Jan Tobochnik, Harvey Gould, and Jon Machta, "Understanding the temperature and the chemical potential through computer simulations," Am. J. Phys. **73** (8), 708–716 (2005). This paper extends the demon algorithm to compute the chemical potential.

Simon Trebst, David A. Huse, and Matthias Troyer, "Optimizing the ensemble for equilibration in broad-histogram Monte Carlo simulations," Phys. Rev. E **70**, 046701-1–5 (2004). The adaptive algorithm presented in this paper overcomes critical slowing down and improves upon the Wang–Landau algorithm and is another example of the flexibility of Monte Carlo algorithms.

I. Vattulainen, T. Ala-Nissila, and K. Kankaala, "Physical tests for random numbers in simulations," Phys. Rev. Lett. **73**, 2513 (1994).

B. Widom, "Some topics in the theory of fluids," J. Chem. Phys. **39**, 2808–2812 (1963). This paper discusses the insertion method for calculating the chemical potential.

CHAPTER

16 Quantum Systems

We discuss numerical solutions of the time-independent and time-dependent Schrödinger equation and describe several Monte Carlo methods for estimating the ground state of quantum systems.

16.1 ■ INTRODUCTION

So far we have simulated the microscopic behavior of physical systems using Monte Carlo methods and molecular dynamics. In the latter method, the classical trajectory (the position and momentum) of each particle is calculated as a function of time. However, in quantum systems the position and momentum of a particle cannot be specified simultaneously. Because the description of microscopic particles is intrinsically quantum mechanical, we cannot directly *simulate* their trajectories on a computer (see Feynman).

Quantum mechanics does allow us to *analyze* probabilities, although there are difficulties associated with such an analysis. Consider a simple probabilistic system described by the one-dimensional diffusion equation (see Section 7.2)

$$\frac{\partial P(x,t)}{\partial t} = D\frac{\partial^2 P(x,t)}{\partial x^2}, \tag{16.1}$$

where $P(x,t)$ is the probability density of a particle being at position x at time t. One way to convert (16.1) to a difference equation and obtain a numerical solution for $P(x,t)$ is to make x and t discrete variables. Suppose we choose a mesh size for x such that the probability is given at p values of x. If we choose p to be order 10^3, a straightforward calculation of $P(x,t)$ would require approximately 10^3 data points for each value of t. In contrast, the corresponding calculation of the dynamics of a single particle based on Newton's second law would require one data point.

The limitations of the direct computational approach become even more apparent if there are many degrees of freedom. For example, for N particles in one dimension, we would have to calculate the probability $P(x_1, x_2, \ldots, x_N, t)$, where x_i is the position of particle i. Because we need to choose a mesh of p points for each x_i, we need to specify N^p values at each time t. For the same level of precision, p will be proportional to the length of the system (for particles confined to one dimension). Consequently, the calculation time and memory requirements grow exponentially with the length of the system. For example, for 10 particles on a mesh of 100 points, we would need to store 10^{100} numbers to represent P, which is already much more than any computer today can store. In two and three dimensions the growth is even faster.