CHAPTER 12

Molecular Motors: Theory

Alex Mogilner, Tim Elston, Hongyun Wang and George Oster

Evolution has created a class of proteins that have the ability to convert chemical energy into mechanical force. Some of these use the free energy of nucleotide hydrolysis as fuel, while others employ ion gradients. Some are "walking motors," others rotating engines. Some are reversible, others are unidirectional. Could there be any common principles amongst such diversity?

The conversion of chemical energy into mechanical work is one of the main themes of modern biology. Biochemists characterize energy transduction schemes by free energy diagrams. But thermodynamics tells us only what *cannot* happen. Recent advances in laser trap and optical technology, along with advances in molecular structure determination can augment traditional biochemical kinetic and thermodynamic analyses to make possible a more mechanistic view of how protein motors function. The result of these advances has been data that yield load-velocity curves and motion statistics for single molecular motors. This sort of data enables a more detailed, mechanistic level of modeling.

At first, the mechanics of proteins may seem counterintuitive because their motions are dominated by Brownian motion, the name given to the frequent changes in velocity of a macromolecule as it is buffeted about by random thermal motions of surrounding water molecules. In addition to "smearing out" deterministic trajectories, Brownian motion serves an effective "lubricant," allowing molecules to pass over high energy barriers that would arrest a deterministic system. More subtly, it makes possible "uphill" motions against an opposing force by "capturing" occasional large thermal fluctuations.

In this chapter we will discuss protein motions on the molecular scale and derive a mathematical formalism to model such motions. To illustrate the formalism, in the next chapter, we will analyze (i) a "switch" controlling the direction of the bacterial flagellar motor, (ii) a polymerization ratchet, and (iii) a "toy" model related to the F_o

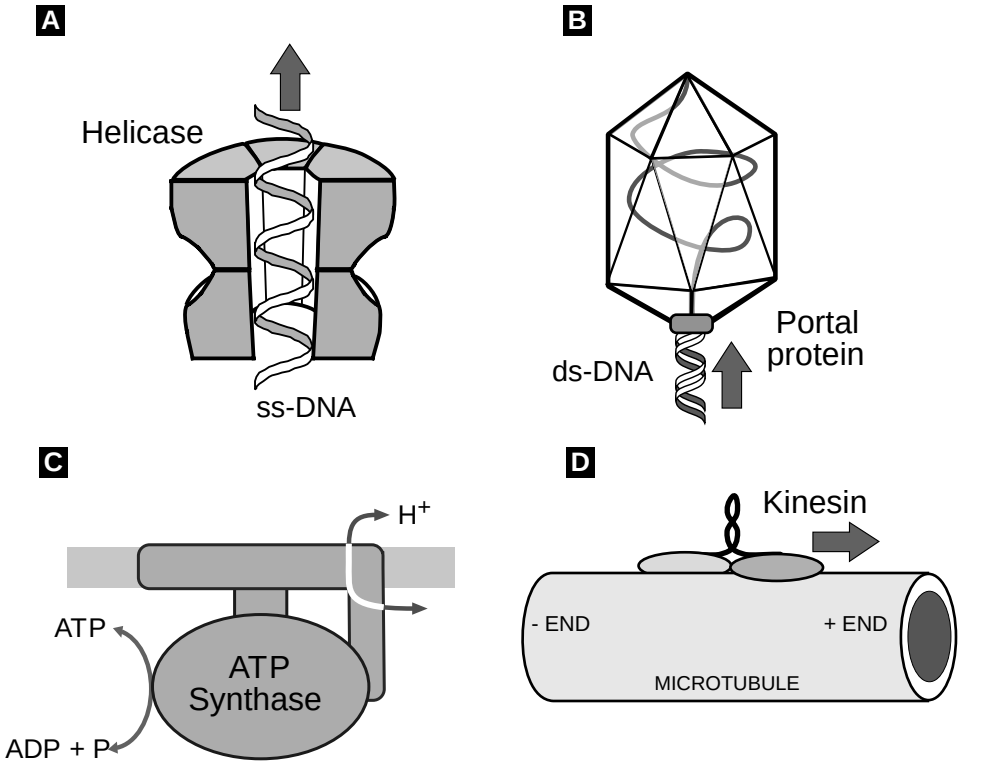

Figure 12.1 Amazing variety of molecular motors: (A) Rotary motor DNA helicase translocates unidirectionally along the DNA strand using nucleotide hydrolysis as a "fuel". (B) Another rotary motor hydrolyzing ATP, bacteriophage portal protein, drives DNA in and out. (C) Reversible rotary motor ATP synthase either produces ATP using ion gradient, or pumps protons hydrolyzing ATP. (D) Linear motor kinesin is a "walking enzyme". Utilizing chemical energy stored in ATP, it moves "head-over-head" toward the plus end of the microtubule "track." Some of these motors are discussed in this chapter.

motor of ATP synthase. These models are simple enough to yield analytical as well as numerical results, and to illustrate many of the principles involved in mechanochemical energy transduction.

There is some ambiguity in what one calls a "motor." Here we take the narrow view that the principal - and proximate - function of a molecular motor is to convert chemical energy into mechanical force. This excludes, for example, ion pumps, which are surely protein machines that generate forces, but whose purpose is not force production. Chemical energy comes in various forms, for example, in transmembrane ion gradients and in the covalent bonds of nucleotides such as ATP and GTP, and the designs of motor proteins are tailored to each energy form.

Energy stored in one form frequently is converted into intermediate forms before being released as mechanical work. For example, a polymerizing actin filament or microtubule can generate a protrusive force capable of deforming a lipid vesicle

or pushing out the leading edge of a cell [Honda et al., 1999, Fygenson et al., 1997, Dogterom and Yurke, 1997]. The energy source in this process is the free energy of binding monomers to the polymer tip. This energy is used to rectify the Brownian motion of the load against which the polymer is pushing. Strictly speaking, the force is generated by thermal fluctuations of the load, and the binding free energy is used to rectify its thermal displacements. Energy conversion here is relatively direct. However, in the acrosomal process of the *Limulus* sperm, thermal fluctuations are first trapped as elastic strain energy in the actin polymer by the binding of an auxiliary protein, scruin. Later, this strain energy is released to generate the force required push the actin rod into the egg cortex [Mahadevan and Matsudaira, 2000].

Many motors use nucleotide hydrolysis to generate mechanical forces, and it is frequently stated that the energy is stored in the γ-phosphate covalent bond. But releasing this energy to perform mechanical work can be quite indirect. The F_1 motor of ATP synthase uses nucleotide hydrolysis to generate a large rotary torque [Yasuda et al., 1998]. However, the actual force generating step takes place during the binding of ATP to the catalytic site; the role of the hydrolysis step is to release the hydrolysis products, allowing the cycle to repeat [Wang and Oster, 2000, Oster and Wang, 2000]. In some motors, not all of the nucleotide binding energy is used immediately for force production; some energy is stored in elastic deformation of the protein to be released later as mechanical work. So energy transduction need not be a "pay as you go" process; deferred payments are permissible and common.

The bacterial flagellar motor and the F_o motor of ATP synthase both use transmembrane ion gradients to generate a rotary torque [Berg, 2000]. Models of this process show how the chemical reaction of binding an ion onto a charged site creates an unbalanced electrostatic field that rectifies the Brownian motion of the motor and/or creates an electrostatic driving torque [Elston and Oster, 1997, Elston et al., 1998]. Although the proximal energy transduction process is a chemical binding event, the motion itself is produced by electrostatic forces and Brownian motion.

Thus a common theme in energy transduction is that *chemical reactions power mechanical using free energy* released during binding events, but the final producttion of mechanical force may involve a number of intermediate energy transductions.

The most important quality of molecular motors that distinguishes them from macroscopic motors is the overwhelming importance of thermal fluctuations. For this reason, all protein motors must be regarded as "Brownian machines." This means that carelessly applying macroscopic physics, where Brownian motion is negligible, to microscopic situations inevitably leads to incorrect conclusions. Therefore, we must begin our discussion by examining how to model molecular motions dominated by thermal fluctuations.

12.1 Molecular motions as stochastic processes

12.1.1 Protein motion as a simple random walk

Generally, a stochastic process refers to a random variable that evolves in time. An example is a one-dimenisonal coordinate, $x(t)$, locating a protein diffusing in an aqueous solution. We will begin by approximating the coordinate, $x(t)$, by a discrete random variable. The rationale for this is twofold. Discrete random variables are conceptually simpler than their continuous counterparts. The results for the discrete case are applicable when studying continuous random processes because continuous random variables represent limiting behavior of their discrete counterparts. Our discussion is restricted to Markov processes. A Markov process is a mathematical idealization in which the future state of a protein is affected by its current state but is independent of its past. That is, the system has no memory of how it arrived at its current state. To a very good approximation, all systems considered in this text satisfy the Markov property. The mathematics involved with studying stochastic processes that are non-Markovian is considerably more complicated.

In the discrete model, a protein is initially started at $x = 0$. In each time interval Δt, it takes one step of length Δx to the right with probability $1/2$ or to the left with probability $1/2$. Because the length of the step that the protein takes is always the same, this example is referred to as a simple random walk. Let x_n denote the protein's position at time $t = n\Delta t$ and define the set of random variables z_m with $m = 1, 2, \ldots, n$ to be independent and identically distributed with $\text{Prob}[z_m = 1] = 1/2$ and $\text{Prob}[z_m = -1] = 1/2$. Then we have

$$x_n = \Delta x(z_1 + z_2 + \cdots z_n). \tag{12.1}$$

The collection of random variables $x = \{x_0, x_1, x_2, \cdots\}$ represents a spatially and temporally discrete stochastic process. In Exercise 1, (12.1) is used to verify that $\langle x_n \rangle = 0$ and $\text{Var}[x_n] = (\Delta x)^2 n = ((\Delta x)^2/\Delta t)t$. Here we use the notation $\langle \cdot \rangle$ to denote the average (expectation), and $\text{Var}[\cdot]$ to denote the variance: $\text{Var}[x] \equiv \langle x^2 \rangle - \langle x \rangle^2$. Note that the variance in x grows linearly with time. This is a characteristic of diffusion; below we show in what sense the quantity $D = (\Delta x)^2/(2\Delta t)$ can be interpreted as a diffusion coefficient.

To fully characterize x requires knowledge of the probability density for finding the particle at position x_n after k steps of size Δx: $p_k(n) = \text{Prob}[x_n = k\Delta x]$. Note that $p_k(n) = 0$, if $n < |k|$. This comes from the fact that the protein can only take one step per time interval. At any time $n\Delta t$, the total number of steps taken by the protein is $n = R_n + L_n$, where R_n is the number of steps taken to the right and L_n is the number of steps taken to the left. Clearly, R_n is binomially distributed, like the number of "heads" in n flips of a coin:

$$\text{Prob}[R_n = m] = \left(\begin{array}{c} n \\ m \end{array} \right) \left(\frac{1}{2}\right)^n = \frac{n!}{n!(n-m)!} \left(\frac{1}{2}\right)^n \tag{12.2}$$

Using these definitions, x_n is written as

$$x_n = \Delta x(R_n - L_n) = \Delta x(2R_n - n), \tag{12.3}$$

or equivalently

$$R_n = \frac{1}{2}(\frac{x_n}{\Delta x} + n). \tag{12.4}$$

Thus $x_n/\Delta x = k$ if and only if $R_n = \frac{1}{2}(n+k)$. Furthermore, $x_n/\Delta x$ must be even if n is even and odd if n is odd, since R_n must be an integer. Therefore, we immediately find that the distribution for x_n is

$$p_k(n) = \left(\begin{array}{c} n \\ (k+n)/2 \end{array} \right) \left(\frac{1}{2}\right)^n \tag{12.5}$$

for $n \geq |k|$ and k and n either both even or both odd.

Note that in (12.1) x_n is written as the sum of n independent and identically distributed random variables. Therefore, the central limit theorem of probability theory guarantees that as n gets large the distribution for x_n becomes normal with $\langle x_n \rangle = 0$ and $\mathrm{Var}[x_n] = ((\Delta x)^2/\Delta t)t = 2Dt$. That is,

$$\frac{p_k(n)}{\Delta x} \approx p(x,t) = \frac{1}{\sqrt{4\pi Dt}} \exp\left(-\frac{x^2}{4Dt}\right), \quad x = k\Delta x, \quad t = n\Delta t. \tag{12.6}$$

Fig. 12.2 shows the probability distribution for x_n and the normal approximation for various values of n. By the time $n = t/\Delta t = 15$, the distribution of x_n is close to normal and the agreement gets better as n is increased. Physically, the normal approximation amounts to a "coarse graining" of the process in which only length scales much larger than Δx and time scales much larger than Δt are resolved. In this limit the random variable x_n, which is discrete in both space and time, is approximated by $x(t)$, a random variable that is continuous in space and time. The value of Δt can be approximated well by the "thermalization" time, $\tau = 10^{-13}$ sec, described in Section 12.2.1. Thus, the continuous and discrete models of a protein's motion are equivalent at all time and distance scales of interest to us.

As an exercise, the reader is asked to verify that $p(x,t)$ satisfies the diffusion equation

$$\frac{\partial p(x,t)}{\partial t} = D\frac{\partial^2 p(x,t)}{\partial x^2}. \tag{12.7}$$

justifying our association of the quantity $(\Delta x)^2/(2\Delta t)$ with a diffusion coefficient.

12.1.2 Polymer growth

Let us consider another example of a stochastic process: the number of monomers, $N(t)$, in a polymerizing biopolymer. There is an important distinction between the stochastic variables $x(t)$ and $N(t)$. In the first example, $x(t)$ is a continuous random variable since

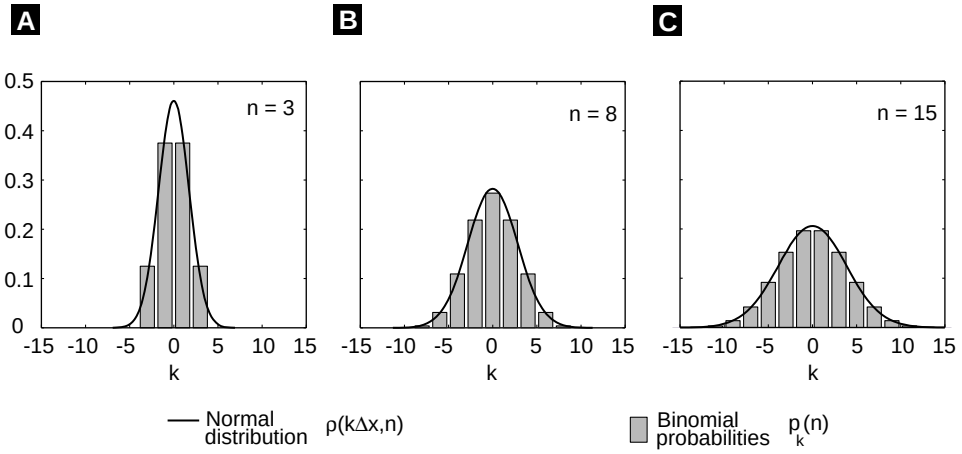

Figure 12.2 In the limit of large n, the binomial distribution is well approximated by a normal distribution. In all three panels the bar graph represents the binomial probabilities and the solid line is the normal approximation. (A) n = 3, (B) n = 8, (C) n = 15

it can take on any real value. On the other hand, the number of subunits in a growing polymer is restricted to the positive integers, so that $N(t)$ is a discrete random variable.

Markov processes in which the random variable is discrete are often referred to as Markov *chains* because they can be represented as a sequence of jumps between discrete states. The simple random walk is an example of a spatially and temporally discrete Markov chain. As an example of a Markov chain in which time is continuous we consider a polymerizing biological polymer (filament), e.g. an actin filament or a microtubule. Fig. 12.3A depicts the type of process we have in mind. In this example, two events change the length of the polymer by one monomer: polymerization and depolymerization. Mathematically, the state of the system is specified by a single number $N(t)$, the number of monomers in the filament at time t. $N(t)$ is a random variable, because we have no way to predict when the next polymerization or depolymerization event will occur. A diagram of the Markov chain for this process is shown in Fig. 12.3B.

There are two equivalent, but conceptually different, levels at which stochastic processes can be studied. The first is at the level of individual sample paths or realizations of the process. To understand what is meant by a sample path, suppose that, at $t = 0$, we start with three filaments that are each exactly 5 monomers long. As time goes on, we observe that the number of monomers in each filament instantaneously changes or "jumps" by ± 1 at random times. Fig. 12.4 graphically illustrates this behavior. Even though each sample path starts with $N(0) = 5$, they all evolve differently, illustrating the randomness of the process. The sample paths of a large ensemble of such filaments can be used to determine the statistics of $N(t)$.

The second approach is to ask how the probability $p_n(t)$ of having exactly n monomers in the filament at time t changes in time. If $p_n(t)$ can be determined for

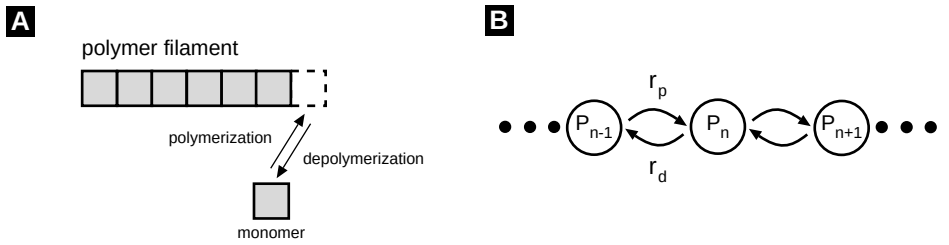

Figure 12.3 A discrete Markov process (Markov chain). (A) A polymer filament grows by incorporating monomers from the solution onto its tip (polymerization). The process is stochastic. The monomer on the tip may dissociate from the filament into the solution (depolymerization). (B) A Markov chain model for the filament polymerization. P_n represents the state of the filament when its length is n monomers. r_p is the polymerization rate. r_d is the depolymerization rate. If $r_p > r_d$, the filament will grow over long times.

all t and n, then we have a complete characterization of the process. Both approaches are equally valid and are useful methods for studying stochastic processes. The advantages of staying at the level of sample paths are that in general it is easy to numerically generate single realizations of the process and sample paths allow us to see the dynamics of the system. The advantage of working directly with the probability distribution is that it fully characterizes the system without having to average over many sample paths to compute the statistics. Of course, there is no free lunch: obtaining all this information comes at a computational price. Below we describe numerical techniques for treating both cases.

To begin our discussion we derive an equation that governs the evolution of $p_n(t)$. Let us assume that we know $p_n(t)$ for a specific value of t. At a slightly later time $t+\Delta t$, we expect $p_n(t+\Delta t)$ to be equal to $p_n(t)$ plus a small correction. The key is to assume that Δt is so small that the probability of two events in the interval $(t, t+\Delta t)$ is very unlikely. Here an event means polymerization or depolymerization. Then we can write:

$$\begin{aligned} p_n(t+\Delta t) = &\text{Prob}\left[N(t) = n \text{ and no event occurs in } (t, t+\Delta t)\right] \\ &+ \text{Prob}\left[N(t) = n-1 \text{ and polym. occurs in } (t, t+\Delta t)\right] \\ &+ \text{Prob}\left[N(t) = n+1 \text{ and depol. occurs in } (t, t+\Delta t)\right], \end{aligned} \tag{12.8}$$

where the right-hand-side follows from the fact that the three events described in the square brackets are mutually exclusive. Next we make the reasonable assumption that the probability of polymerization or depolymerization is independent of the length of the filament. We also assume that these probabilities are proportional to Δt, and let $r_p\Delta t$ and $r_d\Delta t$ be the probability of polymerization and depolymerization, respectively, in $(t, t+\Delta t)$. Under these assumptions (12.8) can be written as

$$\begin{aligned} p_n(t+\Delta t) &= p_n(t)(1-(r_p+r_d)\Delta t) + p_{n-1}(t)r_p\Delta t + p_{n+1}(t)r_d\Delta t \\ &= p_n(t) + \Delta t\left[r_p p_{n-1}(t) + r_d p_{n+1}(t) - (r_p+r_d)p_n(t)\right], \end{aligned} \tag{12.9}$$

where $(1-(r_p+r_d)\Delta t)$ is probability of no event in Δt. There are two important points to be drawn from (12.9). First, it is clear that if we know $p_n(t)$ for all n at a given time, then we have a mechanism for updating the probabilities at all later times. This illustrates the Markov property. Secondly (12.9) represents a numerical algorithm for updating $p_n(t)$. That is, once a Δt is chosen, we can write a computer program to generate $p_n(t+k\Delta t)$, where k is positive integer. We now take the limit $\Delta t \to 0$ in (12.9):

$$\lim_{\Delta t \to 0} \frac{p_n(t+\Delta t)-p_n(t)}{\Delta t} = \frac{dp_n(t)}{dt} = \\ -(r_p+r_d)p_n(t)+r_p p_{n-1}(t)+r_d p_{n+1}(t). \tag{12.10}$$

Therefore, (12.9) is an algorithm for numerically solving the ordinary differential equation given by (12.10). This algorithm is called the Forward Euler method, and is a very useful numerical tool that works adequately for many situations. However, problems may arise when using this scheme, as discussed below. Also, note that (12.10) can be interpreted as a chemical rate equation, so that r_p and r_d are the rates of polymerization and depolymerization, respectively.

12.1.3 Sample paths of the process

The next question we address is how to numerically generate sample paths that are consistent with (12.10). To analyze this problem consider the following experiment. At time $t=0$, we start with a filament containing exactly m monomers. That is, $p_m(0)=1$. Next we watch the filament until the first event occurs (either polymerization or depolymerization). When this event occurs we record the time and start the experiment again. After doing this experiment many times, we find that the amount of time we must wait for the first event to occur is a random variable. Let's call it T. We are after the probability density $f_T(t)$ for T. Let $q(t)$ be the probability that no event has occurred in $(0,t)$. Under the conditions of the experiment and from the derivation of

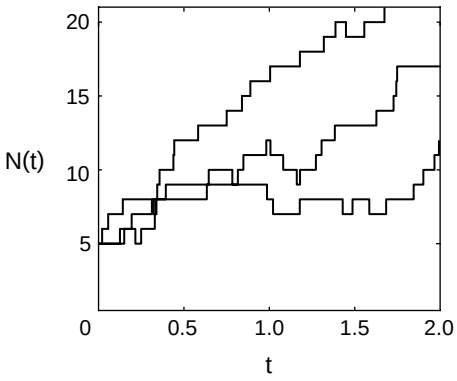

Figure 12.4 Three sample trajectories of the tip of a growing filament. The polymerization process is stochastic with occasional depolymerization events.

(12.10), we have:

$$\frac{dq}{dt} = -(r_p + r_d)q. \tag{12.11}$$

Solving this equation, we find $q(t) = \exp(-(r_p + r_d)t)$. $q(t)$ starts at 1 and decreases to 0 as time goes on. The probability that at least one event has occurred in $(0, t)$ is $(1 - q(t))$. Hence, we can use $(1 - q(t))$ to define a probability density funtion, $f_T(t)$, for the *waiting time* distribution:

$$1 - q(t) = \text{Prob [Waiting time } T < t] = 1 - q(t) = \int_0^t f_T(t')dt', \tag{12.12}$$

By differentiating (12.12), we find the relationship:

$$f_T(t) = -\frac{dq(t)}{dt} = (r_p + r_d)\exp(-(r_p + r_d)t). \tag{12.13}$$

That is, the waiting time until the next event occurs has an exponential distribution with mean $1/(r_p + r_d)$. Thus, to produce realizations of $N(t)$, we need to be able to generate samples of an exponential random variable.

Most programming languages have built-in random number generators that produce numbers that are uniformly distributed between 0 and 1. If R is such a random variable, its probability density function is $f_R(t) = 1$ in $[0, 1]$. The transformation that converts R to an exponential random variable with mean $1/(r_p + r_d)$ is

$$T(R) = -\frac{1}{(r_p + r_d)}\ln R. \tag{12.14}$$

This can be verified mathematically as follows:

$$\int_0^t f_T(t)dt = \text{Prob } [T(R) < t] \tag{12.15}$$

$$= \text{Prob } [R > \exp(-(r_p + r_d)t)]$$

$$= 1 - \exp(-(r_p + r_d)t) = 1 - q(t). \tag{12.16}$$

Given this way to compute when the next transition occurs, the next thing we need to determine is whether polymerization or depolymerization takes place. Remember that in any time interval of length Δt the probability of polymerization is $r_p\Delta t$. Likewise, the probability of depolymerization is $r_d\Delta t$. Therefore, given that an event has occurred, the probability that it was polymerization is:

$$P \text{ [polymerization|an event occurred at } t] = \frac{r_p}{r_p + r_d}. \tag{12.17}$$

We may now generate sample paths of the stochastic process as follows: Start N with a given value. Generate an exponentially distributed random number using (12.14). This determines when the next event takes place. To determine the type of event that occurred, generate a uniformly distributed random number R_2. If $R_2 < r_p/(r_p + r_d)$,

then let $N \to N+1$, otherwise $N \to N-1$. Repeat the process and plot N as a function of time. The trajectories shown in Fig. 12.4 were generated in this way.

Before using this method to simulate protein motions, we briefly discuss the statistical behavior of polymer growth.

12.1.4 The statistical behavior of polymer growth

Since the intervals of time between events of monomers assembly and/or disassembly are random, one can measure only the statistical behavior of polymer growth, such as the average velocity of the polymer's tip, $\langle V \rangle = L\langle N \rangle / t$, where L is the size of a monomer. Much useful information is buried in the statistical fluctuations about this mean velocity. One quantity that can be monitored as the polymer grows is the variance of the tip's displacement about the mean:

$$\mathrm{Var}[x(t)] \equiv \langle x^2 \rangle - \langle x \rangle^2 = L^2(\langle N^2 \rangle - \langle N \rangle^2).$$

It is easy to show (see Exercise 4) that the average velocity of the polymer tip and the variance of its displacement are:

$$\langle V \rangle = L(r_p - r_d), \quad \mathrm{Var}[x(t)] = L^2(r_p + r_d)t.$$

Thus the variance grows linearly with time. In fact, a plot of $\mathrm{Var}[x(t)]$vs$t$ can be used to define an effective diffusion coefficient: $D_{eff} \equiv \mathrm{Var}[x(t)]/2t$ [Wang et al., 1998] (see also (**??**)). D_{eff} can be combined with the average velocity $\langle V \rangle$ to form a "randomness parameter" [Schnitzer and Block, 1995]:

$$r \equiv \frac{2D_{eff}}{L \cdot \langle V \rangle}. \tag{12.18}$$

As an example of the utility of this randomness parameter, let us consider the case when there is no depolymerization: $r_d = 0$. Then, $\mathrm{Var}[x(t)] = L^2 r_p t$, $D_{eff} = L^2 r_p/2$, $\langle V \rangle = L r_p$, and $r = 1$. Now suppose that each polymerization event involves a sequence of reaction processes. Since chemical reactions are also stochastic processes, an additional variance is added to the spatial diffusion, so that the total variance will grow faster, and the randomness parameter is greater than 1. In this case, $1/r$ gives a lower bound on the number of reaction processes per step (i.e., $1/r <$ number of reaction processes per step) [Schnitzer and Block, 1995].

Similar arguments are applicable to some "walking" motors, e.g. kinesin, that take a spatial step of constant size at random times. This time is determined by a sequence of hydrolysis reactions. If there is only one reaction, the walking motor is equivalent to the polymerizing filament, and is called "Poisson stepper." Such a stepper is characterized by randomness parameter $r = 1$. In the next chapter we will show that the average velocity of a molecular motor is a function of the load force resisting the motor's advancement. The importance of considering the effective diffusion coefficient (or, equivalently, the randomness parameter), is that just as load-velocity data gives in-

formation about the motor performance, load-variance data can provide independent estimates of model parameters (see, for example [Peskin and Oster, 1995]).

12.2 Modeling molecular motions

The botanist Robert Brown first observed Brownian movement in 1827. While studying a droplet of water under a microscope, he noticed tiny specks of plant pollen dancing around. Brown first guessed, and later proved, that these were not living, although at the time he had no clue as to the mechanism of their motion. It was not until Einstein contemplated the phenomenon 75 years later that a quantitative explanation emerged. In order to develop an intuition about molecular dynamics we begin with some simple remarks on Brownian motion of proteins in aqueous solutions.

12.2.1 The Langevin equation

The radius of a water molecule is about 0.1 nm, while proteins are two orders of magnitude larger, in the range 2 - 10 nm. This size difference suggests that we can view the fluid as a continuum. A protein moving through the fluid is acted on by frequent and uncorrelated momentum impulses arising from the thermal motions of the fluid. We model these fluctuations as a time-dependent random "Brownian force," $f_B(t)$, whose statistical properties can be mimicked by a random number generator in a computer in a fashion described below. At the same time, the fluid continuum exerts on the moving protein a frictional drag force, f_d, proportional to the protein's velocity: $f_d = -\zeta v$, where ζ is the frictional drag coefficient (see Section 12.6.1). Thus we can write Newton's law for the motion, $x(t)$, of a protein moving in a one-dimensional domain of length L:

$$\frac{dx}{dt} = v, \quad m\frac{dv}{dt} = -\zeta v + f_B(t), \quad 0 \le x(t) \le L. \tag{12.19}$$

The mass, m, of a typical protein is about 10^{-21} kg, and the drag coefficient is about 10^{-7} pN·sec/nm.

If we multiply (12.19) by $x(t)$ and use the chain rule, we get:

$$\frac{m}{2}\frac{d^2(x^2)}{dt^2} - mv^2 = -\frac{\zeta}{2}\frac{d(x^2)}{dt} + x \cdot f_B(t). \tag{12.20}$$

In order to see the consequences of (12.20) for molecular motions we first must average (12.20) over a large number of proteins so that the peculiarities of any particular trajectory are averaged out. We use the notation $\langle\cdot\rangle$ to denote this *ensemble* average:

$$\frac{m}{2}\frac{d^2\langle x^2\rangle}{dt^2} - \langle mv^2\rangle = -\frac{\zeta}{2}\frac{d\langle x^2\rangle}{dt} + \langle x \cdot f_B(t)\rangle. \tag{12.21}$$

Next we take advantage of a central result from statistical mechanics called the *Equipartition Theorem*(Section 12.6.2), which states that each *degree of freedom* of a

Brownian particle carries an average energy

$$\langle E \rangle = \frac{1}{2} k_B T, \quad \text{[Equipartition Theorem]} \tag{12.22}$$

where k_B is Boltzmann's constant and T the absolute temperature [Landau et al., 1980]. Therefore, the second term in (12.21) is just twice the average kinetic energy of the protein: $\langle mv^2 \rangle = k_B T$. At room temperature, the quantity $k_B T \simeq 4.1$ pN·nm is the "unit" of thermal energy.

Because the random impulses from the water molecules are uncorrelated with position, $\langle x(t) \cdot f_B(t) \rangle = 0$. Introducing these two facts into (12.21) and integrating twice between $t = (0, t)$ with $x(0) = 0$:

$$\frac{d\langle x^2 \rangle}{dt} = \frac{2k_B T}{\zeta}(1 - e^{-t/\tau}), \quad \langle x^2 \rangle = \frac{2k_B T}{\zeta}[t - \tau(1 - e^{-t/\tau})], \tag{12.23}$$

where we have introduced the time constant $\tau = m/\zeta$.

For very short times, $t \ll \tau$, we can expand the exponential in (12.23) to second order to obtain:

$$\langle x^2 \rangle = \frac{k_B T}{m} t^2 \quad (t \ll \tau). \tag{12.24}$$

That is, at very short times the protein behaves as a ballistic particle moving with a velocity $v = \sqrt{k_B T/m}$. For a protein with $m \simeq 10^{-21}$ kg [$= 10^{-18}$ pN·sec^2/nm], $v \simeq 2$ m/s. However, in a fluid the protein moves at this velocity only for a time $\tau \sim m/\zeta = 10^{-13}$ sec, much shorter than any motion of interest in a molecular motor. During this short time the protein travels a distance $v \cdot \tau \sim 0.01$ nm before it collides with another molecule. This is only a fraction of a diameter of water molecule, so the ballistic regime is very short lived indeed! Very quickly, the kinetic energy of the protein comes into thermal equilibrium (is "thermalized") with the fluid environment. Thus when $t \gg \tau$, the exponential term disappears and (12.23) becomes

$$\langle x^2 \rangle = 2\frac{k_B T}{\zeta} t \quad (t \gg \tau). \tag{12.25}$$

Einstein recognized that the frictional drag on a moving body is caused by random collisions with the fluid molecules, which is the same effect as the Brownian force, $f_B(t)$ that gives rise to the diffusive motion of the body. Therefore, there must be a connection between the drag coefficient and diffusive motion. By comparing (12.25) to the relation we previously derived between the mean square displacement of a diffusing particle and its diffusion coefficient,

$$\langle x^2 \rangle = 2Dt, . \tag{12.26}$$

we arrive at the famous relation derived by Einstein in 1905:

$$D = k_B T/\zeta, \quad \text{[Einstein Relation]} \tag{12.27}$$

where D is the diffusion coefficient of the protein, typically $D \sim 10^7$ nm^2/sec. For diffusion in 2 and 3 dimensions, respectively, the relation is

$$\langle \mathbf{x} \cdot \mathbf{x} \rangle = 2\nu \cdot Dt = 2\nu \frac{k_B T}{\zeta} t,$$

where $\nu = 2, 3$.

If an external force, F, acts on the protein, this can be added to (12.19), so that the equation of motion for a protein becomes $\zeta \cdot dx/dt = F(x,t) + f_B(t)$ (the inertial term has been neglected; see Section 12.6.1). In general, forces acting on proteins can be characterized by a potential, $F(x,t) = -\partial\phi(x,t)/\partial x$, so the equation of motion for a protein moving through a fluid becomes:

$$\zeta \frac{dx}{dt} = -\frac{\partial \phi(x,t)}{\partial x} + f_B(t). \tag{12.28}$$

(12.28) is frequently referred to as a *Langevin equation*, although this term more properly applies to the corresponding (12.19) that includes inertia.

12.2.2 Numerical simulation of the Langevin equation

Here we show how the stochastic algorithms developed above can be applied to a continuous Markov process describing a protein diffusing in water. According to Langevin's equation (12.28). We want a numerical algorithm for generating sample paths of (12.28). Let us integrate both sides of this equation over the interval $(t, t+\Delta t)$:

$$\begin{aligned} x(t+\Delta t) &= x(t) - \frac{1}{\zeta}\int_t^{t+\Delta t} \frac{\partial \phi(x,t')}{\partial x} dt' + \frac{1}{\zeta}\int_t^{t+\Delta t} f_B(t')dt' \\ &\approx x(t) - \frac{1}{\zeta}\frac{\partial \phi(x,t)}{\partial x}\Delta t + \frac{1}{\zeta}\int_t^{t+\Delta t} f_B(t')dt'. \end{aligned} \tag{12.29}$$

In Section 12.6.3 we demonstrate that the way to include the effect from the Brownian force, $f_B(t)$, is to use the following numerical method for simulating (12.29):

$$x(t+\Delta t) \approx x(t) - \frac{1}{\zeta}\frac{\partial \phi}{\partial x}\Delta t + \sqrt{2D\Delta t}Z, \tag{12.30}$$

where Z is a standard normal random variable, i.e. with mean 0 and variance 1. Many numerical software packages have built-in random number generators that will generate samples of a standard normal distribution. If one is not available, then a standard normal random variable Z can be generated from two independent uniform random variables R_1 and R_2:

$$Z = -\sqrt{-2\ln R_1}\cos(2\pi R_2). \tag{12.31}$$

A derivation of this result is similar to the one presented above for generating an exponential random variable.

Although simulating (12.30) on a computer is easy (see Exercise 6), it is also easy to generate erroneous results, e.g. numerical instabilities which look very like random

displacements due to Brownian motion, or currents that do not vanish at equilibrium. In order to derive a numerical method of simulating random motions that does not have these problems, we have to consider an alternative description of the molecular motion.

12.2.3 The Smoluchowski model

Consider a protein moving under the influence of a constant external force, for example, an electric field. Because of Brownian motion, no two trajectories will look the same. Moreover, even a detailed examination of the path cannot distinguish whether a particular displacement "step" was caused by a Brownian fluctuation or the effect of the field. Only by tracking the particle for a long time and computing the average position vs. time can one detect that the diffusion of the particle exhibits a "drift velocity" in the direction of the force. Therefore, a better way to think about stochastic motion is to imagine a large collection of independent particles moving together. Then we can define the *concentration* of particles at position x and time t as $c(x,t)$ [#/nm], and track the evolution of this *ensemble*.

As the cloud of particles diffuses and drifts, we can write an expression for the *flux* of particles passing through a unit area, J [#/area/time]; in one dimension J_x has dimensions [#/sec]. The diffusive motion of the particles is modeled well by Fick's law: $J_x = -D\partial c/\partial x$. The external field exerts a force on each particle, $F = -\partial\phi/\partial x$ which, in the absence of any diffusive motion, would impart a drift velocity proportional to the field: $v = F/\zeta$. Thus the motion of the body is the sum of the Brownian diffusion and the field-driven drift: $J_x = -D\partial c/\partial x + v \cdot c$, which can be written in several ways (see also Section 12.6.4:

$$J_x = \underbrace{-D\frac{\partial c}{\partial x}}_{\text{Diffusion flux}} - \underbrace{\overbrace{(\frac{D}{k_BT}\cdot\frac{\partial\phi}{\partial x})}^{\text{Drift velocity}}\, c}_{\text{Drift flux}} = -D(\frac{\partial c}{\partial x} + \frac{\partial(\phi/k_BT)}{\partial x}\cdot c)$$

$$= -\frac{1}{\zeta}(k_BT\frac{\partial c}{\partial x} + c\frac{\partial\phi}{\partial x}). \qquad (12.32)$$

At equilibrium the flux vanishes: $(k_BT/c_{eq})(\partial c_{eq}/\partial x)+\partial\phi/\partial x = 0$. Integrating with respect to x, shows that the concentration of particles at equilibrium in an external field, $\phi(x)$, is given by the *Boltzmann distribution*:

$$c_{eq} = c_0 e^{-\phi/k_BT}. \quad \text{[Boltzmann distribution]} \qquad (12.33)$$

Since the number of particles in the swarm remains constant, $c(x,t)$ must obey a *conservation law*. This is simply a balance on a small volume element, Δx:

$$\frac{\partial}{\partial t}(c\Delta x) = \text{Net Flux into } \Delta x = J_x(\text{in}) - J_x(\text{out}) = J_x(x) - J_x(x+\Delta x)$$

or, taking the limit as $\Delta x \to 0$:

$$\frac{\partial c}{\partial t} = -\frac{\partial J_x}{\partial x}. \quad \text{[conservation of particles]} \tag{12.34}$$

Rather than focussing our attention on the swarm of particles, we can rephrase our discussion in terms of the *probability* of finding a *single* particle at (x,t). To do this we normalize the concentration in (12.32) by dividing by the total population, $p(x,t) \equiv c(x,t)/(\int_0^L c(x,t)dx)$. Inserting (12.32) expressed in terms of $p(x,t)$ into the conservation law ((12.34)) yields the *Smoluchowski equation*:

$$\frac{\partial p}{\partial t} = D[\underbrace{\frac{\partial}{\partial x}(p\frac{\partial(\phi/k_BT)}{\partial x})}_{\text{Drift}} + \underbrace{\frac{\partial^2 p}{\partial x^2}}_{\text{Diffusion}}]. \quad \text{[Smoluchowski Equation]} \tag{12.35}$$

Comparing this with the Langevin equation 12.28 shows that the Brownian force is replaced by the diffusion term and the effect of the deterministic forcing is captured by the drift term.

We can nondimensionalize (12.35) by defining time and space scales. If the domain $0 \le x \le L$, the spatial variable can be normalized as x/L. A time scale can be defined by $\tau = L^2/D$. Introducing the space and time scales, (12.35) can be written in dimensionless form as

$$\frac{\partial p}{\partial t} = \frac{\partial}{\partial x}(p\frac{\partial \phi}{\partial x}) + \frac{\partial^2 p}{\partial x^2}, \tag{12.36}$$

where where t and x are now dimensionless, and the potential, ϕ, is measured in units of k_BT.

(12.36) must be augmented by appropriate boundary conditions specifying the value of $p(x=0,t), p(x=L,t)$, and $p(x,t=0)$, where $p(x,t)$ is defined on the interval $[0,L]$. These will depend on the system being modeled.

12.2.4 First passage time

A very useful quantity in modeling protein motions is the average time it takes for a diffusing protein to first reach an absorbing boundary located at $x = L$, starting from position $0 \le x \le L$ [Berg, 1993, Weiss, 1967]. Denote the mean first passage time (MFPT) to position L starting from position x by $T(x,L)$. The equation governing $T(x,L)$ is derived as follows. A particle released at position x can diffuse either to the right or to the left. After a time τ, it covers an average distance Δ, so that it is located at $x \pm \Delta$ with equal probability $1/2$. The MFPT to L from the new positions are $T(x+\Delta, L)$ and $T(x-\Delta, L)$. The average value of $T(x,L)$ is just $T(x,L) = \tau + (1/2)[T(x+\Delta,L) + T(x-\Delta,L)]$. This equation can be re-written in the form:

$$\frac{1}{\Delta}(\frac{(T(x+\Delta,L) - T(x))}{\Delta} - \frac{(T(x) - T(x-\Delta,L))}{\Delta}) + \frac{2\tau}{\Delta^2} = 0.$$

Taking the limit as $\Delta \to 0$ and $\tau \to 0$ (so that $\Delta^2/2\tau = \text{const}$) and recognizing that $\Delta^2/2\tau$ is just the diffusion coefficient, D, the MFPT equation becomes:

$$D\frac{\partial^2 T}{\partial x^2} = -1. \quad \text{[MFPT Equation]} \tag{12.37}$$

The boundary conditions for this equation are simple. At an absorbing boundary, $T = 0$ (it takes no time to get there). At a reflecting boundary, T is unchanged (i.e. a constant), so $\frac{\partial T}{\partial x} = 0$. For example, releasing a particle at a position x with a reflecting boundary at $x = L$ and an absorbing boundary at $x = 0$, the MFPT is $T(x, L) = (2Lx - x^2)/2D$. The special case when $x = L$ (releasing the particle at the reflecting boundary) is just $T = L^2/2D$. Note the resemblance to the familiar equation $\langle x^2 \rangle = 2Dt$. This gives the mean squared distance diffused in time t, whereas the MFPT gives $\langle t \rangle = x^2/2D$, the mean time to diffuse a distance x. This suggests that the MFPT equation might be related to the Smoluchowski diffusion equation; in fact they are adjoints of one another (see, for example [Lindenberg and Seshadri, 1979]). We will use this result to compute the average velocity of the perfect Brownian ratchet in the next chapter.

12.3 Modeling chemical reactions

So far, we paid attention exclusively to protein mechanics. To understand molecular motors, we have to consider chemical reactions, which supply the energy to drive molecular motors. Two of the most common energy sources are nucleotide hydrolysis and transmembrane protonmotive force. The former uses the energy stored in the covalent bond that attaches the terminal phosphate (γ-phosphate) to the rest of the nucleotide. The latter uses the electrical and entropic energy arising from a difference in ion concentrations across a lipid bilayer. Hydrolysis is a complicated process, still incompletely understood. Therefore, we will introduce the reaction model using the simple example of a positively charged ion (e.g. H^+) binding to a negatively charged amino acid: $H^+ + A^- \longleftrightarrow H \cdot A$. If we focus our attention on the amino acid, we see it exists in two states: charged (A^-) and neutral ($H \cdot A \equiv A^0$, so that the neutralization reaction from the viewpoint of the amino acid is simply

$$A^- \underset{k_{-1}}{\overset{k_1 \cdot [H^+]}{\rightleftharpoons}} A^0 \tag{12.38}$$

Here we use the chemists' convention of denoting concentrations in brackets: $k_1 \cdot [H^+]$ and k_{-1} are the forward and reverse rate constants; the forward rate constant depends on the ion concentration, $[H^+]$, which we will treat as a constant parameter (i.e. we shorten our notation to $k_1 \cdot [H^+] \equiv k_1^*$, where k_1^* is called a *pseudo*-first order rate constant).

The rate constants in reaction (12.38) conceal a great deal of physics, for the process of even as simple a reaction as this is quite complex at the atomic level. To model this reaction at a more microscopic level involves introducing additional coordinates to describe the process by which an ionic chemical bond is made and broken. These coordinates have a spatial scale much smaller than the motion of the motor itself (e.g. angstroms vs. nanometers), and a time scale much faster than any motion of the motor (picoseconds vs. microseconds). This is because all reactions involve a redistribution of electrons, and electrons, being very small, move very rapidly. Moreover, in all but the simplest cases, their movements are governed by quantum mechanics rather than classical mechanics. Nevertheless, it is instructive to use the Smoluchowski model to derive a more detailed expression for the rate constants. A deeper discussion can be found in [Billing and Mikkelsen, 1996, Warshel, 1991, Naray-Szabo and Warshel, 1997].

The fundamental concept underlying the modeling of reactions is the notion of a "reaction coordinate," which we denote by ξ. In molecular dynamics simulations, this is actually a 1-D path through a very high dimensional state space along which the system moves from reactants to products [Billing and Mikkelsen, 1996, Warshel, 1991, Naray-Szabo and Warshel, 1997]. For the reaction (12.38), $\xi(t)$ is the distance between the ion (H^+) and the amino acid charge, (A^-). The spatial scale of this coordinate is much smaller (i.e. angstroms) than the spatial scale of the motor's motion, but we can imagine a "super-microscopic" view of the process as shown in Fig. 12.5A, where we have plotted the free energy change, ΔG, during a reaction as a function of the reaction coordinate, ξ. The reason for using free energy is because there are many "hidden" degrees of freedom that must be handled statistically, as will become clear presently. Here the chemical states of the amino acid, $A^-$and A^0, are pictured as energy wells separated by barriers of heights ΔG_1 and ΔG_2, and whose difference in depth is ΔG. The "transition state" (TrSt) is located at the top of the pass between the two wells.

For a fixed H^+concentration, the forward chemical reaction $A^- \rightarrow A^0$ proceeds with a rate $k_1^* \cdot [A^-]$ [#/sec]. However, this rate is a statistical average over many "hidden" events. For a particular reaction to take place, the proton must diffuse to within a few angstroms of the amino acid charge so that the electrostatic attraction between them is felt. Moreover, if the amino acid is located within a protein, there will be steric diffusion barriers that must be circumvented before the two ions "see" each other electrostatically. (Actually, protons inside proteins move by "hopping" along strings of water molecules, or "water wires.") As the concentration of H^+increases, there will be more "tries" at neutralization (i.e. hops from the left well to the right well).

Similarly, the reverse reaction, $A^- \leftarrow A^0$ takes place when a thermal fluctuation confers enough kinetic energy on the proton to overcome the electrostatic attraction. Even then, the "free" proton will more often than not "jump" back and rebind to the amino acid, especially if the route between the solution and the amino acid is tortuous. Only when the proton manages a successful escape into solution (the left well) does it count in computing k_{-1}.

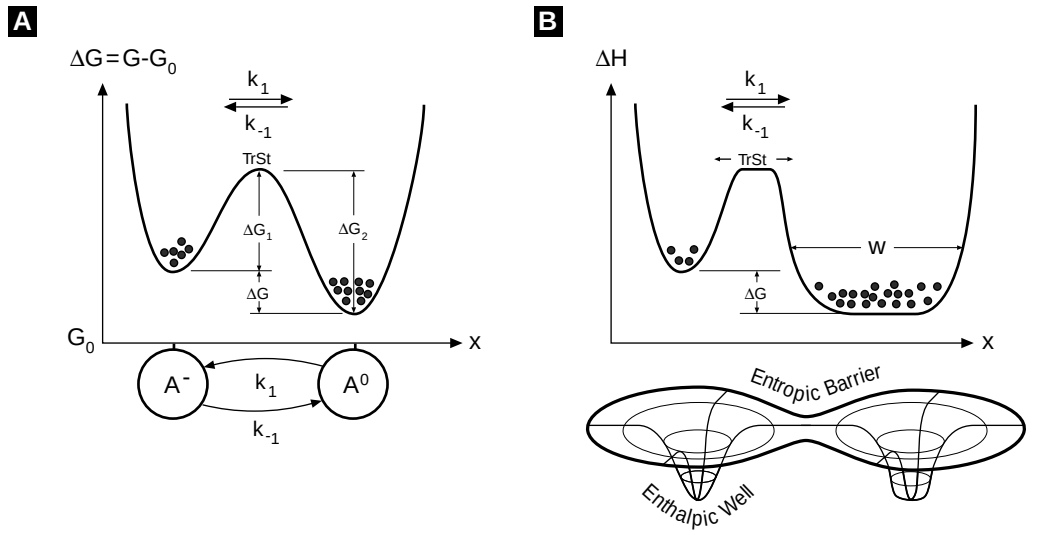

Figure 12.5 (A) Free energy diagram illustrating the chemical reaction $A \longleftrightarrow B$ and the corresponding Markov model. The transition state, TrSt, is ΔG_1^+ above the left well and ΔG_2^+ above the right well. ΔG is the free energy difference between the well bottoms. The equilibrium distribution between the wells depends only on ΔG. (B) The effect of entropic factors on the reaction $A \longleftrightarrow B$. Potential, rather than free, energy is shown as the fuction of ξ, effective one-dimensional reaction coordinate that involves concerted changes in both the chemical state and physical position along the path of the chemical reaction. The equilibrium populations in each well remain the same, but the transition rates between the wells are different due to the entropic effects of widening the transition state, TrSt, and the width, W, of the right well.

The *net* flux over the barrier is

$$J_\xi = k_1^* \cdot [\mathrm{A}^-] - \mathrm{k}_{-1} \cdot [\mathrm{A}^0]. \tag{12.39}$$

After a long time the net flux between the two wells will vanish: $J_\xi = 0$, so that the population of neutral and charged sites will distribute themselves between the wells in a fixed ratio, which we denote by K_{eq} (the equilibrium constant): $K_{eq} \equiv [\mathrm{A}^0_{\mathrm{eq}}]/[\mathrm{A}^-_{\mathrm{eq}}] = \mathrm{k}_1^*/\mathrm{k}_{-1}$. If the transition state is high, then we can assume that population apportions between the two wells according to the Boltzmann distribution (12.33): $K_{eq} = \exp(\Delta G/k_B T)$, where ΔG is a free energy. The value of ΔG determines *how far the reaction goes*, but says nothing about the *rate* of the reaction. Now we know that $\Delta G = \Delta H - T\Delta S$ (c.f. Section 12.6.4). The enthalpy term, ΔH, is due to the electrostatic attraction between the proton and the charged site. The entropic term, $T\Delta S$, incorporates all the effects that influence the diffusion of the proton to the site and its escape from it, the "hidden coordinates." Thus we see that a thermodynamic equilibrium state $\Delta G = 0$ comprises a compromise between energy (ΔH) and randomness ($T\Delta S$). The role of entropic factors is discussed further in Section 12.6.5.

There is one very significant effect in bio chemical reactions that illustrates the importance of entropic effects: hydration. Before a charged ion can bind to the amino acid, it must divest itself of several "waters of hydration". This is because water, being a dipole, will tend to cluster about ions in solution, hindering them from binding to a charged site which is also insulated by its own hydration shell. Suppose for the sake of illustration that the energies binding the waters to the two reactants are just equal to the electrostatic energy of binding between the reactants. Binding seems unfavorable since the ion will loose its translational and rotational degrees of freedom ($\sim 3k_BT$ according to the equipartition theorem, Section 12.6.2). The binding reaction can still proceed strongly because the liberation of the hydration waters is accompanied by a large entropy increase since each water gains $\sim 3k_BT$ of rotational and translational energy, and so the term $-T\Delta S$ is strongly negative.

All of this means that the rate constants summarize the statistical behavior of a large number of "hidden" coordinates that are very difficult to compute explicitly, but may be easy to measure phenomenologically (see, for example [Hanggi et al., 1990]). For our purposes, we shall adopt this phenomenological view of chemical reactions, and assume that the rate constants can be specified, so that the only entropic effect we need to deal with explicitly is the concentrations of the reactants, such as H^+in (12.38). Therefore, we can treat reactions using Markov chain theory, as indicated by the 2-state model shown at the bottom of Fig. 12.5A, whose equations of motion are:

$$\frac{d[\mathrm{A}^0]}{\mathrm{dt}} = -\frac{d[\mathrm{A}^-]}{\mathrm{dt}} = \text{net flow over the energy barrier} = J_\xi = k_1^*[\mathrm{A}^-] - \mathrm{k}_{-1}[\mathrm{A}^0],$$

or in the vector form:

$$\frac{d}{dt}\mathbf{P} = \mathbf{J}_\xi = \mathbf{K}\cdot\mathbf{P}, \quad \mathbf{P} = \begin{pmatrix} p_- \\ p_0 \end{pmatrix}, \quad \mathbf{K} = \begin{pmatrix} k_1^* & -k_{-1} \\ -k_1^* & k_{-1} \end{pmatrix}. \tag{12.40}$$

Here p_- and p_0 are the probabilities to have a negatively charged and neutral amino acid, respectively. In general, the reaction flux will have the form $\mathbf{J}_\xi = \mathbf{K}(\mathbf{P})\cdot\mathbf{P}$, where the matrix $\mathbf{K}(\mathbf{P})$ is the matrix of transition rates, i.e. pseudo-first order rate constants which may contain reactant concentrations that are held parametrically constant. Implicit in this formulation are the assumptions that (i) the actual reaction takes place instantaneously (electronic rearrangements are very fast), so that a substance remains in a chemical state for an exponentially distributed mean time before jumping (reacting) to another state; (ii) the transition out of a state depends only on the state itself, and not on any previous history.

12.4 A mechanochemical model

An important generalization is necessary to model molecular motors. We have spoken of the potential, $\phi(x,t)$, that provides the deterministic forcing as an *external* force. However, for a molecular motor $\phi(x,t)$ generally includes forces generated *internally*

by the motor itself which drive the motor forward. Thus the potential term in (12.36) must be broken into two parts:

$$\phi(x,t) = \underbrace{\phi_I(x,t)}_{\text{Internally generated forces}} + \underbrace{\phi_L(x,t)}_{\text{External load forces}},$$

where $\phi_I(x,t)$ is internally generated force, and $\phi_L(x,t)$ is the external load force. A common situation is when F_L is a constant load force, in which case $\phi_L = F_L \cdot x$, so that $-\partial\phi/\partial x = -F_L$; i.e. the load force acts to oppose the motor's forward progress. The internally generated force potential will generally depend on the chemical state of the system. That is, the mechanical evolution of the system's geometrical coordinates governed by (12.36) is coupled to the chemical reactions described by a Markov chain (12.40). Each *chemical state* is characterized by its own probability distribution, $p_k(x,t)$, where k ranges over all the chemical states, and each chemical state is typically characterized by a separate driving potential, $\phi_k(x,t)$. Thus there will be a Smoluchowski equation (12.36) for each chemical state, and these equations must be solved simultaneously to obtain the motor's motion.

For the neutralization reaction considered above, the total change in probability, $p(x,\xi,t)$, is given by

$$\begin{aligned}
\frac{\partial}{\partial t}\begin{pmatrix} p_1 \\ p_2 \end{pmatrix} &= \text{Net flow in space} + \text{Net flow along reaction coordinates} \\
&= \overbrace{-\begin{pmatrix} (\partial/\partial x_1)J_{x_1} \\ (\partial/\partial x_2)J_{x_2} \end{pmatrix}} + \overbrace{\begin{pmatrix} J_{\xi_1} \\ J_{\xi_2} \end{pmatrix}} \\
&= -D\begin{pmatrix} -(\partial/\partial x_1)[p_1\partial(\phi_1/k_BT)/\partial x_1 + (\partial p_1/\partial x_1)] \\ -(\partial/\partial x_2)[p_2\partial(\phi_2/k_BT)/\partial x_2 + (\partial p_2/\partial x_2)] \end{pmatrix} + \begin{pmatrix} k_{-1}p_2 - k_1^* p_1 \\ k_1^* p_1 - k_{-1}p_2 \end{pmatrix},
\end{aligned} \tag{12.41}$$

where the probability densities $p_i(x_i,t), i = 1,2$ now keep track of the motion along the spatial and reaction coordinates, and $J_{\xi_1} = -J_{\xi_2}$ keeps track of flux along the reaction coordinate (since the reaction is first order, i.e. has only two states). We can visualize the mechanochemical coupling by plotting the spatial and reaction coordinates as shown in Fig. 12.6.

12.5 Numerical simulation of protein motion

We return to the problem of simulating the protein's motion numerically. (12.30) is a very useful and easy to implement numerical scheme. However, one of its shortcomings is that it does not preserve the property of detailed balance. Detailed balance is the constraint placed on $c_{eq}(x)$ to ensure that systems in equilibrium do not experience a net drift. That is, when a system is in equilibrium, J in (12.32) is required to be

identically zero. Detailed balance ensures that the equilibrium density has a Boltzmann distribution.

It is important to understand the distinction between steady state and equilibrium. If we watch sample paths of $x(t)$, it is consistent for the trajectories to move with a mean velocity and for the system to have a steady state. In equilibrium the sample paths must not exhibit a mean velocity. It can be shown that (12.30) produces sample paths that show a net drift when the real system satisfies detailed balance [Elston and Doering, 1995]. Clearly, it is desirable to have an algorithm that preserves detailed balance in equilibrium, and can be used to simulate both equilibrium and nonequilibrium processes.

12.5.1 Numerical algorithm that preserves detailed balance

To obtain an algorithm that has detailed balance built in, we convert the problem into a Markov chain and and use the procedures described above to numerically simulate it. The numerical algorithm given by (12.30) is based on the discretization of time. To convert the problem into a Markov chain requires that we discretize space. Let $x_n = (n - 1/2)\Delta x$ for $n = 0, \pm 1, \pm 2, \pm 3, ...$ be the discrete sites on which the protein can reside. Site x_n is represented by the interval $[x_n - \Delta x/2, x_n + \Delta x/2]$. That is, when the protein is in the interval $[x_n - \Delta x/2, x_n + \Delta x/2]$, we treat it as being at site x_n. If the molecule is at site x_n, then it can jump to either x_{n+1} or x_{n-1}. A diagram of this process is shown in Fig. 12.7. The notation that we have adopted is that $F_{n+1/2}$ is the rate at

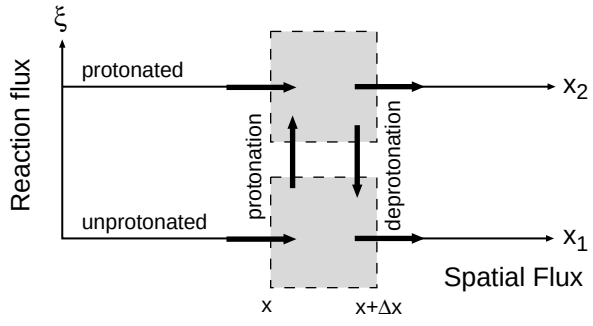

Figure 12.6 The mechanochemical phase plane. A point is defined by its spatial and reaction coordinates $(x(t), \xi(t))$. The flow of probability in the spatial direction is given by the Smoluchowski model (12.35), and the flow in the reaction direction is given by the Markov model (12.40).

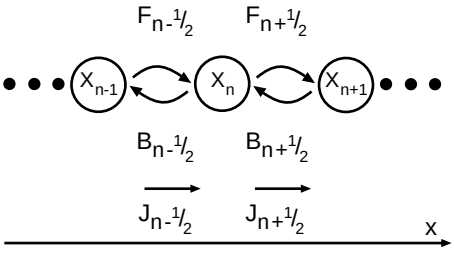

Figure 12.7 The numerical discretization in spatial dimension. A continuous Markov process (the Langevin equation) is approximated by a discrete Markov process. The particle is restricted to a set of discrete sites (x_n) and is allowed to jump only to the neighboring sites (x_{n-1}, x_{n+1}). The site x_n can be viewed as to represent the interval $[x_n - \Delta x/2, x_n + \Delta x/2]$.

which the protein jumps from x_n to x_{n+1} (F refers to a "forward" jump). Similarly $B_{n+1/2}$ is the rate at which the protein jumps from x_{n+1} to x_n (B for 'backward').

For small enough Δx, we have $p_n(t) \approx p(x,t)\Delta x$, where $p_n(t)$ is the probability that the protein is at x_n at time t. The governing equation for $p_n(t)$ is

$$\frac{dp_n}{dt} = -(B_{n-1/2} + F_{n+1/2})p_n + F_{n-1/2}p_{n-1} + B_{n+1/2}p_{n+1}$$
$$= (F_{n-1/2}p_{n-1} - B_{n-1/2}p_n) - (F_{n+1/2}p_n - B_{n+1/2}p_{n+1}) = J_{n-1/2} - J_{n+1/2}, \quad (12.42)$$

where $J_{n+1/2}$ is the net flux between the points x_n and x_{n+1}.

In addition to preserving detailed balance, our numerical scheme must approximate the actual dynamics of the protein. In Section 12.6.6 we demonstrate that the following jump rates preserve the mean drift motion as well as detailed balance:

$$F_{n+1/2} = \frac{D}{(\Delta x)^2} \cdot \frac{\frac{\Delta\phi_{n+1/2}}{k_BT}}{\exp\left(\frac{\Delta\phi_{n+1/2}}{k_BT}\right) - 1}, \quad (12.43)$$

$$B_{n+1/2} = \frac{D}{(\Delta x)^2} \cdot \frac{\frac{-\Delta\phi_{n+1/2}}{k_BT}}{\exp\left(\frac{-\Delta\phi_{n+1/2}}{k_BT}\right) - 1}, \quad (12.44)$$

where

$$\Delta\phi_{n+1/2} = \phi(x_{n+1}) - \phi(x_n). \quad (12.45)$$

12.5.2 Boundary conditions

The algorithm described above must be complemented by boundary conditions. We discuss three types of boundary conditions: periodic, reflecting and absorbing. In each case the total number of grid points within the interval is M, and $\Delta x = L/M$, where L is the length of the spatial domain. The placement of the grid has been chosen such that $x_n = (n - 1/2)\Delta x$ for $n = 1, 2 \ldots M$.

Periodic

Periodic boundary conditions require that $p_{M+1}(t) = p_1(t)$ and $p_0(t) = p_M(t)$. Using these two equalities in (12.42) for $p_1(t)$ and $p_M(t)$ produces:

$$\frac{dp_1}{dt} = -(B_{1/2} + F_{3/2})p_1 + F_{M+1/2}p_M + B_{3/2}p_2, \quad (12.46)$$

$$\frac{dp_M}{dt} = -(B_{M-1/2} + F_{M+1/2})p_M + F_{M-1/2}p_{M-1} + B_{1/2}p_1, \quad (12.47)$$

where we have also made use of the fact that $B_{1/2} = B_{M+1/2}$ and $F_{1/2} = F_{M+1/2}$.

Reflecting

Fig. 12.8A shows a reflecting boundary condition located midway between the grid points M and $M + 1$. A reflecting boundary requires that

$$J_{M+1/2}(t) = F_{M+1/2}p_M(t) - B_{M+1/2}p_{M+1}(t) = 0. \quad (12.48)$$

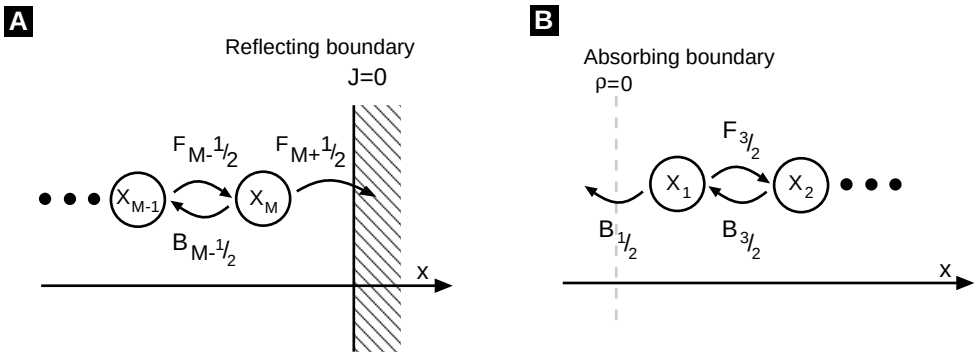

Figure 12.8 Numerical treatments for two types of boundaries. (A) At a reflecting boundary, the particle is not allowed to jump through the boundary and thus the flux through the boundary is zero. (B) At an absorbing boundary, the probability density is zero. Once the particle jumps out of the boundary, it should not be allowed to come back. However, in the numerical discretization, blocking the particle from coming back is not enough. The rate of the particle jumping out of the boundary has to be modified.

We know that $p_{M+1}(t) = 0$, since it is located outside of the reflecting boundary and is inaccessible to the protein. Therefore, to enforce no flux through the boundary, we set $F^{\text{reflect}}_{M+1/2} = 0$. The equation for p_M is then

$$\frac{dp_M}{dt} = -B_{M-1/2}p_M + F_{M-1/2}p_{M-1}. \tag{12.49}$$

Absorbing

If the protein reaches an absorbing boundary it is instantaneously removed from the solution. Therefore, the probability of finding the protein at an absorbing boundary is zero. Fig. 12.8B illustrates an absorbing boundary at $x = 0$. Thus, we must enforce the condition $p(0,t) = 0$. In **Appendix 7** we derive the appropriate jump rate at this boundary:

$$B^{\text{absorb}}_{1/2} = \frac{D}{(\Delta x)^2} \cdot \frac{\alpha^2}{\exp(\alpha) - 1 - \alpha}, \qquad \alpha = \frac{\phi_0 - \phi_1}{k_B T}. \tag{12.50}$$

The equation for p_1 is then:

$$\frac{dp_1}{dt} = -(B^{\text{absorb}}_{1/2} + F_{3/2})p_1 + B_{3/2}p_2. \tag{12.51}$$

It can be shown that this treatment of the absorbing boundary is accurate to second order in Δx and that it preserves the velocity of a perfect Brownian ratchet subject to any load force (see next Chapter).

Now we are ready to numerically integrate p_n. However, before we turn to examples of implementing the algorithm, we must address the issue of numerical stability and introduce an implicit method for the time integration as an alternative to Euler's method.

12.5.3 Numerical stability

For notational convenience let $p_n(k\Delta t) = p_n^k$. Then Euler's method has the form:

$$p_n^{k+1} \approx p_n^k - \Delta t \left[(B_{n-1/2} + F_{n+1/2}) p_n^k + F_{n-1/2} p_{n-1}^k + B_{n+1/2} p_{n+1}^k \right]. \tag{12.52}$$

Euler's method is called an explicit method because p_n^{k+1} can be written explicitly in terms of p_n^k. Each time we use the above technique to update p_n^k, we introduce a small error due to the finite size of Δt and round-off error. In the absence of round-off error, we can achieve any desired accuracy by decreasing Δt. However, there are some problems with this approach. Usually the biggest problem is the amount of computer time required when we chose a very small Δt. However, the round-off error incurred in each step does not decrease with Δt; rather it accumulates. It is possible that if Δt is too small, the total error is dominated by the round-off error. In that situation, the more steps we take the larger the accumulated error. So a careful choice of time step is important.

Numerical stability is another issue with which we have to contend. That is, we do not want our numerical solutions to run off to $\pm\infty$, when the real solution is bounded for all time. It is possible to show that Euler's method is stable only if

$$\Delta t < \Delta t_c = \max_n \left(\frac{1}{F_{n+1/2} + B_{n+1/2}} \right), \tag{12.53}$$

where max in the above equation means to use the value of n that produces the largest value of the quantity in the parentheses. Fig. 12.9 illustrates this change in stability by using time steps slightly above and below Δt_c. To get an intuitive feel for this instability,

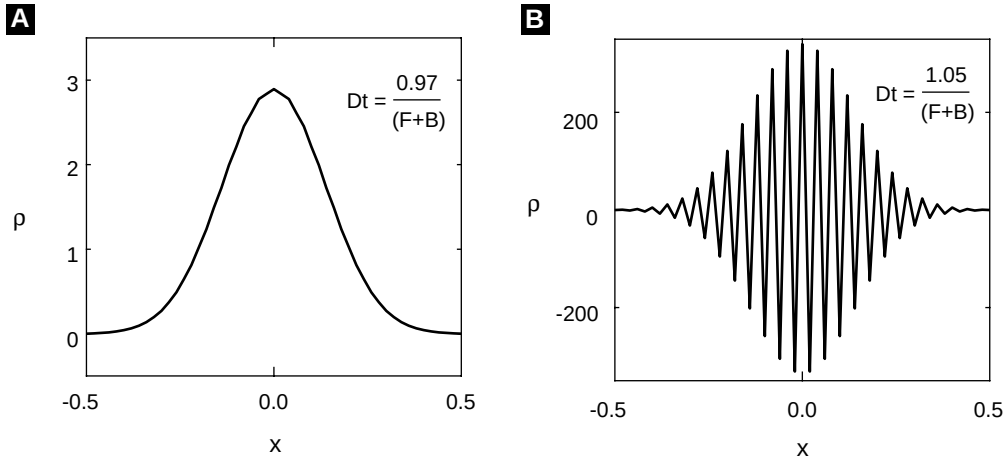

Figure 12.9 Numerical stability/instability. (A) When the time step is slightly below the critical step size, the numerical solution is stable. (B) When the time step is slightly above the critical step size, the numerical solution is unstable.

let us consider one time step of the numerical scheme. At $t = 0$, we take $p_m^0 = 1$ and $p_n^0 = 0$ for $n \neq m$. From (12.52) we have:

$$p_m^1 = [1 - \Delta t(B_{m-1/2} + F_{m+1/2})]\, p_m^0. \tag{12.54}$$

It is clear that if $\Delta t > \Delta t_c$, then p_m^1 will be negative. Since p_m^1 is a probability, negative values clearly do not make sense. The condition on Δt for stability is rather restrictive. Using the jump rates given in (12.84) and (12.85)), it is possible to show that

$$\Delta t_c < \frac{(\Delta x)^2}{2D}. \tag{12.55}$$

This implies that in order to reduce the spatial step by a factor of 10 (which could be necessary to model accurately spatial fluctuations of the force, for example), the time step must be reduced by a factor of 100.

12.5.4 Implicit discretization

We now improve upon Euler's method in two ways. First, we use a second order algorithm that improves the accuracy of the solution for fixed Δt. Secondly, we choose an implicit method that is unconditionally stable. The implicit second order algorithm we employ is called the Crank-Nicolson method. For a simple one dimensional differential equation $dx/dt = h(x)$, the Crank-Nicolson method has the form:

$$\frac{x^{k+1} - x^k}{\Delta t} = \frac{h(x^{k+1}) + h(x^k)}{2}. \tag{12.56}$$

For (12.42), this scheme becomes:

$$\begin{aligned}\frac{p_n^{k+1} - p_n^k}{\Delta t} = &-(B_{n-1/2} + F_{n+1/2}) \cdot \frac{p_n^{k+1} + p_n^k}{2} \\ &+ F_{n-1/2} \cdot \frac{p_{n-1}^{k+1} + p_{n-1}^k}{2} + B_{n+1/2} \cdot \frac{p_{n+1}^{k+1} + p_{n+1}^k}{2}.\end{aligned} \tag{12.57}$$

If we now bring all the p^{k+1} terms to the left-hand-side and use the vector notation:

$$\mathbf{p}^k = \begin{pmatrix} p_1^k \\ p_2^k \\ \vdots \\ p_M^k \end{pmatrix}. \tag{12.58}$$

(12.57) can be written in matrix form as

$$\mathbf{A}\ \mathbf{p}^{k+1} = \ \mathbf{C}\ \mathbf{p}^k, \tag{12.59}$$

where $\mathbf{A}$ and $\mathbf{C}$ are tridiagonal matrices with elements:

$$A_{nn} = 1 + \frac{\Delta t}{2}(B_{n-1/2} + F_{n+1/2}),\ A_{n,n-1} = -\frac{\Delta t}{2}F_{n-1/2},\ A_{n,n+1} = -\frac{\Delta t}{2}B_{n+1/2}, \tag{12.60}$$

and

$$C_{nn} = 1 - \frac{\Delta t}{2}(B_{n-1/2} + F_{n+1/2}),\ C_{n,n-1} = \frac{\Delta t}{2}F_{n-1/2},\ C_{n,n+1} = \frac{\Delta t}{2}B_{n+1/2}. \quad (12.61)$$

(12.59) reveals why this method is called implicit. At each time step we must solve a linear set of coupled equations. Luckily $\mathbf{A}$ is a sparse matrix and it is therefore computationally fast to solve (12.59) for $\mathbf{p}^{k+1}$.

12.6 Derivations and comments

12.6.1 The drag coefficient

The natural units of distance and force on the molecular scale are nanometers (1 nm = 10^{-9} m) and piconewtons (1 pN = 10^{-12} N), respectively. In these units, the viscosity of water at room temperature is $\eta \simeq 10^{-9}$ pN·sec/nm^2. Then, a typical value for the hydrodynamic drag coefficient of a sphere of radius $R = 10$ nm is $\zeta = 6\pi\eta R \simeq 10^{-7}$ pN·sec/nm. A dimensionless number that measures the ratio of inertial to viscous forces is the Reynolds number: $Re \equiv \rho v R/\eta$, where ρ is the density of water (10^3 kg/m^3 = 10^{-21} pN·sec^2/nm^4) [Happel and Brenner, l986, Berg, 1993]. Typical velocities of molecular motors are $v < 10^3$ nm/sec, so on the molecular scale, the Reynolds number is very small indeed: $Re \sim 10^{-8}$. This confirms our conjecture that we can safely ignore the inertial term in (12.19). If the fluid can truly be viewed as a continuum, then ζ can be computed from hydrodynamics [Happel and Brenner, l986]. The frictional drag coefficient ζ depends on the particle shape and size as well as the fluid viscosity: ζ =(dimensionless geometric drag coefficient)×(size factor)× (shape factor). For a sphere, $\zeta = 6\pi\eta R$; drag coefficients for other shapes are given in [Berg, 1993], a good source of intuition on Brownian motion.

12.6.2 The equipartition theorem

Let us consider a collision of two particles of masses m_1 and m_2 with velocities v_1 and v_2 before the collision, and with velocities v_1' and v_2' after the collision, respectively. Conservation of energy and momentum guarantee conservation of the velocity of the center-of-mass after the collision, as well as of the absolute value of the relative velocity [Feynman et al., 1963]. One of the central assumptions of statistical mechanics is that the velocities of the scattered particles are uncorrelated.

From this one can show that $\langle m_1 v_1'^2\rangle = \langle m_2 v_2'^2\rangle$. A more general result that can be derived from statistical mechanics is that each quadratic degree of freedom (e.g. linear or angular momentum) of a particle carries an average amount of energy $\langle E\rangle = k_B T/2$ [Reif, 1965]. Thus the mean kinetic energy of a point particle moving in the x direction is $\langle m v_x^2/2\rangle = k_B T/2$, or $\langle v_x^2\rangle = k_B T/m$. If the particle is moving in a harmonic potential well (i.e. on a spring), its mean potential energy is $k\langle x^2\rangle/2 = k_B T/2$. Thus the mean total energy is $\langle E\rangle = \langle E_{kin}\rangle + \langle E_{pot}\rangle = k_B T$.

12.6.3 A numerical method for the Langevin equation

For (12.29) to be useful, we must specify the statistical properties of $f_B(t)$. Since $f_B(t)$ models the net effect of many protein-water interactions, from the central limit theorem, it seems reasonable to assume that $f_B(t)$ is normally distributed. Physically, we also want $\langle f_B(t) \rangle = 0$, since a protein that is undergoing pure diffusion does not experience a net force. All that is left to fully characterize $f_B(t)$, is to specify its covariance $\text{cov}[f_B(t) f_B(s)]$. Remember a necessary condition for two random variables to be independent is that their covariance is zero. Since the motion of the water molecules is very fast as compared with the motion of the diffusing protein, we take $f_B(t)$ and $f_B(s)$ to be statistically independent whenever $t \neq s$. When $\phi = 0$, we should recover diffusive motion. That is, the variance of Brownian particle started at $x(0) = 0$ is

$$\text{Var}[x(t)] = \langle x(t)^2 \rangle = 2Dt. \tag{12.62}$$

We claim that an appropriate choice for the covariance is

$$\text{cov}[f_B(t) f_B(s)] = \langle f_B(t) f_B(s) \rangle = 2k_B T \zeta \delta(t-s). \tag{12.63}$$

The Dirac delta function $\delta(t-s)$ in (12.63) is a mathematical concept that is best understood as the limit of a normal distribution centered at s as the variance goes to zero:

$$\delta(t-s) = \lim_{\sigma \to 0} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(t-s)^2}{2\sigma^2} \right). \tag{12.64}$$

This is illustrated in Fig. 12.10. The only property of the Dirac delta function that we need is $\int_{-\infty}^{\infty} g(t)\delta(t-s)dt = g(s)$, which is easily understood when the Dirac delta function is interpreted as a sharply peaked probability density. We now have a complete description of $f_B(t)$. A random variable described as such is referred to as *Gaussian white noise*.

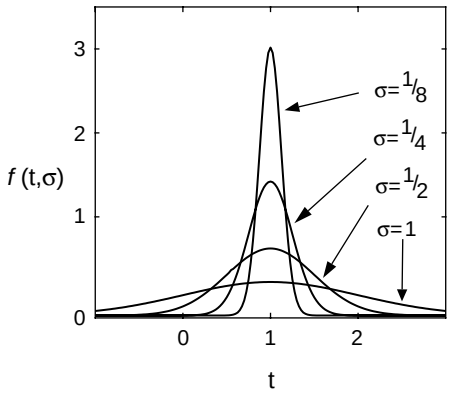

Figure 12.10 The delta function can be viewed as the limit of a sequence of normal probability density functions as the standard deviation goes to zero.

Next we illustrate that our choice of $f_B(t)$ makes sense both mathematically and physically. Note that from (12.28) with $\phi = 0$ and $x(0) = 0$, we have:

$$x(t) = \frac{1}{\zeta}\int_0^t f_B(t')dt', \tag{12.65}$$

from which it is clear that $\langle x(t)\rangle = 0$. Assume without loss of generality that $t > s$. One can show (from the theory of Dirac's delta function) that:

$$\begin{aligned}\mathrm{cov}[x(t)x(s)] &= \frac{1}{\zeta^2}\langle\left(\int_0^t f_b(t')dt'\right)\left(\int_0^s f_b(t'')dt''\right)\rangle \\ &= 2D\int_0^t\int_0^s \delta(t'-t'')dt'dt'' = 2Ds,\end{aligned} \tag{12.66}$$

which is consistent with (12.62), when $s = t$. Furthermore, since $f_B(t)$ is a normal stochastic variable, so is $x(t)$.

If we define the new random variable $W(t) = x(t)/\sqrt{2D}$, then $W(t)$ is a normal random variable characterized by

$$\langle W(t)\rangle = 0, \quad \mathrm{cov}[W(t)W(s)] = \min(t,s), \tag{12.67}$$

where the min in (12.67) means use the value of t or s that is smaller. The random variable $W(t)$ is referred to as a *Weiner process* and possesses some interesting mathematical properties, which we will not go into here. Note that from (12.29) what we need for our numerical algorithm is actually the incremental Weiner process defined as

$$\Delta W(t) = W(t+\Delta t) - W(t) = \frac{1}{\sqrt{2k_BT\zeta}}\int_t^{t+\Delta t} f_B(t')dt'. \tag{12.68}$$

Following a procedure similar to that used in (12.66), it is straightforward to show that $\Delta W(t)$ is a normal random variable with mean zero and standard deviation $\sqrt{\Delta t}$. It is also possible to show that all $\Delta W(t)$ and $\Delta W(s)$ are statistically independent for $t \neq s$. This gives us a way to generalize Euler's method to include Gaussian white noise. That is, a numerical method for simulating (12.28) is (12.30).

12.6.4 Some connections with thermodynamics

Note that the flux ((12.32)) can also be written as $J_x = -(c/\zeta)\partial/\partial x(k_BT\ln c + \phi)$. There can be many *steady states* characterized by a constant flux: $J_x = \text{const}$; one of these is the special case of *equilibrium*: $J_x = 0$. At equilibrium, one can define the quantity $\mu = (k_BT\ln c_{eq} + \phi)$ called the *chemical potential*. The equilibrium distribution of $c_{eq}(x)$ can be computed by setting the gradient in chemical potential to zero, so that $\mu =$const; this is exactly equivalent to enforcing a Boltzmann distribution, ((12.33)).

The chemical potential is also the free energy *per mole*, $G = \mu N$, where N is the mole number. A mole is an Avogadro's number N_a of objects (e.g. molecules), where $N_a = 6.02 \cdot 10^{23}[\#/\text{mol}]$. At *equilibrium* we can define the *entropy*, $S \equiv -k_BN\ln c_{eq}$ and the *enthalpy* as $H = \phi N$. Then we arrive at the definition of the free energy:

$G = H - TS$. These definitions will prove useful when we discuss chemical reactions. Here we note simply that diffusion smoothes out the concentration leading to an increase in entropy. Thus entropic increase accompanying the motion of the ensemble is handled by the Fickian diffusion term in the flux ((12.32)).

When the particles are charged (e.g. protons, H^+), then the chemical potential difference between two states, or across a membrane, is written as: $\Delta\mu = \mu_2 - \mu_1 = (\phi_2 - \phi_1) + k_B T(\ln c_2 - \ln c_1) = \Delta\phi - 2.3\ k_B T\ \Delta\text{pH}$. Here $\text{pH} = -\log_{10} c_{H^+}$, where c_{H^+} is the proton concentration. The protonmotive force is defined as p.m.f. $= \Delta\mu/e = \Delta\psi - 2.3\ (k_B T/e)\ \Delta\text{pH}$, where e is the electronic charge, and $\Delta\psi = \Delta\phi/e$, [mV], is the transmembrane electric potential.

Consider the simple case when a protein motor is propelled by an internally generated motor force, f_M, and opposed by a constant load force, f_L. For example, $f_M = \Delta G/l$, where l is the length of the power stroke and ΔG is the free energy drop accompanying one cycle of the chemical reaction that is supplying the energy to the motor. (This would be an ideal motor: it uses *all* of the chemical energy to produce a constant force power stroke!) Then the Langevin equation (12.19) becomes

$$\frac{dx}{dt} = v, \quad m\frac{dv}{dt} = -\zeta v + f_M - f_L + f_B(t). \tag{12.69}$$

The diffusion equation associated with (12.69) for the probability density $p(x, v, t)$ is called the *Kramers equation*:

$$\frac{\partial p}{\partial t} = -\frac{\partial}{\partial x}(vp) - \frac{1}{m}\frac{\partial}{\partial v}[((f_M - f_L) - \zeta v)p - (\zeta\frac{k_B T}{m})\frac{\partial p}{\partial v}]. \tag{12.70}$$

The Smoluchowski diffusion equation (12.35) is a special case of the Kramers equation; both are generically referred to as *Fokker-Planck equations* [Doi and Edwards, 1986, Risken, 1989, Gardiner, 1985, Reif, 1965]. However, deriving (12.35) from (12.70) is not trivial: it requires a *singular perturbation treatment* that is beyond the scope of this chapter [Doering, 1990, Risken, 1989].

We set $\partial p/\partial t = 0$ in (12.70) to look for the steady state. Multiplying by v^2 and taking the average by integrating over x and v, and using the equipartition theorem, we obtain:

$$0 = \underbrace{-\zeta < v^2 >}_{1} + \underbrace{F_M < v >}_{2} - \underbrace{F_L < v >}_{3} + \underbrace{\frac{k_B T}{m}\zeta}_{4}. \tag{12.71}$$

At constant temperature, the terms in (12.71) (see also Fig. 12.11) have the following interpretation:

- The rate at which the motor dissipates energy via frictional drag with the fluid.
- The rate at which energy is being absorbed by the motor from the chemical reaction.
- The rate of work done by the load force against the motor.
- The rate at which the motor absorbs energy from the thermal fluctuations of the fluid.

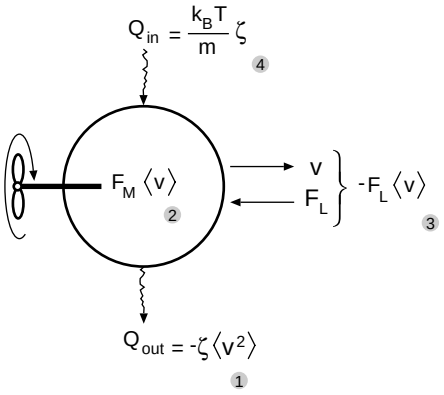

Figure 12.11 Energy balance on a protein motor. Q_{out} is the heat dissipated by the motion of the motor (term 1). Q_{in} is the heat supplied to the motor by thermal fluctuations of the fluid (term 4). The rate of work done by the load force is $-F_L \cdot < v >$ (term 3). The rate of work done by the chemical reaction driving the motor is $F_M \cdot < v >$ (term 2).

Thus we see that when the chemical reaction is turned off, $\Delta G = 0$, the heat absorbed by the motor from the thermal environment (term 4) is just equal to the heat returned to the environment by frictional drag (term 1) in the absence of the load force. If the reaction driving the motor were *endothermic*, then it is possible for the motor to move by taking heat from the environment without violating the 2^{nd} Law of Thermodynamics.

12.6.5 Jumping beans and entropy

An analogy may make the role of entropic factors more clear. Imagine that the left enthalpic well in Fig. 12.5B is filled with Mexican jumping beans whose hops are random in height and angle. We can vary the equilibrium populations of beans in each well without altering the height of the enthalpic barrier by simply increasing the width of the transition state or of one of the wells. This is shown in Fig. 12.5B. Now a bean in the right well may execute many more futile jumps before hurdling the barrier: if it jumps from the right side of the well it will fall back into the well even if its jump is high enough, or if it reaches the transition state it must diffuse (hop) along the plateau randomly with a high probability of hopping back into the right hand well. Both of these effects make it more difficult to escape from the right well, and so the equilibrium population there will increase, as will K_{eq}, the equilibrium population ratio.

The rate at which beans can pass the barrier from right to left will have the form

$$k_{-1} = \nu \cdot \exp(\Delta G_2^+ / k_B T) =$$

$$\underbrace{\nu}_{1} \cdot \underbrace{\exp(\Delta S / k_B)}_{2} \cdot \underbrace{\exp(-\Delta H / k_B T)}_{3} \cdot \underbrace{\exp(-\Delta F_L L / k_B T)}_{4},$$

where (1) ν is a frequency factor (number of jumps/unit time). For reactions that involve an atomic vibration, this is approximately $k_B T/\hbar$, where $\hbar$ is Planck's constant. For diffusion controlled reactions this is of order D/L^2, where D is the diffusion coefficient and L a characteristic dimension. (2) The entropic term, $e^{\Delta S/k_B}$, accounts for

geometric and "hidden variables" effects. (3) The enthalpic term, $e^{-\Delta H/k_B T}$, is a free energy "payoff" for a successful jump; it accounts for the electrostatic and/or hydrophobic interactions. (4) If the reaction involves a mechanical step that is opposed by a load force, F_L, then the fourth term accounts for the penalty exacted by performing work against the load. All of these effects are contained in the kinetic rate constants, and can estimated from more detailed models [Hanggi et al., 1990, Risken, 1989].

Note that the height of the free energy barrier, $\Delta G_2^\dagger$ determines *how fast the reaction goes*. The exponent $\exp(\Delta G_2^\dagger/k_B T)$ is called the *Arrhenius factor*. Because of this factor the reaction rate depends dramatically on the height of the free energy barrier.

12.6.6 Jump rates

Here we determine the appropriate values of $F_{n+1/2}$ and $B_{n+1/2}$ in (12.42). Suppose the system, with proper boundary restrictions, attains an equilibrium as time goes to infinity: $p_n^{(eq)} = \lim_{t\to\infty} p_n(t)$. Then the rates should preserve the property of detailed balance. That is,

$$J_{n+1/2} = F_{n+1/2} p_n^{(eq)} - B_{n+1/2} p_{n+1}^{(eq)} = 0. \tag{12.72}$$

Making use of the equilibrium distribution given by (12.33), we have

$$\frac{F_{n+1/2}}{B_{n+1/2}} = \frac{p_{n+1}^{(eq)}}{p_n^{(eq)}} = \frac{p_{eq}(x_{n+1})}{p_{eq}(x_n)} = \exp\left(\frac{-\Delta\phi_{n+1/2}}{k_B T}\right). \tag{12.73}$$

where

$$\Delta\phi_{n+1/2} = \phi(x_{n+1}) - \phi(x_n). \tag{12.74}$$

(12.73) is our first constraint on the jump rates.

Besides preserving detailed balance, our numerical scheme must of course approximate the actual dynamics of the protein. Let us consider the two simplest statistical properties of the random variable $x(t)$, namely, the mean and the variance. To simplify the presentation, we make the assumption that $\phi(x) = -fx$. That is, our Brownian particle feels a constant force of strength f. For this problem (12.28) reduces to:

$$\frac{dx}{dt} = \frac{f}{\zeta} + \frac{f_B(t)}{\zeta}. \tag{12.75}$$

Assuming $x(0) = 0$, the above equation can be integrated to produce:

$$x(t) = \frac{f}{\zeta} t + \frac{1}{\zeta}\int_0^t f_B(s)\, ds. \tag{12.76}$$

Using (12.76) the mean and variance of $x(t)$ are found to be:

$$\langle x(t)\rangle = \frac{f}{\zeta} t, \tag{12.77}$$

$$\mathrm{Var}[x(t)] = 2Dt. \tag{12.78}$$

Remember in the discrete version, $x(t) = \Delta x n(t)$. Since the force acting on the protein is constant, the forward and backward rates are independent of n. Therefore, we drop the subscripts and use F and B. Using (12.42), it is straightforward to show that

$$\langle x(t) \rangle = \Delta x \langle n(t) \rangle = (F - B)t, \tag{12.79}$$

$$\mathrm{Var}[x(t)] = (\Delta x)^2 \mathrm{Var}[n(t)] = (\Delta x)^2 (F + B)t. \tag{12.80}$$

Equating the mean and the variance given in (12.77) and (12.78) with those of (12.79) and (12.80) gives us two more constraints on the jump rates. To summarize, we would like F and B to satisfies the following three equations:

$$\frac{F}{B} = \exp\left(\frac{f\Delta x}{k_B T}\right), \quad \text{[Detailed Balance]} \tag{12.81}$$

$$(F - B)\Delta x = \frac{f}{\zeta}, \quad \text{[Mean]} \tag{12.82}$$

$$(F + B)(\Delta x)^2 = 2D. \quad \text{[Variance]} \tag{12.83}$$

Generally, it is impossible to satisfy three equations with two unknowns. Let us ignore the constraint on the variance for the time being. The rates that satisfy (12.81) and (12.82) are

$$F = \frac{D}{(\Delta x)^2} \cdot \frac{-\frac{f\Delta x}{k_B T}}{\exp\left(-\frac{f\Delta x}{k_B T}\right) - 1} \tag{12.84}$$

$$B = \frac{D}{(\Delta x)^2} \cdot \frac{\frac{f\Delta x}{k_B T}}{\exp\left(\frac{f\Delta x}{k_B T}\right) - 1} \tag{12.85}$$

Additionally, this set of jump rates satisfies (12.83) with an error of $O((\Delta x)^2)$. We point out that this choice of F and B is an improvement over the rates used by Elston and Doering [Elston and Doering, 1995], since the mean is exactly preserved and F and B have finite values as $k_B T \to 0$.

In general, the force f will not be constant, but will depend on x. In this case the jump rates will depend on n, and are given by (12.43) and (12.44).

12.6.7 Jump rates at an absorbing boundary

To derive an appropriate jump rate at this boundary, we approximate (12.35) near $x = 0$ by

$$\frac{\partial p}{\partial t} = D\frac{\partial}{\partial x}\left(-\frac{f}{k_B T}p + \frac{\partial p}{\partial x}\right), \tag{12.86}$$

where $f = -(\phi_1 - \phi_0)/\Delta x$ is an approximation for $-\partial\phi/\partial x$ in $(0, \Delta x)$. To derive a second order treatment of the boundary, we only need a first order approximation for this derivative.

Next we assume that at any given time $p(x,t)$ in the interval $(0,\Delta x)$ is approximately at steady-state. This assumption is valid because at small length scales diffusion is the dominant effect. The time scale for a particle with diffusion coefficient D to diffuse a distance Δx is $\Delta t_{dif} = (\Delta x)^2/2D$, which is proportional to $(\Delta x)^2$. The time scale for a flow with velocity v to travel a distance of Δx is $\Delta t_{flow} = \Delta x/v$, which is proportional to Δx. For small Δx, $\Delta t_{diff} \ll \Delta t_{flow}$. At small length scales, diffusion relaxes the system to steady state immediately after it is disturbed by the flow. Thus, at any given time, the local structure of the solution is given approximately by the steady-state solution. At an absorbing boundary, the steady-state assumption in $(0,\Delta x)$ can also be justified mathematically by examining (12.86) at $x = 0$. Because $p(0,t) = 0$, the left-hand-side of (12.86) is exactly zero at $x = 0$. Therefore, we set the left-hand-side of (12.86) to zero in the interval $(0,\Delta x)$ and solve the resulting ordinary differential equation subject to the following two conditions:

$$p(0) = 0, \quad \int_0^{\Delta x} p(x)dx = p_1. \tag{12.87}$$

The solution is

$$p(x) = p_1 \frac{\exp(\frac{fx}{k_BT}) - 1}{\frac{k_BT}{f}[\exp(\frac{f\Delta x}{k_BT}) - 1] - \Delta x}. \tag{12.88}$$

Using the above expression for $p(x)$, the flux is found to be

$$J = D\frac{f}{k_BT}p - D\frac{\partial p}{\partial x} = -p_1\frac{D}{(\Delta x)^2} \cdot \frac{\alpha^2}{\exp(\alpha) - 1 - \alpha}, \qquad \alpha = \frac{f\Delta x}{k_BT}. \tag{12.89}$$

In the numerical scheme, the flux at the boundary is

$$J_{1/2} = -p_1 B_{1/2}. \tag{12.90}$$

This equation reflects the fact that once the protein is absorbed, it does not return to the fluid. Comparing (12.89) with (12.90), we get (12.50).

Suggestions for further reading

- *Elementary Applications of Probability Theory*, Henry Tuckwell. A good introduction to probability theory and stochastic processes with some applications to biology. [Tuckwell, 1995]
- *Random Walks in Biology*, Howard Berg. Introductory text on applying stochastic processes to cellular and molecular biological systems. This book is written for biologists. [Berg, 1993]
- *Handbook of Stochastic Methods*, Crispin Gardiner. A good reference that covers most important topics for studying stochastic processes. However, it is not a good book for learning the subject. [Gardiner, 1997]

Alex Mogilner, Tim Elston, Hongyun Wang and George Oster

Exercises

1. Use (12.1) to verify that $\langle x_n \rangle = 0$ and $\text{Var}[x_n] = (\Delta x)^2 n = ((\Delta x)^2/\Delta t)t$.
2. Use (12.6) to compute $\partial p(x,t)/\partial t$ and $\partial^2 p(x,t)/\partial x^2$. Substitute your results for these two expression into (12.7) to directly verify that $p(x,t)$ satisfies the diffusion equation.
3. Let $p_n(t)$ be the probability that a biological filament has n subunits at time t. Assume that the depolymerazation rate of the growing filament is zero. Write down the equations that govern $p_n(t)$. Assuming that $p_1(0) = 1$ solve the equations for $p_1(t)$, $p_2(t)$ and $p_3(t)$. Can you generalize your results for $p_n(t)$? Compute the first passage time density for the time it takes the filament to grow 4 subunits long.
4. Use (12.10) to verify that $\langle x(t) \rangle = L\langle N(t) \rangle = L(r_p - r_d)t$ and $\text{Var}[x(t)] = L^2\text{Var}[N(t)] = L^2(r_p + r_d)t$.
5. For a 10nm sphere of mass $m = 10^{-21}$kg moving in water, compute the thermalization time τ.
6. Nondimensionalize (12.30) by choosing L as the spatial scale and a characteristic time to diffuse across the domain $[-2L, 2L]$ as the time scale. Simulate the resulting equation on the computer for the double well potential $\phi(x) = Ak_BT[(x/L)^4 - (x/L)^2], A = 0.1, 1, 10$ and $x(0) = -L$ and no flux conditions at $x = \pm 2L$. Justify your choice of the time steps. Run simulations until (i) $t_{end} = L^2/D$, (ii) $t_{end} = 10L^2/D$, (iii) $t_{end} = 100L^2/D$. Discuss the results.
7. Use the computer to plot the Boltzmann distributions in the case of the double well potential $\phi(x) = Ak_BT[(x/L)^4 - (x/L)^2]$, $-2L < x < 2L$, for $A = 0.1, 1, 10$. Discuss the results.
8. Solve equation (12.36) numerically on the interval $[0, 1]$ with potential $\phi(x) = x$, no flux boundary and arbitrary initial conditions. Use any standard numerical method. Run the simulations until the transients die out. Compare the solutions with the corresponding Boltzmann distribution and discuss the results.
9. Derive (12.84) and (12.85) from (12.81) and (12.82).
10. Verify that the steady state probability density given in (12.88) satisfies (12.86) and the conditions (12.87).
11. An Ornstein-Uhlenbeck process is characterized by the following stochastic differential equation:

$$m\frac{dV}{dt} = -\zeta V + f_B(t), \langle f_B(t) \rangle = 0, \text{Cov}[f_B(t)f_B(s)] = 2k_BT\zeta\delta(t-s).$$

Write down the diffusion equation that corresponds to this process. Solve the diffusion equation for the equilibrium density. Using the numerical algorithm described in this chapter, generate sample paths for this process. Use the sample paths to

generate a histogram of the particles velocity. Compare the histogram with the analytic result for the equilibrium density.

12. Write a program to numerically solve the diffusion equation (non-dimensionalize first!):

$$\frac{\partial p(x,t)}{\partial t} = D\left[\frac{\partial}{\partial x}\left(A\sin\left(\frac{2\pi x}{L}\right)+F\right)\frac{p(x,t)}{k_BT}+\frac{\partial^2 p(x,t)}{\partial x^2}\right],$$

subject to periodic boundary conditions at $x=0$ and $x=L$ and the initial condition $p(x,0)=\delta(x-L/2)$. Plot the distribution at various times to observe the relaxation to steady-state. Use the steady-state distribution to compute the average velocity. Investigate how the average velocity changes as the parameters F, A and D are varied.