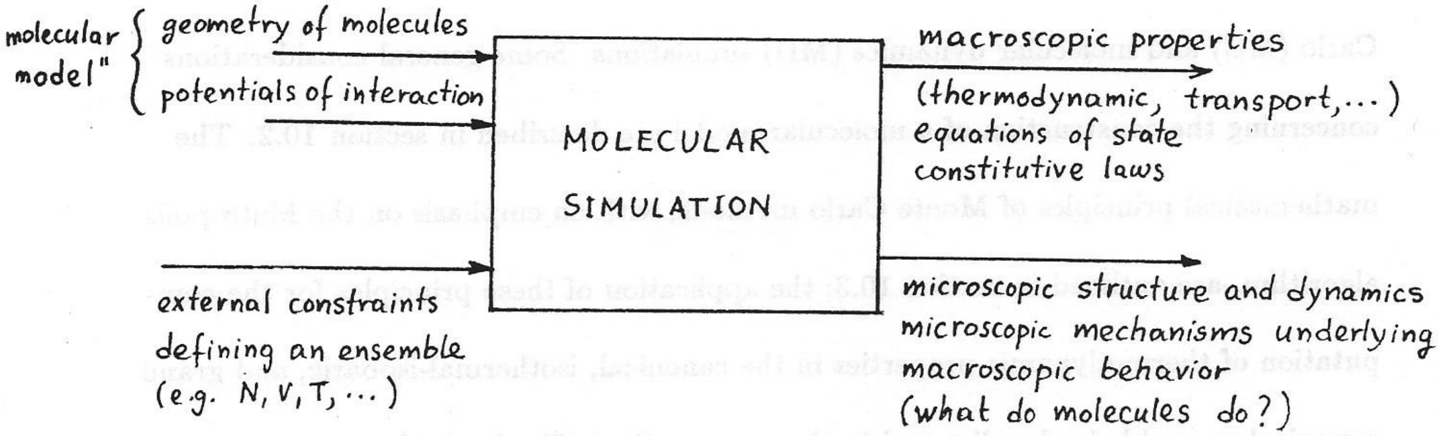

Figure 10.1 Inputs and outputs of a typical molecular simulation.

A simulation is a "computer experiment." The inputs and outputs typically involved in such an experiment are shown schematically in Figure 10.1. In applying simulations to real-life material systems, one is faced with two important problems. First, one's knowledge of the interaction potentials among the sites that constitute a material system is usually limited, especially in systems where strong and specific ("chemical") interactions may be present. Quantum mechanics-based calculations of the potential energy hypersurface as a function of nuclear coordinates can address this problem. Second, molecular simulations are extremely compute-intensive. Even when state-of-the-art supercomputers are used, for example, conventional atomistic molecular dynamics simulations can only track the evolution of model systems of length scale $\mathcal{O}(100\text{Å})$ over time scales of $\mathcal{O}(1 \text{ ns})$. The development of coherent hierarchies of modelling and simulation techniques that can successfully bridge the wide range of time- and length scales characterizing real-life materials is a very serious challenge that has to be met successfully for computer-aided materials design to become a reality.

In this chapter we will briefly address the principles and applications of Monte Carlo (MC) and molecular dynamics (MD) simulations. Some general considerations concerning the construction of a molecular model are described in section 10.2. The mathematical principles of Monte Carlo methods, with an emphasis on the Metropolis algorithm, are outlined in section 10.3; the application of these principles for the computation of thermodynamic properties in the canonical, isothermal-isobaric, and grand canonical ensemble is also dicussed in the same section. The formulation and common algorithms used for molecular dynamics in the microcanonical ensemble are introduced in section 10.5, followed by a discussion of molecular dynamics in generalized coordinates and constraint dynamics methods for molecular systems, as well as of extended ensemble methods for simulating systems in ensembles different from the microcanonical. Section 10.4 reviews ways of analyzing the results of molecular simulations to obtain structural information, thermodynamic, and transport properties. The discussion is intended to give the reader a flavor of how to perform molecular simulations, but is necessarily brief. For more details, the reader is referred to the book by Allen and Tildesley (1987).

10.2 Construction of a molecular model

10.2.1 Molecular Representation

The level of detail built into a molecular model varies, depending on the objectives of the investigator. For example, in modelling polymer solutions and melts, simple lattice models have been used to test scaling relations for chain conformation and to explore how multicomponent phase diagrams and structure at interfaces depend on the architecture and interactions of constituent chains. More detailed continuum models consisting of tangent hard spheres have been used to test theories for the pair distribution function and the equation of state. Detailed atomistic models taking into account the detailed geometry and interactions of functional groups are necessary for the quantitative prediction of the values of properties such as density, cohesive energy, surface and interfacial tension, permeability by gases, and elastic constants in the glassy state.

In the latter, more realistic simulations, a material is represented as a set of interaction sites (atoms, groups, or centers of charge). The total potential energy is written as a function $\mathcal{V}(\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_N)$ of the positions $\mathbf{r}_i$ of interaction sites. It is convenient to write the potential energy function as a sum of pairwise, three-body, ..., interactions among sites:

$$\mathcal{V}(\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_N) = \sum_{i <} \sum_{j} \mathcal{V}_{pair}(\mathbf{r}_i, \mathbf{r}_j) + \sum_{i <} \sum_{j <} \sum_{k} \mathcal{V}_{\substack{\text{three}\\\text{body}}}(\mathbf{r}_i, \mathbf{r}_j, \mathbf{r}_k) + \cdots \qquad (10.1)$$

In systems of nonpolar molecules interacting via dispersive forces, such a decomposition of the potential energy function is supported by quantum calculations of the attractive forces arising due to fluctuations of the electronic clouds. Such calculations lead to the pairwise London potential and to the three-body triple-dipole Axilrod-Teller potential [see Maitland *et al.* (1981)]. Pairwise interactions constitute the dominant contribution to $\mathcal{V}$. Nevertheless, three- and higher-body interactions are far from negligible; they are quite significant at liquid densities, and even more so in the solid state. As an indication, 10% of the lattice energy of solid argon is due to higher-than-pair interactions.

In practice, computing three- and higher-body interactions involves triple- and higher loops over all interaction sites, and is therefore quite time consuming. It is thus preferable to approximate $\mathcal{V}$ as a sum of *effective* pair potentials:

$$\mathcal{V}(\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_N) = \sum_{i<}\sum_{j} \mathcal{V}_{pair}^{eff}(\mathbf{r}_i, \mathbf{r}_j) \tag{10.2}$$

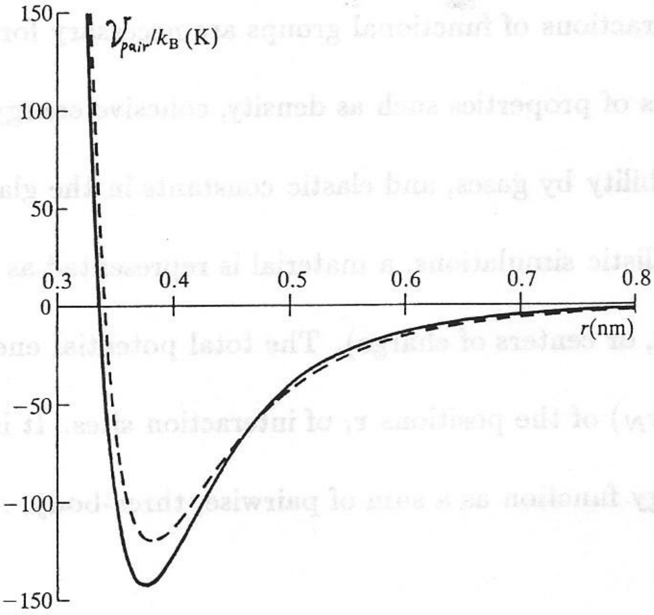

Figure 10.2 Comparison between the exact BBMS pair potential $\mathcal{V}_{pair}$ (solid line) and the Lennard-Jones effective pair potential $\mathcal{V}_{pair}^{eff}$ used in computer simulations of liquid argon (dashed line) [after Allen and Tildesley (1987)].

Contributions from three- and higher-body interactions are absorbed into the effective pair potential. As a consequence, $\mathcal{V}_{pair}^{eff}$ is rather different from the rigorous pair potential $\mathcal{V}_{pair}$ between molecules. This is shown characteristically in Figure 10.2, which compares the very accurate Barker-Bobetic-Maitland-Smith (BBMS) $\mathcal{V}_{pair}$ potential function for argon with an effective $\mathcal{V}_{pair}^{eff}$ frequently used in simulations of liquid argon. The BBMS function is based on molecular beam scattering, spectroscopic measurements on argon dimers, inversion of the temperature dependence of the second virial coefficient [cf. Eq. (9.44)], solid-state properties, and some theoretical calculations. $\mathcal{V}_{pair}^{eff}$, on the other hand, is a 12-6 Lennard-Jones (LJ) potential that works well in reproducing structure and thermodynamic properties in the liquid state. A simulation based on the exact $\mathcal{V}_{pair}$ alone would give poorer results, as it would disregard the effect of higher than two-body interactions.

Molecules with no pronounced spatial separation of electric charge can be represented as sets of Lennard-Jones interaction sites. For example, a butane molecule can be represented as a set of four carbon and ten hydrogen interaction sites [*explicit* representation, Fig. 10.3(a). A less detailed, but more economical representation that is satisfactory in fluid-phase (but not solid-phase) simulations is the *united atom* representation, wherein each CH_3 and each CH_2 group is represented as a single interaction site [Fig. 10.3 (b)].

In addition to nonbonded site-site interactions, the model must represent the bonded geometry of the molecule realistically. The potential energy contribution associated with the distortion of each bond length is often assumed to be of the harmonic form

$$\mathcal{V}_b(\ell) = \frac{1}{2}\, k_b\, (\ell - \ell^0)^2 \qquad (10.3)$$

where ℓ the bond length, (see Fig. 10.3), ℓ^0 its equilibrium value, and k_b the bond force constant. More elaborate forms may be used (*e.g.*, Morse potential). In practice, unless the computation of vibrational spectra is desired, it is satisfactory to consider bond lengths as fixed at their equilibrium value, *i.e.*, assume $k_b \rightarrow \infty$.

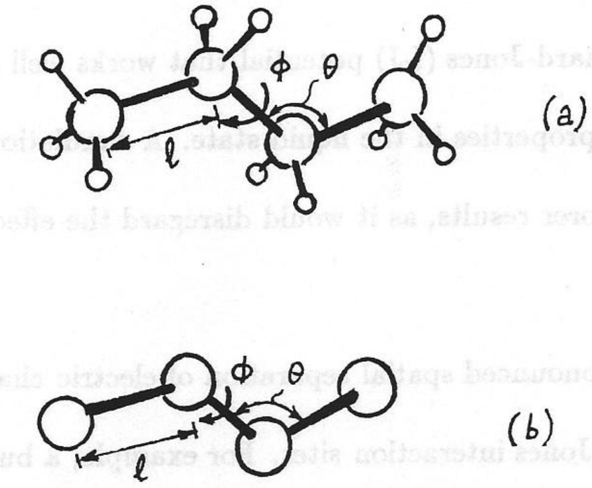

Figure 10.3 Molecular models for *n*-butane: (*a*) explicit representation; (*b*) united atom representation. Interaction sites are sketched as spheres of radius roughly one third their van der Waals radius; the van der Waals spheres of interaction sites interpenetrate [compare Fig. 9.18 (*a*)].

The potential energy associated with bond angle distortions is attributed a form

$$\mathcal{V}_\theta(\theta) = \frac{1}{2}\, k_\theta \,(\theta - \theta^0)^2$$

where θ the bond engle and θ^0 its equilibrium value. More elaborate forms involving the torsion angles, may be used. For molecules with no internal torsional degrees of freedom, bond angles are often assumed fixed ($k_\theta \rightarrow \infty$). Fixed bond angles should not be used for flexible molecules if an accurate representation of the dynamics of these molecules is desired.

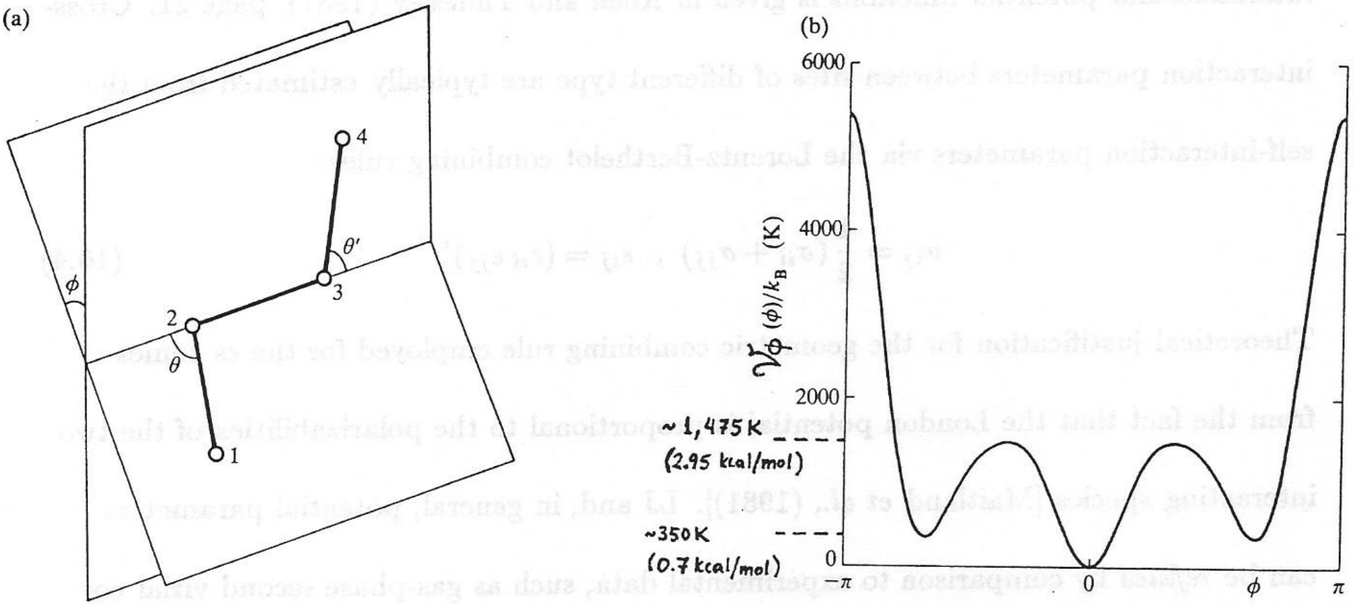

Figure 10.4 (*a*) Definition of the dihedral angle ϕ for the middle skeletal bond of n-butane. By convention, ϕ is zero in the *trans* conformation, and assumes positive values as the bond is turned away from the *trans* conformation in a direction that would lead to unscrewing of a right-handed screw [Flory (1969)]. (*b*) Torsional potential $\mathcal{V}_\phi$ as a function of the torsion angle ϕ. The *gauche* minima (around $\phi = \pm\frac{2\pi}{3}$) are higher than *trans* ($\phi = 0$) by approximately 0.7kcal/mol. The barrier between *gauche* and *trans* is around 2.95kcal/mol.

In flexible molecules, a torsional potential $\mathcal{V}_\phi(\phi)$ is used for every rotatable bond. $\mathcal{V}_\phi$ depends on the dihedral angle ϕ of the bond. As an example, we show the torsional potential used with a united atom representation of n-butane by Maréchal and Ryckaert (1983) in Figure 10.4

A list of atom-atom LJ potentials that can be used to construct crude first- guess intermolecular potential functions is given in Allen and Tildesley (1987), page 21. Cross-interaction parameters between sites of different type are typically estimated from the self-interaction parameters via the Lorentz-Berthelot combining rules:

$$\sigma_{ij} = \frac{1}{2}(\sigma_{ii} + \sigma_{jj}) \ , \ \epsilon_{ij} = (\epsilon_{ii}\,\epsilon_{jj})^{1/2} \tag{10.4}$$

Theoretical justification for the geometric combining rule employed for the ϵs comes from the fact that the London potential is proportional to the polarizabilities of the two interacting species [Maitland *et al.*, (1981)]. LJ and, in general, potential parameters can be *refined* by comparison to experimental data, such as gas-phase second virial coefficients, internal energies at given density and temperature, heats of vaporization, solubility parameters, equation of state behavior (*e.g.*, density at given temperatures and pressures), lattice energy in the crystalline form, and experimental symmetry and lattice parameters of the crystal.

For molecules with expressed separation of charges ("polar" molecules), a representation based upon LJ sites and centers of *partial charge* is appropriate. The partial charges used to represent such molecules may be based on experimentally measured multipole moments of the molecule (dipoles, quadrupoles, *etc.*; see Hirschfelder *et al.* (1954)); on Mulliken population analysis of the electronic charge distribution of an isolated molecule computed by Hartree-Fock methods, or on fitting its charge density distribution as obtained by Density Functional Theory calculations; or (best) on fitting *ab* *initio* calculated values of the potential energy of interaction between two molecules at representative distances and relative orientations of the molecules. Partial charges interact *via* Coulomb forces:

$$\mathcal{V}^{zz}(r_{ij}) = \frac{z_i\, z_j}{4\,\pi\,\epsilon_0\, r_{ij}} \tag{10.5}$$

As an example of molecular representations for a polar molecule that can establish strong and specific interactions (hydrogen bonding), we present two models of water that have been used in past simulation work (Figure 10.5). The two models are of widely different level of complexity. In the TIPS ("transferable intermolecular potentials") model of Jorgensen (1981) [Fig. 10.5 (*a*)], the H_2O molecule is represented in terms of a single LJ site centered at the oxygen nucleus, and three partial charges centered at the oxygen and the two hydrogen nuclei. With the potential truncation procedures employed by Jorgensen, this simple model reproduces the liquid-state behavior of water reasonably well. Note that the dipole moment calculated from the geometry and partial charges employed in the model, 2.25Debye, does not coincide, but is in fair agreement with the experimental dipole moment of the water molecule, 1.85Debye. The model of Matsuoka, Clementi, and Yoshimine (1976), on the other hand, [Fig. 10.5(*b*)] was derived by fitting the potential energy hypersurface of a water dimer, calculated through an *ab initio* configuration-interaction (CI) Hartree-Fock method. This model is much more complex. The oxygen and two hydrogens are all considered as sources of excluded volume repulsive interactions, which decay exponentially with increasing distance. Attractive interactions that also vary exponentially with distance are present between the hydrogens on one molecule and the oxygen on the other. In addition, there are Coulombic interactions involving all partial charges. The nuclei of the hydrogens are considered as centers of positive charge q, while the center of negative charge $-2q$ is *displaced* from the nuclear center of the oxygen, being placed on the bisector of the $H_1O_5H_2$ angle at a distance $R(O_5 - M_7)$ from it. The expression for the dimer energy at a particular configuration is

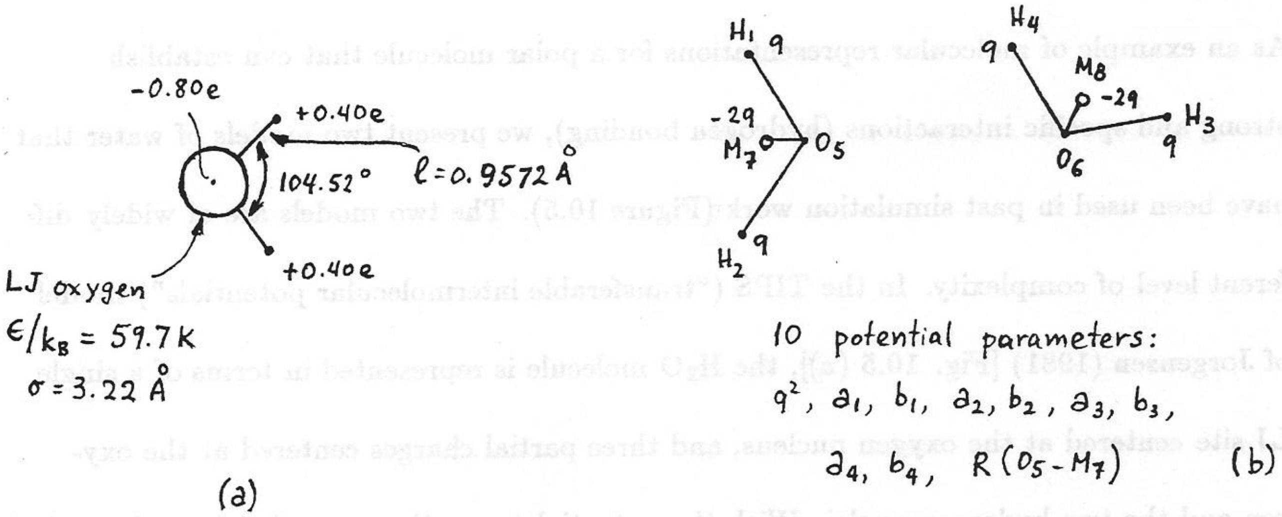

Figure 10.5 Models for water: (*a*) Simple TIPS model of Jorgensen (1981). (*b*) Matsuoka - Clementi - Yoshimine model, based on *ab initio* quantum mechanical calculations of energy of the water dimer.

$$
\begin{aligned}
\mathcal{V}_{dimer} = {} & q^2\left(\frac{1}{r_{13}} + \frac{1}{r_{14}} + \frac{1}{r_{23}} + \frac{1}{r_{24}}\right) + \frac{4\,q^2}{r_{78}} \\
& -2q^2\left(\frac{1}{r_{18}} + \frac{1}{r_{28}} + \frac{1}{r_{37}} + \frac{1}{r_{47}}\right) + a_1 \exp(-b_1\, r_{56}) \\
& + a_2\left[\exp(-b_2\, r_{13}) + \exp(-b_2\, r_{14}) + \exp(-b_2\, r_{23}) + \exp(-b_2\, r_{24})\right] \\
& + a_3\left[\exp(-b_3\, r_{16}) + \exp(-b_3\, r_{26}) + \exp(-b_3\, r_{35}) + \exp(-b_3\, r_{45})\right] \\
& - a_4\left[\exp(-b_4\, r_{16}) + \exp(-b_4\, r_{26}) + \exp(-b_4\, r_{35}) + \exp(-b_4\, r_{45})\right] \qquad (10.6)
\end{aligned}
$$

10.2.2 Summation of the total potential energy function

Any molecular simulation calls for frequent evaluations of the potential energy function $\mathcal{V}$, given the configuration of the model system. Typically, $\mathcal{V}$ is expressed as a sum of effective pair potential contributions $\mathcal{V}_{pair}^{eff}$, as in Eq. (10.2). In the following we will use the simple notation $v(r)$ to indicate a sphericaly symmetric effective pair potential. Pair potentials such as the Lennard-Jones are characterized by weak attractive "tails" extending out to infinite distance. Clearly, it is not practical to sum interactions over infinite pairs of molecules. Thus, a *finite range modification* of the pair potential has to be used in molecular simulations.

The simplest way to create a finite range modification of a pair potential $v(r)$ is to truncate the potential tail at a *cutoff distance* r_c, *i.e.* use

$$v^{modified}(r) = \begin{cases} v(r), & \text{if } r \leq r_c \\ 0, & \text{if } r > r_c \end{cases} \tag{10.7}$$

and deal with attractive interactions at distances beyond r_c in a mean field sense. A typical value of r_c used in MC simulations of Lennard-Jones liquids is 2.5σ, with σ the collision diameter of the LJ potential. A truncated LJ potential generated according to the prescription of Eq. (10.7) is shown in Figure 10.6(a). A problem with this modified potential is that it generates infinite forces at r_c, causing energy conservation to deteriorate within a MD simulation.

A better solution is provided by the *truncated and shifted* potential, shown schematically in Figure 10.6(b). Here, after cutting the potential tail at r_c, the whole pair potential curve is raised by $-v_c = -v(r_c)$:

$$v^{modified}(r) = \begin{cases} v(r) - v_c, & \text{if } r \leq r_c \\ 0, & \text{if } r > r_c \end{cases} \tag{10.8}$$

Note that this modification leaves the forces at distances $r \leq r_c$ unaffected, while the modified force at r_c changes discontinuously from $f(r_c) = -\left(\frac{dv}{dr}\right)_{r_c}$ to 0.

The discontinuous change in the force at r_c tends to create instabilities in the integration of MD equations. To correct for this, some investigators have chosen to use a *shifted force* potential, as shown in Figure 10.6 (c). In this modification, one truncates the tail of the *force curve* at r_c and shifts the whole curve upwards by $-f_c = -f(r_c)$. The pair potential is then obtained by integration of the shifted force curve:

$$f^{modified}(r) = \begin{cases} -\frac{dv}{dr} - f_c, & \text{if } r \leq r_c \\ 0, & \text{if } r > r_c \end{cases} \tag{10.9}$$

$$v^{modified}(r) = \begin{cases} v(r) - v_c + f_c(r - r_c), & \text{if } r \leq r_c \\ 0, & \text{if } r > r_c \end{cases} \tag{10.10}$$

With this modification, both the force and the potential are continuous, a discontinuity appearing only in the first derivative of the force. A problem with the shifted force potential, however, is that, by modifying forces throughout the r-domain, it changes the interaction force law substantially. The thermodynamics of the original system, governed by the unmodified potential $v(r)$, can only be recovered through a perturbation scheme.

A simpler finite range modification that leads to continuous potential, force and spatial derivative of the force, is shown in Figure 10.6(d). Here the potential tail beyond r_1 is substituted by a fifth-order polynomial, whose value, first derivative, and second derivative are equal to v, $\frac{dv}{dr}$, *and* $\frac{d^2v}{dr^2}$, respectively, at r_1, and equal to 0 at r_2

[Theodorou and Suter (1985a)]. A choice of $r_1 = 1.45\sigma$ and $r_2 = 2.30\sigma$ has given good results with LJ systems.

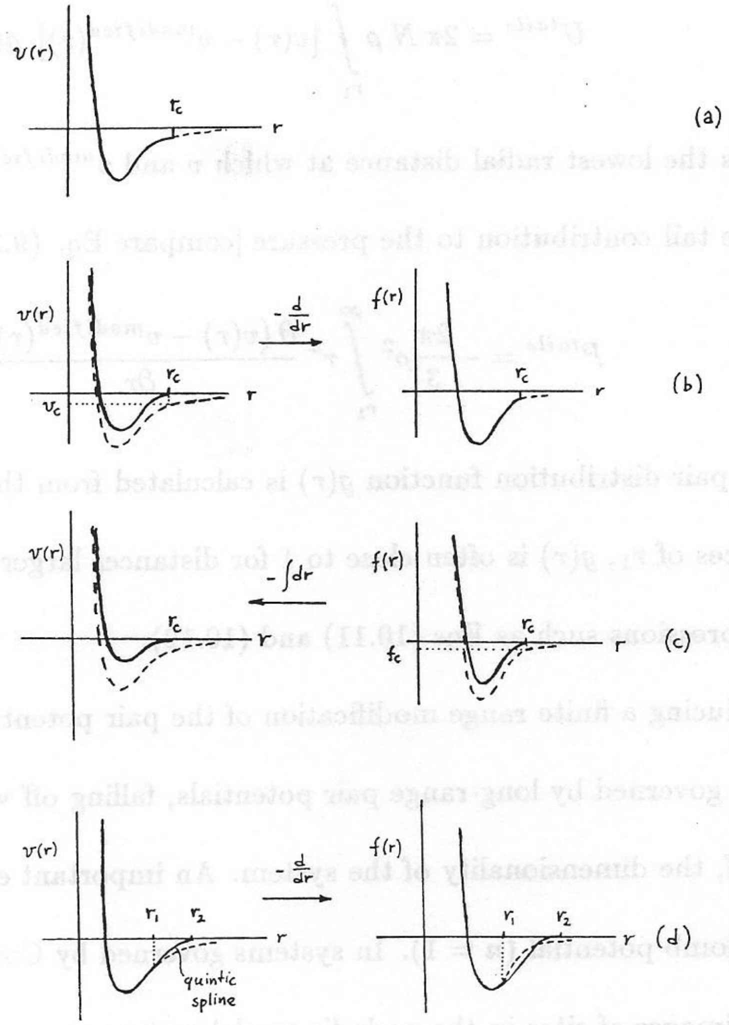

Figure 10.6 Various ways of creating a finite range modification of the Lennard-Jones potential. (a) potential truncation at r_c; (b) truncated and shifted potential; (c) truncated and shifted force; (d) substitution of the potential tail by a quintic spline.

Whenever a modified pair potential is used, the contribution from the omitted or modified "tails" to thermodynamic properties must be calculated by direct integration.

For example, the tail contribution to the internal energy [compare Eq. (9.31)] is

$$U^{tails} = 2\pi\, N\, \rho \int\limits_{r_1}^{\infty} \left[v(r) - v^{modified}(r)\right]\; g(r)\, r^2\, dr \tag{10.11}$$

where r_1 is the lowest radial distance at which v and $v^{modified}$ start deviating from each other. The tail contribution to the pressure [compare Eq. (9.34)] is

$$P^{tails} = -\frac{2\pi}{3}\rho^2 \int\limits_{r_1}^{\infty} r^3\, \frac{\partial\left(v(r) - v^{modified}(r)\right)}{\partial r}\, g(r)\, dr \tag{10.12}$$

where the pair distribution function $g(r)$ is calculated from the simulation. With the usual choices of r_1, $g(r)$ is often close to 1 for distances larger than r_c and is substituted by 1 in expressions such as Eqs (10.11) and (10.12).

Introducing a finite range modification of the pair potential is a poor approximation in systems governed by long-range pair potentials, falling off with radial distance as r^{-n} with $n \leq d$, the dimensionality of the system. An important example of such a potential is the Coulomb potential ($n = 1$). In systems governed by Coulombic forces, interactions among all images of sites in the periodic model system are summed with the technique of Ewald summation [see Allen and Tildesley (1987)].

10.2.3 Periodic boundary conditions and minimum image convention.

Common molecular simulations employ $N = 10$ to 10^4 interaction sites, although simulations of millions of sites have been attempted on state of the art vector and parallel machines. Using large model systems is taxing from the point of view of storage requirements and CPU time (the force/potential evaluation involves loops of order

$\mathcal{O}(N^2)$). Thus, it is advantageous to use in production runs the smallest model system size that can provide reliable estimates of the desired properties. For simple liquids interacting through short-range forces, simulation averages have been found to be very weakly dependent on system size for $N \geq 100$. A much larger number of interaction sites may be required in more complex systems (*e.g.*, polymers) where macroscopic behavior is affected by structure at large length scales. The seasoned simulator always confirms the system size-independence of results by undertaking runs at different system sizes.

In some applications, the behavior of a very small amount of material (a droplet, a nanocrystal) may be of interest; in such cases, a model system of finite extent is used. More often, however, one is interested in the properties of extended bulk phases of material. If a finite model box filled with interaction sites were used in such cases, the predicted behavior would be dominated by surface effects. As an indication, if 1000 molecules are arranged on the nodes of a $10 \times 10 \times 10$ cubic lattice, 488 of these molecules will appear on the cube faces. In simulating bulk behavior, therefore, the trick of *periodic boundary conditions* is used. The model box (*primary* box) is envisioned as being surrounded by images of itself in all directions. Sites in the primary box interact with sites in the image boxes. Whenever, in the course of the simulation, an atom exits the primary box, an identical atom enters through the opposite face. Thus, the primary box is like a unit cell, by periodic continuation of which one creates an infinite medium with no surfaces. A simple schematic of periodic boundary conditions in two dimensions is given in Figure 10.7. The primary box is shown in the center (bounded by bold lines,) along with the immediately surrounding $3^2 - 1 = 8$ images of itself. In three dimensions, the primary box is immediately surrounded by $3^3 - 1 = 26$ images of itself. Site 1 in the primary box interacts with all sites or site images lying within a sphere of radius r_c (the potential cutoff distance) centered at 1.

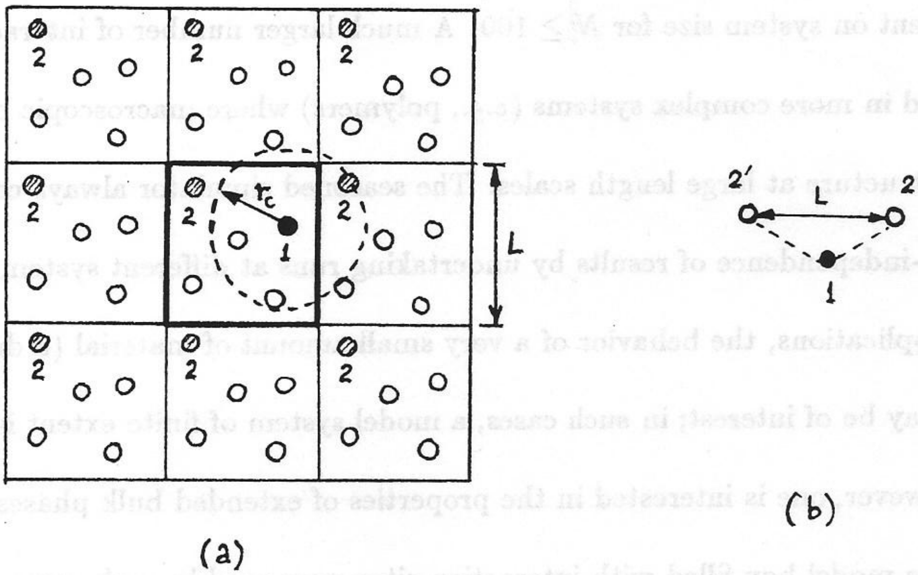

Figure 10.7 (*a*) Schematic of periodic boundary conditions in two dimensions. If the minimum image convention is satisfied, there is at most one image of 2 interacting with 1. (*b*) Simple figure for the explanation of the minimum image convention.

Strictly, this periodicity is not compatible with the absence of long-range structure in amorphous systems, such as liquids and glasses. If the box edge length, L, is sufficiently larger than the largest length scale governing the properties of interest, however, this artificial periodicity does not significantly perturb the results. For example, in a molecular simulation of a LJ liquid, periodic boundary conditions do not affect the thermodynamics significantly if $L \simeq 6\sigma$, as individual sites do not "sense" the periodic geometry. In simulating a polymer melt, on the other hand, one must choose L to be considerably larger than the chain radius of gyration $\langle s^2 \rangle^{1/2}$, if one wishes the long-range conformational characteristics of chains and the properties depending on them not to be affected by the periodic boundary conditions (compare section 9.4).

Periodic boundary conditions necessarily cut off long-wavelength fluctuations (density waves with wavelength $> L$). Given the divergence of the correlation length for structure at a critical point, the reliable simulation of near-critical systems is thus very difficult. Periodic boundary conditions also introduce spurious features in time correlation functions for times $\geq \frac{L}{c}$, where c the sound velocity in the studied medium; $\frac{L}{c}$ is the time it takes a density wave to travel through the simulation box.

Consider two atoms, 1 and 2, in the primary box. In three dimensions, there are 27 images of 2 (in the primary box and the immediately surrounding boxes) that could interact with 1.

Theorem:

$$\text{If} \quad L/2 \geq r_c \tag{10.13}$$

then *at most one* of the images of 2 can interact with 1. The condition (10.13) is known as *minimum image convention.*

Proof: Assume that condition (10.13) is satisfied, and there are two images of 2 interacting with 1. We label these images 2 and 2′ [see Figure 10.7(b)]. Since 1 interacts with 2, we have $r_{12} < r_c$. Since 1 interacts with 2′, we have $r'_{12} < r_c$. Consequently,

$$r_{12} + r'_{12} < 2\,r_c \tag{10.14}$$

On the other hand, by the triangle inequality on the triangle $1\,2\,2'$, we obtain immediately

$$r_{12} + r_{12'} \geq L \geq 2\,r_c \tag{10.15}$$

where the last inequality stems from Eq. (10.13). The simultaneous validity of Eqs. (10.14) and (10.15) is clearly impossible. Thus, the hypothesis that there are more than one images of 2 interacting with 1 is not true, *i.e.* there can be no more than one image of 2 interacting with 1, Q.E.D.

For any two sites 1 and 2 in the primary box, the image of 2 lying closest to 1 is called *minimum image of* 2 *with respect to* 1. If the minimum image distance is less than r_c, then the minimum image pair is interacting.

The following is a simple code for identifying interacting images and calculating the total energy through a double loop over all sites in a cubic primary box:

```
      DO I = 1, N-1
C
              RXI = RX(I)

              RYI = RY(I)

              RZI = RZ(I)
C
              DO J = I+1, N
C Components of actual distance vector from J to I
                      RXIJ = RXI - RX(J)

                      RYIJ = RYI - RY(J)
```

```
            RZIJ = RZI - RZ(J)
C Minimum image distance vector from J to I
            RXIJ = RXIJ - BOXL*ANINT(RXIJ/BOXL)
            RYIJ = RYIJ - BOXL*ANINT(RYIJ/BOXL)
            RZIJ = RZIJ - BOXL*ANINT(RZIJ/BOXL)
C Squared minimum image distance between J and I
            RIJSQ = RXIJ**2 + RYIJ**2 + RZIJ**2
C Screening of squared minimum image distance against
C squared cutoff distance
            IF(RIJSQ.LT.RCUTSQ) THEN
                 ... compute I-J interaction ...
                 ... accumulate energy and forces ...
            ENDIF
     END DO
```

BOXL and RCUTSQ are the values of L and r_c^2, respectively, and the real function ANINT is defined as follows:

$$\text{ANINT}(x) = \begin{cases} +1, & \text{if } 0.5 < x < 1 \\ 0, & \text{if } -0.5 < x < 0.5 \\ -1, & \text{if } -1 < x < -0.5 \end{cases} \tag{10.16}$$

If the IF statement in the innermost loop prevents vectorization, the code can be modified by use of the "conditional vector merge" statement (see Allen and Tildesley).

10.2 Monte Carlo methods

The name *Monte Carlo* was first applied to a class of stochastic numerical methods by scientists working for the development of nuclear weapons at the Los Alamos National Laboratory in the 1940s. The essence of these methods is the invention of games of chance, whose behavior and outcome can be used to find the solution to a mathematical problem. Here we are only presenting a brief overview of Monte Carlo methods, with emphasis on the Metropolis algorithm for calculating averages with respect to a multivariate probability density function. For more details, the reader is referred to the books by Kalos and Whitlock (1986) and by Allen and Tildesley (1987).

10.3.1 Simple stochastic experiments: "Hit or miss Monte Carlo"

Early applications of Monte Carlo techniques were genuinely experimental. Perhaps one of the most interesting ones was Lazzerini's calculation of the number π by applying Buffon's theorem. In 1777, the French naturalist Georges-Louis Leclerc, Comte de Buffon, discovered the following theorem in geometrical probability: Consider a needle of length ℓ being thrown at random onto a set of equally spaced parallel lines, which are a distance d apart from one another ($d > \ell$) [see Figure 10.8 (a)]. The probability that the needle will cross a line is $\dfrac{2\,\ell}{\pi\,d}$. In 1901, the Italian mathematician Lazzerini set out to estimate π by the following stochastic experiment: He spun round and dropped a needle onto a plane surface where he had drawn parallel lines $N_{trials} = 3407$ times. He counted the number of times N_{hits} that the needle crossed a line. He then used his experimental results to estimate π as

$$\pi = \frac{2\,\ell}{d}\,\frac{N_{trials}}{N_{hits}} \tag{10.17}$$

Clearly, Lazzerini's experiment relies upon accumulating an estimate of the probability of crossing a line and then using Buffon's theoretical result to relate this probability to the sought π and to the geometrical parameters of the problem. Today, a computer can be used to simulate Lazzerini's experiment. One needs $\mathcal{O}(10^7)$ trials in order to estimate π to 3 significant figures. The error in the estimate decreases with increasing trials in proportion to $N_{trials}^{-1/2}$. Another "hit or miss" experimental method for calculating π is shown schematically in Figure 10.8(b). Envision two pans of the same height being left out on a rainy day. One pan is square, the other circular, with the diameter of the circular pan being equal to the edge length of the square pan. If N_{trials} is the number of raindrops (or weight of water) collected by the square pan and N_{hits} that collected in the circular pan, one can estimate π as

$$\pi = 4\,\frac{N_{hits}}{N_{trials}} \tag{10.18}$$

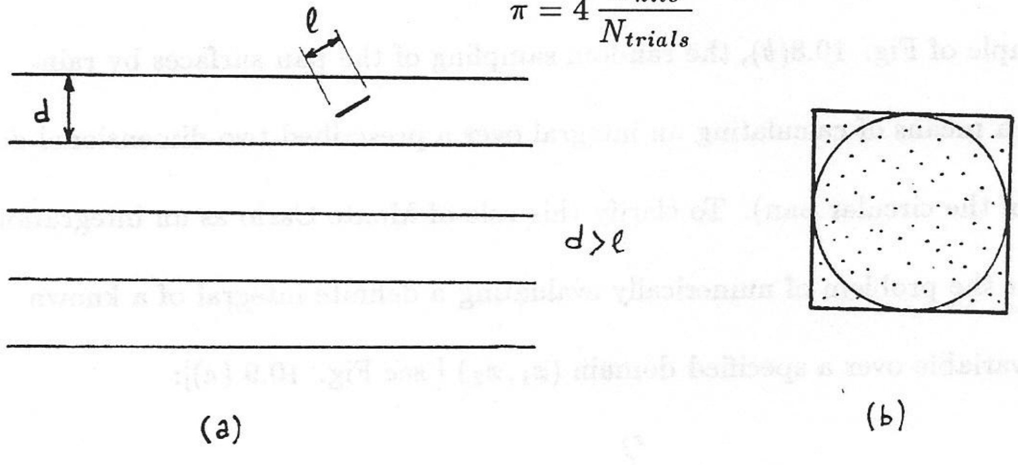

Figure 10.8 (a) Geometry of the Buffon needle experiment. (b) Estimation of π by collecting rainwater on a circular and a square pan.

10.3.2 Monte Carlo integration

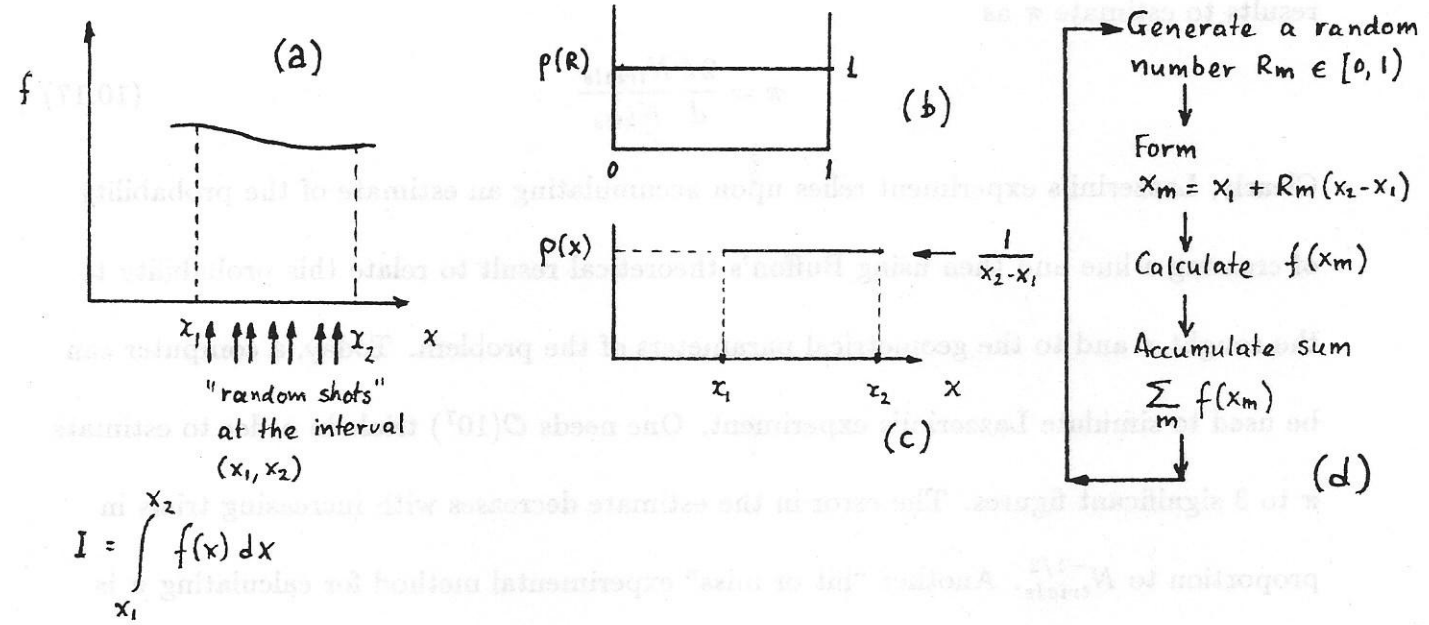
(a)
f
x_1
x_2
x
"random shots"
at the interval
(x_1, x_2)
$I = \int_{x_1}^{x_2} f(x)\,dx$
p(R)
1
(b)
0
1
p(x)
$\frac{1}{x_2 - x_1}$
x_1
x_2
x
(c)
Generate a random number $R_m \in [0,1)$
Form $x_m = x_1 + R_m(x_2 - x_1)$
Calculate $f(x_m)$
Accumulate sum $\sum_m f(x_m)$
(d)

Figure 10.9 (*a*) Schematic of the evaluation of a one- dimensional definite integral by Monte Carlo sample mean integration. (*b*) Uniform distribution in $[0, 1)$ sampled by pseudorandom numbers R. (*c*) Uniform distribution in the integration domain $[x_1, x_2)$ followed by the generated samples x. (*d*) Flow diagram of the Monte Carlo integration calculation.

In the example of Fig. 10.8(*b*), the random sampling of the pan surfaces by raindrops is used as a means of calculating an integral over a prescribed two-dimensional domain (the area of the circular pan). To clarify this role of Monte Carlo as an integration method, consider the problem of numerically evaluating a definite integral of a known function of one variable over a specified domain (x_1, x_2) [see Fig. 10.9 (*a*)]:

$$I = \int_{x_1}^{x_2} f(x)\,dx \tag{10.19}$$

One solution to this problem is provided by "Monte Carlo sample mean integration", shown in the simple flow diagram of Fig. 10.9 (d). One generates a large number of pseudorandom numbers R_m in the interval [0,1). These numbers R_m are samples from a uniform distribution $\rho(R)$ in the interval [0,1) [Fig. 10.9(b)]. (See Knuth (1973) for reliable algorithms that generate pseudorandom numbers.) For each R_m sampled, one forms the quantity $x_m = x_1 + R_m(x_2 - x_1)$. The values x_m are samples from a uniform distribution on the integration domain $[x_1, x_2)$ [Fig. 10.9 (c)]. For each such formed x_m, one computes the value $f_m = f(x_m)$ of the function to be integrated and accumulates the sum $\sum_m f_m$ over all sampled points. After sampling N_{trials} values of x in this way, one has an estimate of the integral I as

$$I \simeq \frac{(x_2 - x_1)}{N_{trials}} \sum_{m=1}^{N_{trials}} f_m \tag{10.20}$$

Theoretical justification for Eq. (10.20) is provided by the mean value theorem of integral calculus. As an example, application of this integration method to the function $f(x) = (1 - x^2)^{1/2}$ between $x_1 = 0$ and $x_2 = 1$ allows one to estimate π to within 4 significant figures in 10^7 trials.

As a numerical technique, sample mean integration cannot compete with more traditional numerical integration methods (Simpson method, Gaussian quadrature) in one or two dimensions. It becomes superior to them, however, when one has to perform *multidimensional* integrations. As an example, consider the evaluation of the configurational integral for N molecules in a cubic container of volume $V = L^3$ at temperature T

$$Z = \int d^{3N}r \; \exp\left[-\beta \mathcal{V}(\mathbf{r})\right] = \int d^{3N}r \, f(\mathbf{r}) \tag{10.21}$$

where the integrand function

$$f(\mathbf{r}) \equiv f(\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_N) \equiv \exp\left[-\beta\mathcal{V}(\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_N)\right] \tag{10.22}$$

Z is a $3N$-dimensional integral. For $N = \mathcal{O}(10)$, an estimate of Z can be obtained by the multidimensional analogue of Eq. (10.20) as

$$Z \simeq \frac{V^N}{N_{trials}} \sum_{m=1}^{N_{trials}} \exp\left[-\beta\mathcal{V}(\mathbf{r}_1^{(m)}, \mathbf{r}_2^{(m)}, \ldots, \mathbf{r}_N^{(m)})\right] \tag{10.23}$$

where $\left(\mathbf{r}_1^{(m)}, \mathbf{r}_2^{(m)}, \ldots, \mathbf{r}_N^{(m)}\right) \equiv \mathbf{r}^{(m)}$ is a randomly selected point in the $3N$-dimensional configuration space of the system. In practice, $\mathbf{r}^{(m)}$ is generated as N triplets of random numbers, each number being drawn from a uniform distribution in $[0, L)$. A Simpson technique would require evaluating $f(\mathbf{r})$ at all nodes of a regular $3N$-dimensional grid passed through the entire configuration space. If 10 points (nodes) per coordinate are used, this would entail 10^{3N} function evaluations. With Monte Carlo integration, one can get a reasonable estimate of Z using a number N_{trials} of function evaluations much smaller than this. Although the Monte Carlo estimate is subject to some uncertainty, due to the stochastic nature of the numerical method, meaningful error bounds can be established by examining how the estimate changes as a function of N_{trials}. In systems of six or more degrees of freedom, Monte Carlo becomes clearly preferable to the conventional quadrature techniques for the same amount of CPU time.

The direct calculation of configurational integrals (and therefore free energies) through Eq. (10.23) becomes impractical for $N = \mathcal{O}(100)$; for such dimensionalities, an inordinately large number of configurations sampled from a uniform distribution would be required for a reliable estimation of Z. Fortunately, estimates of thermodynamic properties for systems of such high dimensionality can still be derived through *importance sampling.*

10.3.3 Importance sampling for the calculation of ensemble averages

In chapters 4, 5, 6, and 9 we saw that important thermodynamic properties can be expressed as ensemble averages, of the general form

$$\langle f \rangle_{ens} = \frac{\int d^{3N}r\, f(\mathbf{r})\, \rho^{ens}(\mathbf{r})}{\int d^{3N}r\, \rho^{ens}(\mathbf{r})} \tag{10.24}$$

For example, when $\rho^{ens}(\mathbf{r}) = \rho^{NVT}(\mathbf{r}) = \frac{1}{Z}\exp\left[-\beta\mathcal{V}(\mathbf{r})\right]$,

$$f = \mathcal{V} \rightarrow \langle f \rangle_{NVT} = U^{ex} \equiv U(\rho,T) - U^{ig}(\rho,T)$$

$$f = \frac{1}{3}\sum_i \mathbf{r}_i \cdot \mathbf{F}_i^{int} \rightarrow \langle f \rangle_{NVT} = PV - N\,k_B\,T$$

$$f = \exp\left(-\beta\mathcal{V}_{test}\right) \rightarrow \langle f \rangle_{NVT} = \exp(-\beta\mu^{ex})$$

One could try to estimate $\langle f \rangle$ in Eq. (10.24) by applying the Monte Carlo integration method outlined in section 2.3.2 to both the numerator and the denominator, *i.e.*,

$$\langle f \rangle \simeq \frac{\sum_{m=1}^{N_{trials}} f(m)\,\rho^{ens}(m)}{\sum_{m=1}^{N_{trials}} \rho^{ens}(m)}$$

where m labels randomly generated configurations (points in configuration space). This, however, would be very inefficient: Some regions of the multidimensional configuration space (regions of high ρ^{ens}) contribute *much more* to the average than other regions. For example, in the NVT ensemble, relatively confined regions of low $\mathcal{V}$ have a large weight in the averaging, while more extended regions of high energy contribute very little. If we pick the sampled points by randomly "shooting" at the configuration space (Monte Carlo integration), we have an overwhelming probability of hitting states with very high energy, which would contribute almost nothing to the density. To realize why such a random selection of configurations would be very inefficient in sampling low-energy states, envision a process of generating configurations of a dense liquid by randomly picking N triads of coordinates for the molecular centers within a volume V. The probability that at least one overlap between the repulsive cores of the molecules will occur is overwhelming.

In calculating the ensemble average of Eq. (10.24), it would be advantageous to sample configuration space *nonuniformly*, so that *more probable states* (low energy configurations) are sampled *more frequently than less probable states* (high energy configurations). Such a nonuniform sampling is called *importance sampling.* In fact, it would be most advantageous to devise a sampling scheme such that *the frequency of sampling each state is proportional to* ρ^{ens} *for that state.* If we do this, then the calculation of $\langle f \rangle$ will amount to taking an arithmetic average over the sampled states:

$$\langle f \rangle = \frac{1}{N_{trials}} \sum_{m=1}^{N_{trials}} f(m) \tag{10.25}$$

An ingenious algorithm for sampling a multidimensional space according to a (nonuniform) probability distribution and for calculating averages with respect to that distribution was introduced by Metropolis *et al.* (1953). Before describing the Metropolis algorithm, we will briefly discuss Markov chains.

10.3.4 Markov chains

A Markov chain [Kemeny and Snell (1960), Allen and Tildesley (1987)] is a sequence of trials (stochastic process) that satisfies two conditions:

a. The outcome of each trial belongs to a finite set of outcomes, called the *state space.*

b. The outcome of each trial depends only on the oucome of the *immediately preceding trial.*

Central to a Markov chain is the concept of a *transition probability.* In these notes we will define the transition probability P_{nm} *from state m to state n* as the probability that a trial produces the state n, given that the previous trial has resulted in state m. P_{nm} depends only on the origin and destination states m, n and is independent of *where* within the sequence the considered trial lies. This is as a result of the definition of a Markov chain, whereby memory goes only one step back.

By collecting the values of P_{nm} for all possible n and m we can form a *transition probability matrix.* This matrix contains all information about the "dynamics" governing the "evolution" of the Markov chain.

To clarify the concept of a Markov chain, let us consider the following example: The weather pattern in some equatorial country can be well approxinated by the following simplistic rules:

- If it does not rain on one day, there is a 60% chance it will not rain on the next day.
- If it rains on one day, there is a 70% chance that it will rain on the following day.

We are interested in calculating the chances of rain over an extended period of time.

This system is described by a state space with two states: S (sunny, *i.e.*, no rain) and R (rainy). The transition probabilities are

$$P_{SS} = 0.60 \ , \quad P_{RS} = 0.40$$

$$P_{RR} = 0.70 \ , \quad P_{SR} = 0.30$$

They can be arranged in a *transition probability matrix* **P** as follows:

$$\mathbf{P} = \begin{bmatrix} P_{SS} & P_{SR} \\ P_{RS} & P_{RR} \end{bmatrix} = \begin{bmatrix} 0.60 & 0.30 \\ 0.40 & 0.70 \end{bmatrix} \tag{10.26}$$

Eq. (10.26) illustrates an important property of transition probability matrices, namely that

$$\sum_n P_{nm} = 1 \ \text{ for any } m \tag{10.27}$$

No matter what the origin state is, at the end of a trial the system must find itself in some state. Consequently, the elements of each column of **P** must sum to 1. A matrix **P** with this property is called a *stochastic matrix*.

Returning to our weather problem, assume that the weather forecast for December 31 is that there will be 50% chance of rain. For the initial a priori state probabilities we have:

$$\mathbf{q}^{(0)} = \begin{bmatrix} q_S^{(0)} \\ q_R^{(0)} \end{bmatrix} = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} \tag{10.28}$$

We now ask, what are the *a priori* probabilities $q_S^{(1)}$, $q_R^{(1)}$ of sunny/rainy weather on January 1? Clearly, by the definition of transition probabilities,

$$q_S^{(1)} = q_S^{(0)} \, P_{SS} + q_R^{(0)} \, P_{SR}$$

$$q_R^{(1)} = q_S^{(0)} P_{RS} + q_R^{(0)} P_{RR}$$

or

$$\mathbf{q}^{(1)} = \begin{bmatrix} P_{SS} & P_{SR} \\ P_{RS} & P_{RR} \end{bmatrix} \begin{bmatrix} q_S^{(0)} \\ q_R^{(0)} \end{bmatrix} = \mathbf{P}\,\mathbf{q}^{(0)} \tag{10.29}$$

Similarly, for the weather on January 2 we can write

$$\mathbf{q}^{(2)} = \mathbf{P}\,\mathbf{q}^{(1)} = \mathbf{P}^2\,\mathbf{q}^{(0)}$$

and, in general, for the t^{th} day of the year,

$$\mathbf{q}^{(t)} = \mathbf{P}\,\mathbf{q}^{(t-1)} = \mathbf{P}^t\,\mathbf{q}^{(0)} \tag{10.30}$$

Note the distinction between the *a priori* probabilities $\mathbf{q}$ and the *conditional* probabilities $\mathbf{P}$. Application of Eq. (10.30) for various t leads to the following results:

	$t=0$	$t=1$	$t=2$	$t=3$	$t=4$	$t=5$	$t=6$	$\cdots$	$t=20$	$\cdots$
q_S	0.50	0.45	0.435	0.4305	0.4291	0.4287	0.4286	$\cdots$	0.4286	$\cdots$
q_R	0.50	0.55	0.565	0.5695	0.5709	0.5713	0.5714	$\cdots$	0.5714	$\cdots$

Clearly, the Markov chain approaches a *limiting probability distribution* among the states, which we will symbolize by

$$\mathbf{\Pi} = \lim_{t \to \infty} \mathbf{q}^{(t)}$$

The limiting probability distribution satisfies the *stationary state* condition

$$\mathbf{P}\,\mathbf{\Pi} = \mathbf{\Pi} \tag{10.31}$$

This means that $\mathbf{\Pi}$ is an *eigenvector of the stochastic matrix* $\mathbf{P}$, corresponding to an eigenvalue of 1. Note that $\mathbf{\Pi}$ is completely determined by $\mathbf{P}$ and is not at all influenced by where the chain started (*i.e.*, by $\mathbf{q}^{(0)}$). All memory of the initial state has been effaced.

Markov chains in which one can ultimately go in a finite number of steps from any state to any other state are called *ergodic*, or *irreducible*. Such chains always have a limiting probability distribution (compare discussion in Section 10.2.3). One can deduce the nature of a Markov chain from its transition probability matrix. If the matrix $\mathbf{P}$ is block diagonal, the chain is not ergodic. Such chains may converge (get "trapped") in one of several stationary state distributions, depending on the initial state. On the other hand, if the transition probability matrix is full, the chain is ergodic.

10.3.5 The Metropolis Monte Carlo algorithm: A numerical method for calculating averages with respect to a multivariate probability density function

We now turn to the problem of sampling from a given probability distribution function. Given is a probability distribution in a multidimensional space. The distribution is defined by the probabilities Π_m for being at each point m of the multidimensional space. Note that the state space in question may be discrete or continuous. In the latter case, $\Pi_m = \rho(\mathbf{r}_m)\, d^{3N}r$, where $\rho(\mathbf{r}_m)$ is the probability density at point $\mathbf{r}_m$ and $d^{3N}r$ is an elementary volume in the multidimensional space centered at $\mathbf{r}_m$. Sought is an efficient numerical procedure for *sampling* the multidimensional state space according to the probability distribution $\{\Pi_m\}$. By "sampling" we mean picking a finite set of points (states)

$$m_1, m_2, \ldots, m_t, \ldots, m_{N_{trials}}$$

(where a given state may appear more than once in the sequence) such that the probability of finding each state m_t in the sequence is practically equal to the prescribed Π_{m_t}. If we accomplish this sampling (*i.e.*, generate such a sequence), then we can immediately calculate the average of any function defined on the space of interest as an arithmetic average, according to Eq. (10.25).

A solution to this problem was devised by Metropolis, Rosenbluth, Rosenbluth, Teller, and Teller (1953) and is known as the *Metropolis* or MR^2T^2 algorithm. The representative points are generated as a *sequence*, each from the previous one, according to stochastic rules that render the generated sequence a *Markov chain.*

$$m_1 \rightarrow m_2 \rightarrow \ldots m_t \rightarrow \ldots m_{N_{trials}}$$

The transition matrix for this Markov chain is selected as

$$P_{nm} = \begin{cases} C_{nm}, & \text{if } \Pi_n \geq \Pi_m, \quad n \neq m \\ C_{nm} \dfrac{\Pi_n}{\Pi_m}, & \text{if } \Pi_n < \Pi_m, \quad n \neq m \end{cases} \qquad (10.32a)$$

$$P_{nn} = 1 - \sum_{n \neq m} P_{nm} \qquad (10.32b)$$

where **C**, referred to as the *underlying matrix of the Markov chain,* is a symmetric stochastic matrix, satisfying the conditions:

$$C_{nm} = C_{mn} \quad \text{for every } m, n \qquad (10.33a)$$

$$\sum_n C_{nm} = 1 \qquad (10.33b)$$

Eqs. (10.32a) for the transition probability can be written in the form

$$P_{nm} = C_{nm} \min\left(1, \frac{\Pi_n}{\Pi_m}\right) \quad (m \neq n) \qquad (10.34)$$

C_{nm} is interpreted as an *attempt* probability. A move from state m to state n is attempted with probability C_{nm}. In practice, moves are usually attempted only between close-lying states in the multidimensional state space. That is, most of the elements of $\mathbf{C}$ are zero, except for some elements that correspond to pairs of states (m, n) that lie close to each other. Usually, but not always, C_{nm} is uniform in a small region of the space around state m; *i.e.*, one attempts to move from a given state m to any of the nearby states n with equal probability. The choice of $\mathbf{C}$ must ensure that the resulting Markov chain of states is ergodic. The second factor of the transition probability in Eq. (10.34) is interpreted as an *acceptance probability*. Having attempted the move $m \rightarrow n$ with probability C_{nm}, one accepts this move with probability $\min\left(1, \frac{\Pi_n}{\Pi_m}\right)$. Note that, if the trial state n is more probable than the origin state m, then n is accepted as the next state in the Markov chain. If the trial state n is less probable than the origin state m, then n is accepted only *some of the time*, with probability $\frac{\Pi_n}{\Pi_m} < 1$. If the trial state, n, is not accepted, then the original state m is retained as the next state in the Markov chain according to the requirements of Eq. (10.32b). The fact that not every attempted move leads to a change in state is a very important aspect of the MR^2T^2 algorithm. To show that the transition probabilities defined by the Metropolis scheme, Eqs. (10.32) and (10.33), will generate a Markov chain of states whose *a priori* probability distribution in state space asymptotically converges to the given distribution $\mathbf{\Pi}$, we prove three statements concerning the transition matrix $\mathbf{P}$.

Statement 1: The matrix **P** is stochastic. *i.e.*

$$\sum_n P_{nm} = 1 \tag{10.35}$$

This follows immediately from Eq. (10.32b), which is part of the definition of the transition probabilities.

Statement 2: **P** and **Π** satisfy the condition of *microscopic reversibility* or *detailed balance*, *i.e.*,

$$\Pi_m P_{nm} = \Pi_n P_{mn} \tag{10.36}$$

Physically, Eq. (10.36) informs us that the overall probability of encountering a move $m \rightarrow n$ somewhere in the Markov chain equals the probability of encountering the reverse move $n \rightarrow m$.

To prove Eq. (10.36), assume that, for the two states in question, $\Pi_n \geq \Pi_m$. Then, following the general prescription, Eq. (10.32),

$$\Pi_m P_{nm} = \Pi_m C_{nm}$$

$$\Pi_n P_{mn} = \Pi_n \left(C_{mn} \frac{\Pi_m}{\Pi_n} \right) = C_{mn} \Pi_m = \Pi_m C_{mn}$$

while, from Eq. (10.33a), $C_{nm} = C_{mn}$.

From the above three equations, $\Pi_m P_{nm} = \Pi_n P_{mn}$, Q.E.D. The proof in the case $\Pi_n < \Pi_m$ proceeds exactly as above, with the role of m and n reversed. Note that correct Monte Carlo algorithms can be designed even with a nonsymmetric matrix **C** of attempt probabilities, provided the acceptance probabilities are redefined in such a way as to ensure that the condition of microscopic reversibility, Eq. (10.36), is satisfied. For such "smart" Monte Carlo alghorithms, see Allen and Tildesley (1987).

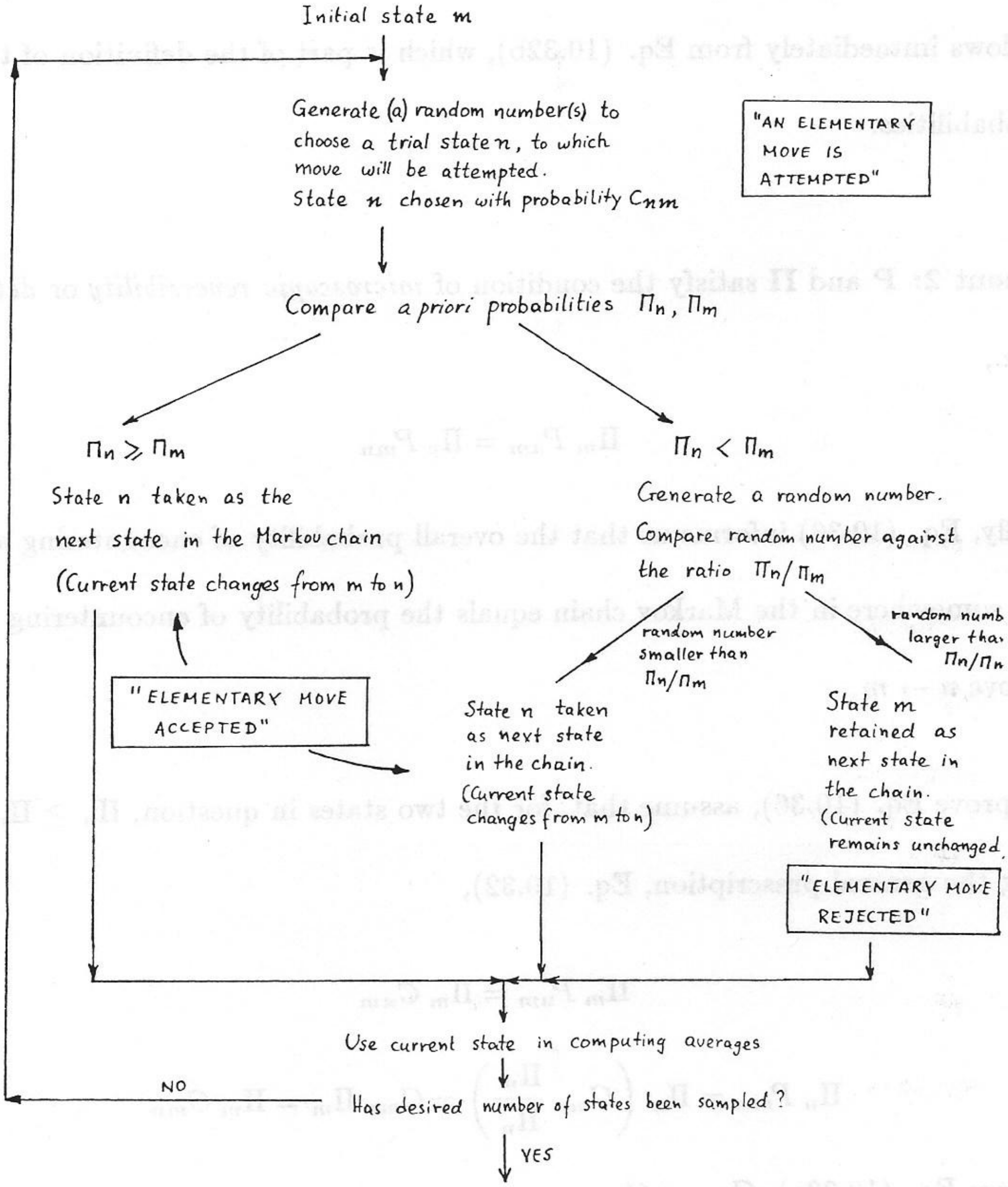
Initial state m
Generate (a) random number(s) to choose a trial state n, to which move will be attempted.
State n chosen with probability C_{nm}
"AN ELEMENTARY MOVE IS ATTEMPTED"
Compare *a priori* probabilities Π_n, Π_m
$\Pi_n \geqslant \Pi_m$
State n taken as the next state in the Markov chain (Current state changes from m to n)
"ELEMENTARY MOVE ACCEPTED"
$\Pi_n < \Pi_m$
Generate a random number. Compare random number against the ratio Π_n/Π_m
random number smaller than Π_n/Π_m
random numb larger tha Π_n/Π_n
State n taken as next state in the chain. (Current state changes from m to n)
State m retained as next state in the chain. (Current state remains unchanged
"ELEMENTARY MOVE REJECTED"
Use current state in computing averages
NO
Has desired number of states been sampled ?
YES

Figure 10.10 Flow of calculations in a Metropolis Monte Carlo algorithm for sampling the probability distribution **Π**.

Statement 3: **Π** is the limiting distribution corresponding to the transition matrix **P**. To show this statement, we must prove that **Π** is an eigenvector of matrix **P** corresponding to eigenvalue 1 [compare Eq. (10.31)]. Considering the i^{th} element of the product $\mathbf{P}\,\mathbf{\Pi}$ and using the condition of microscopic reversibility, Eq. (10.36), which we just proved, we obtain:

$$\sum_m P_{im}\,\Pi_m = \sum_m P_{mi}\,\Pi_i = \Pi_i \sum_m P_{mi} = \Pi_i\,1 = \Pi_i$$

where the last transformation is possible by the stochastic property of matrix **P**, Eq. (10.35). Applying the above reasoning for all i, we are led to

$$\mathbf{P}\,\mathbf{\Pi} = \mathbf{\Pi} \quad \text{Q.E.D.} \tag{10.31}$$

Eq. (10.31) guarantees that, whatever the initial state in the generated Markov chain of states, a chain that is long enough will asymptotically sample the probability distribution of interest, defined by $\mathbf{\Pi}$. How long it will take for this asymptotic situation to be reached will depend on the choice of the matrix **C** of attempt probabilities. Good Monte Carlo algorithms will move quickly through state space, sampling states of high probability efficiently, and quickly effacing all memory of the initial state. It is noteworthy that, to define the transition matrix **P** according to Eqs. (10.32), one need only know $\mathbf{\Pi}$ up to a multiplicative constant. Only *ratios* of *a priori* probabilities appear in the Metropolis scheme; this is particularly convenient in applications of the algorithm to statistical mechanics.

The general flow of calculations in a Metropolis algorithm designed to sample a multivariate probability distribution $\mathbf{\Pi}$ and to compute averages with respect to that distribution is shown in Figure 10.10.

10.3.6 *NVT* Monte Carlo by the Metropolis algorithm

As an example, we consider how the Metropolis scheme is applied for the simulation of a simple liquid in the canonical ensemble. To fix ideas, consider a simulation of liquid argon, modelled by a pairwise additive LJ potential such as the one displayed in Figure 10.2. Our objective is to generate a sequence of configurations of N molecules in a box of constant volume V with periodic boundary conditions, such that the sequence asymptotically samples the probability density of the canonical ensemble at temperature T.

In transcribing the general terminology and notation of the previous section to the simulation in question, we select the configuration space $(\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_N)$ as the multidimensional state space to be sampled. States correspond to configurations. The *a priori* probability Π_m of a configuration m equals $\rho^{NVT}(\mathbf{r}_1^{(m)}, \mathbf{r}_2^{(m)}, \ldots, \mathbf{r}_N^{(m)})\, d^{3N}r$, with ρ^{NVT} the equilibrium probability density of the canonical ensemble and $d^{3N}r$ an elementary volume in configuration space that is assigned to state m. The ratio of state probabilities entering the Metopolis selection criteria becomes

$$\frac{\Pi_m}{\Pi_n} = \frac{\rho^{NVT}(\mathbf{r}_1^{(n)}, \ldots, \mathbf{r}_N^{(n)})}{\rho^{NVT}(\mathbf{r}_1^{(m)}, \ldots, \mathbf{r}_N^{(m)})} \equiv \frac{\rho_n^{NVT}}{\rho_m^{NVT}} \qquad (10.37)$$

Now,

$$\rho_m^{NVT} = \frac{\exp(-\beta \mathcal{V}_m)}{Z(N, V, T)} \qquad (10.38)$$

so

$$\frac{\rho_n^{NVT}}{\rho_m^{NVT}} = \exp\left[-\beta\left(\mathcal{V}_n - \mathcal{V}_m\right)\right] = \exp\left[-\beta \Delta \mathcal{V}_{m \to n}\right] \qquad (10.39)$$

The acceptance probability becomes

$$\min\left(1, \frac{\rho_n^{NVT}}{\rho_m^{NVT}}\right) = \begin{cases} 1, & \text{if } \mathcal{V}_n \leq \mathcal{V}_m \\ \exp\left[-\beta \Delta \mathcal{V}_{m \rightarrow n}\right], & \text{if } \mathcal{V}_n > \mathcal{V}_m \end{cases} \tag{10.40}$$

The Metropolis selection criterion involves only the potential energy change. Downhill moves in energy are accepted always, while uphill moves are accepted with a probability equal to the Boltzmann factor of the potential energy change they bring about.

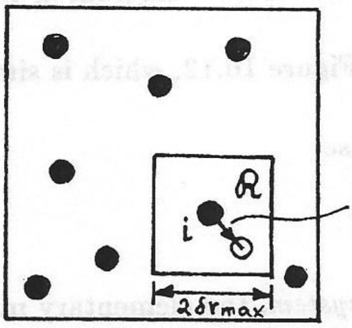

Figure 10.11 Schematic of displacement move attempted in a Metropolis Monte Carlo simulation of a simple fluid.

As regards the attempt probabilities, the elementary move employed is a translation of a randomly chosen molecule by a random vector. The components of the random displacement vector are uniformly distributed within an interval $(-\delta r_{max}, \delta r_{max})$. In this way, the attempted new positions form a uniform distribution within a cube $\mathcal{R}$ of edge length $2\,\delta r_{max}$, centered at the original position of the molecule (see Figure 10.11). The element C_{nm} of the underlying matrix of the Markov chain is thus

$$C_{nm} = \begin{cases} \dfrac{1}{N}\dfrac{1}{N_{\mathcal{R}}}, & \text{if } \Delta \mathbf{r}_i \in \mathcal{R} \\ 0, & \text{if } \Delta \mathbf{r}_i \notin \mathcal{R} \end{cases} \tag{10.41}$$

where N is the total number of molecules in the box and $N_{\mathcal{R}}$ the total number of numerically distinguishable positions in the cube of edge length $2\,\delta r_{max}$. The first factor in C_{nm} corresponds to the choice of the molecule to be displaced, while the second factor corresponds to the choice of its displacement vector. Note that the matrix $\mathbf{C}$ is by definition *symmetric*, as C_{nm} does not depend on the identity of states m or n.

Eqs. (10.40) and (10.41) summarize the Metropolis Monte Carlo algorithm for a simple fluid in the canonical ensemble. A flow diagram of the computations involved in a step of the algorithm is given in Figure 10.12, which is simply a transcription of Figure 10.10 for the canonical ensemble case.

When simulating a *molecular system*, the elementary moves attempted should change all the configurational degrees of freedom. They should include rigid translation of a randomly chosen molecule by a random vector, rigid rotation (random reorientation) of a randomly chosen molecule, and torsion around a randomly chosen rotatable bond of a randomly chosen molecule by a random dihedral angle for flexible molecules. For the acceptance criteria for reorientation moves, see Allen and Tildesley (1987).

The question of the optimal choice of δr_{max} has received some attention. This quantity should be chosen so as to give maximal efficiency in the exploration of configuration space (*e.g.*, maximal mean squared displacement of molecules for a given number of moves). Traditionally, δr_{max} is chosen so as to lead to 50% acceptance of the attempted moves (50% acceptance ratio). There is indication, however, especially in simulations of more complex systems, that bolder moves (with lower probability of acceptance than 50%) are more efficient.

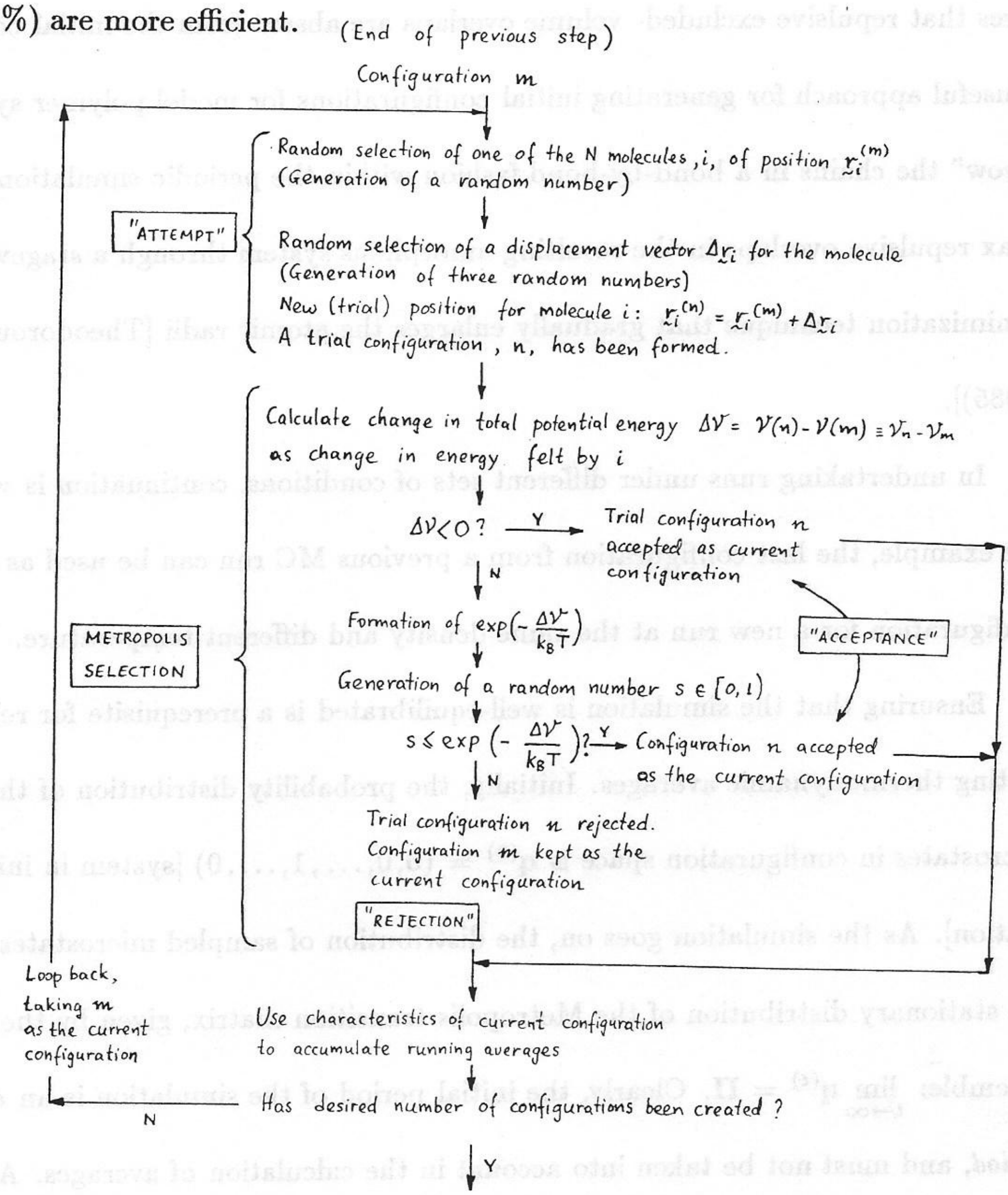
(End of previous step)
Configuration m
Random selection of one of the N molecules, i, of position $r_i^{(m)}$
(Generation of a random number)
"ATTEMPT"
Random selection of a displacement vector Δr_i for the molecule
(Generation of three random numbers)
New (trial) position for molecule i: $r_i^{(n)} = r_i^{(m)} + \Delta r_i$
A trial configuration, n, has been formed.
Calculate change in total potential energy $\Delta \mathcal{V} = \mathcal{V}(n) - \mathcal{V}(m) \equiv \mathcal{V}_n - \mathcal{V}_m$
as change in energy felt by i
$\Delta \mathcal{V} < 0$?
Y
Trial configuration n accepted as current configuration
N
Formation of $\exp(-\frac{\Delta \mathcal{V}}{k_B T})$
METROPOLIS SELECTION
"ACCEPTANCE"
Generation of a random number $s \in [0,1)$
$s \leq \exp(-\frac{\Delta \mathcal{V}}{k_B T})$?
Y
Configuration n accepted as the current configuration
N
Trial configuration n rejected.
Configuration m kept as the current configuration
"REJECTION"
Loop back, taking m as the current configuration
Use characteristics of current configuration to accumulate running averages
N
Has desired number of configurations been created?
Y

Figure 10.12 Flow diagram of the calculations involved in a Metropolis Monte Carlo simulation of a simple fluid in the canonical ensemble.

As regards the choice of a starting configuration, it is advantageous that this be a configuration of high probability under the conditions of interest, to minimize the time spent in equilibration. The traditional approach for simple liquids is to start from an *fcc* lattice of the required density and "melt" it to obtain a liquid; in this way, one ensures that repulsive excluded- volume overlaps are absent from the initial configuration. A useful approach for generating initial configurations for model polymer systems is to "grow" the chains in a bond-by-bond fashion within the periodic simulation box and relax repulsive overlaps in the resulting amorphous system through a stagewise energy minimization technique that gradually enlarges the atomic radii [Theodorou and Suter (1985)].

In undertaking runs under different sets of conditions, continuation is very effective. For example, the last configuration from a previous MC run can be used as a starting configuration for a new run at the same density and different temperature.

Ensuring that the simulation is well-equilibrated is a prerequisite for reliably estimating thermodynamic averages. Initially, the probability distribution of the sampled microstates in configuration space is $\mathbf{q}^{(0)} = (0, 0, \ldots, 1, \ldots, 0)$ [system in initial configuration]. As the simulation goes on, the distribution of sampled microstates approaches the stationary distribution of the Metropolis transition matrix, given by the simulated ensemble: $\lim_{t\to\infty} \mathbf{q}^{(t)} = \mathbf{\Pi}$. Clearly, the initial period of the simulation is an *equilibration period*, and must not be taken into account in the calculation of averages. At the end of the equilibration period, *all memory of the initial configuration should have been lost.* Equilibration can be monitored in several ways. Quantities that can be readily monitored are potential energy and pressure. The equilibration period should be extended until they show no systematic drift as functions of the number of attempted moves, but oscillate about steady mean values. In the case of a liquid simulation that has been initiated at a lattice configuration, one must ensure that all vestiges of order have disappeared before starting the thermodynamic averaging; translational and orientational order parameters (see section 10.5) must show no signs of long-range order. The observation of self-diffusive behavior (mean squared displacement of molecules growing linearly with time) is another indication of equilibration. As a rule of thumb, $500N - 1000N$ MC steps are typically sufficient for equilibrating a simple liquid, having started from a liquidlike configuration. More time is required if one starts from a lattice or if a phase transition is close to the simulated thermodynamic state point. Much more time is necessary for equilibration in systems of associating liquids or polymer systems, where severe constraints on configurational rearrangement are placed by the strong and specfic interactions of the chain connectivity.

10.3.7 Isothermal-isobaric (NPT) Monte Carlo

As a second application of the Metropolis algorithm we consider the simulation of a simple fluid in the isothermal-isobaric ensemble [Wood (1968)]. Such NPT simulations are convenient in that they permit tracking the thermodynamic properties of a model system along constant-pressure conditions that closely correspond to the conditions used in experimental measurements. Also, since it explicitly sets pressure, temperature, and composition, *i.e.*, all intensive properties that fix the thermodynamic properties of a single phase, this type of simulation is much less likely to lead to phase separation within the simulation box than a canonical ensemble simulation, in which total volume is fixed, but pressure can fluctuate.

The acceptance criterion of MC moves is dictated, as always, by the probability density of the ensemble of interest:

$$\rho^{NPT}(\mathbf{r};\, V) = \frac{\exp\left[-\beta\left(\mathcal{V}(\mathbf{r}) + P\,V\right)\right]}{Z_{NPT}} \tag{10.42}$$

where $\mathbf{r}$ stands, as usual, for the $3N$-dimensional vector of positions of the N molecules. Fluctuations in both the system volume and the molecular coordinates must be introduced to sample configuration space. A complication in using $(\mathbf{r}, V)$ as degrees of freedom is that the limits between which the components of $\mathbf{r}$ take on values depend on V. To avoid this interdependence among the degrees of freedom, we introduce *scaled coordinates*

$$\mathbf{s} \equiv (\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_N) \quad \text{where}$$

$$\mathbf{s} = \frac{1}{L}\mathbf{r} \tag{10.43}$$

with $L \equiv V^{1/3}$ the edge length of the simulation box. The scaled coordinates $\mathbf{s}$ are dimensionless. If the center of the primary box is put at the origin and is retained constant while fluctuating V, the components $s_{\alpha\, i}$, $\alpha = x, y, z$, $i = 1(1)N$ all satisfy

$$-1/2 \leq s_{\alpha\, i} \leq 1/2$$

The probability density in the transformed coordinate set, $\rho^{NPT}(\mathbf{s}, V)$, is found by equating the state probabilities in the original and transformed representations:

$$\rho^{NPT}(\mathbf{s}; V)\, d^{3N}s\, dV = \rho^{NPT}(\mathbf{r}; V)\, d^{3N}r\, dV = \rho^{NPT}(\mathbf{r}, V)\, L^{3N}\, d^{3N}s\, dV, \quad \text{or}$$

$$\rho^{NPT}(\mathbf{s};\, V) = \frac{\exp\left[-\beta\left(\mathcal{V}(\mathbf{s}; V) + P\, V\right)\right]}{Z_{NPT}}\, V^N = \frac{\exp\left[-\beta\left(\mathcal{V}(\mathbf{s}; V) + P\, V\right) + N \ln V\right]}{Z_{NPT}} \tag{10.44}$$

The factor V^N appearing in Eq. (10.44) is the Jacobian of transformation (ratio of elementary volumes in configuration space) between new and old coordinates. The ratio of state probabilities used in the Metropolis selection criteria is

$$\frac{\rho_n}{\rho_m} = \frac{\exp\left[-\beta\left(\mathcal{V}_n + P\, V_n\right) + N \ln V_n\right]}{\exp\left[-\beta\left(\mathcal{V}_m + P\, V_m\right) + N \ln V_m\right]} =$$

$$\exp\left[-\beta\left(\mathcal{V}_n - \mathcal{V}_m + P\left(V_n - V_m\right) - \frac{N}{\beta} \ln \frac{V_n}{V_m}\right)\right] =$$

$$\exp\left[-\beta\left(\Delta\mathcal{V} + P\, \Delta V - \frac{N}{\beta} \ln \frac{V_n}{V_m}\right)\right] \tag{10.45}$$

The quantity $\Delta\mathcal{V} + P\, \Delta V - \dfrac{N}{\beta} \ln \dfrac{V_n}{V_m}$ replaces the $\Delta\mathcal{V}$ used in the canonical algorithm.

The elementary moves attempted in the NPT MC algorithm are shown schematically in Figure 10.13. The molecule translation moves (change in $\mathbf{s}_i$ under constant V) are carried out exactly as in NVT MC. The volume fluctuation moves change V under constant s. They bring about a uniform dilation/contraction of the primary box and its images under preservation of all scaled coordinates; the accompanying deformation of the model system, which occurs under preservation of the scaled distances between interaction sites, is sometimes referred to as *affine*. The acceptance probability obtained from the general criterion, Eq. (10.45), for each of these types of moves is shown in Fig. 10.13

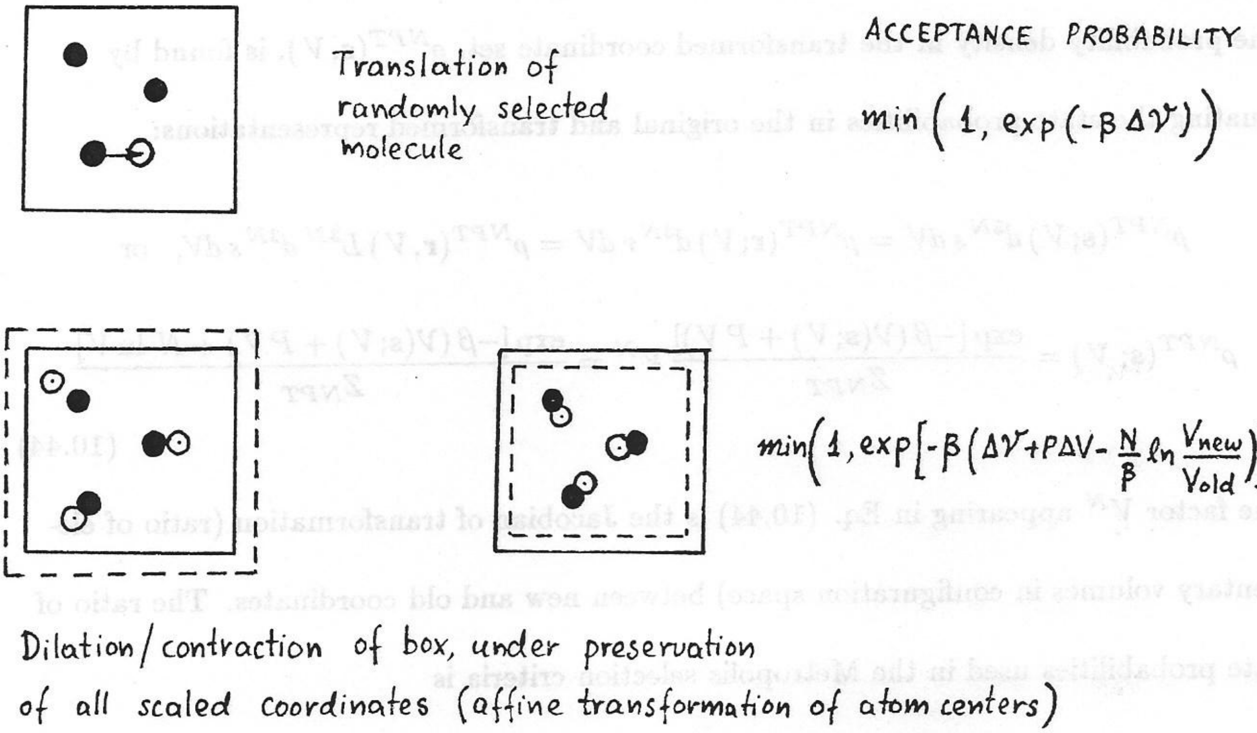
ACCEPTANCE PROBABILITY
Translation of randomly selected molecule
$\min\left(1, \exp(-\beta \Delta \mathcal{V})\right)$
$\min\left(1, \exp\left[-\beta\left(\Delta\mathcal{V} + P\Delta V - \frac{N}{\beta} \ln \frac{V_{new}}{V_{old}}\right)\right]\right),$
Dilation/contraction of box, under preservation of all scaled coordinates (affine transformation of atom centers)

Figure 10.13 Elementary moves employed in isothermal-isobaric MC simulation of a simple fluid, and associated Metropolis selection criteria.

NPT MC changes the the density of the model system. "Tail" contributions to the total potential energy [compare Eq. (10.11)] must therefore be recomputed at each step and taken into account in the Metropolis selection rules.

In systems with exclusively pairwise additive LJ interactions between sites, the calulation of the energy change upon introducing a volume fluctuation can be accomplished very fast, based on a simple scaling argument. The potential energy in the original state m can be written as

$$\mathcal{V}_m = 4\,\epsilon \sum_i \sum_{j>i} \left(\frac{\sigma}{L_m\, s_{ij}^{(m)}} \right)^{12} - 4\,\epsilon \sum_i \sum_{j>i} \left(\frac{\sigma}{L_m\, s_{ij}^{(m)}} \right)^{6} \equiv \mathcal{V}_m^{(12)} + \mathcal{V}_m^{(6)} \qquad (10.46)$$

Given that the scaled coordinates remain unchanged in a colume fluctuation move, the potential energy in the new state n, $\mathcal{V}_n$, is obtainable as

$$\mathcal{V}_n = \mathcal{V}_m^{(12)} \left(\frac{L_m}{L_n} \right)^{12} + \mathcal{V}_m^{(6)} \left(\frac{L_m}{L_n} \right)^{6} \qquad (10.47)$$

Clearly, this argument does not hold for molecular systems where bonded potentials are used, or where bond lengths and/or bond angles are kept fixed, meaning that scaled intramolecular distances are not preserved in a volume fluctuation move.

The relative frequency of translation (rotation, conformational change) and volume (edgelength) fluctuation moves in an NPT MC simulation is set so that equilibration is fastest. Volume fluctuations are typically much less frequent than moves changing the molecular coordinates, so that the model system is given the chance to adjust itself to the modified density. As a rule of thumb, Jorgensen suggests introducing 1 volume fluctuation every $6N$ translations in a simple fluid simulation.

10.3.8 Grand canonical (μVT) Monte Carlo algorithm

The general principles of Metropolis Monte Carlo can be applied relatively straightforwardly to simulate a fluid in the grand canonical ensemble. Such simulations are valuable in studying sorption equilibria and interfacial structure and thermodynamics. The first Grand Canonical Monte Carlo (GCMC) algorithm was proposed by Norman and Filinov (1969).

In the case of GCMC, state space is comprised of both the number N of particles present in the system and the coordinates $\mathbf{r}^N$ of these particles (compare Chapter 6). The probability density to be sampled is

$$\rho^{\mu VT}(\mathbf{r}^N;N) = \frac{1}{N!}\frac{\exp(N\beta\mu)}{\Lambda^{3N}} \exp\left[-\beta\,\mathcal{V}(\mathbf{r}_1,\ldots,\mathbf{r}_N)\right]\frac{1}{\Xi} \tag{10.48}$$

Clearly, $\mathbf{r}^N$ and N are not independent, as the dimensionality of $\mathbf{r}^N$ depends on N. In fact, the *dimensions* of the density function $\rho^{\mu VT}(\mathbf{r}^N;N)$ depend on N, which complicates the consideration of density ratios in designing selection criteria for the algorithm. To eliminate this problem, we introduce again scaled coordinates $(\mathbf{s}_1,\ldots,\mathbf{s}_N)$, where $s_{\alpha\,i} = V^{-1/3}\,r_{\alpha\,i}$, $0 \le s_{\alpha\,i} \le 1$. The probability density in the transformed state space is

$$\rho^{\mu VT}(\mathbf{s}^N;N) = \rho^{\mu VT}(\mathbf{r}^N;N)\,V^N \quad \text{or}$$

$$\rho(\mathbf{s}^N;N) = \frac{\exp\left[-\beta\left(\mathcal{V}(\mathbf{s}^N;N) - N\,\mu\right) - \ln N! - 3N\,\ln\Lambda + N\,\ln V\right]}{\Xi} \tag{10.49}$$

Note that $\rho^{\mu VT}(\mathbf{s}^N;N)$ is dimensionless. Taking the ratio of $\rho^{\mu VT}(\mathbf{s}^N;N)$ at two different states,

$$\frac{\rho_n}{\rho_m} = \frac{\exp\left[-\beta\left(\mathcal{V}_n - N_n\,\mu\right) - \ln N_n! - 3\,N_n\,\ln\Lambda + N_n\,\ln V\right]}{\exp\left[-\beta\left(\mathcal{V}_m - N_m\,\mu\right) - \ln N_m! - 3\,N_m\,\ln\Lambda + N_m\,\ln V\right]} =$$

$$\exp\left[-\beta\left(\mathcal{V}_n - \mathcal{V}_m\right) + \beta\,\mu\left(N_n - N_m\right) - \ln\frac{N_n!}{N_m!} - 3\left(N_n - N_m\right)\ln\Lambda + \left(N_n - N_m\right)\ln V\right] \tag{10.50}$$

In particular, if $N_n - N_m = 1$ (creation of a molecule),

$$\frac{\rho_n}{\rho_m} = \exp\left[-\beta\left(\mathcal{V}_n - \mathcal{V}_m\right) + \beta\mu - \ln\left(N_m + 1\right) - 3\,\ln\Lambda + \ln V\right] =$$

$$\exp\left[-\beta\left(\mathcal{V}_n-\mathcal{V}_m\right)+\ln\left(\frac{\beta f V}{N_m+1}\right)\right] \tag{10.51}$$

where $f=\dfrac{1}{\beta}\dfrac{\exp(\beta\mu)}{\Lambda^3}$ is the fugacity. If $N_n - N_m = -1$ (destruction of a molecule),

$$\frac{\rho_n}{\rho_m}=\exp\left[-\beta\left(\mathcal{V}_n-\mathcal{V}_m\right)-\beta\mu+\ln N_m+3\ln\Lambda-\ln V\right]=$$

$$\exp\left[-\beta\left(\mathcal{V}_n-\mathcal{V}_m\right)+\ln\left(\frac{N_m}{\beta f V}\right)\right] \tag{10.52}$$

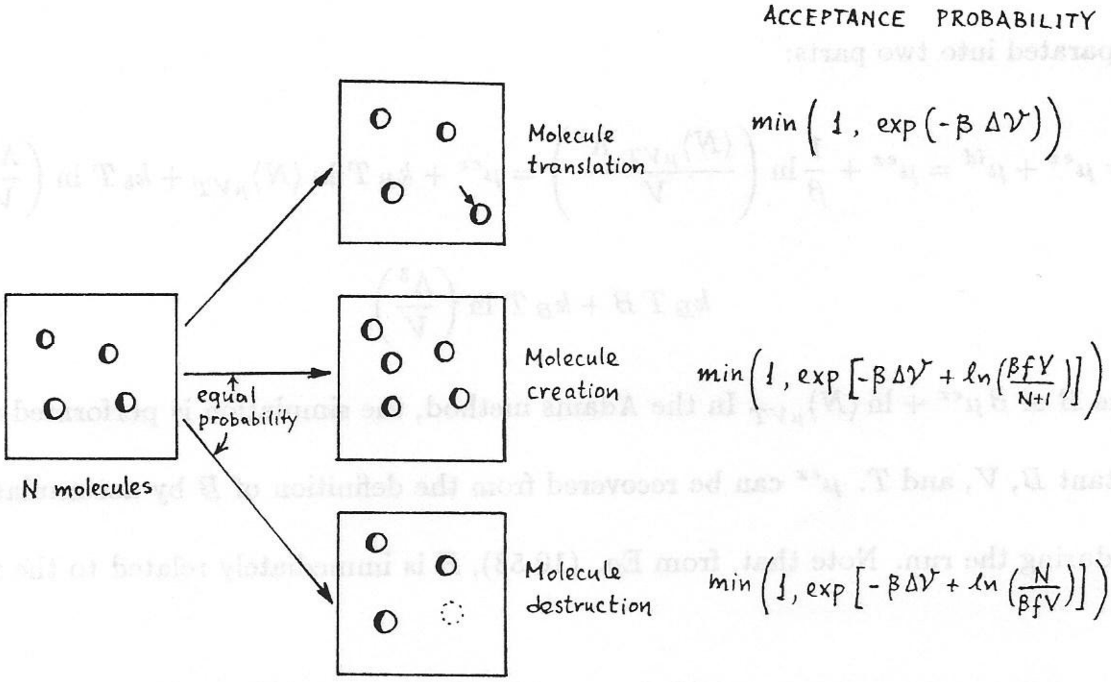
ACCEPTANCE PROBABILITY
Molecule translation
$\min\left(1,\ \exp\left(-\beta\,\Delta\mathcal{V}\right)\right)$
Molecule creation
$\min\left(1,\exp\left[-\beta\Delta\mathcal{V}+\ln\left(\frac{\beta f V}{N+1}\right)\right]\right)$
equal probability
Molecule destruction
$\min\left(1,\exp\left[-\beta\Delta\mathcal{V}+\ln\left(\frac{N}{\beta f V}\right)\right]\right)$
N molecules

Figure 10.14 Types of moves used in Grand Canonical Monte Carlo and corresponding selection criteria.

As used in practice, GCMC attempts three types of moves: Translation of a randomly chosen molecule by a random vector; creation of a new molecule in a randomly chosen position of the simulation box; and destruction of a randomly chosen molecule. Creation and destruction moves are attempted with the same frequency, to ensure microscopic reversibility. Molecule displacement moves are conducted exactly as in NVT

MC simulations, and are subject to the same selection criterion as in NVT MC [obtainable from Eq. (10.50) by setting $N_n = N_m$]. Attempts to create or destroy a molecule are assessed with criteria (10.51) and (10.52), respectively. The three types of moves are shown schematically in Figure 10.14, along with the corresponding selection criteria.

Adams (1974) introduced a modification of the GCMC algorithm in which a quantity B related to the *excess* chemical potential is used as input. The chemical potential is separated into two parts:

$$\mu = \mu^{ex} + \mu^{id} = \mu^{ex} + \frac{1}{\beta} \ln \left(\frac{\langle N \rangle_{\mu V T} \, \Lambda^3}{V} \right) = \mu^{ex} + k_B \, T \ln \langle N \rangle_{\mu V T} + k_b \, T \ln \left(\frac{\Lambda^3}{V} \right) =$$

$$k_B \, T \, B + k_B \, T \ln \left(\frac{\Lambda^3}{V} \right) \tag{10.53}$$

where $B \equiv \beta \, \mu^{ex} + \ln \langle N \rangle_{\mu V T}$ In the Adams method, the simulation is performed under constant B, V, and T. μ^{ex} can be recovered from the definition of B by accumulating $\langle N \rangle$ during the run. Note that, from Eq. (10.53), B is immediately related to the fugacity:

$$\beta f = \frac{\exp(\beta \mu)}{\Lambda^3} = \frac{\exp(B) \, \Lambda^3 / V}{\Lambda^3} = \frac{\exp(B)}{V} \tag{10.54}$$

Clearly, constancy of B and T implies constancy of μ and T, and Adams' BVT simulation is a GCMC simulation. The selection criteria, Eqs. (10.51) and (10.52), can be recast in terms of B as

$$\frac{\rho_n}{\rho_m} = \exp \left[-\beta \, \Delta \mathcal{V} + \ln \left(\frac{e^B}{N+1} \right) \right] \quad \text{(molecule creation)} \tag{10.55}$$

$$\frac{\rho_n}{\rho_m} = \exp \left[-\beta \, \Delta \mathcal{V} + \ln \left(\frac{N}{e^B} \right) \right] \quad \text{(molecule destruction)} \tag{10.56}$$