3

Simulations of Molecular Ensembles

3.1 Relationship Between MM Optima and Real Systems

As noted in the last chapter within the context of comparing theory to experiment, a minimum-energy structure, i.e., a local minimum on a PES, is sometimes afforded more importance than it deserves. Zero-point vibrational effects dictate that, even at 0 K, the molecule probabilistically samples a range of different structures. If the molecule is quite small and is characterized by fairly 'stiff' molecular coordinates, then its 'well' on the PES will be 'narrow' and 'deep' and the range of structures it samples will all be fairly close to the minimum-energy structure; in such an instance it is not unreasonable to adopt the simple approach of thinking about the 'structure' of the molecule as being the minimum energy geometry. However, consider the case of a large molecule characterized by many 'loose' molecular coordinates, say polyethyleneglycol, (PEG,$-(OCH_2CH_2)_n-$), which has 'soft' torsional modes: What is the structure of a PEG molecule having $n = 50$? Such a query is, in some sense, ill defined. Because the probability distribution of possible structures is not compactly localized, as is the case for stiff molecules, the very concept of structure as a time-independent property is called into question. Instead, we have to accept the flexibility of PEG as an intrinsic characteristic of the molecule, and any attempt to understand its other properties must account for its structureless nature. Note that polypeptides, polynucleotides, and polysaccharides all are *also* large molecules characterized by having many loose degrees of freedom. While nature has tended to select for particular examples of these molecules that are less flexible than PEG, nevertheless their utility as biomolecules sometimes derives from their ability to sample a wide range of structures under physiological conditions, and attempts to understand their chemical behavior must address this issue.

Just as zero-point vibration introduces probabilistic weightings to single-molecule structures, so too thermodynamics dictates that, given a large collection of molecules, probabilistic distributions of structures will be found about *different* local minima on the PES at non-zero absolute temperatures. The relative probability of clustering about any given minimum is a function of the temperature and some particular thermodynamic variable characterizing the system (e.g., Helmholtz free energy), that variable depending on what experimental conditions are being held constant (e.g., temperature and volume). Those variables being held constant define the 'ensemble'.

Essentials of Computational Chemistry, 2nd Edition Christopher J. Cramer
© 2004 John Wiley & Sons, Ltd ISBNs: 0-470-09181-9 (cased); 0-470-09182-7 (pbk)

We will delay a more detailed discussion of ensemble thermodynamics until Chapter 10; indeed, in this chapter we will make use of ensembles designed to render the operative equations as transparent as possible without much discussion of extensions to other ensembles. The point to be re-emphasized here is that the vast majority of *experimental* techniques measure molecular properties as averages – either time averages or ensemble averages or, most typically, both. Thus, we seek computational techniques capable of accurately reproducing these aspects of molecular behavior. In this chapter, we will consider Monte Carlo (MC) and molecular dynamics (MD) techniques for the simulation of real systems. Prior to discussing the details of computational algorithms, however, we need to briefly review some basic concepts from statistical mechanics.

3.2 Phase Space and Trajectories

The state of a classical system can be completely described by specifying the positions and momenta of all particles. Space being three-dimensional, each particle has associated with it six coordinates – a system of N particles is thus characterized by $6N$ coordinates. The $6N$-dimensional space defined by these coordinates is called the 'phase space' of the system. At any instant in time, the system occupies one point in phase space

$$\mathbf{X}' = (x_1, y_1, z_1, p_{x,1}, p_{y,1}, p_{z,1}, x_2, y_2, z_2, p_{x,2}, p_{y,2}, p_{z,2}, \ldots) \tag{3.1}$$

For ease of notation, the position coordinates and momentum coordinates are defined as

$$\mathbf{q} = (x_1, y_1, z_1, x_2, y_2, z_2, \ldots) \tag{3.2}$$

$$\mathbf{p} = (p_{x,1}, p_{y,1}, p_{z,1}, p_{x,2}, p_{y,2}, p_{z,2}, \ldots) \tag{3.3}$$

allowing us to write a (reordered) phase space point as

$$\mathbf{X} = (\mathbf{q}, \mathbf{p}) \tag{3.4}$$

Over time, a dynamical system maps out a 'trajectory' in phase space. The trajectory is the curve formed by the phase points the system passes through. We will return to consider this dynamic behavior in Section 3.2.2.

3.2.1 Properties as Ensemble Averages

Because phase space encompasses every possible state of a system, the average value of a property A at equilibrium (i.e., its expectation value) for a system having a constant temperature, volume, and number of particles can be written as an integral over phase space

$$\langle A \rangle = \iint A(\mathbf{q}, \mathbf{p}) P(\mathbf{q}, \mathbf{p}) d\mathbf{q} d\mathbf{p} \tag{3.5}$$

where P is the probability of being at a particular phase point. From statistical mechanics, we know that this probability depends on the energy associated with the phase point according to

$$P(\mathbf{q},\mathbf{p}) = Q^{-1}e^{-E(\mathbf{q},\mathbf{p})/k_{\mathrm{B}}T} \tag{3.6}$$

where E is the total energy (the sum of kinetic and potential energies depending on $\mathbf{p}$ and $\mathbf{q}$, respectively) k_{B} is Boltzmann's constant, T is the temperature, and Q is the system partition function

$$Q = \int\int e^{-E(\mathbf{q},\mathbf{p})/k_{\mathrm{B}}T}d\mathbf{q}d\mathbf{p} \tag{3.7}$$

which may be thought of as the normalization constant for P.

How might one go about evaluating Eq. (3.5)? In a complex system, the integrands of Eqs. (3.5) and (3.7) are unlikely to allow for analytic solutions, and one must perforce evaluate the integrals numerically. The numerical evaluation of an integral is, in the abstract, straightforward. One determines the value of the integrand at some finite number of points, fits those values to some function that *is* integrable, and integrates the latter function. With an increasing number of points, one should observe this process to converge to a particular value (assuming the original integral is finite) and one ceases to take more points after a certain tolerance threshold has been reached.

However, one must remember just how vast phase space is. Imagine that one has only a very modest goal: One will take only a single phase point from each 'hyper-octant' of phase space. That is, one wants all possible combinations of signs for all of the coordinates. Since each coordinate can take on two values (negative or positive), there are 2^{6N} such points. Thus, in a system having $N = 100$ particles (which is a very small system, after all) one would need to evaluate A and E at 4.15×10^{180} points! Such a process might be rather time consuming ...

The key to making this evaluation more tractable is to recognize that phase space is, for the most part, a wasteland. That is, there are enormous volumes characterized by energies that are far too high to be of any importance, e.g., regions where the positional coordinates of two different particles are such that they are substantially closer than van der Waals contact. From a mathematical standpoint, Eq. (3.6) shows that a high-energy phase point has a near-zero probability, and thus the integrand of Eq. (3.5) will also be near-zero (as long as property A does not go to infinity with increasing energy). As the integral of zero is zero, such a phase point contributes almost nothing to the property expectation value, and simply represents a waste of computational resources. So, what is needed in the evaluation of Eqs. (3.5) and (3.7) is some prescription for picking *important* (i.e., high-probability) points.

The MC method, described in Section 3.4, is a scheme designed to do exactly this in a pseudo-random fashion. Before we examine that method, however, we first consider a somewhat more intuitive way to sample 'useful' regions of phase space.

3.2.2 Properties as Time Averages of Trajectories

If we start a system at some 'reasonable' (i.e., low-energy) phase point, its energy-conserving evolution over time (i.e., its trajectory) seems *likely* to sample relevant regions of phase space.

Certainly, this is the picture most of us have in our heads when it comes to the behavior of a real system. In that case, a reasonable way to compute a property average simply involves computing the value of the property periodically at times t_i and assuming

$$\langle A \rangle = \frac{1}{M} \sum_{i}^{M} A(t_i) \tag{3.8}$$

where M is the number of times the property is sampled. In the limit of sampling continuously and following the trajectory indefinitely, this equation becomes

$$\langle A \rangle = \lim_{t \to \infty} \frac{1}{t} \int_{t_0}^{t_0+t} A(\tau) d\tau \tag{3.9}$$

The 'ergodic hypothesis' assumes Eq. (3.9) to be valid and independent of choice of t_0. It has been proven for a hard-sphere gas that Eqs. (3.5) and (3.9) are indeed equivalent (Ford 1973). No such proof is available for more realistic systems, but a large body of empirical evidence suggests that the ergodic hypothesis is valid in most molecular simulations.

This point being made, we have not yet provided a description of how to 'follow' a phase-space trajectory. This is the subject of molecular dynamics, upon which we now focus.

3.3 Molecular Dynamics

One interesting property of a phase point that has not yet been emphasized is that, since it is defined by the positions and momenta of all particles, it *determines* the location of the next phase point in the absence of outside forces acting upon the system. The word 'next' is used loosely, since the trajectory is a continuous curve of phase points (i.e., between any two points can be found another point) – a more rigorous statement is that the forward trajectory is completely determined by the initial phase point. Moreover, since time-independent Hamiltonians are necessarily invariant to time reversal, a single phase point completely determines a full trajectory. As a result, phase space trajectories cannot cross themselves (since there would then be two *different* points leading away (in both time directions) from a single point of intersection). To illuminate further some of the issues involved in following a trajectory, it is helpful to begin with an example.

3.3.1 Harmonic Oscillator Trajectories

Consider a one-dimensional classical harmonic oscillator (Figure 3.1). Phase space in this case has only two dimensions, position and momentum, and we will define the origin of this phase space to correspond to the ball of mass m being at rest (i.e., zero momentum) with the spring at its equilibrium length. This phase point represents a stationary state of the system. Now consider the dynamical behavior of the system starting from some point other than the origin. To be specific, we consider release of the ball at time t_0 from

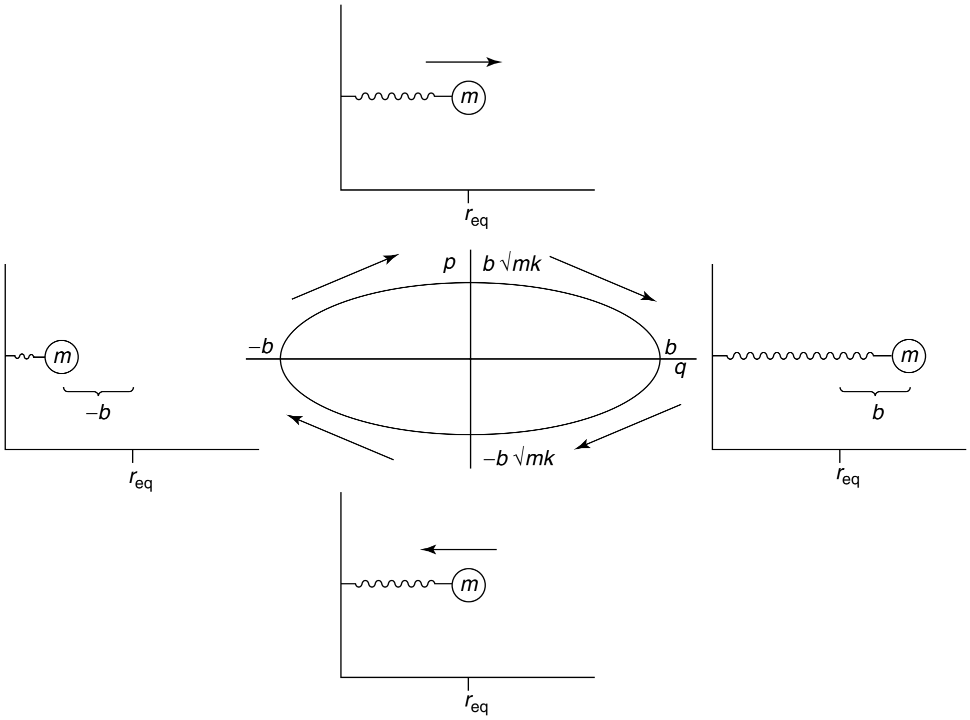

Figure 3.1 Phase-space trajectory (center) for a one-dimensional harmonic oscillator. As described in the text, at time zero the system is represented by the rightmost diagram ($q = b$, $p = 0$). The system evolves clockwise until it returns to the original point, with the period depending on the mass of the ball and the force constant of the spring

a position b length units displaced from equilibrium. The frictionless spring, characterized by force constant k, begins to contract, so that the position coordinate decreases. The momentum coordinate, which was 0 at t_0, also decreases (momentum is a vector quantity, and we here define negative momentum as movement towards the wall). As the spring passes through coordinate position 0 (the equilibrium length), the *magnitude* of the momentum reaches a maximum, and then decreases as the spring begins resisting further motion of the ball. Ultimately, the momentum drops to zero as the ball reaches position $-b$, and then grows increasingly positive as the ball moves back towards the coordinate origin. Again, after passing through the equilibrium length, the magnitude of the momentum begins to decrease, until the ball returns to the same point in phase space from which it began.

Let us consider the phase space trajectory traced out by this behavior beginning with the position vector. Over any arbitrary time interval, the relationship between two positions is

$$q(t_2) = q(t_1) + \int_{t_1}^{t_2} \frac{p(t)}{m} dt \tag{3.10}$$

where we have used the relationship between velocity and momentum

$$v = \frac{p}{m} \tag{3.11}$$

Similarly, the relationship between two momentum vectors is

$$p(t_2) = p(t_1) + m\int_{t_1}^{t_2} a(t)dt \tag{3.12}$$

where a is the acceleration. Equations (3.10) and (3.12) are Newton's equations of motion. Now, we have from Newton's Second Law

$$a = \frac{F}{m} \tag{3.13}$$

where F is the force. Moreover, from Eq. (2.13), we have a relationship between force and the position derivative of the potential energy. The simple form of the potential energy expression for a harmonic oscillator [Eq. (2.2)] permits analytic solutions for Eqs. (3.10) and (3.12). Applying the appropriate boundary conditions for the example in Figure 3.1 we have

$$q(t) = b\cos\left(\sqrt{\frac{k}{m}}t\right) \tag{3.14}$$

and

$$p(t) = -b\sqrt{mk}\sin\left(\sqrt{\frac{k}{m}}t\right) \tag{3.15}$$

These equations map out the oval phase space trajectory depicted in the figure.

Certain aspects of this phase space trajectory merit attention. We noted above that a phase space trajectory cannot cross itself. However, it *can* be periodic, which is to say it can trace out the same path again and again; the harmonic oscillator example is periodic. Note that the complete set of *all* harmonic oscillator trajectories, which would completely fill the corresponding two-dimensional phase space, is composed of concentric ovals (concentric circles if we were to choose the momentum metric to be $(mk)^{-1/2}$ times the position metric). Thus, as required, these (periodic) trajectories do not cross one another.

3.3.2 Non-analytical Systems

For systems more complicated than the harmonic oscillator, it is almost never possible to write down analytical expressions for the position and momentum components of the phase space trajectory as a function of time. However, if we *approximate* Eqs. (3.10) and (3.12) as

$$\mathbf{q}(t + \Delta t) = \mathbf{q}(t) + \frac{\mathbf{p}(t)}{m}\Delta t \tag{3.16}$$

and

$$\mathbf{p}(t + \Delta t) = \mathbf{p}(t) + m\mathbf{a}(t)\Delta t \tag{3.17}$$

(this approximation, Euler's, being exact in the limit of $\Delta t \rightarrow 0$) we are offered a prescription for *simulating* a phase space trajectory. [Note that we have switched from the scalar notation of the one-dimensional harmonic oscillator example to a more general vector notation. Note also that although the approximations in Eqs. (3.16) and (3.17) are introduced here from Eqs. (3.10) and (3.12) and the definition of the definite integral, one can also derive Eqs. (3.16) and (3.17) as Taylor expansions of **q** and **p** truncated at first order; this is discussed in more detail below.]

Thus, given a set of initial positions and momenta, and a means for computing the forces acting on each particle at any instant (and thereby deriving the acceleration), we have a formalism for 'simulating' the true phase-space trajectory. In general, initial positions are determined by what a chemist thinks is 'reasonable' – a common technique is to build the system of interest and then energy minimize it partially (since one is interested in dynamical properties, there is no point in looking for an absolute minimum) using molecular mechanics. As for initial momenta, these are usually assigned randomly to each particle subject to a temperature constraint. The relationship between temperature and momentum is

$$T(t) = \frac{1}{(3N - n)k_\mathrm{B}} \sum_{i=1}^{N} \frac{|\mathbf{p}_i(t)|^2}{m_i} \tag{3.18}$$

where N is the total number of atoms, n is the number of constrained degrees of freedom (vide infra), and the momenta are relative to the reference frame defined by the motion of the center of mass of the system. A force field, as emphasized in the last chapter, is particularly well suited to computing the accelerations at each time step.

While the use of Eqs. (3.16) and (3.17) seems entirely straightforward, the finite time step introduces very real practical concerns. Figure 3.2 illustrates the variation of a single momentum coordinate of some arbitrary phase space trajectory, which is described by a smooth curve. When the acceleration is computed for a point on the true curve, it will be a vector tangent to the curve. If the curve is not a straight line, any mass-weighted step along the tangent (which is the process described by Eq. (3.17)) will necessarily result in a point *off* the true curve. There is no guarantee that computing the acceleration at this new point will lead to a step that ends in the vicinity of the true curve. Indeed, with each additional step, it is quite possible that we will move further and further away from the true trajectory, thereby ending up sampling non-useful regions of phase space. The problem is compounded for position coordinates, since the velocity vector being used is already only an estimate derived from Eq. (3.17), i.e., there is no guarantee that it will even be tangent to the true curve when a point on the true curve is taken. (The atomistic picture, for those finding the mathematical discussion opaque, is that if we move the atoms in a single direction over too long a time, we will begin to ram them into one another so that they are far closer than van der Waals contact. This will lead to huge repulsive forces, so that still larger atomic movements will occur over the next time step, until our system ultimately looks like a nuclear furnace, with

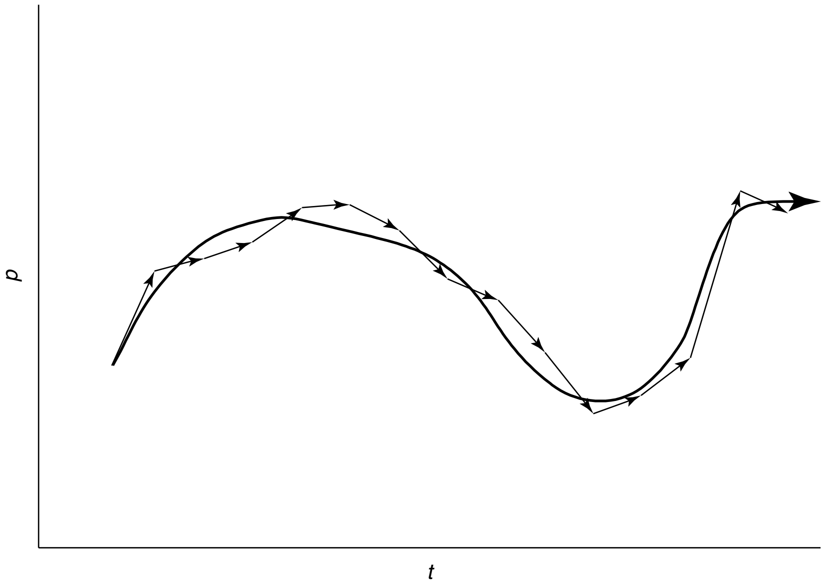

Figure 3.2 An actual phase-space trajectory (bold curve) and an approximate trajectory generated by repeated application of Eq. (3.17) (series of arrows representing individual time steps). Note that each propagation step has an identical Δt, but individual Δp values can be quite different. In the illustration, the approximate trajectory hews relatively closely to the actual one, but this will not be the case if too large a time step is used

atoms moving seemingly randomly. The very high energies of the various steps will preclude their contributing in a meaningful way to any property average.)

Of course, we know that in the limit of an infinitesimally small time step, we will recover Eqs. (3.10) and (3.12). But, since each time step requires a computation of all of the molecular forces (and, presumably, of the property we are interested in), which is computationally intensive, we do not want to take too *small* a time step, or we will not be able to propagate our trajectory for any chemically interesting length of time. What then is the optimal length for a time step that balances numerical stability with chemical utility? The general answer is that it should be at least one and preferably two orders of magnitude smaller than the fastest periodic motion within the system. To illustrate this, reconsider the 1-D harmonic oscillator example of Figure 3.1: if we estimate the first position of the mass after its release, given that the acceleration will be computed to be towards the wall, we will estimate the new position to be displaced in the negative direction. But, if we take too large a time step, i.e., we keep moving the mass towards the wall without ever accounting for the change in the acceleration of the spring with position, we might end up with the mass at a position more negative than $-b$. Indeed, we could end up with the mass behind the wall!

In a typical (classical) molecular system, the fastest motion is bond vibration which, for a heavy-atom–hydrogen bond has a period of about 10^{-14} s. Thus, for a system containing such bonds, an integration time step Δt should not much exceed 0.1 fs. This rather short time step means that modern, large-scale MD simulations (e.g., on biopolymers in a surrounding solvent) are rarely run for more than some 10 ns of simulation time (i.e., 10^7 computations of energies, forces, etc.) That many interesting phenomena occur on the microsecond timescale or longer (e.g., protein folding) represents a severe limitation to the application of MD to these phenomena. Methods to efficiently integrate the equations of motion over longer times are the subject of substantial modern research (see, for instance, Olender and Elber 1996; Grubmüller and Tavan 1998; Feenstra, Hess and Berendsen 1999).

3.3.3 Practical Issues in Propagation

Using Euler's approximation and taking integration steps in the direction of the tangent is a particularly simple integration approach, and as such is not particularly stable. Considerably more sophisticated integration schemes have been developed for propagating trajectories. If we restrict ourselves to consideration of the position coordinate, most of these schemes derive from approximate Taylor expansions in **r**, i.e., making use of

$$\mathbf{q}(t+\Delta t) = \mathbf{q}(t) + \mathbf{v}(t)\Delta t + \frac{1}{2!}\mathbf{a}(t)(\Delta t)^2 + \frac{1}{3!}\left.\frac{d^3\mathbf{q}(\tau)}{dt^3}\right|_{\tau=t}(\Delta t)^3 + \cdots \tag{3.19}$$

where we have used the abbreviations **v** and **a** for the first (velocity) and second (acceleration) time derivatives of the position vector **q**.

One such method, first used by Verlet (1967), considers the sum of the Taylor expansions corresponding to forward and reverse time steps Δt. In that sum, all odd-order derivatives disappear since the odd powers of Δt have opposite sign in the two Taylor expansions. Rearranging terms and truncating at second order (which is equivalent to truncating at third-order, since the third-order term has a coefficient of zero) yields

$$\mathbf{q}(t+\Delta t) = 2\mathbf{q}(t) - \mathbf{q}(t-\Delta t) + \mathbf{a}(t)(\Delta t)^2 \tag{3.20}$$

Thus, for any particle, each subsequent position is determined by the current position, the previous position, and the particle's acceleration (determined from the forces on the particle and Eq. (3.13)). For the very first step (for which no position $\mathbf{q}(t-\Delta t)$ is available) one might use Eqs. (3.16) and (3.17).

The Verlet scheme propagates the position vector with no reference to the particle velocities. Thus, it is particularly advantageous when the position coordinates of phase space are of more interest than the momentum coordinates, e.g., when one is interested in some property that is independent of momentum. However, often one wants to control the simulation temperature. This can be accomplished by scaling the particle velocities so that the temperature, as defined by Eq. (3.18), remains constant (or changes in some defined manner), as described in more detail in Section 3.6.3. To propagate the position and velocity vectors in a *coupled* fashion, a modification of Verlet's approach called the leapfrog algorithm has been proposed. In this case, Taylor expansions of the position vector truncated at second order (not third) about $t + \Delta t/2$ are employed, in particular

$$\mathbf{q}\left(t + \frac{1}{2}\Delta t + \frac{1}{2}\Delta t\right) = \mathbf{q}\left(t + \frac{1}{2}\Delta t\right) + \mathbf{v}\left(t + \frac{1}{2}\Delta t\right)\frac{1}{2}\Delta t + \frac{1}{2!}\mathbf{a}\left(t + \frac{1}{2}\Delta t\right)\left(\frac{1}{2}\Delta t\right)^2 \tag{3.21}$$

and

$$\mathbf{q}\left(t + \frac{1}{2}\Delta t - \frac{1}{2}\Delta t\right) = \mathbf{q}\left(t + \frac{1}{2}\Delta t\right) - \mathbf{v}\left(t + \frac{1}{2}\Delta t\right)\frac{1}{2}\Delta t + \frac{1}{2!}\mathbf{a}\left(t + \frac{1}{2}\Delta t\right)\left(\frac{1}{2}\Delta t\right)^2 \tag{3.22}$$

When Eq. (3.22) is subtracted from Eq. (3.21) one obtains

$$\mathbf{q}(t + \Delta t) = \mathbf{q}(t) + \mathbf{v}\left(t + \frac{1}{2}\Delta t\right)\Delta t \tag{3.23}$$

Similar expansions for $\mathbf{v}$ give

$$\mathbf{v}\left(t + \frac{1}{2}\Delta t\right) = \mathbf{v}\left(t - \frac{1}{2}\Delta t\right) + \mathbf{a}(t)\Delta t \tag{3.24}$$

Note that in the leapfrog method, position depends on the velocities as computed one-half time step out of phase, thus, scaling of the velocities can be accomplished to control temperature. Note also that no force-field calculations actually take place for the fractional time steps. Forces (and thus accelerations) in Eq. (3.24) are computed at integral time steps, half-time-step-forward velocities are computed therefrom, and these are then used in Eq. (3.23) to update the particle positions. The drawbacks of the leapfrog algorithm include ignoring third-order terms in the Taylor expansions and the half-time-step displacements of the position and velocity vectors – both of these features can contribute to decreased stability in numerical integration of the trajectory.

Considerably more stable numerical integration schemes are known for arbitrary trajectories, e.g., Runge–Kutta (Press *et al.* 1986) and Gear predictor-corrector (Gear 1971) methods. In Runge–Kutta methods, the gradient of a function is evaluated at a number of different intermediate points, determined iteratively from the gradient at the current point, prior to taking a step to a new trajectory point on the path; the 'order' of the method refers to the number of such intermediate evaluations. In Gear predictor-corrector algorithms, higher order terms in the Taylor expansion are used to predict steps along the trajectory, and then the actual particle accelerations computed for those points are compared to those that were predicted by the Taylor expansion. The differences between the actual and predicted values are used to correct the position of the point on the trajectory. While Runge–Kutta and Gear predictor-corrector algorithms enjoy very high stability, they find only limited use in MD simulations because of the high computational cost associated with computing multiple first derivatives, or higher-order derivatives, for every step along the trajectory.

A different method of increasing the time step without decreasing the numerical stability is to remove from the system those degrees of freedom having the highest frequency (assuming, of course, that any property being studied is independent of those degrees of freedom). Thus, if heavy-atom–hydrogen bonds are constrained to remain at a constant length, the next highest frequency motions will be heavy-atom–heavy-atom vibrations; these frequencies are typically a factor of 2–5 smaller in magnitude. While a factor of 2 is of only marginal utility, reducing the number of available degrees of freedom generally offers some savings in time and integration stability. So, when the system of interest is some solute immersed in a large bath of surrounding solvent molecules, it can be advantageous to freeze some or all of the degrees of freedom within the solvent molecules.

A commonly employed algorithm for eliminating these degrees of freedom is called SHAKE (Ryckaert, Ciccotti, and Berendsen 1977). In the context of the Verlet algorithm, the formalism for freezing bond lengths involves defining distance constraints d_{ij} between atoms i and j according to

$$\left|\mathbf{r}_{ij}\right|^2 - d_{ij}^2 = 0 \tag{3.25}$$

where $\mathbf{r}_{ij}$ is the instantaneous interatomic distance vector. The position constraints can be applied iteratively in the Verlet algorithm, for example, by first taking an *unconstrained* step according to Eq. (3.20). The constraints are then taken account of according to

$$\mathbf{r}_i(t + \Delta t) = \mathbf{r}_i^0(t + \Delta t) + \Delta\mathbf{r}_i(t) \tag{3.26}$$

where $\mathbf{r}_i^0(t + \Delta t)$ is the position after taking the unconstrained step, and $\Delta\mathbf{r}_i(t)$ is the displacement vector required to satisfy a set of coupled constraint equations. These equations are defined as

$$\Delta\mathbf{r}_i(t) = \frac{2(\Delta t)^2}{m_i}\sum_j \lambda_{ij}\mathbf{r}_{ij}(t) \tag{3.27}$$

where the Lagrange multipliers λ_{ij} are determined iteratively following substitution of Eqs. (3.25) and (3.26) into Eq. (3.20).

Finally, there are a number of entirely mundane (but still very worthwhile!) steps that can be taken to reduce the total computer time required for a MD simulation. As a single example, note that any force on a particle derived from a force-field non-bonded energy term is induced by some *other* particle (i.e., the potential is pairwise). Newton's Third Law tells us that

$$\mathbf{F}_{ij} = -\mathbf{F}_{ji} \tag{3.28}$$

so we can save roughly a factor of two in computing the non-bonded forces by only evaluating terms for $i < j$ and using Eq. (3.28) to establish the rest.

3.3.4 Stochastic Dynamics

When the point of a simulation is not to determine accurate thermodynamic information about an ensemble, but rather to watch the dynamical evolution of some particular system immersed in a larger system (e.g., a solute in a solvent), then significant computational savings can be had by modeling the larger system stochastically. That is, the explicit nature of the larger system is ignored, and its influence is made manifest by a continuum that interacts with the smaller system, typically with that influence including a degree of randomness.

In Langevin dynamics, the equation of motion for each particle is

$$\mathbf{a}(t) = -\zeta\mathbf{p}(t) + \frac{1}{m}[\mathbf{F}_{\text{intra}}(t) + \mathbf{F}_{\text{continuum}}(t)] \tag{3.29}$$

where the continuum is characterized by a microscopic friction coefficient, ζ, and a force, **F**, having one or more components (e.g., electrostatic and random collisional). Intramolecular forces are evaluated in the usual way from a force field. Propagation of position and momentum vectors proceeds in the usual fashion.

In Brownian dynamics, the momentum degrees of freedom are removed by arguing that for a system that does not change shape much over very long timescales (e.g., a molecule, even a fairly large one) the momentum of each particle can be approximated as zero relative to the rotating center of mass reference frame. Setting the l.h.s. of Eq. (3.29) to zero and integrating, we obtain the Brownian equation of motion

$$\mathbf{r}(t) = \mathbf{r}(t_0) + \frac{1}{\zeta}\int_{t_0}^{t}[\mathbf{F}_{\text{intra}}(\tau) + \mathbf{F}_{\text{continuum}}(\tau)]\mathrm{d}\tau \tag{3.30}$$

where we now propagate only the position vector.

Langevin and Brownian dynamics are very efficient because a potentially very large surrounding medium is represented by a simple continuum. Since the computational time required for an individual time step is thus reduced compared to a full deterministic MD simulation, much longer timescales can be accessed. This makes stochastic MD methods quite attractive for studying system properties with relaxation times longer than those that can be accessed with deterministic MD simulations. Of course, if those properties involve the surrounding medium in some explicit way (e.g., a radial distribution function involving solvent molecules, vide infra), then the stochastic MD approach is not an option.

3.4 Monte Carlo

3.4.1 Manipulation of Phase-space Integrals

If we consider the various MD methods presented above, the Langevin and Brownian dynamics schemes introduce an increasing degree of stochastic behavior. One may imagine carrying this stochastic approach to its logical extreme, in which event there are no equations of motion to integrate, but rather phase points for a system are selected entirely at random. As noted above, properties of the system can then be determined from Eq. (3.5), but the integration converges very slowly because most randomly chosen points will be in chemically meaningless regions of phase space.

One way to reduce the problem slightly is to recognize that for many properties A, the position and momentum dependences of A are separable. In that case, Eq. (3.5) can be written as

$$\langle A\rangle = \int A(\mathbf{q})\left[\int P(\mathbf{p},\mathbf{q})d\mathbf{p}\right]d\mathbf{q} + \int A(\mathbf{p})\left[\int P(\mathbf{p},\mathbf{q})d\mathbf{q}\right]d\mathbf{p} \tag{3.31}$$

Since the Hamiltonian is also separable, the integrals in brackets on the r.h.s. of Eq. (3.31) may be simplified and we write

$$\langle A\rangle = \int A(\mathbf{q})P(\mathbf{q})d\mathbf{q} + \int A(\mathbf{p})P(\mathbf{p})d\mathbf{p} \tag{3.32}$$

where $P(\mathbf{q})$ and $P(\mathbf{p})$ are probability functions analogous to Eq. (3.6) related only to the potential and kinetic energies, respectively. Thus, we reduce the problem of evaluating a $6N$-dimensional integral to the problem of evaluating two $3N$-dimensional integrals. Of course, if the property is independent of either the position or momentum variables, then there is only one $3N$-dimensional integral to evaluate.

Even with so large a simplification, however, the convergence of Eq. (3.32) for a realistically sized chemical system and a random selection of phase points is too slow to be useful. What is needed is a scheme to select important phase points in a biased fashion.

3.4.2 Metropolis Sampling

The most significant breakthrough in Monte Carlo modeling took place when Metropolis *et al.* (1953) described an approach where 'instead of choosing configurations randomly, then weighting them with $\exp(-E/k_{\mathrm{B}}T)$, we choose configurations with a probability $\exp(-E/k_{\mathrm{B}}T)$ and weight them evenly'.

For convenience, let us consider a property dependent only on position coordinates. Expressing the elegantly simple Metropolis idea mathematically, we have

$$\langle A\rangle = \frac{1}{X}\sum_{i=1}^{X} A(\mathbf{q}_i) \tag{3.33}$$

where X is the total number of points $\mathbf{q}$ sampled according to the Metropolis prescription. Note the remarkable similarity between Eq. (3.33) and Eq. (3.8). Equation (3.33) resembles an ensemble average from an MD trajectory where the order of the points, i.e., the temporal progression, has been lost. Not surprisingly, as time does not enter into the MC scheme, it is not possible to establish a time relationship between points.

The Metropolis prescription dictates that we choose points with a Boltzmann-weighted probability. The typical approach is to begin with some 'reasonable' configuration $\mathbf{q}_1$. The value of property A is computed as the first element of the sum in Eq. (3.33), and then $\mathbf{q}_1$ is randomly perturbed to give a new configuration $\mathbf{q}_2$. In the constant particle number, constant volume, constant temperature ensemble (*NVT* ensemble), the probability p of 'accepting' point $\mathbf{q}_2$ is

$$p = \min\left[1, \frac{\exp(-E_2/k_\mathrm{B}T)}{\exp(-E_1/k_\mathrm{B}T)}\right] \tag{3.34}$$

Thus, if the energy of point $\mathbf{q}_2$ is not higher than that of point $\mathbf{q}_1$, the point is always accepted. If the energy of the second point *is* higher than the first, p is compared to a random number z between 0 and 1, and the move is accepted if $p \geq z$. Accepting the point means that the value of A is calculated for that point, that value is added to the sum in Eq. (3.33), and the entire process is repeated. If second point is *not* accepted, then the first point 'repeats', i.e., the value of A computed for the first point is added to the sum in Eq. (3.33) a second time and a new, random perturbation is attempted. Such a sequence of phase points, where each new point depends only on the immediately preceding point, is called a 'Markov chain'.

The art of running an MC calculation lies in defining the perturbation step(s). If the steps are very, very small, then the volume of phase space sampled will increase only slowly over time, and the cost will be high in terms of computational resources. If the steps are too large, then the rejection rate will grow so high that again computational resources will be wasted by an inefficient sampling of phase space. Neither of these situations is desirable.

In practice, MC simulations are primarily applied to collections of molecules (e.g., molecular liquids and solutions). The perturbing step involves the choice of a single molecule, which is randomly translated and rotated in a Cartesian reference frame. If the molecule is flexible, its internal geometry is also randomly perturbed, typically in internal coordinates. The ranges on these various perturbations are adjusted such that 20–50% of attempted moves are accepted. Several million individual points are accumulated, as described in more detail in Section 3.6.4.

Note that in the MC methodology, only the energy of the system is computed at any given point. In MD, by contrast, forces are the fundamental variables. Pangali, Rao, and Berne (1978) have described a sampling scheme where forces are used to choose the direction(s) for molecular perturbations. Such a force-biased MC procedure leads to higher acceptance rates and greater statistical precision, but at the cost of increased computational resources.

3.5 Ensemble and Dynamical Property Examples

The range of properties that can be determined from simulation is obviously limited only by the imagination of the modeler. In this section, we will briefly discuss a few typical properties in a general sense. We will focus on structural and time-correlation properties, deferring thermodynamic properties to Chapters 10 and 12.

As a very simple example, consider the dipole moment of water. In the gas phase, this dipole moment is 1.85 D (Demaison, Hütner, and Tiemann 1982). What about water in liquid water? A zeroth order approach to answering this problem would be to create a molecular mechanics force field defining the water molecule (a sizable number exist) that gives the correct dipole moment for the isolated, gas-phase molecule at its equilibrium geometry, which moment is expressed as

$$\mu = \sum_{i=1}^{3} \frac{q_i}{\mathbf{r}_i} \tag{3.35}$$

where the sum runs over the one oxygen and two hydrogen atoms, q_i is the partial atomic charge assigned to atom i, and $\mathbf{r}_i$ is the position of atom i (since the water molecule has no net charge, the dipole moment is independent of the choice of origin for $\mathbf{r}$). In a liquid simulation (see Section 3.6.1 for more details on simulating condensed phases), the expectation value of the moment would be taken over *all* water molecules. Since the liquid is isotropic, we are not interested in the average vector, but rather the average magnitude of the vector, i.e.,

$$\langle|\mu|\rangle = \frac{1}{N}\sum_{n=1}^{N}\left|\sum_{i=1}^{3}\frac{q_{i,n}}{\mathbf{r}_{i,n}}\right| \tag{3.36}$$

where N is the number of water molecules in the liquid model. Then, to the extent that in liquid water the average geometry of a water molecule changes from its gas-phase equilibrium structure, the expectation value of the magnitude of the dipole moment will reflect this change. Note that Eq. (3.36) gives the ensemble average for a single snapshot of the system; that is, the 'ensemble' that is being averaged over is intrinsic to each phase point by virtue of their being multiple copies of the molecule of interest. By MD or MC methods, we would generate multiple snapshots, either as points along an MD trajectory or by MC perturbations, so that we would finally have

$$\langle|\mu|\rangle = \frac{1}{M{\bullet}N}\sum_{m=1}^{M}\sum_{n=1}^{N}\left|\sum_{i=1}^{3}\frac{q_{i,n,m}}{\mathbf{r}_{i,n,m}}\right| \tag{3.37}$$

where M is the total number of snapshots. [If we were considering the dipole moment of a solute molecule that was present in only one copy (i.e., a dilute solution), then the sum over N would disappear.]

Note that the expectation value compresses an enormous amount of information into a single value. A more complete picture of the moment would be a probability distribution, as depicted in Figure 3.3. In this analysis, the individual water dipole moment magnitudes (all $M{\bullet}N$ of them) are collected into bins spanning some range of dipole moments. The moments are then plotted either as a histogram of the bins or as a smooth curve reflecting the probability of being in an individual bin (i.e., equivalent to drawing the curve through the midpoint of the top of each histogram bar). The width of the bins is chosen so as to give maximum resolution to the lineshape of the curve without introducing statistical noise from underpopulation of individual bins.

Note that, although up to this point we have described the expectation value of A as though it were a scalar value, it is also possible that A is a function of some experimentally (and computationally) accessible variable, in which case we may legitimately ask about its expectation value at various points along the axis of its independent variable. A good

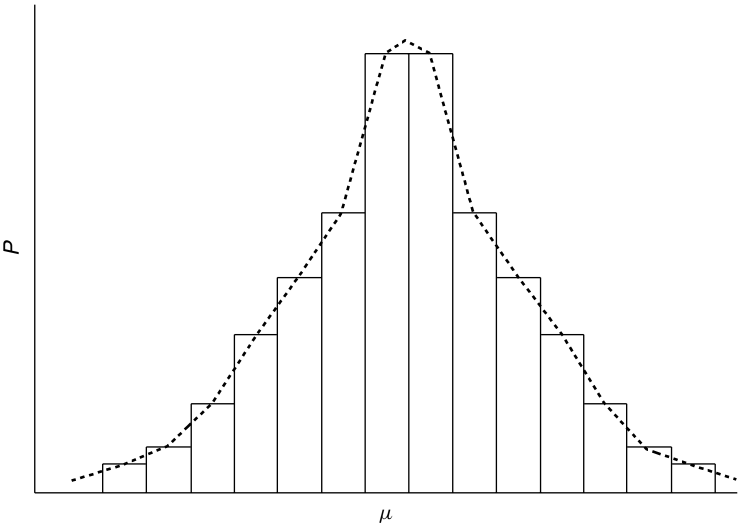

Figure 3.3 Hypothetical distribution of dipole moment magnitudes from a simulation of liquid water. The dashed curve is generated by connecting the tops of histogram bins whose height is dictated by the number of water molecules found to have dipole moments in the range spanned by the bin. Note that although the example is illustrated to be symmetric about a central value (which will thus necessarily be $\langle\mu\rangle$) this need not be the case

example of such a property is a radial distribution function (r.d.f.), which can be determined experimentally from X-ray or neutron diffraction measurements. The r.d.f. for two atoms A and B in a spherical volume element is defined by

$$\frac{1}{V}g_{\mathrm{AB}}(r) = \frac{1}{N_{\mathrm{A}}\bullet N_{\mathrm{B}}}\left\langle \sum_{i=1}^{N_{\mathrm{A}}}\sum_{j=1}^{N_{\mathrm{B}}}\delta\left[r - r_{\mathrm{A}_i\mathrm{B}_j}\right]\right\rangle \tag{3.38}$$

where V is the volume, N is the total number of atoms of a given type within the volume element, δ is the Dirac delta function (the utility of which will become apparent momentarily), and r is radial distance. The double summation within the ensemble average effectively counts for each distance r the number of AB pairs separated by that distance. If we integrate over the full spherical volume, we obtain

$$\begin{aligned}\frac{1}{V}\int g_{\mathrm{AB}}(r)d\mathbf{r} &= \frac{1}{N_{\mathrm{A}}\bullet N_{\mathrm{B}}}\left\langle \sum_{i=1}^{N_{\mathrm{A}}}\sum_{j=1}^{N_{\mathrm{B}}}\int\delta\left[r - r_{\mathrm{A}_i\mathrm{B}_j}\right]d\mathbf{r}\right\rangle \\ &= 1\end{aligned} \tag{3.39}$$

where we have made use of the property of the Dirac delta that its integral is unity. As there are $N_{\mathrm{A}}\bullet N_{\mathrm{B}}$ contributions of unity to the quantity inside the ensemble average, the r.h.s. of Eq. (3.39) is 1, and we see that the $1/V$ term is effectively a normalization constant on g.

We may thus interpret the l.h.s. of Eq. (3.39) as a probability function. That is, we may express the probability of finding two atoms of A and B within some range Δr of distance r from one another as

$$P\{\mathrm{A}, \mathrm{B}, r, \Delta r\} = \frac{4\pi r^2}{V} g_{\mathrm{AB}}(r)\Delta r \tag{3.40}$$

where, in the limit of small Δr, we have approximated the integral as $g_{\mathrm{AB}}(r)$ times the volume of the thin spherical shell $4\pi r^2 \Delta r$.

Note that its contribution to the probability function makes certain limiting behaviors on $g_{\mathrm{AB}}(r)$ intuitively obvious. For instance, the function should go to zero very rapidly when r becomes less than the sum of the van der Waals radii of A and B. In addition, at very large r, the function should be independent of r in homogeneous media, like fluids, i.e., there should be an equal probability for any interatomic separation because the two atoms no longer influence one another's positions. In that case, we could move g outside the integral on the l.h.s. of Eq. (3.39), and then the normalization makes it apparent that $g = 1$ under such conditions. Values other than 1 thus indicate some kind of structuring in a medium – values greater than 1 indicate preferred locations for surrounding atoms (e.g., a solvation shell) while values below 1 indicate underpopulated regions. A typical example of a liquid solution r.d.f. is shown in Figure 3.4. Note that with increasing order, e.g., on passing from a liquid to a solid phase, the peaks in g become increasingly narrow and the valleys increasingly wide and near zero, until in the limit of a motionless, perfect crystal, g would be a spectrum of Dirac δ functions positioned at the lattice spacings of the crystal.

It often happens that we consider one of our atoms A or B to be privileged, e.g., A might be a sodium ion and B the oxygen atom of a water and our interests might focus

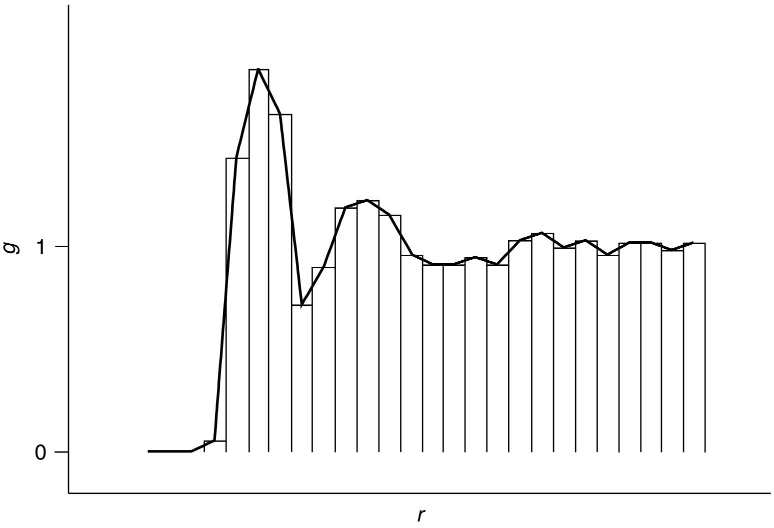

Figure 3.4 A radial distribution function showing preferred ($g > 1$) and disfavored ($g < 1$) interparticle distances. Random fluctuation about $g = 1$ is observed at large r

on describing the solvation structure of water about sodium ions in general. Then, we can define the total number of oxygen atoms n_{B} within some distance range about *any* sodium ion (atom A) as

$$n_{\mathrm{B}}\{r, \Delta r\} = N_{\mathrm{B}} P\{\mathrm{A}, \mathrm{B}, r, \Delta r\} \tag{3.41}$$

We may then use Eq. (3.40) to write

$$n_{\mathrm{B}}\{r, \Delta r\} = 4\pi r^2 \rho_{\mathrm{B}} g_{\mathrm{AB}}(r) \Delta r \tag{3.42}$$

where ρ_{B} is the number density of B in the total spherical volume. Thus, if instead of $g_{\mathrm{AB}}(r)$ we plot $4\pi r^2 \rho_{\mathrm{B}} g_{\mathrm{AB}}(r)$, then the area under the latter curve provides the number of molecules of B for arbitrary choices of r and Δr. Such an integration is typically performed for the distinct peaks in $g(r)$ so as to determine the number of molecules in the first, second, and possibly higher solvation shells or the number of nearest neighbors, next-nearest neighbors, etc., in a solid.

Determining $g(r)$ from a simulation involves a procedure quite similar to that described above for determining the continuous distribution of a scalar property. For each snapshot of an MD or MC simulation, all A–B distances are computed, and each occurrence is added to the appropriate bin of a histogram running from $r = 0$ to the maximum radius for the system (e.g., one half the narrowest box dimension under periodic boundary conditions, vide infra). Normalization now requires taking account not only of the total number of atoms A and B, but also the number of snapshots, i.e.,

$$g_{\mathrm{AB}}(r) = \frac{V}{4\pi r^2 \Delta r M N_{\mathrm{A}} N_{\mathrm{B}}} \sum_{m=1}^{M} \sum_{i=1}^{N_{\mathrm{A}}} \sum_{j=1}^{N_{\mathrm{B}}} Q_m\left(r; r_{\mathrm{A}_i\mathrm{B}_j}\right) \tag{3.43}$$

where Δr is the width of a histogram bin, M is the total number of snapshots, and Q_m is the counting function

$$Q\left(r; r_{\mathrm{A}_i\mathrm{B}_j}\right) = \begin{cases} 1 & \text{if } r - \Delta r/2 \le r_{\mathrm{A}_i\mathrm{B}_j} < r + \Delta r/2 \\ 0 & \text{otherwise} \end{cases} \tag{3.44}$$

for snapshot m.

The final class of dynamical properties we will consider are those defined by time-dependent autocorrelation functions. Such a function is defined by

$$C(t) = \langle a(t_0) a(t_0 + t) \rangle_{t_0} \tag{3.45}$$

where the ensemble average runs over *time* snapshots, and hence can only be determined from MD, *not* MC. Implicit in Eq. (3.45) is the assumption that C does not depend on the value of t_0 (since the ensemble average is over different choices of this quantity), and this will only be true for a system at equilibrium. The autocorrelation function provides a measure of the degree to which the value of property a at one time influences the value at a later time. An autocorrelation function attains its maximum value for a time delay of zero (i.e., no time delay at all), and this quantity, $\langle a^2 \rangle$ (which *can* be determined from MC simulations since no time correlation is involved) may be regarded as a normalization constant.

Now let us consider the behavior of C for long time delays. In a system where property a is not periodic in time, like a typical chemical system subject to effectively random thermal fluctuations, two measurements separated by a sufficiently long delay time should be completely uncorrelated. If two properties x and y are uncorrelated, then $\langle xy \rangle$ is equal to $\langle x \rangle \langle y \rangle$, so at long times C decays to $\langle a \rangle^2$.

While notationally burdensome, the discussion above makes it somewhat more intuitive to consider a reduced autocorrelation function defined by

$$\hat{C}(t) = \frac{\langle [a(t_0) - \langle a \rangle][a(t_0 + t) - \langle a \rangle] \rangle_{t_0}}{\langle [a - \langle a \rangle]^2 \rangle} \tag{3.46}$$

which is normalized and, because the arguments in brackets fluctuate about their mean (and thus have individual expectation values of zero) decays to zero at long delay times. Example autocorrelation plots are provided in Figure 3.5. The curves can be fit to analytic expressions to determine characteristic decay times. For example, the characteristic decay time for an autocorrelation curve that can be fit to $\exp(-\zeta t)$ is ζ^{-1} time units.

Different properties have different characteristic decay times, and these decay times can be quite helpful in deciding how long to run a particular MD simulation. Since the point of a simulation is usually to obtain a statistically meaningful sample, one does not want to compute an average over a time shorter than several multiples of the characteristic decay time.

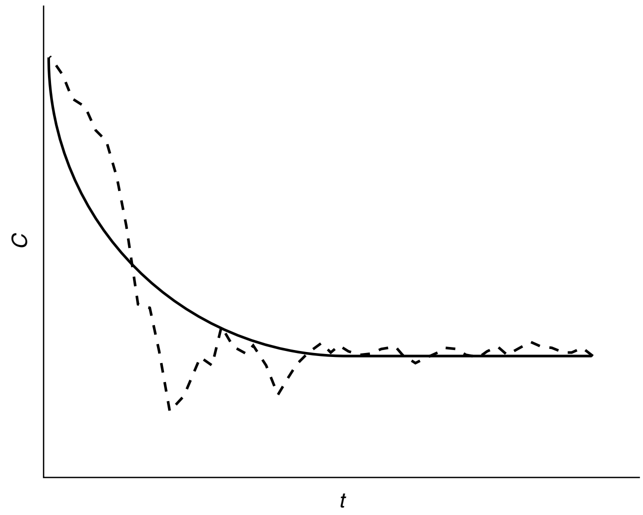

Figure 3.5 Two different autocorrelation functions. The solid curve is for a property that shows no significant statistical noise and appears to be well characterized by a single decay time. The dashed curve is quite noisy and, at least initially, shows a slower decay behavior. In the absence of a very long sample, decay times can depend on the total time sampled as well

As for the properties themselves, there are many chemically useful autocorrelation functions. For instance, particle position or velocity autocorrelation functions can be used to determine diffusion coefficients (Ernst, Hauge, and van Leeuwen 1971), stress autocorrelation functions can be used to determine shear viscosities (Haile 1992), and dipole autocorrelation functions are related to vibrational (infrared) spectra as their reverse Fourier transforms (Berens and Wilson 1981). There are also many useful correlation functions between two *different* variables (Zwanzig 1965). A more detailed discussion, however, is beyond the scope of this text.

3.6 Key Details in Formalism

The details of MC and MD methods laid out thus far can realistically be applied in a rigorous fashion only to systems that are too small to meaningfully represent actual chemical systems. In order to extend the technology in such a way as to make it useful for interpreting (or predicting) chemical phenomena, a few other approximations, or practical simplifications, are often employed. This is particularly true for the modeling of condensed phases, which are macroscopic in character.

3.6.1 Cutoffs and Boundary Conditions

As a spherical system increases in size, its volume grows as the cube of the radius while its surface grows as the square. Thus, in a truly macroscopic system, surface effects may play little role in the chemistry under study (there are, of course, exceptions to this). However, in a typical simulation, computational resources inevitably constrain the size of the system to be so small that surface effects may *dominate* the system properties. Put more succinctly, the modeling of a cluster may not tell one much about the behavior of a macroscopic system. This is particularly true when electrostatic interactions are important, since the energy associated with these interactions has an r^{-1} dependence.

One approach to avoid cluster artifacts is the use of ‘periodic boundary conditions’ (PBCs). Under PBCs, the system being modeled is assumed to be a unit cell in some ideal crystal (e.g., cubic or orthorhombic, see Theodorouo and Suter 1985). In practice, cut-off distances are usually employed in evaluating non-bonded interactions, so the simulation cell need be surrounded by only one set of nearest neighbors, as illustrated in Figure 3.6. If the trajectory of an individual atom (or a MC move of that atom) takes it outside the boundary of the simulation cell in any one or more cell coordinates, its image simultaneously enters the simulation cell from the point related to the exit location by lattice symmetry.

Thus, PBCs function to preserve mass, particle number, and, it can be shown, total energy in the simulation cell. In an MD simulation, PBCs also conserve linear momentum; since linear momentum is *not* conserved in real contained systems, where container walls disrupt the property, this is equivalent to reducing the number of degrees of freedom by 3. However, this effect on system properties is typically negligible for systems of over 100 atoms. Obviously, PBCs do *not* conserve angular momentum in the simulation cell of an MD simulation, but over time the movement of atoms in and out of each wall of the cell will be such that

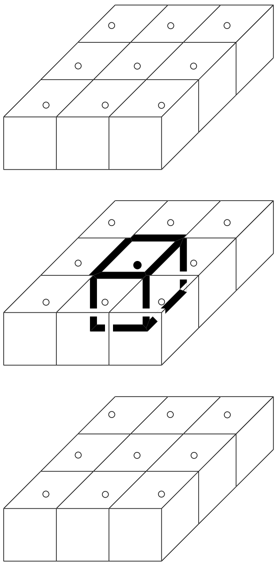

Figure 3.6 Exploded view of a cubic simulation cell surrounded by the 26 periodic images generated by PBCs. If the solid particle translates to a position that is outside the simulation cell, one of its periodic images, represented by open particles, will translate in

fluctuations will take place about a well-defined average. The key aspect of imposing PBCs is that no molecule within the simulation cell sees 'vacuum' within the range of its interaction cutoffs, and thus surface artifacts are avoided. Other artifacts associated with periodicity may be introduced, particularly with respect to correlation times in dynamics simulations (Berne and Harp 1970; Bergdorf, Peter, and Hunenberger 2003), but these can in principle be eliminated by moving to larger and larger simulation cells so that periodicity takes place over longer and longer length scales.

Of course, concerns about periodicity only relate to systems that are *not* periodic. The discussion above pertains primarily to the simulations of liquids, or solutes in liquid solutions, where PBCs are a useful approximation that helps to model solvation phenomena more realistically than would be the case for a small cluster. If the system truly is periodic, e.g., a zeolite crystal, then PBCs are integral to the model. Moreover, imposing PBCs can provide certain advantages in a simulation. For instance, Ewald summation, which accounts for electrostatic interactions to infinite length as discussed in Chapter 2, can only be carried out within the context of PBCs.

An obvious question with respect to PBCs is how large the simulation cell should be. The simple answer is that all cell dimensions must be at least as large as the largest cut-off length employed in the simulation. Otherwise, some interatomic interactions would be at least double counted (once within the cell, and once with an image outside of the cell). In practice, one would like to go well beyond this minimum requirement if the system being modeled is supposedly homogeneous and non-periodic. Thus, for instance, if one is modeling a large, dilute solute in a solvent (e.g., a biomolecule), a good choice for cell size might be the dimensions of the molecule plus at least twice the largest cut-off distance. Thus, no two solute molecules interact with one another nor does any solvent molecule see two copies of the solute. (Note, however, that this does not change the fundamentally periodic nature of the system; it simply increases the number of molecules over which it is made manifest.)

As already noted in Chapter 2, for electrostatic interactions, Ewald sums are generally to be preferred over cut-offs because of the long-range nature of the interactions. For van der Waals type terms, cut-offs do not introduce significant artifacts provided they are reasonably large (typically 8–12 Å).

Because of the cost of computing interatomic distances, the evaluation of non-bonded terms in MD is often handled with the aid of a 'pairlist', which holds in memory all pairs of atoms within a given distance of one another. The pairlist is updated periodically, but less often than every MD step. Note that a particular virtue of MC compared to MD is that the only changes in the potential energy are those associated with a moved particle – all other interactions remain constant. This makes evaluation of the total energy a much simpler process in MC.

3.6.2 Polarization

As noted in Chapter 2, computation of charge–charge (or dipole–dipole) terms is a particularly efficient means to evaluate electrostatic interactions because it is pairwise additive. However, a more realistic picture of an actual physical system is one that takes into account the polarization of the system. Thus, different regions in a simulation (e.g., different functional groups, or different atoms) will be characterized by different local polarizabilities, and the local charge moments, by adjusting in an iterative fashion to their mutual interactions, introduce many-body effects into a simulation.

Simulations including polarizability, either only on solvent molecules or on all atoms, have begun to appear with greater frequency as computational resources have grown larger. In addition, significant efforts have gone into introducing polarizability into force fields in a general way by replacing fixed atomic charges with charges that fluctuate based on local environment (Winn, Ferenczy and Reynolds 1999; Banks *et al.* 1999), thereby preserving the simplicity of a pairwise interaction potential. However, it is not yet clear that the greater 'realism' afforded by a polarizable model greatly improves the accuracy of simulations. There are certain instances where polarizable force fields seem better suited to the modeling problem. For instance, Dang *et al.* (1991) have emphasized that the solvation of ions, because of their concentrated charge, is more realistically accounted for when surrounding solvent molecules are polarizable and Soetens *et al.* (1997) have emphasized its importance in the computation of ion–ion interaction potentials for the case of two guanidinium ions in water.

In general, however, the majority of properties do not yet seem to be more accurately predicted by polarizable models than by unpolarizable ones, provided adequate care is taken in the parameterization process. Of course, if one wishes to examine issues associated with polarization, it must necessarily be included in the model. In the area of solvents, for instance, Bernardo *et al.* (1994) and Zhu and Wong (1994) have carefully studied the properties of polarizable water models. In addition, Gao, Habibollazadeh, and Shao (1995) have developed alcohol force fields reproducing the thermodynamic properties of these species as liquids with a high degree of accuracy, and have computed the polarization contribution to the total energy of the liquids to be 10–20%.

However, the typically high cost of including polarization is not attractive. Jorgensen has argued against the utility of including polarization in most instances, and has shown that bulk liquid properties can be equally well reproduced by fixed-charge force fields given proper care in the parameterization process (see, for instance, Mahoney and Jorgensen 2000). A particularly interesting example is provided by the simple amines ammonia, methylamine, dimethylamine, and trimethylamine. In the gas phase, the basicity of these species increases with increasing methylation in the expected fashion. In water, however, solvation effects compete with intrinsic basicity so that the four amines span a fairly narrow range of basicity, with methylamine being the most basic and trimethylamine and ammonia the least. Many models of solvation (see Chapters 11 and 12 for more details on solvation models) have been applied to this problem, and the failure of essentially all of them to correctly predict the basicity ordering led to the suggestion that in the case of explicit models, the failure derived from the use of non-polarizable force fields. Rizzo and Jorgensen (1999), however, parameterized non-polarizable classical models for the four amines that accurately reproduced their liquid properties and then showed that they further predicted the correct basicity ordering in aqueous simulations, thereby refuting the prior suggestion. [As a point of philosophy, the above example provides a nice illustration that a model's failure to accurately predict a particular quantity does *not* necessarily imply that a more expensive model needs to be developed – sometimes all that is required is a more careful parameterization of the existing model.] At least for the moment, then, it appears that errors associated with other aspects of simulation technology typically continue to be as large or larger than any errors introduced by use of non-polarizable force fields, so the use of such force fields in everyday simulations seems likely to continue for some time.

3.6.3 Control of System Variables

Our discussion of MD above was for the 'typical' MD ensemble, which holds particle number, system volume, and total energy constant – the *NVE* or 'microcanonical' ensemble. Often, however, there are other thermodynamic variables that one would prefer to hold constant, e.g., temperature. As temperature is related to the total kinetic energy of the system (if it is at equilibrium), as detailed in Eq. (3.18), one could in principle scale the velocities of each particle at each step to maintain a constant temperature. In practice, this is undesirable because the adjustment of the velocities, occasionally by fairly significant scaling factors, causes the trajectories to be no longer Newtonian. Properties computed over such trajectories are less likely to be reliable. An alternative method, known as Berendsen coupling (Berendsen *et al.* 1984), slows the scaling process by envisioning a connection between the system and a surrounding bath that is at a constant temperature T_0. Scaling of each particle velocity is accomplished by including a dissipative Langevin force in the equations of motion according to

$$\mathbf{a}_i(t) = \frac{\mathbf{F}_i(t)}{m_i} + \frac{\mathbf{p}_i(t)}{m_i\tau}\left[\frac{T_0}{T(t)} - 1\right] \tag{3.47}$$

where $T(t)$ is the instantaneous temperature, and τ has units of time and is used to control the strength of the coupling. The larger the value of τ the smaller the perturbing force and the more slowly the system is scaled to T_0 (i.e., τ is an effective relaxation time).

Note that, to start an MD simulation, one must necessarily generate an initial snapshot. It is essentially impossible for a chemist to simply 'draw' a large system that actually corresponds to a high-probability region of phase space. Thus, most MD simulations begin with a so-called 'equilibration' period, during which time the system is allowed to relax to a realistic configuration, after which point the 'production' portion of the simulation begins, and property averages are accumulated. A temperature coupling is often used during the equilibration period so that the temperature begins very low (near zero) and eventually ramps up to the desired system temperature for the production phase. This has the effect of damping particle movement early on in the equilibration (when there are presumably very large forces from a poor initial guess at the geometry).

In practice, equilibration protocols can be rather involved. Large portions of the system may be held frozen initially while subregions are relaxed. Ultimately, the entire system is relaxed (i.e., all the degrees of freedom that are being allowed to vary) and, once the equilibration temperature has reached the desired average value, one can begin to collect statistics.

With respect to other thermodynamic variables, many experimental systems are not held at constant volume, but instead at constant pressure. Assuming ideal gas statistical mechanics and pairwise additive forces, pressure P can be computed as

$$P(t) = \frac{1}{V(t)}\left[Nk_{\mathrm{B}}T(t) + \frac{1}{3}\sum_{i}^{N}\sum_{j>1}^{N} F_{ij}r_{ij}\right] \tag{3.48}$$

where V is the volume, N is the number of particles, F and r are the forces and distances between particles, respectively. To adjust the pressure in a simulation, what is typically modified is the volume. This is accomplished by scaling the location of the particles, i.e., changing the size of the unit cell in a system with PBCs. The scaling can be accomplished in a fashion exactly analogous with Eq. (3.47) (Andersen 1980).

An alternative coupling scheme for temperature and pressure, the Nosé–Hoover scheme, adds new, independent variables that control these quantities to the simulation (Nosé 1984; Hoover 1985). These variables are then propagated along with the position and momentum variables.

In MC methods, the 'natural' ensemble is the *NVT* ensemble. Carrying out MC simulations in other ensembles simply requires that the probabilities computed for steps to be accepted or rejected reflect dependence on factors other than the internal energy. Thus, if we wish to maintain constant pressure instead of constant volume, we can treat volume as a variable (again, by scaling the particle coordinates, which is equivalent to expanding or contracting the unit cell in a system described by PBCs). However, in the *NPT* ensemble, the deterministic thermodynamic variable is no longer the internal energy, but the enthalpy (i.e., $E + PV$) and, moreover, we must account for the effect of a change in system volume (three dimensions) on the total volume of phase space ($3N$ dimensions for position) since probability is related to phase-space volume. Thus, in the *NPT* ensemble, the probability for accepting a new point 2 over an old point 1 becomes

$$p = \min\left\{1, \frac{V_2^N \exp[-(E_2 + PV_2)/k_B T]}{V_1^N \exp[-(E_1 + PV_1)/k_B T]}\right\} \tag{3.49}$$

(lower case 'p' is used here for probability to avoid confusion with upper case 'P' for pressure).

The choices of how often to scale the system volume, and by what range of factors, obviously influence acceptance ratios and are adjusted in much the same manner as geometric variables to maintain a good level of sampling efficiency. Other ensembles, or sampling schemes other than those using Cartesian coordinates, require analogous modifications to properly account for changes in phase space volume.

Just as with MD methods, MC simulations require an initial equilibration period so that property averages are not biased by very poor initial values. Typically various property values are monitored to assess whether they appear to have achieved a reasonable level of convergence prior to proceeding to production statistics. Yang, Bitetti-Putzer, and Karplus (2004) have offered the rather clever suggestion that the equilibration period can be defined by analyzing the convergence of property values starting from the *end* of the simulation, i.e., the time arrow of the simulation is reversed in the analysis. When an individual property value begins to depart from the value associated with the originally late, and presumably converged, portion of the trajectory, it is assumed that the originally early region of the trajectory should not be included in the overall statistics as it was most probably associated with equilibration. We now focus more closely on this issue.

3.6.4 Simulation Convergence

Convergence is defined as the acquisition of a sufficient number of phase points, through either MC or MD methods, to thoroughly sample phase space in a proper, Boltzmann-weighted fashion, i.e., the sampling is ergodic. While simple to define, convergence is *impossible* to prove, and this is either terribly worrisome or terribly liberating, depending on one's personal outlook.

To be more clear, we should separate the analysis of convergence into what might be termed 'statistical' and 'chemical' components. The former tends to be more tractable than the latter. Statistical convergence can be operatively defined as being *likely* to have been achieved when the average values for all properties of interest appear to remain roughly constant with increased sampling. In the literature, it is fairly standard to provide one or two plots of some particular properties as a function of time so that readers can agree that, to their eyes, the plots appear to have flattened out and settled on a particular value. For instance, in the simulation of macromolecules, the root-mean-square deviation (RMSD) of the simulation structure from an X-ray or NMR structure is often monitored. The RMSD for a particular snapshot is defined as

$$\mathrm{RMSD} = \sqrt{\frac{\sum_{i=1}^{N}(r_{i,\mathrm{sim}} - r_{i,\mathrm{expt}})^2}{N}} \tag{3.50}$$

where N is the number of atoms in the macromolecule, and the positions r are determined in a coordinate system having the center of mass at the origin and aligning the principle moments of inertia along the Cartesian axes (i.e., the simulated and experimental structures are best aligned prior to computing the RMSD). Monitoring the RMSD serves the dual purpose of providing a particular property whose convergence can be assessed and also of offering a quantitative measure of how ‘close’ the simulated structure is to the experimentally determined one. When no experimental data are available for comparison, the RMSD is typically computed using as a reference either the initial structure or the average simulated structure. A typical RMSD plot is provided in Figure 3.7.

[Note that the information content in Figure 3.7 is often boiled down, when reported in the literature, to a single number, namely ⟨RMSD⟩. However, the magnitude of the fluctuation about the mean, which can be quantified by the standard deviation, is also an important quantity, and should be reported wherever possible. This is true for all expectation values

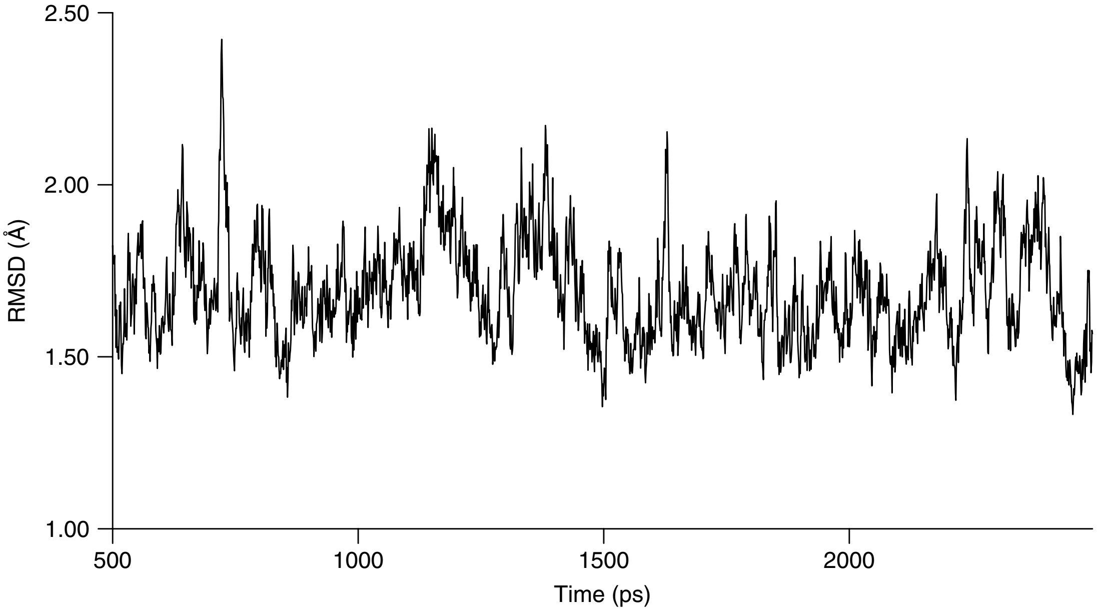

Figure 3.7 RMSD plot after 500 ps of equilibration for a solvated tRNA microhelix relative to its initial structure (Nagan *et al.* 1999)

derived from simulation. The standard deviation can be interpreted as a combination of the statistical noise (deriving from the limitations of the method) and the thermal noise (deriving from the 'correct' physical nature of the system). Considerably more refined methods of error analysis for average values from simulations have been promulgated (Smith and Wells 1984; Straatsma, Berendsen and Stam 1986; Kolafa 1986; Flyvberg and Petersen 1989).]

A more detailed decomposition of macromolecular dynamics that can be used not only for assessing convergence but also for other purposes is principal components analysis (PCA), sometimes also called essential dynamics (Wlodek *et al.* 1997). In PCA the positional covariance matrix **C** is calculated for a given trajectory after removal of rotational and translational motion, i.e., after best overlaying all structures. Given M snapshots of an N atom macromolecule, **C** is a $3N \times 3N$ matrix with elements

$$C_{ij} = \frac{1}{M}\sum_{k=1}^{M}(q_{i,k} - \langle q_i \rangle)\,(q_{j,k} - \langle q_j \rangle) \tag{3.51}$$

where $q_{i,k}$ is the value for snapshot k of the ith positional coordinate (x, y, or z coordinate for one of the N atoms), and $\langle q_i \rangle$ indicates the average of that coordinate over all snapshots. Diagonalization of **C** provides a set of eigenvectors that describe the dynamic motions of the structure; the associated eigenvalues may be interpreted as weights indicating the degree to which each mode contributes to the full dynamics.

Note that the eigenvectors of **C** comprise an orthogonal basis set for the macromolecular $3N$-dimensional space, but PCA creates them so as to capture as much structural dynamism as possible with each successive vector. Thus, the first PCA eigenvector may account for, say, 30 percent of the overall dynamical motion, the second a smaller portion, and so on. The key point here is that a surprisingly large fraction of the overall dynamics may be captured by a fairly small number of eigenvectors, each one of which may be thought of as being similar to a macromolecular vibrational mode. Thus, for example, Sherer and Cramer (2002) found that the first three PCA modes for a set of related RNA tetradecamer double helices accounted for 68 percent of the total dynamics, and that these modes were well characterized as corresponding to conceptually simple twisting and bending motions of the helix (Figure 3.8 illustrates the dominant mode). Being able in this manner to project the total macromolecular motion into PCA spaces of small dimensionality can be very helpful in furthering chemical analysis of the dynamics.

Returning to the issue of convergence, as noted above the structure of each snapshot in a simulation can be described in the space of the PCA eigenvectors, there being a coefficient for each vector that is a coordinate value just as an x coordinate in three-dimensional Cartesian space is the coefficient of the **i** Cartesian basis vector (1,0,0). If a simulation has converged, the distribution of coefficient values sampled for each PCA eigenvector should be normal, i.e., varying as a Gaussian distribution about some mean value.

Yet another check of convergence in MD simulations, as alluded to in Section 3.5, is to ensure that the sampling length is longer than the autocorrelation decay time for a particular property by several multiples of that time. In practice, this analysis is performed with less regularity than is the simple monitoring of individual property values.

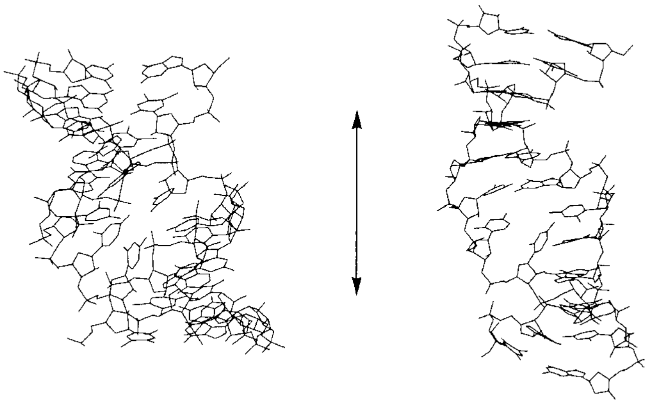

Figure 3.8 Twisted/compressed and untwisted/elongated double helices corresponding to minimal and maximal coefficient values for the corresponding PCA eigenvector.

It must be borne in mind, however, that the typical simulation lengths that can be achieved with modern hardware and software are very, very rarely in excess of 1 μs. It is thus quite possible that the simulation, although it appears to be converged with respect to the analyses noted above, is trapped in a metastable state having a lifetime in excess of 1 μs, and as a result the statistics are not meaningful to the true system at equilibrium. The only way to address this problem is either to continue the simulation for a longer time or to run one or more additional simulations with different starting conditions or both. Entirely separate trajectories are more likely to provide data that are statistically uncorrelated with the original, but they are also more expensive since equilibration periods are required prior to collecting production mode data.

Problems associated with statistical convergence and/or metastability are vexing ones, but more daunting still can be the issue of chemical convergence. This is probably best illustrated with an example. Imagine that one would like to simulate the structure of a protein in water at pH 7 and that the protein contains nine histidine residues. At pH 7, the protein could, in principle, exist in many different protonation states (i.e., speciation) since the pK_a of histidine is quite near 7. Occam's razor and a certain amount of biochemical experience suggest that, in fact, only one or two states are likely to be populated under biological conditions, but how to choose which one(s) for simulation, since most force fields will not allow for protonation/deprotonation to take place? If the wrong state is chosen, it may be possible to acquire very good statistical convergence for the associated region of phase space, but that region is statistically unimportant compared to other regions which were *not* sampled.

3.6.5 The Multiple Minima Problem

A related problem, and one that is commonly encountered, has to do with molecules possessing multiple conformations. Consider N-methylacetamide, which can exist in E and Z forms. The latter stereoisomer is favored over the former by about 3 kcal/mol, but the barrier to interconversion is in excess of 18 kcal/mol. Thus, a simulation of N-methylacetamide starting with the statistically less relevant E structure is highly unlikely ever to sample the Z form, either using MD (since the high barrier implies an isomerization rate that will be considerably slower than the simulation time) or MC (since with small steps, the probability of going so far uphill would be very low, while with large steps it might be possible for the isomers to interconvert, but the rejection rate would be enormous making the simulation intractable). A related example with similar issues has to do with modeling phase transfer by MC methods, e.g., the movement of a solute between two immiscible liquids, or of a molecule from the gas phase to the liquid phase. In each case, the likelihood of moving a molecule in its entirety is low.

A number of computational techniques have been proposed to address these limitations. The simplest approach conceptually, which can be applied to systems where all possible conformations can be readily enumerated, is to carry out simulations for each one and then weight the respective property averages according to the free energies of the conformers (means for estimating these free energies are discussed in Chapter 12). This approach is, of course, cumbersome when the number of conformers grows large. This growth can occur with startling rapidity. For example, 8, 18, 41, 121, and 12 513 distinct minima have been identified for cyclononane, -decane, -undecane, -dodecane, and -heptadecane, respectively (Weinberg and Wolfe 1994). And cycloalkanes are relatively simple molecules compared, say, to a protein, where the holy grail of conformational analysis is prediction of a properly folded structure from only sequence information. Nevertheless, fast heuristic methods continue to be developed to rapidly search low-energy conformational space for small to medium-sized molecules. For example, Smellie *et al.* (2003) have described an algorithm that performed well in generating collections of low-energy conformers for 97 000 drug-like molecules with an average time of less than 0.5 s per stereoisomer.

A different approach to the identification of multiple minima is to periodically heat the system to a very high temperature. Since most force fields do not allow bond-breaking to occur, high temperature simply has the effect of making conformational interconversions more likely. After a certain amount of time, the system is cooled again to the temperature of interest, and statistics are collected. In practice, this technique is often used for isolated molecules in the gas phase in the hope of finding a global minimum energy structure, in which case it is referred to as 'simulated annealing'. In condensed phases, it is difficult to converge the statistical weights of the different accessed conformers. Within the context of MC simulations, other techniques to force the system to jump between minimum-energy wells in a properly energy-weighted fashion have been proposed (see, for instance, Guarnieri and Still 1994; Senderowitz and Still 1998; Brown and Head-Gordon 2003).

An alternative to adjusting the temperature to help the system overcome high barriers is to artificially lower the barrier by adding an external potential energy term that is large and positive in regions where the 'normal' potential energy is large and negative (i.e., in the regions of minima). This summation effectively counterbalances the normal potential energy barrier. For instance, if the barrier is associated with a bond rotation, a so-called 'biasing potential' can be added such that the rotational potential becomes completely flat. The system can now sample freely over the entire range of possible rotations, but computed properties must be corrected for the proper free energy difference(s) in the absence of the biasing potential(s) (Straatsma, T. P.; McCammon, J. A.; Andricioaei and Straub 1996). In the absence of already knowing the shape of the PES, however, it may be rather difficult to construct a useful biasing potential *ab initio*. Laio and Parrinello (2002) have described a protocol whereby the biasing potential is history-dependent, filling in minima as it goes along in a coarse-grained space defined by collective coordinates. Collective coordinates have also been used by Jaqaman and Ortoleva (2002) to explore large-scale conformational changes in macromolecules more efficiently and by Müller, de Meijere, and Grubmüller (2002) to predict relative rates of unimolecular reactions.

Another method to artificially lower barrier heights in certain regions of phase space is to artificially expand that space by a single *extra* coordinate introduced for just that purpose–an idea analogous to the way catalysts lower barrier heights without affecting local minima (Stolovitzky and Berne 2000). In a related fashion, Nakamura (2002) has shown that barriers up to 3000 kcal mol^{-1} can be readily overcome simply by sampling in a logarithmically transformed energy space followed by correction of the resulting probability distribution.

An interesting alternative suggested by Verkhivker, Elber, and Nowak (1992) is to have multiple conformers present *simultaneously* in a 'single' molecule. In the so-called 'locally enhanced sampling' method, the molecule of interest is represented as a sum of different conformers, each contributing fractionally to the total force field energy expression. When combined with 'softened' potentials, Hornak and Simmerling (2003) have shown that this technology can be useful for crossing very high barriers associated with large geometric rearrangements.

Just as with statistical convergence, however, there can be no *guarantee* that any of the techniques above will provide a thermodynamically accurate sampling of phase space, even though on the timescale of the simulation various property values may *appear* to be converged. As with most theoretical modeling, then, it is best to assess the likely utility of the predictions from a simulation by first comparing to experimentally well-known quantities. When these are accurately reproduced, other predictions can be used with greater confidence. As a corollary, the modeling of systems for which few experimental data are available against which to compare is perilous.

3.7 Force Field Performance in Simulations

As discussed in Chapter 2, most force fields are validated based primarily on comparisons to small molecule data and moreover most comparisons involve what might be called static properties, i.e., structural or spectral data for computed fixed conformations. There are a few noteworthy exceptions: the OPLS and TraPPE force fields were, at least for molecular solvents, optimized to reproduce bulk solvent properties derived from simulations, e.g., density, boiling point, and dielectric constant. In most instances, however, one is left with the question of whether force fields optimized for small molecules or molecular fragments will perform with acceptable accuracy in large-scale simulations.

This question has been addressed with increasing frequency recently, and several useful comparisons of the quality of different force fields in particular simulations have appeared. The focus has been primarily on biomolecular simulations. Okur *et al.* (2003) assessed the abilities of the force fields of Cornell *et al.* and Wang, Cieplak, and Kollman (see Table 2.1) to predict correctly folded vs. misfolded protein structures; they found both force fields to suffer from a bias that predicts helical secondary structure to be anomalously too stable and suggested modifications to improve the more recent of the two force fields. Mu, Kosov, and Stock (2003) compared six different force fields in simulations of trialanine, an oligopeptide for which very high quality IR and NMR data are available. They found the most recent OPLS force field to provide the best agreement with experiment for the relative populations of three different conformers, while CHARMM, GROMOS, and force fields coded in the AMBER program systematically overstabilized an α-helical conformer. They also found that the timescales associated with transitions between conformers differed by as much as an order of magnitude between different force fields, although in this instance it is not clear which, if any, of the force fields is providing an accurate representation of reality. Finally, Zamm *et al.* (2003) compared six AA and UA force fields with respect to their predictions for the conformational dynamics of the pentapeptide neurotransmitter Met-enkephalin; they found AA force fields to generally give more reasonable dynamics than UA force fields.

Considering polynucleotides, Arthanari *et al.* (2003) showed that nOe data computed from an unrestrained 12 ns simulation of a double-helical DNA dodecamer using the force field of Cornell *et al.* agreed better with solution NMR experiments than data computed using either the X-ray crystal structure or canonical A or B form structures. Reddy, Leclerc, and Karplus (2003) exhaustively compared four force fields for their ability to model a double-helical DNA decamer. They found the CHARMM22 parameter set to incorrectly favor an A-form helix over the experimentally observed B form. The CHARMM27 parameter set gave acceptable results as did the BMS force field and that of Cornell *et al.* (as modified by Cheatham, Cieplak, and Kollman (1999) to improve performance for sugar puckering and helical repeat).

In conclusion, it appears that the majority of the most modern force fields do well in predicting structural and dynamical properties within wells on their respective PESs. However, their performance for non-equilibrium properties, such as timescales for conformational interconversion, protein folding, etc., have not yet been fully validated. With the increasing speed of both computational hardware and dynamics algorithms, it should be possible to address this question in the near future.

3.8 Case Study: Silica Sodalite

Synopsis of Nicholas *et al.* (1991) ‘Molecular Modeling of Zeolite Structure. 2. Structure and Dynamics of Silica Sodalite and Silicate Force Field’.

Zeolites are mesoporous materials that are crystalline in nature. The simplest zeolites are made up of Al and/or Si and O atoms. Also known as molecular sieves, they find use as drying agents because they are very hygroscopic, but from an economic standpoint they are of greatest importance as size-selective catalysts in various reactions involving hydrocarbons and functionalized molecules of low molecular weight (for instance, they can be used to convert methanol to gasoline). The mechanisms by which zeolites operate are difficult to identify positively because of the heterogeneous nature of the reactions in which they are involved (they are typically solids suspended in solution or reacting with gas-phase molecules), and the signal-to-noise problems associated with identifying reactive intermediates in a large background of stable reactants and products. As a first step toward possible modeling of reactions taking place inside the zeolite silica sodalite, Nicholas and co-workers reported the development of an appropriate force field for the system, and MD simulations aimed at its validation.

The basic structural unit of silica sodalite is presented in Figure 3.9. Because there are only two atomic types, the total number of functional forms and parameters required to define a force field is relatively small (18 parameters total). The authors restrict themselves to an overall functional form that sums stretching, bending, torsional, and non-bonded interactions, the latter having separate LJ and electrostatic terms. The details of the force field are described in a particularly lucid manner. The Si–O stretching potential is chosen to be quadratic, as is the O–Si–O bending potential. The flatter Si–O–Si bending potential is modeled with a fourth-order polynomial with parameters chosen to fit a bending potential computed from *ab initio* molecular orbital calculations (such calculations are the subject of Chapter 6). A Urey–Bradley Si–Si non-bonded harmonic stretching potential is added to couple the Si–O bond length to the Si–O–Si bond angle. Standard torsional potentials and LJ expressions are used, although, in the former case, a switching function is applied to allow the torsion energy to go to zero if one of the bond angles in the four-atom link becomes linear (which can happen at fairly low energy). With respect to electrostatic interactions, the authors note an extraordinarily large range of charges previously proposed for Si and O in this and related systems (spanning about 1.5 charge units). They choose a value for Si roughly midway through this range (which, by charge neutrality, determines the O charge as well), and examine the sensitivity of their model to the electrostatics by

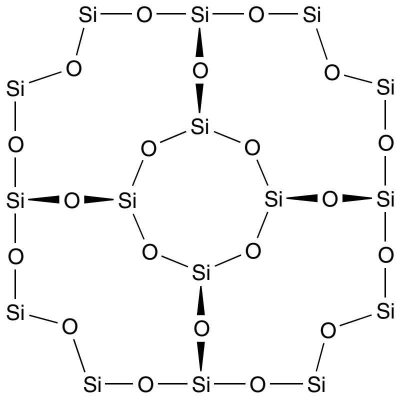

Figure 3.9 The repeating structural unit (with connections not shown) that makes up silica sodalite. What kinds of terms would be required in a force field designed to model such a system?

carrying out MD simulations with dielectric constants of 1, 2, and 5. The simulation cell is composed of 288 atoms (quite small, which makes the simulations computationally simple). PBCs and Ewald sums are used to account for the macroscopic nature of the real zeolite in simulations. Propagation of MD trajectories is accomplished using a leapfrog algorithm and 1.0 fs time steps following 20 ps or more of equilibration at 300 K. Each MD trajectory is 20 ps, which is very short by modern standards, but possibly justified by the limited dynamics available within the crystalline environment.

The quality of the parameter set is evaluated by comparing various details from the simulations to available experimental data. After testing a small range of equilibrium values for the Si–O bond, they settle on 1.61 Å, which gives optimized values for the unit cell Si–O bond length, and O–Si–O and Si–O–Si bond angles of 1.585 Å and 110.1° and 159.9°, respectively. These compare very favorably with experimental values of 1.587 Å and 110.3° and 159.7°, respectively. Furthermore, a Fourier transform of the total dipole correlation function (see Section 3.5) provides a model IR spectrum for comparison to experiment. Again, excellent agreement is obtained, with dominant computed bands appearing at 1106, 776, and 456 cm^{-1}, while experimental bands are observed at 1107, 787, 450 cm^{-1}. Simulations with different dielectric constants showed little difference from one another, suggesting that overall, perhaps because of the high symmetry of the system, sensitivity to partial atomic charge choice was low.

In addition, the authors explore the range of thermal motion of the oxygen atoms with respect to the silicon atoms they connect in the smallest ring of the zeolite cage (the eight-membered ring in the center of Figure 3.9). They determine that motion inward and outward and above and below the plane of the ring takes place with a fair degree of facility, while motion parallel to the Si–Si vector takes place over a much smaller range. This behavior is consistent with the thermal ellipsoids determined experimentally from crystal diffraction.

The authors finish by exploring the transferability of their force field parameters to a different zeolite, namely, silicalite. In this instance, a Fourier transform of the total dipole correlation function provides another model infrared (IR) spectrum for comparison to experiment, and again excellent agreement is obtained. Dominant computed bands appear at 1099, 806, 545, and 464 cm^{-1}, while experimental bands are observed at 1100, 800, 550, and 420 cm^{-1}. Some errors in band intensity are observed in the lower energy region of the spectrum.

As a first step in designing a general modeling strategy for zeolites, this paper is a very good example of how to develop, validate, and report force field parameters and results. The authors are pleasantly forthcoming about some of the assumptions employed in their analysis (for instance, all experimental data derive from crystals incorporating ethylene glycol as a solvent, while the simulations have the zeolite filled only with vacuum) and set an excellent standard for modeling papers of this type.

Bibliography and Suggested Additional Reading

Allen, M. P. and Tildesley, D. J. 1987. *Computer Simulation of Liquids*, Clarendon: Oxford.

Banci, L. 2003. 'Molecular Dynamics Simulations of Metalloproteins', *Curr. Opin. Chem. Biol.*, **7**, 143.

Beveridge, D. L. and McConnell, K. J. 2000. 'Nucleic acids: theory and computer simulation, Y2K' *Curr. Opin. Struct. Biol.*, **10**, 182.

Brooks, C. L., III and Case, D. A. 1993. 'Simulations of Peptide Conformational Dynamics and Thermodynamics' *Chem. Rev.*, **93**, 2487.

Cheatham, T. E., III and Brooks, B. R. 1998. 'Recent Advances in Molecular Dynamics Simulation Towards the Realistic Representation of Biomolecules in Solution' *Theor. Chem. Acc.*, **99**, 279.

Frenkel, D. and Smit, B. 1996. *Understanding Molecular Simulation: From Algorithms to Applications*, Academic Press: San Diego.

Haile, J. 1992. *Molecular Dynamics Simulations*, Wiley: New York.

Jensen, F. 1999. *Introduction to Computational Chemistry*, Wiley: Chichester.

Jorgensen, W. L. 2000. 'Perspective on "Equation of State Calculations by Fast Computing Machines"' *Theor. Chem. Acc.*, **103**, 225.

Lybrand, T. P. 1990. 'Computer Simulation of Biomolecular Systems Using Molecular Dynamics and Free Energy Perturbation Methods' in *Reviews in Computational Chemistry*, Vol. 1, Lipkowitz, K. B. and Boyd, D. B., Eds., VCH: New York, 295.

McQuarrie, D. A. 1973. *Statistical Thermodynamics*, University Science Books: Mill Valley, CA.

Norberg, J. and Nilsson, L. 2003. 'Advances in Biomolecular Simulations: Methodology and Recent Applications', *Quart. Rev. Biophys.*, **36**, 257.

Straatsma, T. P. 1996. 'Free Energy by Molecular Simulation' in *Reviews in Computational Chemistry*, Vol. 9, Lipkowitz, K. B. and Boyd, D. B., Eds., VCH: New York, 81.

References

Andersen, H. C. 1980. *J. Chem. Phys.*, **72**, 2384.

Andricioaei, I. and Straub, J. E. 1996. *Phys. Rev. E*, **53**, R3055.

Arthanari, H., McConnell, K. J., Beger, R., Young, M. A., Beveridge, D. L., and Bolton, P. H. 2003. *Biopolymers*, **68**, 3.

Banks, J. L., Kaminski, G. A., Zhou, R., Mainz, D. T., Berne, B. J., and Friesner, R. A. 1999. *J. Chem. Phys.*, **110**, 741.

Berendsen, H. J. C., Postma, J. P. M., van Gunsteren, W. F., DiNola, A., and Haak, J. R. 1984. *J. Chem. Phys.*, **81**, 3684.

Berens, P. H., and Wilson, K. R. 1981. *J. Chem. Phys.*, **74**, 4872.

Bergdorf, M., Peter, C., and Hunenberger, P. H. 2003. *J. Chem. Phys.*, **119**, 9129.

Bernardo, D. N., Ding, Y., Krogh-Jespersen, K., and Levy, R. M. 1994. *J. Phys. Chem.*, **98**, 4180.

Berne, B. J. and Harp, G. D. 1970. *Adv. Chem. Phys.*, **17**, **63**, 130.

Brown, S. and Head-Gordon, T. 2003. *J. Comput. Chem.*, **24**, 68.

Cheatham, T. E., III, Cieplak, P., and Kollman, P. A. 1999. *J. Biomol. Struct. Dyn.*, **16**, 845.

Dang, L. X., Rice, J. E., Caldwell, J., and Kollman, P. A. 1991. *J. Am. Chem. Soc.*, **113**, 2481.

Demaison, J., Hütner, W., and Tiemann, E. 1982. In: *Molecular Constants, Landolt-Börstein, New Series, Group II*, Vol. 14a, Hellwege, K. -H. and Hellwege, A. M., Eds., Springer-Verlag: Berlin, 584.

Ernst, M. H., Hauge, E. H., and van Leeuwen, J. M. J. 1971. *Phys. Rev. A*, **4**, 2055.

Feenstra, K. A., Hess, B., and Berendsen, H. J. C. 1999. *J. Comput. Chem.*, **20**, 786.

Flyvberg, H. and Petersen, H. G. 1989. *J. Chem. Phys.*, **91**, 461.

Ford, J. 1973. *Adv. Chem. Phys.*, **24**, 155.

Gao, J., Habibollazadeh, D., and Shao, L. 1995. *J. Phys. Chem.*, **99**, 16460.

Gear, C. W. 1971. *Numerical Initial Value Problems in Ordinary Differential Equations*, Prentice-Hall: Englewood Cliffs, N.J.

Grubmüller, H. and Tavan, P. 1998. *J. Comput. Chem.*, **19**, 1534.

Guarnieri, F. and Still, W. C. 1994. *J. Comput. Chem.*, **15**, 1302.

Haile, J. 1992. *Molecular Dynamics Simulations*, Wiley: New York, 291.

Hoover, W. G. 1985. *Phys. Rev. A,* **31**, 1695.

Hornak, V. and Simmerling, C. 2003. *Proteins*, **51**, 577.

Jaqaman, K. and Ortoleva, P. J. 2002. *J. Comput. Chem.*, **23**, 484.

Kolafa, J. 1986. *Mol. Phys.*, **59**, 1035.

Laio, A. and Parrinello, M. 2002. *Proc. Natl. Acad. Sci. USA*, **99**, 12562.

Mahoney, W. and Jorgensen, W. L. 2000. *J. Chem. Phys.*, **112**, 8910.

Metropolis, N., Rosenbluth, A. E., Rosenbluth, M. N., Teller, A. H., and Teller, E. 1953. *J. Chem. Phys.*, **21**, 1087.

Mu, Y., Kosov, D. S., and Stock, G. 2003. *J. Phys. Chem. B*, **107**, 5064.

Müller, E. M., de Meijere, A., and Grubmüller, H. 2002. *J. Chem. Phys.* **116**, 897.

Nagan, M. C., Kerimo, S. S., Musier-Forsyth, K., and Cramer, C. J. 1999. *J. Am. Chem. Soc.*, **121**, 7310.

Nakamura, H. 2002. *J. Comput. Chem.*, **23**, 511.

Nicholas, J. B., Hopfinger, A. J., Trouw, F. R., and Iton, L. E. 1991. *J. Am. Chem. Soc.*, **113**, 4792.

Nosé, S. 1984. *Mol. Phys.*, **52**, 255.

Okur, A, Strockbine, B., Hornak, V., and Simmerling, C. 2003. *J. Comput. Chem.*, **24**, 21.

Olender, R. and Elber, R., 1996. *J. Chem. Phys.*, **105**, 9299.

Pangali, C., Rao, M., and Berne, B. J. 1978. *Chem. Phys. Lett.*, **55**, 413.

Press, W. H., Flannery, B. P., Teukolsky, S. A., and Vetterling, W. T. 1986. *Numerical Recipes*, Cambridge University Press: New York.

Reddy, S. Y., Leclerc, F., and Karplus, M. 2003. *Biophys. J.*, **84**, 1421.

Rizzo, R. C. and Jorgensen, W. L. 1999. *J. Am. Chem. Soc.*, **121**, 4827.

Ryckaert, J. P., Ciccotti, G., and Berendsen, H. J. C. 1977. *J. Comput. Phys.*, **23**, 327.

Senderowitz, H. and Still, W. C. 1998. *J. Comput. Chem.*, **19**, 1736.

Sherer, E. C. and Cramer, C. J. 2002. *J. Phys. Chem. B*, **106**, 5075.

Smellie, A., Stanton, R., Henne, R., and Teig, S. 2003. *J. Comput. Chem.*, **24**, 10.

Smith, E. B. and Wells, B. H. 1984. *Mol. Phys.*, **53**, 701.

Soetens, J.-C., Millot, C., Chipot, C., Jansen, G., Ángyán, J. G., and Maigret, B. 1997. *J. Phys. Chem. B*, **101**, 10910.

Stolovitzky, G. and Berne, B. J. 2000. *Proc. Natl. Acad. Sci. (USA)*, **21**, 11164.

Straatsma, T. P. and McCammon, J. A. 1994. *J. Chem. Phys.*, **101**, 5032.

Straatsma, T. P., Berendsen, H. J. C., and Stam, A. J. 1986. *Mol. Phys.*, **57**, 89.

Theodorouo, D. N. and Suter, U. W. 1985. *J. Chem. Phys.*, **82**, 955.

Verkhivker, G., Elber, R., and Nowak, W. 1992. *J. Chem. Phys.*, **97**, 7838.

Verlet, L. 1967. *Phys. Rev.*, **159**, 98.

Weinberg, N. and Wolfe, S. 1994. *J. Am. Chem. Soc.*, **116**, 9860.

Winn, P. J., Ferenczy, G., and Reynolds, C. A. 1999. *J. Comput. Chem.*, **20**, 704.

Wlodek, S. T., Clard, T. W., Scott, L. R., McCammon, J. A. 1997. *J. Am. Chem. Soc.*, **119**, 9513.

Yang, W., Bitetti-Putzer, R., and Karplus, M. 2004. *J. Chem. Phys.*, **120**, 2618.

Zamm, M. H., Shen, M.-Y., Berry, R. S., and Freed, K. F. 2003. *J. Phys. Chem. B*, **107**, 1685.

Zhu, S.-B. and Wong, C. F. 1994. *J. Phys. Chem.*, **98**, 4695.

Zwanzig, R. 1965. *Ann. Rev. Phys. Chem.*, **16**, 67.