

16

Computer Simulations

Computer simulations play an important role in modern statistical mechanics. The history of the use of computer simulations in science parallels the history of early digital computing. The people and places involved centered around Los Alamos and other U.S. national laboratories where the first digital computers became available for use by scientists after World War II. Early leaders in the development of computer simulation methods included Fermi, Ulam, von Neumann, Teller, Metropolis, Rosenbluth, and others who were also involved in the Manhattan Project (Metropolis, 1987).

Computer simulations in statistical mechanics fall into two broad classes: Monte Carlo (MC) and molecular dynamics (MD), although variants span the range between the two. Both methods involve numerically evolving simple models of materials through a set of microstates in order to determine the thermodynamic averages of measurable quantities. Computer simulations provide a means to study physical systems that is complementary to both experiment and theory. The following are a few of the advantages of computer simulations:

- Computer simulations can provide insight into the equilibrium and nonequilibrium behavior of model systems for ranges of parameters where theoretical approximations are invalid or untested.
- Computer simulations provide a means to test the range of validity of theoretical approximations against specific model systems.
- Computer simulations allow visualization of physical processes that can provide new insights into complex phenomena.
- Computer simulations allow detailed examination of behaviors that might not be accessible experimentally.
- Computer simulations can be used to examine fundamental physical processes that can be used to guide theory.
- Computer simulations can be used to model systems that do not exist in nature to provide assistance in understanding existing materials and engineering new ones.

16.1 Introduction and statistics

While certain critical aspects of computer simulation theory should be followed rigorously, much of computer simulation development and use is an art form. There are many possible simulation approaches for any given problem and some choices will be more effective at

654 Chapter 16 • Computer Simulations

elucidating important physical properties than others. This brief chapter concentrates on equilibrium simulations but computer simulations are also widely used to model dynamical and nonequilibrium processes. The task of determining equilibrium thermodynamic averages of model systems is accomplished by generating a sequence of microstates that are chosen from the equilibrium ensemble of the model. For example, an MD simulation might be used to integrate Newton's equations of motion for generating a time-series of states in phase space as the system explores the constant-energy hypersurface of the Hamiltonian. By comparison, an MC simulation of the same model might generate a sequence of states chosen by a random walk among the configurational microstates of the canonical ensemble. Both methods are examples of *importance sampling*, which focuses computational effort on generating microstates that are representative of the equilibrium ensemble rather than sampling *all* of the phase space. It is this huge improvement in efficiency that makes computer simulations of statistical mechanical models feasible. The sequence of states produced by either method can be used to estimate equilibrium averages. Allen and Tildesley (1990), Binder and Heermann (2002), Frenkel and Smit (2002), and Landau and Binder (2009) provide more detailed discussions of computer simulations and their applications in statistical physics.

Let q represent a microstate of the system and $A(q)$ a thermodynamic observable that is a function of the microstate. In an MC simulation, q might represent the positions of all the particles in the system while in an MD simulation q might represent the positions and momenta of all the particles. The observable $A(q)$ might represent the potential energy, virial contribution to the pressure, pair correlation function, and so on. The initial microstate chosen to start a simulation will generally not be typical of the set of microstates that make up the equilibrium ensemble, but the goal of a simulation is to evolve the microstate through a large enough subset of the microstates of the equilibrium ensemble so that averages of observables approach their equilibrium values. After a simulation has run long enough for the system to approach equilibrium, the simulation then generates a sequence of M configurations, $\{q_j\}_{j=1}^M$, chosen from the set of microstates in the equilibrium ensemble and stores a sequence of values, $\{A(q_j)\}_{j=1}^M$, for each of the thermodynamic variables one wants to measure.¹ Since the microstates are chosen from the equilibrium ensemble, the equilibrium average of A is approximated by a simple average of the set of values $\{A(q_j)\}_{j=1}^M$. Of course, a simulation can only provide a finite sequence of states, so a statistical analysis of the uncertainty of the results is a crucial part of any simulation.

The equilibrium average of the variable A is given by

$$\langle A \rangle = \langle A \rangle_M \pm \sigma_M \quad (1)$$

¹Alternatively, one can store the full configuration set of statistically independent microstates for later analysis. This tactic requires a large amount of storage space but is useful if there is a large computational cost of generating statistically independent configurations and one needs to calculate averages of many different observables at a later time using the stored configurations. This is sometimes done in large-scale lattice quantum chromodynamics simulations.

where the simulation average $\langle A \rangle_M$ and uncertainty σ_M are determined by

$$\langle A \rangle_M = \frac{1}{M} \sum_{j=1}^M A(q_j), \quad (2a)$$

$$\sigma_M = \frac{\sqrt{\langle A^2 \rangle_M - \langle A \rangle_M^2}}{\sqrt{M/(2\tau + 1)}}, \quad (2b)$$

$$\langle A^2 \rangle_M - \langle A \rangle_M^2 = \frac{1}{M} \sum_{j=1}^M [A(q_j) - \langle A \rangle_M]^2. \quad (2c)$$

The “correlation time” τ is defined as follows. Since the states q_j are generated sequentially by the simulation, each new state q_{j+1} is guaranteed to be close to the previous state q_j , so the values $A(q_j)$ in the sequence are highly correlated. The correlations in the values of $A(q_j)$ decrease with the “correlation time” τ which can be calculated from the correlation function $\phi_{AA}(t)$, namely

$$\phi_{AA}(t) = \frac{\langle A(t)A(0) \rangle - \langle A(t) \rangle \langle A(0) \rangle}{\langle A^2 \rangle - \langle A \rangle^2}, \quad (3a)$$

$$\tau = \sum_{t>0} \phi_{AA}(t). \quad (3b)$$

The variable t is a measure of the separation between pairs of configurations in the ordered sequence. In the case of molecular dynamics simulations, τ represents a physical time for the system to move far enough along its trajectory on the energy surface to result in decorrelated values of A . Monte Carlo simulations explore equilibrium microstates in a random walk, so τ does not correspond to physical time but rather the average number of Monte Carlo sweeps needed to give statistically independent values for A . The quantity $M/(2\tau + 1)$ represents the number of statistically independent configurations in the sequence of M values.²

²The correlations $\phi_{AA}(t)$ in equation (16.1.3a) can be measured using subsequences of the M configurations:

$$\langle A(t)A(0) \rangle \approx \frac{1}{M'} \sum_{j=1}^{M'} A(q_{j+t})A(q_j),$$

and

$$\langle A(t) \rangle \approx \frac{1}{M'} \sum_{j=1}^{M'} A(q_{j+t}).$$

By definition, the correlation function $\phi_{AA}(0)$ is unity and the correlations decay to zero as $t \rightarrow \infty$. Once one can place a reliable upper bound on the size of the correlation time τ for a given system from a knowledge of its equilibrium correlations, one can simply skip more than τ configurations between storing values of $A(q_j)$ to ensure that the numbers in the sequence are now approximately statistically independent.

16.2 Monte Carlo simulations

The term *Monte Carlo method*, named for the gambling casinos in Monaco, was coined by Nicholas Metropolis (1987) – “a suggestion not unrelated to the fact that Stan [Ulam] had an uncle who would borrow money from relatives because he ‘just had to go to Monte Carlo’.” The goal of a Monte Carlo simulation in equilibrium statistical mechanics is to use pseudo-random numbers to draw a representative sample of microstates $\{q\}$ from the equilibrium probability distribution

$$P_{\text{eq}}(q) = \frac{\exp(-\beta E(q))}{\sum_{q'} \exp[-\beta E(q')]} \quad (1)$$

This means that “instead of choosing configurations randomly, then weighting them with $\exp(-E/kT)$, we choose configurations with a probability $\exp(-E/kT)$ and weight them evenly” (Metropolis et al., 1953). If a simulation can accomplish this, then thermodynamic averages can be calculated using the simple averages in equation (16.1.2). This importance sampling of the states provides a huge computational advantage over normal random sampling.

The following algorithm accomplishes the goal of randomly selecting microstates q from the set of all microstates with a probability distribution that approaches the equilibrium distribution (1) (Metropolis et al., 1953; Kalos and Whitlock, 1986; Allen and Tildesley, 1990; Frenkel and Smit, 2002; Binder and Heermann, 2002; Landau and Binder, 2009). Consider an ensemble of microstates that has some initial distribution of probabilities $P(q, 0)$ and let the distribution evolve according to the discrete stochastic rate equation

$$P(q, t+1) = \sum_{q'} P(q', t) P(q' \rightarrow q), \quad (2)$$

where $P(q \rightarrow q')$ is the transition probability from state q to state q' . If the transition probability obeys the balance condition

$$\sum_{q'} P_{\text{eq}}(q') P(q' \rightarrow q) = P_{\text{eq}}(q) \sum_{q'} P(q \rightarrow q'), \quad (3)$$

and the random process in equation (2) can reach every microstate from every other microstate in a finite number of steps, then the ensemble probability will approach the equilibrium distribution:

$$\lim_{t \rightarrow \infty} P(q, t) = P_{\text{eq}}(q). \quad (4)$$

In practice, equation (3) is usually implemented using the *detailed* balance condition

$$P_{\text{eq}}(q) P(q \rightarrow q') = P_{\text{eq}}(q') P(q' \rightarrow q). \quad (5)$$

Evaluating $P_{\text{eq}}(q)$ requires summing over all states to determine the partition function, but the ratio $P_{\text{eq}}(q')/P_{\text{eq}}(q)$ depends only on the energy difference $\Delta E = E(q') - E(q)$. Therefore,

the transition rates are related by

$$P(q \rightarrow q') = \exp(-\beta\Delta E) P(q' \rightarrow q). \quad (6)$$

This guarantees that the sequence of states generated by this stochastic process, beginning from any starting configuration, asymptotically becomes equivalent to selecting states by a random walk among the microstates of the equilibrium ensemble. This can be implemented in a computer code, as first proposed by Metropolis et al. (1953), using the acceptance rates

$$\begin{aligned} W(q \rightarrow q') &= 1 && \text{if } \Delta E \leq 0, \\ W(q \rightarrow q') &= \exp(-\beta\Delta E) && \text{if } \Delta E > 0. \end{aligned} \quad (7)$$

Other choices for the acceptance rates are possible but this form, named after Metropolis, is the most commonly used.

16.2.A Metropolis Monte Carlo algorithm

The Metropolis method can be implemented in a computer program by using a pseudorandom number generator `rand()` that returns pseudorandom numbers that are uniformly distributed on the open unit interval $(0,1)$; see Appendix I for a discussion of how pseudorandom numbers are generated. First, initialize the system by choosing a starting state q_0 from the set of all microstates of the model. It is helpful if q_0 is not atypical of the states in the equilibrium ensemble. This reduces the number of steps needed for the system to equilibrate. For example, a disordered liquid-like state would not be the best starting point for a simulation of a crystalline solid.

The Metropolis algorithm is defined by the following steps:

1. Generate a random trial state q_{trial} that is “nearby” the current state q_j of the system. “Nearby” here means that the trial state should be almost identical to the current state except for a small random change made, usually, to a single particle or spin. For example, one can create a trial state of a particle simulation by randomly moving one particle to a nearby location

$$x_i^{\text{trial}} = x_i + \Delta x(\text{rand}() - 0.5), \quad (8)$$

with two more calls to `rand()` to generate y_i^{trial} and z_i^{trial} . The trial state of a spin system usually involves a spin flip or a random rotation of a single spin.³

2. Determine the change in the energy of the trial state compared to the previous state, namely $\Delta E = E(q_{\text{trial}}) - E(q_j)$. If $\Delta E \leq 0$, accept the trial state, that is, set

³There are Monte Carlo algorithms for spin systems that flip spins in large correlated clusters rather than one spin at a time; see Swendsen and Wang (1987) and Wolff (1989). These methods are very effective for simulations of some particular models. Also, one can attempt spin flips of all the spins on noninteracting sublattices at one time since the acceptance of each flip is independent of the other flipped spins. For example, a chessboard pattern update of a spin model in which spins only interact with nearest neighbors of a square lattice can be more efficient for some computer architectures or programming environments.

658 Chapter 16 • Computer Simulations

$q_{j+1} = q_{\text{trial}}$. If $\Delta E > 0$, then accept the trial state with probability $\exp(-\beta\Delta E)$. This is accomplished by using an additional call to the pseudorandom number generator. If $\text{rand}() < \exp(-\beta\Delta E)$, then accept the trial state. If the interactions are short-ranged, the calculation of the energy change will only involve interactions with a few nearby particles or spins. If the trial state is illegal in some way, that is, it is not an allowed state in the set of all configurations, then the state should be rejected. This is equivalent to setting the energy change at $+\infty$. If the trial state is rejected for either reason, then set the new state of the system equal to the previous state $q_{j+1} = q_j$, that is, leave the state at the old value q_j , throw away the trial state, and move on.

3. Perform steps 1 and 2 once for each particle or spin in the system. This is often done randomly to ensure detailed balance.⁴ Steps 1 through 3 define one Monte Carlo sweep.
4. Repeat steps 1 through 3 for M_{eq} Monte Carlo sweeps to let the system equilibrate. The proper choice of M_{eq} is not obvious *a priori*. At the very least, all the measures $A(q)$ studied in the simulation should no longer have any obvious monotonic drift by the end of equilibration. This does not guarantee that the system has reached equilibrium since the system could well be trapped in the vicinity of a long-lived metastable state.
5. Repeat steps 1 through 3 for M Monte Carlo sweeps while keeping track of all the thermodynamic variables one wants to measure, namely $\{A(q_j)\}_{j=1}^M$. Use equations (16.1.1) and (16.1.2) to determine the equilibrium averages and uncertainties.

To determine averages at a different set of parameters (temperature, density, etc.), change the parameters by a small amount and repeat steps 1 through 5, including the equilibration step 4.⁵ Using the last configuration of the previous run as the first configuration of the next run can often reduce the equilibration time. Figure 16.1 shows a Monte Carlo calculation of the specific heat of the two-dimensional Ising model on a 128×128 square lattice, as compared to the exact solution presented in Section 13.4.A.

⁴Sequential and other update methods that violate detailed balance are sometimes used for efficiency but special care should be taken to ensure that detailed balance is maintained on average.

⁵Histogram reweighting methods can sometimes be used to reduce the number of temperatures and fields that need to be simulated (Ferrenberg and Swendsen, 1988). For example, if a spin simulation at coupling K and field h collects a histogram that samples the joint energy-magnetization distribution $P_{K,h}(E, M)$, the distribution at nearby temperatures and fields is given by

$$P_{K+\Delta K, h+\Delta h}(E, M) = \frac{P_{K,h}(E, M) e^{\Delta K E + \Delta h M}}{\sum_{E, M} P_{K,h}(E, M) e^{\Delta K E + \Delta h M}}.$$

Other methods that are now widely used are: parallel tempering, multicanonical Monte Carlo, and “broad histogram” methods. These are particularly effective for studying systems with strongly first-order phase transitions. Monte Carlo renormalization group methods are very powerful for studying critical points. For a survey, see Landau and Binder (2009).

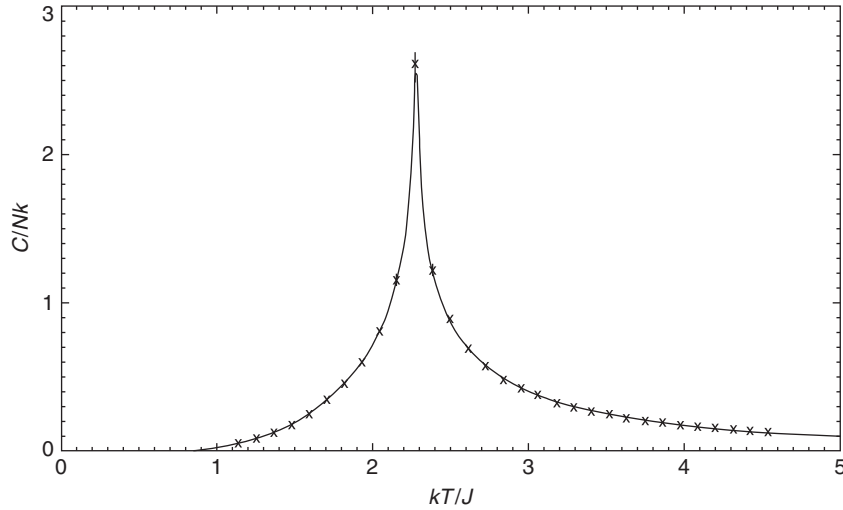


FIGURE 16.1 Monte Carlo specific heat (\times 's) of the two-dimensional Ising model on a 128×128 lattice, as compared to the exact solution (solid line) from Section 13.4.A; see Kaufman (1948), Ferdinand and Fisher (1967), and Beale (1996). The MC error bars are smaller than the symbols used, except near the bulk critical temperature $T_c(\infty)$. Each data point represents an average using 10^5 Monte Carlo sweeps, except at the bulk critical point where 10^6 Monte Carlo sweeps were used to mitigate critical slowing down.

16.3. Molecular dynamics

The purpose of a molecular dynamics simulation is to integrate Newton's equations of motion for the set of particles in the given system. One advantage of MD over MC is that it approximates the time evolution of the equations of motion of the system, so MD can be used to study a host of dynamical properties. MD is usually more efficient at simulating systems with long-range interactions since all the particles are updated together. MD is sometimes easier to implement than MC for complex systems since appropriate MC moves are sometimes difficult to derive. There are MD variants that allow simulations of other ensembles, but the simplest case simulates a microcanonical ensemble in which the microstate of the given system explores its energy surface in the phase space; see Allen and Tildesley (1990) and Frenkel and Smit (2002) for details.

The equations of motion here are

$$\frac{d^2 \mathbf{r}_i}{dt^2} = \frac{1}{m_i} \mathbf{F}_i = -\frac{1}{m_i} \nabla_i U(\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_N), \quad (1)$$

where \mathbf{F}_i is the force on particle i arising from the N -particle potential energy function U . The MD simulation moves the system forward in time by discrete steps Δt . The most commonly used integration method in this context is due to Verlet (1967):

$$\mathbf{r}_i(t + \Delta t) = 2\mathbf{r}_i(t) - \mathbf{r}_i(t - \Delta t) + \frac{(\Delta t)^2}{m_i} \mathbf{F}_i(t). \quad (2)$$

660 Chapter 16 • Computer Simulations

This is equivalent to the leap-frog and velocity Verlet algorithms that update both positions and velocities of the particles; see Frenkel and Smit (2002). The Verlet method preserves the time-reversal symmetry of the Hamiltonian equations of motion and has an error per step of order $(\Delta t)^4$, while only requiring a single determination of the force on each particle, which is usually the most computationally time-consuming part of the simulation. Most importantly, the Verlet algorithm is symplectic, so the integration is equivalent to an exact solution of a “nearby” ghost Hamiltonian, which results in good long-term stability and good conservation of energy properties.

A simulation starts the system in some initial microstate with defined positions and velocities of all the particles, and the integration algorithm steps the positions and velocities of the particles forward in time. The simplest forms of the approximations employed for the velocities and the energy are

$$\mathbf{v}_i(t) = \frac{\mathbf{r}_i(t + \Delta t) - \mathbf{r}_i(t - \Delta t)}{2\Delta t}, \quad (3a)$$

$$E = \sum_{i=1}^N \frac{m_i}{2} \left(\frac{\mathbf{r}_i(t + \Delta t) - \mathbf{r}_i(t - \Delta t)}{2\Delta t} \right)^2 + U[\mathbf{r}_1(t), \mathbf{r}_2(t), \dots, \mathbf{r}_N(t)]. \quad (3b)$$

The time-step Δt is chosen to be small as compared to the shortest fundamental time scale in the Hamiltonian, while not so small as to limit the efficiency of the program. The numerical integration approximates a member of the microcanonical ensemble moving along the constant-energy hypersurface in the phase space. Calculating equilibrium averages properly depends on the Hamiltonian being ergodic;⁶ this allows the system to sample all regions of the constant-energy hypersurface, so the MD time-averages are equivalent to averages over the microcanonical ensemble. If the total energy drifts more than some predetermined amount during the course of the simulation, then all the velocities can be rescaled to shift the total energy back to its initial value. Alternatively, one can use a thermostat to maintain the temperature at a desired value; see Frenkel and Smit (2002).

A commonly used pair interaction for monatomic fluids such as neon and argon is the Lennard-Jones interaction

$$u(r) = 4\varepsilon \left(\left(\frac{D}{r} \right)^{12} - \left(\frac{D}{r} \right)^6 \right), \quad (4)$$

⁶Since an MD simulation creates a time evolution of the model system, one needs some assurance that the system is ergodic, that is, the time averages and the ensemble averages are the same. For example, a system of harmonic oscillators is not ergodic. A system of N particles in d dimensions has a $6N$ -dimensional phase space, so the constant-energy hypersurface has $6N - 1$ dimensions. The normal mode solution for N coupled oscillators has $3N$ constants of the motion, so the system explores only a $3N$ -dimensional hypersurface. Even making the couplings between particles anharmonic does not eliminate the problem as first shown by Fermi, Pasta, and Ulam (1955) and explored theoretically by Kolmogorov (1954), Arnold (1963), and Moser (1962). MD simulations of equilibrium systems presume that the system is ergodic. There are only a few systems that are provably ergodic but, fortunately, most systems with realistic pair potentials appear to behave ergodically in two or more dimensions. In view of this, MD simulations of one-dimensional systems should be treated as suspect from this perspective.

where D is the molecular diameter and ε is the depth of the attractive well. The Lennard-Jones potential is attractive at long distances, and decays as $1/r^6$ to model the van der Waals attraction; at short distances, it diverges as $1/r^{12}$ to model the Pauli repulsion that prevents overlap of the electronic wavefunctions. Simulations are best carried out using dimensionless parameters. In the case of a fluid with Lennard-Jones interactions, all lengths can be measured in units of D , all energies (including kT) in units of ε , all forces in units of ε/D , all pressures in units of ε/D^3 , all times in units of $\sqrt{mD^2/\varepsilon}$, and so on. Simulations then need to be conducted only for single values of reduced temperature kT/ε , the reduced density nD^3 , and so on, while measuring observables in reduced units. Comparisons between simulations and experimental results can then be made using experimental values of D , m , and ε .⁷ In dimensionless units, the Lennard-Jones force between a pair of particles is

$$\mathbf{F} = \pm \frac{\mathbf{r}}{r^2} \left(\frac{48}{r^{12}} - \frac{24}{r^6} \right). \quad (5)$$

Newton's third law of motion can be used to reduce the number of force calculations by a factor of two. The Lennard-Jones model was first studied in an MC simulation by Wood and Parker (1957) and in an MD simulation by Rahman (1964) and Verlet (1967).

16.3.A Molecular dynamics algorithm

First, start the system by choosing an initial state by setting the initial positions and velocities of all the particles. The initial velocities are usually set by choosing each component of the velocity vector of each particle from the Maxwell distribution. In reduced units, this is

$$P_{\text{Maxwell}}(v_x(0)) = \frac{1}{\sqrt{2\pi T}} \exp\left(-\frac{v_x^2(0)}{2T}\right); \quad (6)$$

see Appendix I to see how to use a uniform pseudorandom number generator to select from a Gaussian distribution. The initial velocities can then be used to set the positions of the particles after the first time-step, namely

$$\mathbf{r}_i(\Delta t) = \mathbf{r}_i(0) + \mathbf{v}_i(0)\Delta t + \frac{1}{2} \frac{(\Delta t)^2}{m_i} \mathbf{F}_i(0). \quad (7)$$

⁷For example, the Lennard-Jones parameters appropriate for argon are $\varepsilon/k = 119.8 \text{ K}$ and $D = 0.3405 \text{ nm}$ (Levelt, 1960; Rowley, Nicholson, and Parsonage, 1975). Interaction potentials are almost always cut off at a finite distance between molecules to reduce the number of interactions that need to be considered at each time step. For the Lennard-Jones interaction, this is most commonly done at $r_{\text{max}} = 2.5D$. If the potential is set to zero for distances greater than r_{max} , then this would leave a small discontinuity in the potential. To eliminate this, the potential is often shifted upward by $-u(2.5D) \simeq 0.0163\varepsilon$, so that the potential is zero at r_{max} . This allows a direct comparison between MC and MD simulations. If the shift is not made in the potential, one could not directly compare the results from MC and MD simulations because the discontinuity in the potential would result in a delta-function force that affects the motion in the MD simulation but not the configurations in the MC simulation. Comparisons of MC and MD results with experiments need to include perturbations from the shift and the missing tails of the pair potentials.

662 Chapter 16 • Computer Simulations

1. Next, use equation (2) to move the system forward in time through $M_{\text{eq}} = \tau_{\text{eq}}/\Delta t$ time steps. The equilibration time τ_{eq} must be chosen large enough for the system to equilibrate; see the Monte Carlo discussion in Section 16.2. A thermostat is often used to evolve the system to a state with the desired temperature; see Frenkel and Smit (2002).
2. Now, use equation (2) to move the system forward in time through $M = \tau_{\text{avg}}/\Delta t$ time steps while keeping track of all the thermodynamic variables $\{A(q_j)\}$ one wants to measure. Finally, use equation (16.1.2) to determine the equilibrium averages and uncertainties.

To determine averages at a different set of parameters (temperature, density, etc.), change the parameters by a small amount and repeat steps 1 and 2. Using the last configuration of the previous run as the first configuration of the new run can often reduce the equilibration time.

16.4. Particle simulations

Fluids can be modeled by both MC and MD simulations by placing N particles in a periodic box with volume V interacting via a pair potential. Hansen and McDonald (1986), Allen and Tildesley (1990), and Frenkel and Smit (2002) provide excellent surveys of this topic. Calculating energy changes of trial moves in MC or forces in MD only involves pairs of particles whose closest periodic copies are within the cutoff distance of each other. MC simulations typically sample the canonical ensemble,⁸ so they control the temperature and density, and measure the energy, pressure, and so on. MD simulations typically sample the microcanonical ensemble,⁹ so they control the energy and density, and they measure the temperature, pressure, and so on. The equipartition theorem gives for the temperature T in a d -dimensional system

$$kT = \frac{1}{Nd} \left\langle \sum_{i=1}^N m_i v_i^2 \right\rangle. \quad (1)$$

The virial equation of state in equations (3.7.15) and (10.7.11) can be used to determine the pressure P in either type of simulation:

$$\frac{P}{nkT} = 1 + \frac{1}{Ndkt} \left\langle \sum_{i<j} \mathbf{F}(\mathbf{r}_{ij}) \cdot \mathbf{r}_{ij} \right\rangle = 1 - \frac{n}{2dkT} \int \frac{du}{dr} rg(r) d\mathbf{r}. \quad (2)$$

⁸Monte Carlo simulations of isobaric or grand canonical ensembles are also widely used by adding PV or μN terms to the Hamiltonian (Frenkel and Smit, 2002).

⁹Molecular dynamics simulations of other ensembles are possible. For example, one can include extra dynamical variables that allow the total energy or volume to fluctuate in order to approximate a canonical or isobaric ensemble; see Frenkel and Smit (2002). Variants of MC and MD simulations that span the range between the two include hybrid Monte Carlo methods that mix MC and MD methods into one code to take advantage of the strengths of both methods. Alternatively, one can include Langevin random force terms and damping in an MD simulation to create coupling to a heat bath.

As discussed in Section 10.7, the pair correlation function of a fluid can be used to measure a variety of thermodynamic properties including the pressure, the isothermal compressibility κ_T , and the scattering structure factor $S(k)$; see equations (10.7.18) through (10.7.21). The pair correlation function $g(r)$ defined in equation (10.7.5) can be determined by collecting a histogram of the distances between all pairs of particles periodically during the simulation, accounting for the periodic boundary conditions, and scaling the histogram by an amount proportional to the volume of shells of radius r and thickness Δr . In three dimensions, the pair correlation function is given by

$$g(r) = \frac{2V}{N^2 \left(\frac{4\pi}{3}\right) [(r + \Delta r)^3 - r^3]} \left\langle \sum_{i < j}^N \Delta_{\Delta r}(r_{ij} - r) \right\rangle, \quad (3)$$

where the step function $\Delta_{\Delta r}(\xi)$ is unity for $0 < \xi < \Delta r$ and zero otherwise. This expression is the ratio of the number of events in each bin in the histogram compared to the average number that would be expected for an ideal gas with the same density.

16.4.A Simulations of hard spheres

The system of hard spheres has been studied extensively in both MC and MD simulations, and was the first model studied using either method (Metropolis et al., 1953; Adler and Wainwright, 1957, 1959). The pair potential for hard spheres is

$$u(r) = \begin{cases} 0 & \text{for } r > D, \\ \infty & \text{for } r \leq D. \end{cases} \quad (4)$$

Temperature is an irrelevant parameter for the spatial configurations sampled by this model since the pair potential does not have a finite energy scale. A full exploration of the phase diagram involves only varying the *reduced* number density nD^d . All thermodynamic properties are either independent of temperature, or scale with temperature in a trivial way. For example, the scaled pressure for a system of hard spheres P/nkT is a function only of the reduced number density nD^d . The hard sphere density is often expressed in terms of the packing fraction η , the fraction of the volume of the system actually occupied by the spheres. In three dimensions, the volume fraction is given by $\eta = \pi nD^3/6$. Since the pair potential is singular, the pressure cannot be calculated using the virial equation (2) but the pressure can be determined using the virial equation of state for hard spheres, namely (10.7.12).

An MC code for hard spheres is relatively simple since the energy change in a trial move is either zero or infinity. A trial displacement of a particle is rejected if the trial position of the particle is within a distance D of any other particle, and is accepted otherwise. This was the first statistical physics model ever studied in a computer simulation (Metropolis et al., 1953).

Implementing MD for hard spheres requires a different approach from the standard MD. Finite-difference integration methods will not work here since the potential is not differentiable. Instead, one can exploit the exact solution to the equations of motion. Each particle

664 Chapter 16 • Computer Simulations

travels in a straight line at a constant velocity except at the instants when pairs of particles collide, that is when they are a distance D apart. Due to the singular nature of the potential, the collisions can be uniquely time-ordered. Each collision conserves both kinetic energy and momentum, so the velocities after each collision can be determined analytically from the velocities and the displacement vector between the centers of the two particles at the moment of the collision. The changes in the velocities of the two colliding particles are

$$\Delta \mathbf{v}_i = -\Delta \mathbf{v}_j = \left. \frac{-(\mathbf{r}_{ij} \cdot \mathbf{v}_{ij}) \mathbf{r}_{ij}}{D^2} \right|_{|\mathbf{r}_{ij}|=D}, \quad (5)$$

where \mathbf{r}_{ij} and \mathbf{v}_{ij} are, respectively, the relative positions and relative velocities of the two particles. The simulation moves the particles forward in time from collision to collision and changes the velocities of the pairs of particles involved in the collisions, as given by equation (5). This was the first implementation of the MD method in statistical physics (Alder and Wainwright 1957, 1959).

The pair correlation function and the structure factor in the fluid phase are shown in Figure 16.2 and the phase diagram for hard spheres in Figure 16.3. At low densities, the equilibrium phase is a short-range ordered fluid. At high densities, the equilibrium phase of the model is a long-range ordered, face-centered cubic solid. It is worthwhile to note that an attractive interaction is *not* required for a model to have a crystalline phase. An attractive interaction *is*, however, required for the formation of a liquid–vapor coexistence line and a critical point. For this reason, the low-density phase of the hard sphere model is often referred to as a fluid phase rather than a liquid phase since the model does not have a liquid–vapor coexistence line. The liquid and solid volume fractions at the liquid–solid coexistence line are $\eta_l \simeq 0.491 \pm 0.002$ and $\eta_s \simeq 0.543 \pm 0.002$, respectively. The liquid–solid coexistence pressure is given by $P_{ls}^* = P_{ls} D^3 / kT \simeq 11.55 \pm 0.11$; see Speedy (1997). In the low-density fluid phase for $\eta < \eta_l$, the reduced pressure is accurately modeled by the Carnahan–Starling equation of state (10.3.25)

$$\frac{P}{nkT} = \frac{1 + \eta + \eta^2 - \eta^3}{(1 - \eta)^3}, \quad (6)$$

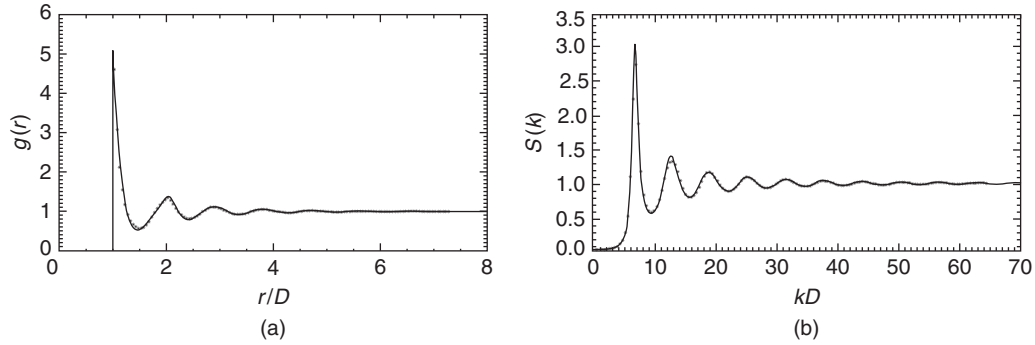


FIGURE 16.2 (a) The pair correlation function $g(r)$ and (b) the static structure factor $S(k)$ for a three-dimensional system of hard spheres at volume fraction $\eta = 0.49$ from a Monte Carlo simulation of 2916 particles (M. Glaser, unpublished). This value of the volume fraction is in the liquid phase close to the solid–liquid coexistence line. The solid lines depict the pair correlation function and the static structure factor from the Percus–Yevick approximation; see Percus and Yevick (1958), Wertheim (1963), and Hansen and McDonald (1986).

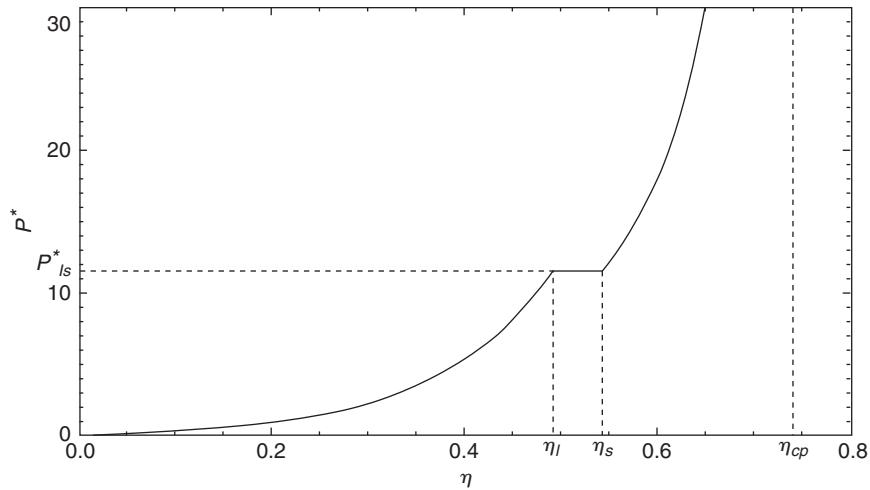


FIGURE 16.3 Sketch of the equilibrium phase diagram for hard spheres in three dimensions. The horizontal axis is the volume fraction $\eta = \pi n D^3/6$ and the vertical axis is the scaled pressure $P^* = P D^3/kT$. There are two equilibrium phases: a low-density fluid phase for $0 < \eta < \eta_l$ and a high-density solid phase for $\eta_s < \eta < \eta_{cp}$.

although there are other good parametrizations as well. In the solid phase the pressure is approximately given by

$$\frac{P}{nkT} = \frac{3}{1 - \eta^*} - 0.5921 \frac{\eta^* - 0.7072}{\eta^* - 0.601}, \quad (7)$$

where $\eta^* (= \eta/\eta_{cp})$ is the ratio of the actual packing fraction to the maximum close-packed value η_{cp} , namely $\pi\sqrt{2}/6 \simeq 0.7405$ (Speedy, 1997; Frenkel and Smit, 2002). The pressure in the solid phase diverges as the density approaches the close-packed density. The model also

exhibits a metastable disordered phase for densities between η_l and the random close-packed volume fraction $\eta_{rcp} \simeq 0.644 \pm 0.005$; see Rintoul and Torquato (1996). For a survey of hard sphere results, see Mulero et al. (2008).

16.5. Computer simulation caveats

Computer simulations are widely used in statistical physics and have played an important role in our understanding of many physical systems. Simulations complement theory and experiment and provide many advantages for the study of systems that are not amenable to exact or approximate theoretical analysis. However, it is important to understand the inherent limitation of this technique.

- Computer simulations necessarily involve a limited number of degrees of freedom, typically hundreds to thousands of particles or spins. This is not nearly large enough to display many of the behaviors that occur in thermodynamically large systems. For example, a model of a dense system of 1,000 particles in a three-dimensional cubic box will only have about 10 particles along each linear dimension of the box, so correlations beyond about five particle diameters are affected by the periodic boundary conditions. Extraction of accurate thermodynamic behavior from such a study will often involve an analysis of the finite-size scaling behavior of the model for a sequence of systems with different sizes.
- Computer simulations necessarily involve a limited time-scale of simulation. Typical molecular time scales are of the order of $t_0 \approx a/v$, where a a microscopic length scale and v a molecular velocity. For atomic scales near room temperature, $a \approx 0.1$ nm and $v \approx 100$ m/s, which gives $t_0 \approx 10^{-12}$ s. In an MD simulation, each time step moves the system forward in time by an amount Δt , which must be much less than t_0 in order to aptly integrate the equations of motion, say $\Delta t \approx 10^{-14}$ s. A simulation that moves the system forward through 10^6 time steps will sample a physical time of only 10 ns, which may not be sufficient to reach many important time-scales of interest in the problem. This is especially problematic when the system has inherently slow time-scales, such as the critical slowing down near second-order phase transitions and the hysteresis near first-order phase transitions. Monte Carlo simulations are similarly hampered in that the simulation must run long enough for the model to explore a sufficiently large region of the phase space to capture the equilibrium behavior. In favorable cases, this and the previous issue can be mitigated by special simulation methods such as coarse graining, cluster update methods, parallel tempering, multicanonical Monte Carlo methods, Monte Carlo renormalization group, and so on; see footnote 5.
- Interactions between particles are usually highly simplified for computational efficiency and the interaction range is usually cut off. Long-range interactions (Coulomb, dipole, van der Waals, etc.) need to be resummed or treated via perturbation theory to try to account for their effects.

- MC and MD simulations do not directly measure the number of microstates available to the system, so one cannot directly calculate the entropy or the free energy in the same way as other observables. If, for example, a determination of the free energy is necessary to locate a phase transition, then it can be determined by a thermodynamic integration to a state with a theoretically known free energy; see Frenkel and Smit (2002).
- MD simulations depend on the ergodicity of the Hamiltonian, so one-dimensional models, models that are weakly perturbed from mechanical equilibrium, and other nearly integrable models may get trapped in low-dimensional orbits that do not fully explore the constant-energy hypersurface of the Hamiltonian; see footnote 6.
- All pseudorandom number generators produce some level of correlation in their sequences. Even subtle correlations in pseudorandom number sequences can produce erroneous results in Monte Carlo simulations. Different classes of generators have different weaknesses, so switching to a generator based on a different algorithm will sometimes cure a problem caused by correlations produced by a particular generator. Testing a generator before using it in a simulation is always a good idea; see Appendix I.
- It is extremely important to confirm the validity of MC and MD simulation codes. This process is rather different from verifying a theoretical calculation. Some code evaluation and verification procedures include: testing the code initially on small systems with known properties, testing the code whenever possible against models with exact solutions or models that have been widely studied in the literature, examining results as a function of system size and run length, and retesting carefully whenever new interactions or code modules are added. In this connection, Frenkel and Smit (2002), Parker (2008), and Landau and Binder (2009) provide lists of good strategies.

Problems

- 16.1.** Write a code to test a uniform pseudorandom number generator. If you do not have a canned generator available, write a generator based on L'Ecuyer's recommended generator in Appendix I. Apply the following tests: average $\langle x \rangle = 1/2$, variance $\langle x^2 \rangle - \langle x \rangle^2 = 1/12$, and the pair correlations test $\langle x_{i+k} x_i \rangle = 1/4$ for $k \neq 0$. Generate a histogram of pairs of numbers on a two-dimensional unit square and test that the distribution is statistically uniform.
- 16.2.** Write a code to test a Gaussian pseudorandom number generator. If you do not have a canned generator available, write a generator based on the Box-Muller algorithm in Appendix I. Apply the following tests: average $\langle x \rangle = 0$, variance $\langle x^2 \rangle = 1$, and the pair correlations test $\langle x_{i+k} x_i \rangle = 0$ for $k \neq 0$. Generate a histogram of pairs of numbers in two dimensions and test that the distribution is statistically Gaussian.
- 16.3.** Define a sequence of correlated random numbers

$$s_k = \alpha s_{k-1} + (1 - \alpha) r_k,$$

where r_k is a unit-variance, uncorrelated, Gaussian pseudorandom number while $0 < \alpha < 1$ defines the range of the correlations. Show that this sequence is Gaussian distributed, with a zero mean. Determine the variance in terms of α and compare your result with equation (16.1.2b). Write a code to determine the correlation function (16.1.3). Plot your measured correlation function and compare it to the exact correlation function.

668 Chapter 16 • Computer Simulations

- 16.4. Write a Monte Carlo code for a system of N hard spheres of diameter D on a one-dimensional ring of length L with periodic boundary conditions. Calculate the pair correlation function and compare it to equations (13.1.6) and (13.1.7). The pressure of the system is given by $P/nkT = 1 + nDg(D^+)$; see equation (10.7.12). Compare your pressure to the one obtained for the exact configurational partition function $Z_N = L(L - ND)^{N-1}/N!$; see equation (13.1.2).
- 16.5. Write a Monte Carlo code for a fluid of N hard spheres in a two-dimensional $L \times L$ square box with periodic boundary conditions in each direction. Calculate the pair correlation function and determine the scaled pressure using equation (10.7.12), namely $P/nkT = 1 + 2\eta g(D^+)$. Compare this pressure to the approximate form $P/nkT = (1 + \eta/8)/(1 - \eta)^2$.
- 16.6. Write a Monte Carlo code for a fluid of N hard spheres in a two-dimensional $L \times L$ square box and include a one-body gravity term $\sum_{i=1}^N mgy_i$ in the algorithm, that is, accept otherwise legal configurations with probability $\exp(-\beta mg\Delta y)$. You will need to use hard-wall boundary conditions on the top and bottom walls. Determine the average number density as a function of the vertical position in the box.
- 16.7. Write a molecular dynamics code for N Lennard-Jones particles in a two-dimensional $L \times L$ square box. Apply periodic boundary conditions in each direction. Determine the scaled pressure using the virial equation (16.4.2). Calculate and plot the pair correlation function of the system.
- 16.8. Write a molecular dynamics code for N Lennard-Jones particles in a two-dimensional $L \times L$ square box, and include a one-body gravity term in the energy: $\sum_{i=1}^N mgy_i$. Apply periodic boundary conditions in the x -direction but a repulsive WCA (Weeks, Chandler, and Andersen, 1971) potential on the top and bottom walls. The WCA potential is the repulsive part of a Lennard-Jones potential for $r/D < (2)^{1/6}$, with the potential shifted up by ε . Show that the average kinetic energy per particle is independent of the height y in the box but the average scaled density nD^2 depends on the vertical position in the box.
- 16.9. Write an MC code to simulate the one-dimensional Ising model on a periodic lattice of length L . Calculate the internal energy and specific heat of the model and compare them to equations (13.2.15) and (13.2.16). Calculate the correlation function $G(n) = \langle s_{i+n}s_i \rangle$ and it compare to equation (13.2.32).
- 16.10. Write an MC code to simulate the two-dimensional nearest-neighbor Ising model on a periodic $L \times L$ lattice in zero field. Calculate the internal energy and the specific heat of the system as functions of temperature and compare them to the exact results in section 13.4.A. See exact results for the two-dimensional Ising model for various lattice sizes at www.elsevierdirect.com.
- 16.11. Write an MC code to simulate the two-dimensional nearest-neighbor Ising model on a periodic $L \times L$ lattice in zero field. Calculate the energy distribution $P(E)$ over a range of temperatures including the critical point. Use this distribution to calculate the internal energy and the specific heat as functions of temperature. See exact results for the two-dimensional Ising model for various lattice sizes at www.elsevierdirect.com.
- 16.12. Write an MC code to simulate the one-dimensional XY model. Calculate the internal energy, the specific heat, the isothermal susceptibility, and the pair correlation function, and compare your results to the analytical results for the $n = 2$ case in Section 13.2.
- 16.13. Write an MC code to simulate the two-dimensional XY model. Calculate the internal energy, specific heat, isothermal susceptibility and the pair correlation function, and compare your results to the theoretical results given in Section 13.7.