CHAPTER 9

Molecular Dynamics

9.1 INTRODUCTION TO THE METHOD: PROPERTIES OF A DILUTE GAS

In this chapter we consider a mechanical approach for studying multiparticle systems. The method we describe is known as molecular dynamics, and it is in many respects complementary to the Monte Carlo method employed in Chapter 8. The Monte Carlo algorithm is very convenient for studying the *equilibrium* properties of a system that is in contact with a heat bath. However, there are many questions that cannot be addressed with such an approach. For example, suppose we are interested in how fast a system will come into equilibrium after a sudden change in the temperature. With the Monte Carlo method the dynamics is governed entirely by the Monte Carlo transition rules (for an Ising model these are the spin-flipping rules), together with the appropriate Boltzmann factors. While these rules are designed to simulate a system in thermal equilibrium, they cannot tell us how fast in real time that a system will move from one particular microstate to another, or if such a transition is even possible.

One way to deal with such questions is to directly simulate the dynamics using the microscopic equations of motion, and this is the philosophy of the molecular dynamics technique. The basic idea is similar to the cream-in-your-coffee problem of Chapter 7, but here we are going to use the appropriate equations of motion to treat the full many-body problem,[1] rather than simulate the motion of an "average" particle. We imagine a box containing a collection of molecules. These molecules move throughout the box as they collide with each other and with the walls of the box. To simulate this process we employ Newton's second law to calculate the positions and velocities of all of the molecules as functions of time. The kinds of questions that can be addressed with this approach include the nature of the melting transition, the rate of equilibration after a sudden addition or loss of energy, and the rate at which molecules diffuse.[2] You may recall that we spent the first part of Chapter 7 arguing that such a microscopic approach provides results that are much too detailed for many of the questions we might like to address, and that claim still stands. However, some questions *do* require such a full-blown microscopic treatment; we will now describe how to carry out such a treatment, along with some of the problems for which it is needed.

While the general simulation scheme we are going to consider in this chapter could apply to any system containing a large number of particles, including droplets in an aerosol, particles in a flame, or stars in a galaxy, molecular dynamics is, as its name implies, usually employed to study the motion of molecules. We might

[1] Albeit for far fewer than 10^{23} particles!

[2] We will now have a way to calculate the value of the diffusion constant that we have encountered in several previous problems.

therefore worry that a *classical* approach involving Newton's second law would not be appropriate, and that we should instead aim for a fully quantum mechanical simulation. A quantum simulation would be *much* more time consuming and computationally difficult than a classical simulation, so it is fortunate that a classical treatment is justifiable for many situations of interest.[3] That this is, in fact, the case can be seen from the following arguments. Let us consider the simulation of a collection of atoms, such as argon. All of the electrons associated with each argon atom are bound fairly tightly to their respective nuclei; the energy required[4] to promote an electron to an excited state, or to remove it entirely from an argon atom, is of order 10 electron volts (eV). This energy is much larger than the typical kinetic energy associated with the center of mass motion of an atom, which is of order 0.1 eV at room temperature. This large energy difference means that collisions between argon atoms will not involve enough energy to have an effect on the electron configuration of either atom. In particular, there is not nearly enough energy available to strip away an electron. For this reason it is a very good approximation to treat each atom as a simple structureless particle. In addition, the DeBroglie wavelength of an argon atom at room temperature is of order 10^{-7} Å. The average spacing between atoms in a solid is of order 1 Å, and we will see shortly that in typical liquids and gasses the atoms never get closer than ~ 1 Å during a collision. Hence, the atomic wavelength is much smaller than the particle separation, which again justifies a classical approach. We can therefore use Newton's second law to calculate the positions of the atoms as a function of time.

It is useful to begin by considering order of magnitude estimates of some of the quantities involved in molecular dynamics. First, the mean number of collisions that a given particle (i.e., molecule) suffers per unit time is related to the particle density n, the mean relative velocity of the particles v_{rel}, and the so-called scattering cross section σ_0 (essentially the target area). The approximate relation is

$$f \approx n v_{rel} \sigma_0 \,. \tag{9.1}$$

It is not surprising that the collision rate (f) is proportional to these quantities as it should increase when n, v_{rel}, or σ_0 increases. The mean length that a particle travels between collisions is called the mean free path ℓ, and this is of order v_{rel}/f, since $1/f$ is the mean time between collisions. Now, if we replace the density n by $1/L^d$ where L is the mean particle separtion in d dimensions, and σ_0 by a^{d-1} where a is the linear dimension of a particle, we get

$$\ell \approx \left(\frac{L}{a}\right)^d a = \left(\frac{L}{a}\right)^{d-1} L \,. \tag{9.2}$$

For gasses and liquids we have $L/a \gg 1$, giving $\ell \gg L \gg a$. It also means that the mean free path is much larger in three dimensions than in two dimensions, for

[3] There are, however, situations where quantum mechanical treatment is necessary and appropriate. An example would be where the chemical bonds between atoms undergo changes. Quantum mechanical molecular dynamics approaches are usually called *ab initio* calculations since the molecular dynamics time evolution is blended with on-the-fly solutions of the Schrödinger equation. You can read more about it in, e.g., Marx and Hutter in the references.

[4] You may recall that the binding energy of an electron in the ground state of a hydrogen atom is ~ 13.6 eV.

comparable L and a, hence requiring more molecular dynamics time steps to simulate a scattering event in higher dimensions. In addition, the number of particles within a given distance from a given particle is much larger in three dimensions than in two if L is comparable. Therefore, on both counts, molecular simulations in three dimensions will be much more costly in computing resources than in two dimensions. We can estimate crudely how much more. Assume that $L/a = 10$, and we take the mean molecular displacement during a time step to be on the order of a itself. In this case, $\ell \approx 10^3$ steps in three dimensions versus 10^2 in two. Suppose further that the calculation involves 10^3 particles in the volume of side length $10L$ in three dimensions, as compared to 10^2 particles in an area of the same side length in two dimensions. Then, taken together, the three dimensional computation will take 100 times longer in this particular scenario than in the two dimensional case. For these reasons, as well as for simpler presentation and coding, we will consider atoms moving in a two-dimensional plane in this chapter.[5]

For each atom i we then have the equations of motion

$$\begin{aligned} \frac{dv_{i,x}}{dt} &= a_{i,x} \ , \\ \frac{dx_i}{dt} &= v_{i,x} \ , \\ \frac{dv_{i,y}}{dt} &= a_{i,y} \ , \\ \frac{dy_i}{dt} &= v_{i,y} \ , \end{aligned} \tag{9.3}$$

where $v_{i,x}$ and $v_{i,y}$ are the components of the velocity of the ith atom, which is located at position (x_i, y_i). The components of the acceleration of each particle, $a_{i,x}$ and $a_{i,y}$, are determined by the forces from all of the other particles in the system. To solve these equations numerically we have several choices. In nearly all of the problems we have encountered so far in this book, either the Euler or Euler-Cromer method has been adequate. However, in molecular dynamics we will be interested in computing the motion over a very large number of time steps, and it turns out that the numerical errors associated with Euler type methods are too big to tolerate. It is therefore necessary to use a slightly more complicated scheme for solving the differential equations arising from Newton's second law. The scheme we will employ here is known as the Verlet method and is introduced in Appendix A. As usual, we discretize time in steps Δt. Letting $x_i(n)$, $v_{i,x}(n)$, and $a_{i,x}(n)$ be the x components of position, velocity, and acceleration of particle i at time-step n, their values at the next time step are given according to the Verlet method as

$$\begin{aligned} x_i(n+1) &\approx 2\,x_i(n) \ - \ x_i(n-1) \ + \ a_{i,x}(n)\,(\Delta t)^2 \ , \\ v_{i,x}(n) &\approx \frac{x_i(n+1) \ - \ x_i(n-1)}{2\,\Delta t} \ , \end{aligned} \tag{9.4}$$

[5]Although an extension of the algorithm to three dimensions is straightforward.

with similar equations for y_i and $v_{i,y}$. These equations are derived in Appendix A, where it is also shown that the numerical errors associated with the Verlet method are much smaller than with the Euler method (for a comparable amount of computing time).

At first glance, (9.4) doesn't look much like our usual Euler or Euler-Cromer expressions. However, the origin of this relation for $x_i(n+1)$ can be appreciated if we recall the finite-difference approximation for a second derivative

$$\frac{d^2x_i}{dt^2} \approx \frac{x_i(n+1) \ + \ x_i(n-1) \ - \ 2\,x_i(n)}{(\Delta t)^2} \ , \tag{9.5}$$

which we have encountered on several previous occasions. If we use the fact that d^2x_i/dt^2 is the acceleration and rearrange (9.5) to solve for $x_i(n+1)$, we obtain precisely (9.4) (a more careful derivation is given in Appendix A). An important feature of the Verlet method is that it conserves energy very well over the course of many time steps, which is crucial if we want to perform an accurate molecular dynamics simulation. It is interesting to note that the position can be calculated directly from the acceleration; we don't really have to calculate the velocity as an intermediate step, as is necessary with the Euler method. However, we will see that the velocity contains some very useful information, so we will always compute it along with the position.[6]

A molecular-dynamics simulation consists of solving the Verlet equations for every particle in the system. A key quantity required in this calculation is the acceleration. Each particle experiences a force from all of the other particles; this is what gives rise to collisions between the particles. To estimate the force between any two particles requires knowledge of the interaction potential, $V(r)$, where r is the separation of the particles. The calculation of $V(r)$ *does* involve quantum mechanics; it depends on what kinds of atoms are involved and the nature of the forces between them. For elements such as argon the situation is relatively simple. For large separations the interaction is due to the Van der Waals force, which is a weak attraction arising from the transient electric dipole moments of the two atoms. We don't have space to give a derivation here;[7] all we really need to know is that this potential varies as r^{-6} and is attractive. When the atoms get close together there is also a repulsive force due to the overlap of their electron clouds. The precise form of this force is hard to calculate, since it involves many electrons. Common practice is to approximate it by a term that varies as r^{-12} and is repulsive. Adding this to the Van der Waals component yields what is known as the *Lennard-Jones* potential

$$V(r) \ = \ 4\,\epsilon \left[\left(\frac{\sigma}{r}\right)^{12} \ - \ \left(\frac{\sigma}{r}\right)^{6} \right] \ , \tag{9.6}$$

where ϵ and σ are constants that set the energy and distance scales associated with the interaction. This function is plotted in Figure 9.1; the associated force is the derivative $F = -\partial V/\partial r$, and is directed along the line connecting the atoms. We

[6]However, the errors involved in the Verlet's method calculation of the velocity are of the same order as in the Euler method; see Appendix A.

[7]See any introductory quantum mechanics text, such as Schiff (1963).

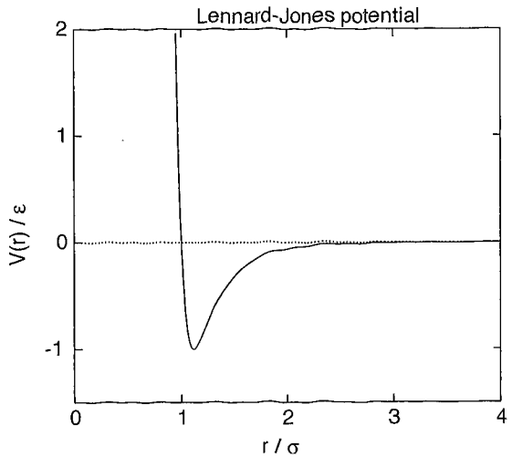

FIGURE 9.1: Lennard-Jones potential.

see from Figure 9.1 that the two atoms experience a significant attractive force when the separation is in the range $\sim$ (1.1-2.0) σ. For separations larger than about 3σ the force is essentially zero, while for $r \leq 1.1\sigma$ the force is very strongly repulsive.

We will follow standard molecular-dynamics practice and use the Lennard-Jones potential in our calculations.[8] Its magnitude is set by ϵ, so it is convenient to work in units for which $\epsilon = 1$. Thus, we effectively measure all energies in terms of ϵ, which for argon is $\epsilon/k_B = 120$ K. Argon is a popular choice for molecular-dynamics simulations, since the Ar-Ar interaction is very well described by the Lennard-Jones potential. Likewise, it is convenient to set the length scale $\sigma = 1$, so that all lengths are measured in units of σ, which has a value of 3.4 Å for argon. These are often referred to in the literature as *reduced units* and are denoted by E^* and r^*, etc.; we will usually just use the symbols E and r for convenience. Finally, standard practice also sets the mass of an atom to unity, so that all masses are measured in terms of the mass of one argon atom. Since energy has units of mass times velocity squared, this leads us to measure time in units of $\tau \equiv \sqrt{m\sigma^2/\epsilon}$ and velocity in units of $\sqrt{\epsilon/m}$, where m is the mass of an argon atom.[9] One (reduced) unit of time for argon is $\approx 1.8 \times 10^{-12}$ s, about 2 picoseconds.

Before we discuss the programming there is one more issue that must be addressed, namely the boundary conditions. Perhaps the most obvious way to enclose

[8]Other potentials are sometimes used, for example, to study systems such as metals in which conduction electrons contribute significantly to the interaction.

[9]Thus, a dimensionless velocity of the order of 1 (i.e., of order $\sim (\sqrt{\epsilon/m})$) corresponds to the kinetic energy of order of ϵ. This makes these units very convenient and reasonable to use in simulations.

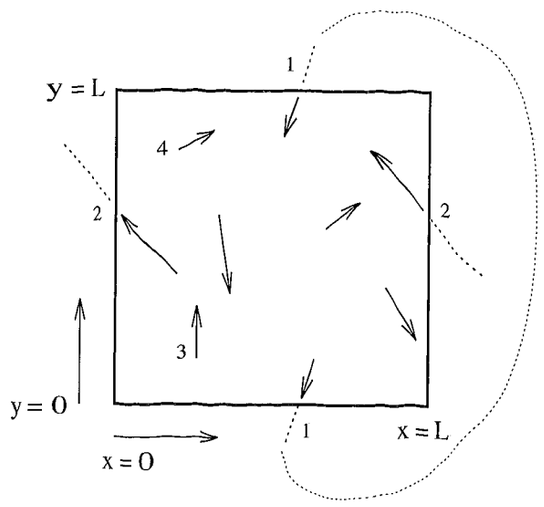

FIGURE 9.2: Periodic boundary conditions for a molecular dynamics simulation using an $L \times L$ box. The arrows denote atoms and their velocities.

the particles is in a box[10] with hard, perfectly reflecting walls. The problem with this approach is that for presently available computers we are limited to systems containing a relatively small number of atoms. In small systems, the collisions with the walls can be a significant fraction of the total number of collisions, in contrast to a real system where the behavior would be dominated by collisions with other particles. Moreover, the *shape* of a small container can greatly affect how the particles are spatially arranged, which can be a serious problem if we want to study a condensed phase (a liquid or a solid). For these reasons it is common practice to use *periodic boundary conditions*. We have already encountered such boundary conditions in our Monte Carlo simulations of the Ising model, but the situation there was a little simpler since the spins did not move. Here periodic boundary conditions work as illustrated schematically in Figure 9.2, which shows a collection of atoms in a square, two-dimensional box with walls at $x = 0, L$ and $y = 0, L$. Whenever an atom encounters a wall it is transported instantly to the *opposite* side of the system. For example, the atom marked 1, which reached the wall at $y = 0$, was transported to the opposite side at $y \approx L$. Note that *only* the y coordinate was affected by this move. The x coordinate and both components of the velocity were unaffected.

One way to view periodic boundary conditions is to imagine that the box is situated on a torus. Two particles that are near opposite faces in a plot such as Figure 9.2 are then actually very close to each other. This avoids all collisions with walls (since there are no walls!), but poses a slight complication when we calculate

[10]We need some sort of enclosure to keep the particle from traveling away and never coming back.

the force between two particles, since this force depends on their separation. For example, particles 3 and 4 in Figure 9.2 are separated by a distance of $\approx 3L/4$ along the y direction. However, if we take advantage of the teleportation of periodic boundary conditions, their separation is only $\approx L/4$. In order for our equations of motion to be consistent, particles should only be allowed to interact once, so we must use the smaller of these two separations to calculate both the magnitude and the *direction* of the force.

We have now described the basic conceptual details associated with molecular dynamics and next consider how to construct a program. Here we will describe only its main components. After setting the size of the box, the number of particles, and the time step, we must pick the initial positions and velocities. These will depend on the particular problem we want to study. As an example, let us consider a fairly dilute gas. Your intuitive picture of a gas probably has the atoms arranged randomly within the box, so it is natural to choose the initial positions randomly. However, this intuitive picture does not take into account the hard-core repulsion of the potential. Even in a gas, the atoms are *not* arranged completely at random. If there is an atom at a particular location (x_1, y_1), the probability of finding a second atom within a distance σ of this position is essentially zero. To take this into account in our initialization it is convenient to give the atoms an approximately regular arrangement so as to guarantee that no two atoms are too close to each other. In the following code example, we do this by first placing the atoms on a square lattice in which the spacing between nearest neighbors is greater than 2σ and then displacing the atoms from these locations at random by a distance $\leq \sigma/2$. This gives a somewhat random initial arrangement and keeps the atoms well separated from each other. The choice of initial velocities is not as complicated. One possibility is to give each particle a speed v_0 in a randomly chosen direction and another is to allow both the direction and the magnitude of the velocity to be random within certain limits. We used the latter approach in the example sketched below. We will consider other possibilities shortly.

EXAMPLE 9.1 Initializing a molecular dynamics simulation

- Choose the number of particles (N), and the size of the system (L).
- Set computational parameters, including Δt (the time step).
- Set initial positions and velocities of the particles:
  - ▷ Set max position component deviation δr from the vertices of a regular array.
  - ▷ Set max initial velocity component magnitude v_0.
  - ▷ Iterate through N particles ($i = 1, 2, ..., N$):
    - Calculate the equidistant grid point coordinates $[g_x(i), g_y(i)]$ for particle i.
    - Displace the particle randomly a bit by setting $x_{curr}(i) = g_x(i) + 2(\mathtt{rnd} - 0.5)\delta r$, $y_{curr}(i) = g_y(i) + 2(\mathtt{rnd} - 0.5)\delta r$.
    - Calculate a randomized initial velocity by $v_{x0}(i) = 2(\mathtt{rnd} - 0.5)v_0$, $v_{y0}(i) = 2(\mathtt{rnd} - 0.5)v_0$.
    - To set the stage for Verlet method calculations, define the (fictitious) position *previous* to this initial time by $x_{prev}(i) = x_0(i) - v_{x0}(i)\Delta t$, $y_{prev}(i) = y_0(i) - v_{y0}(i)\Delta t$.

After initializing the particle positions and velocities, the Verlet method can be used to calculate the subsequent motion of each atom. The acceleration components of particle j are $a_{j,x}$ and $a_{j,y}$, and these are obtained from the Lennard-Jones potential. The components of the acceleration for particle j are obtained by simply summing the individual forces from all of the other particles in the system

$$a_{j,x} = \frac{1}{m}\sum_{k\neq j} f_{k,j}\cos\theta_{k,j}\,, \qquad (9.7)$$

$$a_{j,y} = \frac{1}{m}\sum_{k\neq j} f_{k,j}\sin\theta_{k,j}\,,$$

where $f_{k,j}$ is the force of particle k on particle j (note that in our reduced units $m = 1$), and $\theta_{k,j}$ is the angle that the line between them makes with the x axis. These sums exclude terms like $f_{j,j}$, since a particle does not interact with itself. The pair forces are given by

$$f_{k,j} = -\frac{\partial V}{\partial r_{k,j}} = 24\left(\frac{2}{r_{k,j}^{13}} - \frac{1}{r_{k,j}^{7}}\right),$$

where $r_{k,j}$ is the separation between particles k and j, and we have assumed $\sigma = \epsilon = 1$ so that we are using reduced units as mentioned above. Note that $r_{k,j}$ and the associated angle must be measured with the "minimum" separation rule of periodic boundary conditions. For example, particles 3 and 4 in Figure 9.2 are a distance $\approx L/4$ apart (not $3L/4$). The angular factors in (9.7) can be estimated from the relative positions of the two particles. If their separation along x is Δx, then $\cos\theta_{k,j} = \Delta x/r_{k,j}$, etc. for $\sin\theta_{k,j}$.

At each time step it is necessary to compute the acceleration of every particle. Since each particle interacts with every other particle, this involves the calculation of *many* pair forces $f_{k,j}$. In practice this is the most time-consuming part of the calculation, so it is worth making this part of the program efficient. To this end we first recall from Figure 9.1 that the interaction potential is essentially zero for $r > 3$ (remember that this distance is measured in reduced units, so $r = 1$ corresponds to a real separation of σ). We will therefore take the force to be exactly zero when $r > 3$. This is known as "cutting off" the potential, and speeds things up since we can avoid calculating $f_{k,j}$ for most pairs of particles.[11] Another way to speed

[11] However, our program does calculate $r_{k,j}$ for every pair so that it can decide whether or not to evaluate $f_{k,j}$. We could go a step further and not even bother to calculate $r_{k,j}$ for particles that were very widely separated at the previous time step. This approach will speed things up even more, but the programming is a bit more complicated since we must decide how often to check on the value of $r_{k,j}$.

things up is to note that $|f_{k,j}| = |f_{j,k}|$, which is required from Newton's third law. It is most efficient to calculate all of the $f_{k,j}$ at one time (with two nested loops), being careful that each pair of particles is considered only once.

After calculating the $f_{k,j}$ as just described, our program estimates the new position of particle j using (9.4). At the same time the new velocities are also calculated. Note that with (9.4) we use a symmetric form of the derivative to find the velocities, so when we calculate the positions at step $n+1$ we then obtain the velocities at step n. This will be important[12] when we consider the total energy, since we will want to calculate the kinetic and potential energies at the *same* value of t. After obtaining the new positions we must then check to see if any of the particles have "left" the box. If so, our program uses the periodic boundary condition rules to teleport the particle to the opposite side of the system. To be consistent, we also teleport the previous value of the position since this will be needed in the calculation of the velocity at the next time step. The velocity does not need any adjustment.

We sketch below a subroutine `update` that updates the positions and velocities of the particles implementing these ideas.

EXAMPLE 9.2 Subroutine `update`

- Iterate through N particles ($i = 1, 2, ..., N$) and calculate the new position and velocity for each one.
  - ▷ For particle i, iterate through all other particles, performing the following tasks:
    - Calculate the distance r_{ij} to particle j (taking into account the periodic boundary condition).
    - If $r_{ij} > r_{cut}$ (the cutoff length), skip j.
    - Otherwise, calculate the force f_{ij} on particle i due to j. Add this force vector to the net force $f(i)$ on i from all the other particles.
  - ▷ Use the net force $f(i)$ to update the position and velocity of particle i:
    - $x_{\text{new}}(i) = 2x_{\text{curr}}(i) - x_{\text{prev}}(i) + f(i)_x(\Delta t)^2$, $y_{\text{new}}(i) = 2y_{\text{curr}}(i) - y_{\text{prev}}(i) + f(i)_y(\Delta t)^2$ where the subscripts curr and prev refer to the *current* and *previous* time steps.
    - If the new position falls outside of the system, apply periodic boundary condition and bring it back inside
    - $v_x(i) = (x_{\text{new}}(i) - x_{\text{prev}}(i))/(2\Delta t)$, $v_y(i) = (y_{\text{new}}(i) - y_{\text{prev}}(i))/(2\Delta t)$.
- Again iterate through N particles and this time:
  - ▷ Relabel the current positions as *previous*.
  - ▷ Relabel the newly calculated positions to *current*.

[12] There are several different variations on the Verlet algorithm, as discussed in Heermann (1990), and some of them avoid this problem. Watch out for this when comparing our program with those of other authors.

  - ▷ Calculate and store any quantity you wish to later use, such as the total kinetic and potential energies.

Note that we must calculate the forces acting on all the particles in their current positions first, and then update all of positions at one time; we cannot update a particle's position before calculating the forces on other particles.

This completes one time step of the calculation. The above procedure is then repeated for as many time steps as desired. The results can be monitored and analyzed in several ways, which we will now describe as we consider specific simulations. Figure 9.3 shows the results for 20 particles in a 10×10 box. It is usually not necessary to (permanently) record or display the positions after each time step, since the particles should move little during an interval Δt; otherwise Δt was not chosen small enough in the first place! Here we have plotted the positions sufficiently often that the individual trajectories can be followed without confusing one particle for another. Many pair "collisions" can be noted, although the dots plotted here never actually "touch" each other, since the hard-core repulsion of the Lennard-Jones potential effectively prevents the particles from getting closer than about σ, which is unity in our reduced units. We also see several cases of particles exiting one side of the box and reappearing at the opposite side. A much better feeling for the behavior is obtained from observing the motion in "real time" as the calculation proceeds, but unfortunately we cannot include a movie here. This is also a very good way to find programming errors.

The results in Figure 9.3 show that we can indeed compute the motion of a collection of particles. It remains for us to demonstrate that our simulations have anything in common with a real system. Perhaps the most fundamental issue is whether our system reaches a proper equilibrium state, and if so, how fast it comes into equilibrium. This also raises the question of how such an equilibrium state is described in terms of statistical mechanics. Since the only forces in our problem are those between particles, the total energy must be conserved, corresponding to the microcanonical ensemble of statistical mechanics. There is no external heat bath as we had in our Monte Carlo simulations. Even so, the concept of temperature can still be useful. One way to view this is to imagine a restricted "system" consisting of a small fraction of the particles, with the remaining particles then acting as a heat bath. Thus, we can still use the concept of "temperature," but unlike the case with the Monte Carlo method developed in the previous chapter, the value of T is not an explicitly given parameter. Rather, we will have to calculate T from the behavior of the system. This can be accomplished using the equipartition theorem, which states that for a classical system the average energy of each "quadratic" degree of freedom[13] is $k_BT/2$. This theorem can be used in association with the velocity components of each particle, since the kinetic energy associated with v_x is $mv_x^2/2$, etc., for v_y. The assertion is that our molecular dynamics simulation will describe a system whose temperature can be computed using the equipartition theorem.

To justify this claim we next consider if and how the system in Figure 9.3

[13] That is, each coordinate or velocity that contributes a quadratic term to the energy.

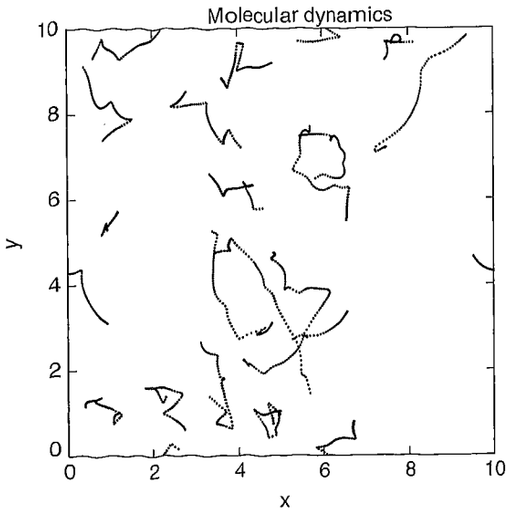

FIGURE 9.3: Trajectories of 20 particles in a 10×10 box with periodic boundary conditions, with initial speeds $v = 1$ in randomly chosen directions. The time step was 0.02 and the positions of the particles were plotted after every third time step.

reaches an equilibrium state. We will focus on the behavior of the particle velocities, since we know that in equilibrium a classical gas is described by the Maxwell distribution. For a two-dimensional gas we have the *speed* distribution

$$P(v) = C \frac{v^2}{k_B T} \exp\left(-mv^2/2k_B T\right) , \tag{9.9}$$

where $P(v)$ is the probability per unit v of finding a particle with speed v, and C is a constant that depends on the mass of the particle. The corresponding *velocity* distribution is

$$P(v_x) = \frac{C_x}{(k_B T)^{1/2}} \exp\left(-mv_x^2/2k_B T\right) , \tag{9.10}$$

with a similar expression for $P(v_y)$. If the molecular dynamics method describes the behavior of a real gas, it should yield velocity and speed distributions that have the Maxwell forms.

In Figure 9.4 we show the speed distribution of the gas in Figure 9.3 calculated by averaging over several different time intervals as the simulation proceeded. Initially all of the particles were given a velocity whose magnitude was unity ($v = 1$ in reduced units) and whose direction was random, yielding a speed distribution as shown by the vertical bar at $v = 1$ in Figure 9.4. We then let the simulation run, and after every 10 time steps the speed distribution was recorded by dividing the v range into bins and tabulating the number of atoms whose speed was in the

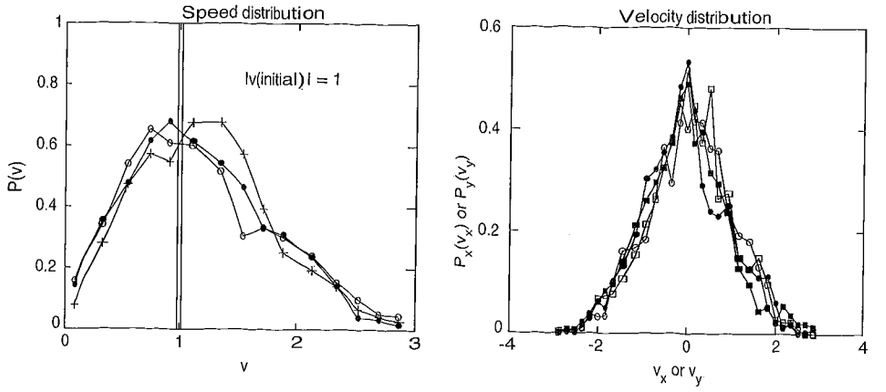

FIGURE 9.4: Left: speed distribution of the gas of particles in Figure 9.3 at various times. At $t = 0$ the distribution was the vertical bar at $v = 1$ (which is not drawn to scale). The other symbols correspond to averages over the reduced time intervals $t =$ 0–20 (solid circles), $t =$ 20–40 (open circles), and $t =$ 40–60 (pluses). Right: velocity distributions obtained in a similar simulation, but using a different initialization of v_x and v_y. These velocity distributions were calculated by averaging over the intervals $t =$ 0–20 and 20–40. Aside from the statistical uncertainties, these results agree with the Maxwell distribution (9.10).

range corresponding to each bin. These histograms[14] were averaged over the time intervals $t =$ 0–20, 20–40, and 40–60, with the results shown in Figure 9.4. We will leave it to the exercises to demonstrate in detail that in all three cases the speed distribution has a form that (to within the statistical fluctuations) is well described by (9.9). The conclusion is that our system did indeed come into thermal equilibrium. Moreover, once reached, this equilibrium distribution was *maintained* at future times, since apart from the statistical fluctuations due to the relatively small number of particles in the system, the distribution remained unchanged as the simulation continued.

The results in Figure 9.4 show that our system reaches the expected equilibrium distribution when started from a particular initial state. It is also interesting to consider the behavior starting from *different* initial states. We know that a real system can be started in different initial states but still reach the *same* final equilibrium state, assuming of course that the temperature, pressure, etc., are kept the same.[15] The same is true for our molecular dynamics model. This can be demonstrated using the speed distribution, as we will explore in the exercises. Here we consider this question using the *velocity* distributions $P_x(v_x)$ and $P_y(v_y)$. While these are closely related to the speed distributions just considered, they contain additional information relating to the direction of the motion. While our results for the speed distribution indicated that the magnitudes of the velocities were distributed properly for a system in thermal equilibrium, the velocity distributions can tell us if the directions are also given correctly.

[14] Note that these distributions have been normalized so that $\int P(v)dv = 1$, etc.

[15] Indeed, this is one of the features that makes the concept of equilibrium so useful in the first place.

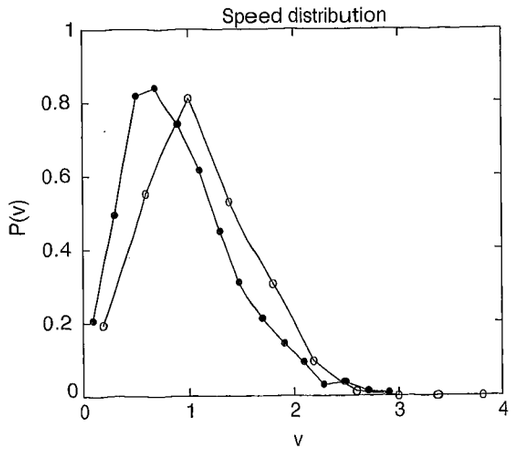

FIGURE 9.5: Speed distributions. Filled circles: obtained after equilibration from a state with initial velocities v_x and v_y chosen randomly in the range ± 1. Open circles: same, but with v_x and v_y chosen randomly in the range ± 2. The system thus had more energy in this case, and the peak in the speed distribution comes at a correspondingly larger value of v. These speed distributions were obtained by averaging over the interval $t = 0 - 20$.

We chose an initial state in which half the particles were given an initial x component of (reduced) velocity chosen randomly in the range $-1 \leq v_x \leq +1$ with $v_y = 0$, and the others were given $v_x = 0$ with $-1 \leq v_y \leq +1$. This is certainly a very peculiar situation, which is nowhere near a Maxwell distribution (9.10), so if the system manages to come into equilibrium starting from this state, we would expect that the same would be found for (most) other initial states (we'll consider some exceptional situations in the exercises). We see from the results on the right in Figure 9.4 that our gas quickly reached the expected distributions for both $P_x(v_x)$ and $P_y(v_y)$; we will leave it to the reader to verify that these distributions do indeed have the Maxwell form (9.10). This provides us with more evidence (and greater confidence) that molecular dynamics provides a way to simulate a mechanical system in thermal equilibrium.

An important quantity for a system in thermal equilibrium is the temperature. As we have already mentioned, T does not enter the simulations as an input parameter. It must instead be "measured," and the results for the speed and velocity distributions give us a way to make this measurement. To demonstrate this we show in Figure 9.5 the results for the speed distributions obtained from two separate simulations. In one case the initial velocity components were both chosen randomly in the range ± 1, while in the second this range was ± 2. Hence, on average, the atoms in the latter case had greater speeds and were, therefore, "hotter." This can be seen from the fact that the speed at which the peak occurs in $P(v)$ is shifted to higher values of v in the system that had the larger initial velocities. Comparing this with the form of the Maxwell distribution, we can see that the

location of this peak is determined directly by the temperature. We could thus use the peak location in conjunction with (9.9) to measure T.

Another approach is to make use of the equipartition theorem. As mentioned above, this theorem states that for a classical system an average energy of $k_B T/2$ is associated with any degree of freedom that enters the energy quadratically. The kinetic energy of an atom in our two-dimensional system is $m(v_x^2 + v_y^2)/2$, so the equipartition theorem tells us that for a system in thermal equilibrium[16]

$$k_B T = \langle \frac{m}{2} (v_x^2 + v_y^2) \rangle , \tag{9.11}$$

where the (reduced) temperature is measured in units of ϵ/k_B, which is ≈ 120 K for argon. Here the angular brackets can be interpreted in two ways. According to one point of view, (9.11) applies to each atom in the system, so we can obtain T by computing the *time* average of the kinetic energy of one particular atom. However, since the atoms are all equivalent, the same result can be obtained by averaging (at a particular instant in time) over the different atoms in the system.[17] In practice, the best computational accuracy will be obtained by combining the two points of view and computing the time average of the kinetic energy per atom averaged over all of the atoms. We will leave it to the exercises to compare the temperature calculated from (9.11) with the value obtained by fitting the Maxwell distribution functions directly to the results in Figures 9.4 and 9.5.

EXERCISES

9.1. Calculate the speed distributions for a dilute gas as in Figure 9.4 and compare the results quantitatively with the Maxwell distribution. (For example, perform the χ^2 analysis described in Appendix G.) This analysis also yields the temperature; compare the value you find with the result calculated directly from the equipartition theorem (9.11).

9.2. Calculate the speed distributions starting from different initial states and show that if the initial kinetic energy is the same and the particles are widely separated so that the potential energy is also the same, the equilibrium distributions are the same. *Hint:* Choose the initial velocities so that the average values of v_x and v_y are both zero. See the next problem for a discussion of why this is important.

***9.3.** Repeat the previous problem, but now consider how (and if) certain "peculiar" initial states approach equilibrium. Consider a dilute gas in a 10×10 box containing 20 particles. Give all of the particles an initial v_x which is positive; you might, for example, choose v_x randomly in the range 0–1. Show that the distribution $P(v_x)$ never assumes the form (9.10), as the average value of v_x will always be positive rather than zero. Explain why this follows from conservation laws. *Hint:* Consider the conservation of momentum.

[16] Note that this assumes that the overall translational kinetic energy of the system as a whole is negligible. For a molecular-dynamics simulation this simply means that the center of mass velocity must be much smaller than that of a typical atom. Otherwise a system in which the atoms were all at rest relative to each other would, if the center of mass velocity were large, have a high temperature, at least according to (9.11). However, we know that this cannot be the case.

[17] This should remind you of the ergodic hypothesis.

9.4. Study the diffusion of particles in a dilute system. For example, take 16 particles in a 10×10 box and calculate the mean-square displacement of an atom as a function of time. Show that the motion is indeed diffusive [that $(\Delta r)^2 \approx Dt$], and find the value of the diffusion constant. You can also study how D varies with density. The diffusion of an atom in a system containing a large number of like atoms is known as self-diffusion. *Hint:* Be sure to allow for the teleportation associated with the periodic boundary conditions when you calculate Δr.

9.5. The first molecular-dynamics simulations (see Alder and Wainwright [1959]) treated a system of hard disks. In this case the potential is zero when the separation exceeds the disk diameter and infinite for smaller separations, and collisions are assumed to be elastic. Perform a simulation for this case and compare the velocity distribution with the Maxwell form. This may remind you of the billiard problem of Chapter 3.

***9.6.** The Lennard-Jones potential (9.6) describes a pairwise interaction between atoms and is a function only of their separation. Explain why this potential would *not* be useful for modeling a system in which the bonding between atoms is covalent. Discuss what new features the potential must have in this case. *Hint:* Consider the role of bond angles.

***9.7.** Investigate the approach to equilibrium of a system containing only a few particles. We have already seen in connection with Figure 9.4 that a gas containing 20 particles will come into equilibrium, as observed using the velocity distribution. However, as an extreme case we know that this will not happen for a gas containing only a single particle, since in this case v_x and v_y will never change (assuming periodic boundary conditions). Perform simulations like those in Figure 9.4 for a system containing 2 particles. Does this system come into equilibrium in the sense of (9.9) and (9.10)? Explain why it does not. *Hint:* Consider how momentum is exchanged in each collision. Repeat the calculation with 3, 4,... particles and explain your results.

9.8. Write a program that uses hard-wall boundary conditions. That is, when a particle hits a wall it should be specularly reflected, as in the billiard problem of Chapter 3. Show that a gas of 20 particles in a 10×10 box reaches an equilibrium state such as that shown in Figure 9.4.

***9.9.** The behavior of the total energy of a gas as a function of temperature can reveal the importance of interactions between the atoms. This can be appreciated by calculating the energy as a function of T for several different densities (a time step of 0.01 is a good choice for all of the simulations below).

(a) Begin by calculating E, the sum of the kinetic and potential energies of all of the particles, for a system containing 16 particles in a box of size 20×20. Show that to a reasonably good approximation E varies linearly with T, with $E \to 0$ as $T \to 0$. Explain why this should be expected for a very dilute gas. *Hint:* When the particles are far apart the potential energy will be negligible, so E is then approximately equal to the kinetic energy. What does the equipartition theorem then tell you about $E(T)$?

(b) Increase the density by confining the same number of particles to a 5×5 box. You should now observe that E does not vanish as $T \to 0$. Explain how this is a result of the interactions between particles. This is an example of a "nonideal" gas.

9.10. In order to fully characterize a gas it is useful to measure (or calculate) the equation of state. For this you need to know the pressure. This can be calculated in a molecular-dynamics simulation in the following way. If the simulation employs hard-wall boundary conditions, the pressure on a wall of the container will be the force per unit area exerted by the particles that are reflected by the wall. This force can be calculated from the momentum change that the wall imparts on each particle it reflects. A simulation that uses periodic boundary conditions will not have any walls or reflections, but the same quantity can be obtained by considering the particles that "pass through" a particular boundary as part of the teleportation process associated with periodic boundary conditions. Every time a particle tries to pass through the surface at $x = +L$ (we assume an $L \times L$ box), it is transported via the periodic boundary conditions to $x = -L$. This particle carries an amount of momentum mv_x in the x direction. If there had been a hard wall at this location, the particle would have been reflected ($v_x \to -v_x$), which would have imparted a momentum $2mv_x$ to the wall (since momentum is conserved in a collision with the wall). The force on the wall is the momentum per unit time that is transferred to it by all collisions, and the pressure is the force per unit area.[18]

Use this approach to calculate the pressure of a dilute gas as a function of temperature. Good parameter choices are 16 particles in a 10×10 box with $\Delta t = 0.02$. Calculate P and show that it varies linearly with T. Explain why this is so. *Hint:* Consider the equation of state for an ideal gas.

9.2 THE MELTING TRANSITION

In Section 9.1 we introduced the technique of molecular dynamics and used it to investigate several properties of a dilute gas and how it approaches equilibrium. In this section we will use the same method to investigate the melting transition. Melting is a phenomenon in which the interactions between particles play a crucial role. The phases involved in melting, the liquid and solid phases, are direct results of these interactions. Hence, in order to provide a quantitative description of melting, a method that treats interparticle interactions in a realistic manner is essential. Molecular dynamics is an ideal choice.

We have already mentioned that melting is a first-order phase transition, so we expect to find an abrupt change in the system when it melts. One of our tasks will be to devise useful measures of "liquidness" and "solidness." This will turn out to be a little more difficult than you might suspect and will force us to think carefully about what is meant by the terms *liquid* and *solid.*

Let us first establish that our molecular-dynamics approach yields a system that is a solid under the appropriate conditions, namely low temperatures and high

[18] Another common way to calculate the pressure and the equation of state involves the application of so-called virial theorem. The virial theorem equation of state for pair-wise interacting particles is discussed in standard statistical mechanics texts such as Chandler and Pathria in the references. In the context of molecular dynamics, it is useful to write the virial theorem in the form of

$$PV = Nk_BT + (1/6)\langle \sum_{i \neq j}^{N} \mathbf{r}_{ij} \cdot \mathbf{F}_{ij} \rangle ,$$

where the summation is over all pairs of particles and $\mathbf{r}_{ij}$ is the displacement vector and $\mathbf{F}_{ij}$ is the force respectively between particles i and j. This method of calculating the pressure P is sometimes preferred since it involves all the particles in the system rather than relying on the few particles that strike a surface as in the method described above.

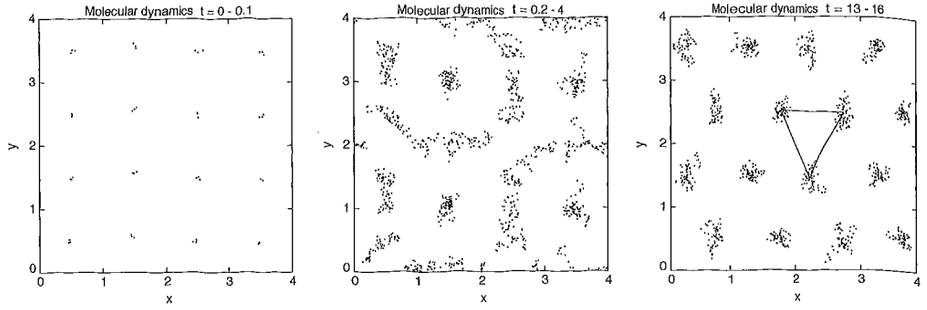

FIGURE 9.6: Snapshots of a system of 16 particles in a 4×4 box at a low temperature. The time step in the simulation was 0.005 in reduced units. These are "time-lapsed" pictures, taken over the time intervals indicated in each figure. The position was recorded after every 10 time steps. On the right we have drawn lines connecting several nearest neighbor atoms to make the triangular structure clear.

density. In order to make T as low as possible, we will start with all of the particles at rest. This does not mean that they will remain at rest, since each will move in response to forces from neighboring particles. Nevertheless, this will give us a low initial temperature. The density is also important, since we expect that for low densities, that is, for a large average particle separation, the system will be a gas. Examining the Lennard-Jones potential in Figure 9.1 we note that the maximum attraction occurs for an interparticle spacing of approximately $1.2\sigma = 1.2$ in reduced units. Hence, we are led to choose a density of approximately 1 particle for each (reduced) unit of area.

Some results for such a dense system are shown in Figure 9.6. Here we consider 16 particles in a 4×4 box, with the particles initially arranged on a square lattice. However, while the particles were at first all at rest, this arrangement was not stable and they immediately began to move; Figure 9.6 shows "time-lapsed" snapshots of the system during various time intervals. The plot on the left shows snapshots taken over the first few time steps, and it is seen that the particles have moved only a little from their initial positions, as the square lattice is still apparent. However, when given some time the particles move substantial amounts, as shown in the plot in the center, which shows a superposition of snapshots taken during the period $t =$ 0.2–4. Eventually an equilibrium state is reached, which is shown in the picture on the far right in Figure 9.6. While the particles are still in motion here, each moves in a region that is only $\approx \pm 0.2$ units in size. This is a crystalline solid.[19]

[19] We need to choose our words carefully here, since the stability of a solid in two dimensions is a rather tricky issue. It turns out that in two dimensions, an infinitely large solid would actually be unstable. That is, an infinitely large two-dimensional crystal would not be the thermodynamically stable phase at any nonzero temperature. This has been established through exact analytic arguments, as described by Mermin (1968) (see also Nelson [1983]). Nevertheless, it appears that finite systems, such as those we study in this section, can be in a state that for nearly all practical purposes is a crystalline solid. The melting transition of an infinitely large two-dimensional system is believed to differ in some subtle ways from that of a three-dimensional solid, as discussed in the references. We will ignore these complications here, as they do not affect the points we wish to make regarding the qualitative aspects of melting. To within the accuracy and resolution of our simulations it is safe to assume, as we will below, that our two dimensional system is an ordinary solid at low temperatures and that the melting transition in two dimensions is an ordinary first-order transition.

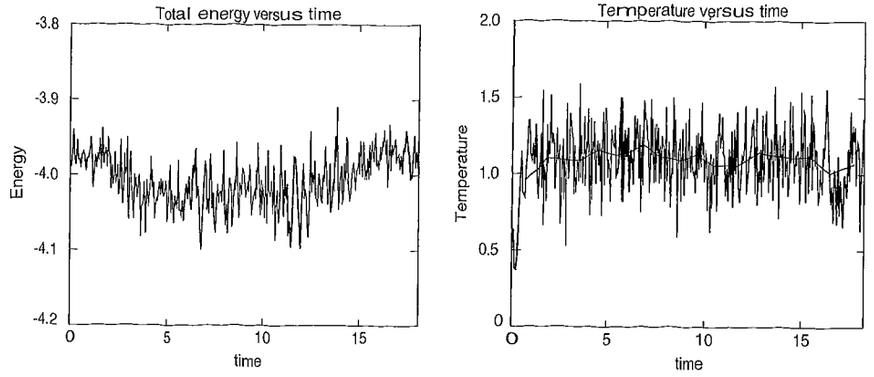

FIGURE 9.7: Energy and temperature as functions of time for the simulation in Figure 9.6. The solid slowly varying line on the right shows the temperature as averaged over intervals of $\Delta t = 1$.

However, we can see that the arrangement of the atoms now is *not* a square lattice (as set up originally). Rather, we have a *triangular* lattice. It turns out that this structure minimizes the energy[20] for particles interacting through the Lennard-Jones potential.

Given the rather small size of our system we might worry that boundary effects could influence the crystal structure. Even though we have employed periodic boundary conditions, this does *not* eliminate the effects of the boundaries. These boundary conditions still impose a "square" shape on the system, since the periodicity conditions have that geometry. The fact that we have obtained a structure that is far from square implies that a triangular structure has a significantly lower energy than a square lattice and that the free energy barrier that separates the two phases is also not too large (this should remind you of our discussion of hysteresis in connection with the Ising model in Chapter 8). Thus, it is probably safe to conclude that in this case a triangular lattice is the most stable solid structure for the Lennard-Jones potential. It is possible to treat the periodic boundary conditions in a more general way, in which the *shape* of the effective box is allowed to vary. Such an approach is more complicated, but is required if we want to determine the most stable solid without any bias imposed by the simulation.

We mentioned earlier that the Verlet method was chosen because it conserves the energy fairly well over the course of many time steps. This claim is justified in Figure 9.7, which shows both the total energy (kinetic plus potential) and the temperature for the simulation that yielded the snapshots in Figure 9.6. Since

[20] Actually it minimizes the free energy, but at the low temperature considered here this is nearly equal to the energy.

there are no external sources of energy, E should be conserved exactly. We see that the energy remained approximately constant (the vertical scale here is greatly expanded), with fluctuations of about ± 2 percent due to numerical errors associated with the Verlet method. These errors could be made smaller by using a smaller time step, but this would require more computer time. The time step used here is suitable for our purposes, but a research calculation would probably strive to keep variations in E below 1 percent.

The plot on the right in Figure 9.7 shows the variation of the temperature for the same simulation. As in the previous section, we have calculated T using the equipartition theorem, (9.11). It is important to realize that this relation holds for any classical system, even one in which the interactions between particles are strong, as they are here. While the fluctuations in the energy are fairly small, the corresponding fluctuations in the temperature are much larger. They are so large, in fact, that we might worry that the concept of temperature may not be useful (or appropriate). However, the difficulty here is not with the concept of temperature, but in our calculation of the averages needed to evaluate (9.11). To estimate T we need to compute the averages of v_x^2 and v_y^2 for all of the particles. In the simulation here the number of particles was fairly small (only 16), so we shouldn't expect these averages to have a high precision. However, this problem can be circumvented by performing an additional time average of the values of T in Figure 9.7, an example of which is shown by the solid line on the right in Figure 9.7. Here we have averaged the values of T over time intervals of size $\Delta t = 1$, and the fluctuations are now much reduced. The message here is that when evaluating the averages that arise in statistical physics, the fact that there is a small number of particles must be kept in mind. Actually, while this is a concern for small-scale simulations like those we have used here to illustrate the molecular-dynamics algorithm, it would usually not be a serious problem in a research calculation, as these typically use many thousands of particles or more.

Now that we have obtained a system that is a solid, we must devise a method to heat it in order to observe melting. This is usually accomplished by increasing the kinetic energy "by hand." That is, we increase the velocities of all of the particles by a factor that is greater than unity. This gives them all some additional kinetic energy and through (9.11) will increase the temperature. After increasing the velocities we must then give the system a chance to come into equilibrium. The additional energy is injected as purely kinetic energy, but we know that as the system comes back into equilibrium this energy will be redistributed among the kinetic and potential energies of the particles.

As a programming note, we must be careful when rescaling the velocities since it is the positions that enter[21] the equations of motion (9.4). With the Verlet algorithm we have the position at both the current and the previous time step. A convenient way to rescale the kinetic energies is to adjust the location at the *previous* time step in the following way. Let $\vec{r}_c$ and $\vec{r}_p$ be the current and previous positions [$\vec{r}_c = (x_c, y_c)$, etc.]. If we want to increase the velocity by a factor of 2,

[21] While the velocities can be calculated at each time step, for the form of the Verlet method that we have described they do not enter into determining the position at the next time step.

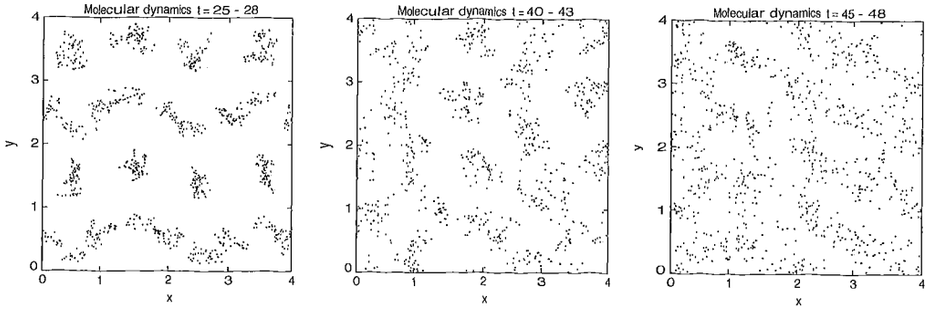

FIGURE 9.8: Time-lapse snapshots at several temperatures as the system considered in Figure 9.6 was heated in stages. The time intervals are given at the top of each plot, and the corresponding energy and temperature are shown in Figure 9.9.

we adjust $\vec{r}_p$ so as to make it twice as far from $\vec{r}_c$. In general, to rescale the velocity by an amount R we take

$$\vec{r}_p \rightarrow \vec{r}_c - R(\vec{r}_c - \vec{r}_p)\,. \tag{9.13}$$

This is not the only way that we could rescale the velocities, but it does have one important property: it does not alter the current positions. If we had, instead, chosen to adjust $\vec{r}_c$, we would affect the current potential energy, as well. However, as long as the time step is small, either approach would be suitable. While we will usually use this rescaling procedure to increase the temperature, it can also be used to decrease the temperature, as would be needed to study freezing or condensation.

In Figure 9.8 we show snapshots of our system as it was heated in stages. The corresponding energy and temperature as functions of time are shown in Figure 9.9, where the abrupt steplike increases in E show the times when the kinetic energy was increased. At each of these points the velocities were increased by a factor of 1.5, and as expected the temperature increases along with E. The fluctuations in T also increased dramatically as the temperature was raised. In our discussion of the Monte Carlo method in Chapter 8, we saw that such fluctuations are related to quantities such as the specific heat (we will leave the exploration of this issue to the industrious reader).

Our goal in this section is to observe the melting transition, and it is time for us to finally tackle that problem. By examining the snapshots of the system at various temperatures in Figure 9.8, we can clearly see by eye that the system became increasingly disordered as the temperature was raised. However, it is not obvious (at least to the authors) how to locate the melting transition from these snapshots alone. In a sense, the information in the snapshots is too "microscopic." Most of the properties that we associate with a solid are based not on the positions of individual atoms, but on thermodynamic types of variables. Unfortunately, thermodynamic variables are often the ones that are most difficult to determine accurately with molecular dynamics.

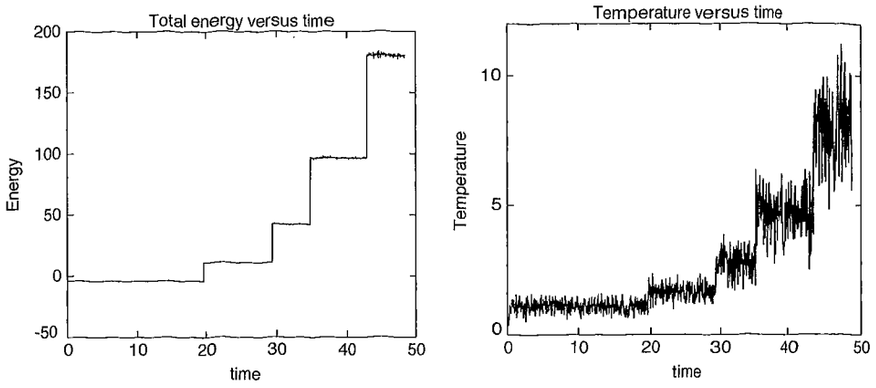

FIGURE 9.9: Energy and temperature as functions of time for the simulation in Figures 9.6 and 9.8. Each abrupt increase in E was produced by rescaling the velocities by a factor of 1.5.

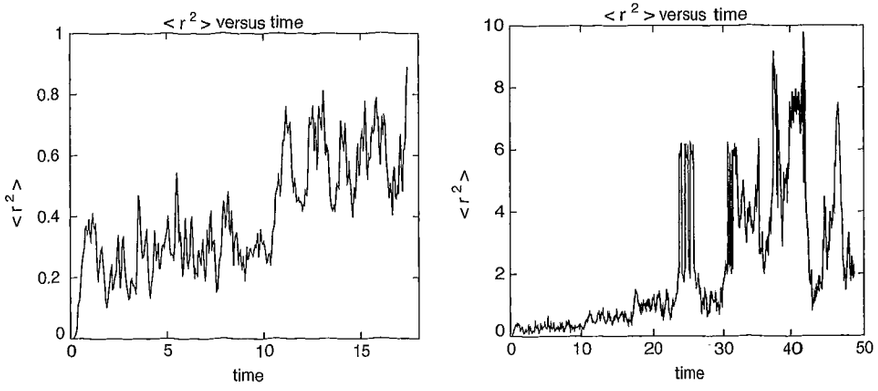

FIGURE 9.10: Motion of a "test" particle in the simulation of Figure 9.8. Left: behavior at early times corresponding to a temperature $T \sim 1.1$; right: variation of $< r^2 >$ as the temperature was increased in stages as shown on the right in Figure 9.9. Note that the vertical scale is very different in the two plots.

What we really need is a quantitative measure of solidness and liquidness; preferably this measure should be closely connected with the motion of individual atoms, since that is the type of information we have at our disposal. It turns out that there are several such measures that could be employed. One is to consider in detail the motion of a particular particle, which might be termed a "test particle" even though it is equivalent to all of the others in the system. To illustrate the approach, we have chosen one particle in Figure 9.8 and recorded its position as a function of time. The square of the displacement from its initial position is shown in Figure 9.10. These values were obtained from the simulation considered in Figure 9.9, so the corresponding temperatures can be read from those results. At the lowest temperature (corresponding to the earliest times) the particle quickly settled into a position a small distance away from its initial position, as the structure changed from the initial square form to a triangular lattice. Our test particle then remained nearly stationary up to $t \approx 11$. While the particle did move a little during this interval, the mean-square displacement was a small fraction of the spacing between atoms (which was ≈ 1). There was an abrupt shift in the particle's position at $t \approx 11$, but the overall displacement was still much less than the average spacing between particles and was probably due to motion of the entire lattice.[22] Overall, the behavior for $t < 11$ is that expected for a particle in a solid.

As the temperature was increased, the fluctuations of the particle's position became larger. During the period $t \approx 23$–26 it shifted back and forth between two fairly well-defined locations, before returning (approximately) to its original location at $t \approx 28$. It thus appears that the particle was moving between several different lattice sites, or perhaps the entire lattice was shifting in space. With either interpretation, while our test particle was certainly more mobile than it was at lower temperatures, the system was probably still a solid, since there appear to be fairly discrete "lattice positions" available to the particle. However, after the temperature increase at $t \approx 30$, the behavior changed qualitatively, as the displacement fluctuated rapidly over distances much greater than a lattice spacing and did not tend to prefer any discrete values. This suggests that the system had become a liquid.[23] These fluctuations increased further in magnitude as the system was heated again at $t \approx 35$ and 42. From the corresponding results for the temperature as a function of time in Figure 9.9 we conclude that the melting transition took place at a temperature somewhere in the range $T \sim 1.5$–2.0.

There are several other measures that we can use to discern whether the system is a solid or a liquid, several of which will be explored in the exercises. We will next consider one that involves the *relative* separation of two atoms. We monitor the square of the separation of two adjacent atoms, $(\Delta r)^2$, as a function of time, the idea being that in a solid this separation will remain (fairly) constant with time, while in a liquid it will not, since in the latter case the atoms will undergo diffusion. Some results of this kind are given in Figure 9.11, which shows $(\Delta r)^2$ as

[22] Precisely what happened at $t \approx 11$ could be determined by examining snapshots before and after the displacement. The beauty (and power) of molecular dynamics is that we have such information in great detail.

[23] Strictly speaking, from these results we really can't distinguish between a liquid and a gas. That would require a study of the phase diagram.

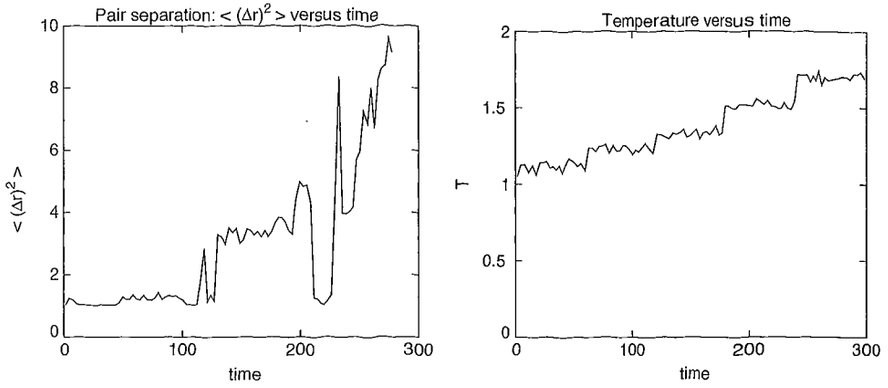

FIGURE 9.11: Left: mean-square separation of a pair of atoms as a function of time while a system similar to that in Figure 9.8 was slowly heated; right: corresponding variation of the temperature. The temperature was increased by rescaling the velocities by a factor of 1.1 at $t = 60, 120, 180, \ldots$.

the system was slowly heated, starting again from the solid phase. The separation was initially ~ 1 (in reduced units) as the atoms were in adjacent lattice sites. This separation remained relatively constant up to about $t \approx 120$, when it increased to $(\Delta r)^2 \sim 3.5$ (recall that here we are plotting the square of the separation), but then it stayed at that value. This indicates that one of the atoms in the pair jumped to a different lattice site, but since the separation remained at a (relatively) "quantized" value, the system was still a solid. Comparing this with the triangular lattice in Figure 9.6, we see that this separation corresponds to the separation of second nearest neighbors in our triangular lattice. [$(\Delta r)^2 \sim 3.5$ in Figure 9.11 implies a spacing of about 1.9 in reduced units.] However, at $t \approx 200$ the separation began to change rapidly and erratically with time, signaling a much enhanced motion of one or both of the atoms. This is precisely what we expect for a liquid, and we conclude that melting occurred at $T \sim 1.5$. This value is a little more accurate than that obtained from Figure 9.10, as in this simulation the system was heated more slowly.

A key feature of these results is that the melting transition was, to within the resolution of these simulations, abrupt. There were no warnings or enhanced fluctuations to indicate that the transition was imminent. This is in accord with the claim we made earlier that the melting transition is *first order*. While this has admittedly been a very rough analysis of melting, our results do show that the behavior changes dramatically as the temperature is increased. To do a better job of locating the transition would require a careful examination of other properties. We will leave this to the exercises (and also to the readers' exploration of references).

EXERCISES

9.11. Study the melting transition at different densities. You should find that the melting temperature drops as the density is reduced. For 16 particles in a box of size 4.3×4.3, you should find melting near $T = 1.0$. Toxvaerd (1978) describes a careful study of the melting curve in two dimensions.

9.12. In our determination of the lattice structure in Figure 9.6, we began the simulation with the particles in an ordered array. How do we know that the triangular lattice we found is really the stable lattice type? One way to address this question is to repeat the simulation with the atoms in different initial arrangements. For example, we could begin with the atoms in a honeycomb lattice (in which all of the atoms have three nearest neighbors), or you could start with a disordered arrangement. Perform this simulation at several temperatures below the melting temperature determined from the results in Figure 9.11 (keeping the same density as in that simulation), and determine the final stable structure of the atoms.

9.13. Investigate melting in a three-dimensional system. First determine the structure of the solid (it should be face-centered cubic), then try to locate the melting temperature.

***9.14.** A useful quantity for studies of structure is the pair correlation function $g(r)$. This is the density of particles per unit distance, at a separation r from any given particle. Calculate this from a simulation like that in Figure 9.8, with 16 particles in a 4×4 box and a time step of 0.001. Take one particle as the "origin" and let r be the distance as measured from this point. Divide the r axis into bins (try ~ 40 or so, from $r = 1$ to 3 for this rather small system), and after every 10 time steps record the number of particles in each bin. After many time steps this will yield a histogram that is proportional to $g(r)$. Compare the results for $g(r)$ in the solid and liquid phases. You should find that at low temperatures (i.e., in the solid phase) the correlation function has a large peak at $r \approx 1.1$, with a slightly smaller peak at $r \approx 1.9$ and a much smaller one at $r \approx 2.8$. These correspond to the spacing of first-, second-, and third-nearest neighbors in the (triangular) lattice (Figure 9.6). The (relative) size of the first peak should decrease and its width should increase when the system melts. Give a qualitative argument to explain this. Why do you expect to find peaks in $g(r)$ even in the liquid state? Rahman (1964) gives results for $g(r)$ for a three-dimensional Lennard-Jones system.

***9.15.** Investigate condensation from a gas. Begin with the system in a fairly dilute high-temperature phase; a system of 16 particles in a 30×30 box is a good choice. Then gradually cool the system and see if you can observe condensation. *Hint:* It is worth thinking carefully about how you can distinguish a liquid from a gas.

***9.16.** Consider again the process of self-diffusion discussed in the exercises in Section 9.1. Here we are interested in using the associated diffusion constant to study melting and, in particular, to locate the melting transition. Calculate the diffusion constant for self-diffusion for a system of 16 particles in a 4×4 box as a function of temperature. Compare its value in the solid and liquid phases. Can it be used to determine when the system melts? *Hint:* To improve your statistical accuracy, average the diffusion constants for all of the particles; this makes sense since they are all identical and should, therefore, diffuse in the same manner. However, be sure to account for the teleportation associated with the periodic boundary conditions when calculating the mean-square displacement of an atom.

9.3 EQUIPARTITION AND THE FERMI-PASTA-ULAM PROBLEM

Molecular dynamics lies at the interface between two of the foundational pillars of physics: classical dynamics and statistical mechanics. On the one hand, from the viewpoint of classical dynamics, the world contains a collection of masses that move according to Newton's laws. These laws can be expressed as differential equations; these equations have deterministic solutions, and are the basis of molecular dynamics. On the other hand, a fundamental basis of statistical mechanics is the ergodic hypothesis, which implies that systems with many degrees of freedom will move to a state of thermodynamic equilibrium. This leads to results such as equipartition, which we saw plays a central role in molecular dynamics. The determinism of classical dynamics may, at first glance, seem to be incompatible with the tendency towards disorder inherent in statistical mechanics. However, there is now a fairly happy marriage of these two areas of physics. The groundwork for this marriage was prepared by Poincaré in his work on the three body problem, which led, eventually, to the understanding of chaos in dynamical systems that we discussed in Chapter 3. The basic idea is that systems with many degrees of freedom can exhibit chaotic behavior, with an extreme sensitivity to initial conditions. This in turn can lead to effectively ergodic behavior. However, this simple picture of the interface between classical dynamics and statistical mechanics is actually a little too simple. This lesson is illustrated nicely by a problem that was introduced 50 years ago.

When general purpose computers first became available, several notable physicists, including von Neumann and Fermi, suggested that computational physics should be used to do "experiments." The idea was to use computational techniques to solve problems that had resisted theoretical attack. The insights gained from the computational results could then be used to guide subsequent analytic work, and thereby move a field forward. This approach has been used very effectively in recent years.

One of the very first problems that was attacked in this way is the Fermi-Pasta-Ulam (FPU) problem. The problem is sketched in Fig. 9.12. A collection of masses are connected by springs to form a one dimensional vibrating system. For simplicity, all of the masses are identical, all of the springs are the same, and the masses at the ends are connected (via springs) to rigid walls. This is a simple vibrating system, and is very similar to several problems that we have studied in other chapters, including vibrating strings and membranes. From that work we know what will happen if the masses in Fig. 9.12 are connected by Hooke's law (linear) springs. In this case the system can be described by normal modes of vibration. Each of these modes acts as an independent oscillator, with its own frequency and displacement pattern. The normal modes of the one dimensional FPU system with linear springs are, in fact, identical to the standing waves that one finds for a vibrating string (Chapter 6).[24] In this case we know that if the system is set up to vibrate in one normal mode, that vibration will continue forever, with the energy staying in the initial mode. In a linear system (Hooke's law springs) each normal mode behaves as a completely independent oscillator, uncoupled from all of the other modes.

[24]These frequencies and displacement vectors can also be obtained by solving the corresponding eigenvalue problem, as we discuss in Chapter 11.

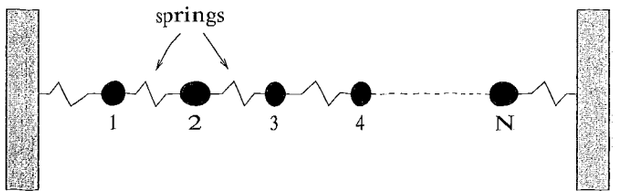

FIGURE 9.12: Fermi-Pasta-Ulam system of masses connected by nonlinear springs. The black dots are each of mass m, and are connected by nonlinear springs described, e.g., by (9.15) or (9.16).

However, the situation is quite different when the springs are nonlinear. Such a nonlinear system does not possess "pure" normal modes. When the nonlinearity is weak, one can think of the system as consisting of weakly coupled modes, and this coupling will cause energy to be exchanged and shared among the modes. Roughly speaking, this sharing is the basis for equipartition of energy and hence ergodicity. Fermi, Pasta, and Ulam wanted to study this energy sharing process in detail. They expected that adding a small nonlinearity to the springs would lead to equipartition of energy, and they wanted to calculate how fast this happens. This calculation can then form the basis for understanding phenomena such as thermal conductivity, a problem that has been studied by subsequent workers. However, we will see that FPU had their hands full with the equipartition problem, as the system in Fig. 9.12 with nonlinear springs did *not* behave in the simple manner expected by FPU.

A simulation of the FPU system is very similar to a molecular dynamics calculation, as we can again use the Verlet method (9.4). The FPU problem is a bit simpler, since (according to the original formulation by FPU) we only need to worry about motion along one direction, which we will call x. If x_i is the displacement of mass i from its equilibrium position in Fig. 9.12, the equation of motion for this mass (just Newton's second law, $F_i = m_i a_i$) has the form

$$m_i \frac{d^2 x_i}{dt^2} = f_{\rm spring}(x_i - x_{i+1}) + f_{\rm spring}(x_i - x_{i-1}) \,, \tag{9.14}$$

which is essentially the same as (9.3). The force is now given by the spring functions $f_{\rm spring}(\Delta x)$, where Δx is the amount that a spring is stretched or compressed. FPU considered several different types of nonlinear functions for $f_{\rm spring}$. Two convenient and much studied choices are a quadratic spring

$$f^{\alpha}_{\rm spring}(\Delta x) = -K(\Delta x) - \alpha(\Delta x)^2 \,, \tag{9.15}$$

and a cubic spring

$$f^{\beta}_{\rm spring}(\Delta x) = -K(\Delta x) - \beta(\Delta x)^3 \,, \tag{9.16}$$

and we will refer to these as alpha and beta-type springs. The two types of springs lead to very similar behavior, so we will only show simulation results for one of them, the beta-type; we'll leave it to you to explore the other type in the exercises.

We can derive the equation of motion for an FPU system with beta-type springs by inserting the force law (9.16) into the equation of motion (9.14). For simplicity (and following the work of FPU) we will take $m = 1$ for all of the masses, and the Hooke's law term $K = 1$ for all of the springs, so that β measures the strength of the nonlinearity. If we have N masses, then the equation of motion (9.14) is actually a system of N equations. For each one we write the second derivative term in finite difference form (9.5), with a time step Δt. Using our usual notation we let $x_i(n)$ be the displacement of mass i at time step n. We can rearrange our finite difference equations to solve for the displacement of each mass at time step $n+1$ in terms of the displacements at previous time steps, with the result

$$x_i(n+1) = 2x_i(n) - x_i(n-1) + \frac{1}{(\Delta t)^2}[x_{i+1}(n) + x_{i-1}(n) - 2x_i(n)] \quad (9.17)$$
$$+ \frac{\beta}{(\Delta t)^2}([x_{i+1}(n) - x_i(n)]^3 + [x_{i-1}(n) - x_i(n)]^3) \ .$$

From a numerical point of view, (9.17) is equivalent to both the Verlet method *and* to the approach we used to deal with waves on a string in Chapter 6. We thus expect (9.17) to be stable and efficient.

We now follow the work of FPU and consider the behavior when we begin in one of the normal modes of the corresponding linear system. As we already mentioned, these normal modes are simply standing waves, and they can be described by

$$x_i = A\sqrt{\frac{2}{N+1}} \sin\left(\frac{ik\pi}{N+1}\right) , \quad (9.18)$$

where N is the number of masses in the system, and k is the mode number ($k = 1, 2, 3, ...N$).[25] The wavelength λ_k of this mode is equal to $2(N+1)/k$. The first three of these normal modes are sketched in Fig. 9.13.

We are now ready to consider the behavior of a FPU system. We consider a system with $N = 32$ masses, with beta-type springs. The system is given an initial displacement corresponding to the lowest mode (mode 1 in Fig. 9.13) and then released. Since there are 32 vibrating masses, there are a number of different ways that we could plot the behavior. One useful way to display the results is to examine the energy in the different modes. An FPU system is analogous to a vibrating string, and the normal modes correspond to standing waves. At any particular instant in time the FPU displacement profile can be written as a sum of these normal mode displacements; i.e., just as the profile of a vibrating string can be written as a collection of standing wave displacements. Since the FPU normal modes are sine waves (9.18), this is also equivalent to writing the FPU displacement as a Fourier series. Hence, the displacements of the FPU masses can be written as

$$x_i = \sqrt{\frac{2}{N+1}} \sum_{k=1}^{N} a_k \sin\left(\frac{ik\pi}{N+1}\right) \ . \quad (9.19)$$

[25] The factor in front of the sin term in (9.18) follows the standard terminology in the FPU literature. Including this factor will make it simpler to compare your results with those of others. A is an arbitrary amplitude factor.

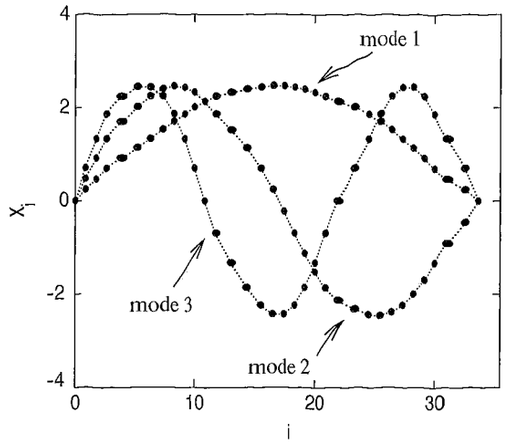

FIGURE 9.13: Lowest three normal modes of a *linear* FPU system with $N = 32$ masses with the amplitude $A = 10$. These are simply standing waves, with zero displacement (nodes) at each end. The black dots represent the masses, and the dashed lines can be thought of as the connecting springs.

Here the sum is over the N normal modes, and a_k is the amplitude (or contribution) of mode k. Equation (9.19) is just a Fourier transform, so there is also an inverse transform

$$a_k = \sqrt{\frac{2}{N+1}} \sum_{i=1}^{N} x_i \sin\left(\frac{ik\pi}{N+1}\right) \ . \quad (9.20)$$

There will be kinetic and potential energies associated with each mode, so the energy of mode k will be given by

$$E_k = \frac{1}{2}\left[\left(\frac{da_k}{dt}\right)^2 + \omega_k^2 a_k^2\right] \ . \quad (9.21)$$

The first term is just the kinetic energy of the mode (proportional to the square of the mode velocity), while the second term is the potential energy. The frequency of mode k is[26]

$$\omega_k = 2\sin\left(\frac{k\pi}{2(N+1)}\right) \ . \quad (9.22)$$

Our FPU simulation yields the displacements of all of the masses, x_i, as functions of time. We can then use (9.20) to calculate the amplitude of each mode a_k as a function of time, and obtain the energy of each mode using (9.21). Some results are shown in Fig. 9.14, which shows very striking and unexpected behavior. The energy starts in mode 1, and a small amount rapidly goes into the higher

[26] Note that this corresponds to $\omega_k = 2\sin(\pi/\lambda_k)$. The similar problem of finding the frequencies of string vibration modes is discussed in Chapters 6 and 11.

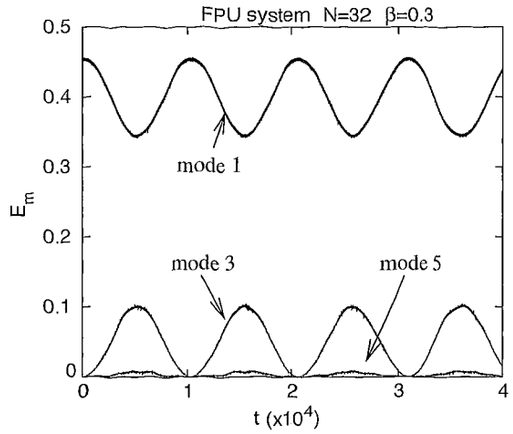

FIGURE 9.14: Energy in modes 1, 3, and 5 as a function of time for an FPU system with $N = 32$ and $\beta = 0.3$. The system was initially given a displacement (9.18) with $m = 1$ (mode 1), with amplitude $A = 10$. For this simulation we used $\Delta t = 0.05$.

modes[27] 3 and 5. However, this energy quickly *returns* to mode 1, and the modal energies appear to oscillate forever. Hence, energy is *not* shared equally between the modes; there is no equipartition of energy. Indeed, the system seems to oscillate in a very regular manner, with no hint of the chaotic behavior we had expected to see for an ergodic system.

This result was very surprising to Fermi, Pasta, and Ulam, and they spent a good deal of time investigating the behavior in many other cases. Figure 9.15 shows results for larger values values of β. Increasing β makes the system more nonlinear, which should produce stronger mixing of the energy between modes, and that is what we seem to see with $\beta = 1$. The oscillations are not quite as regular as at smaller β (Fig. 9.14), and more energy finds its way to the other modes, but the energy is still not distributed in any sort of even way across the modes. However, when the nonlinearity is increased further to $\beta = 3$ the behavior changes dramatically. The system now appears to be chaotic and the energy is shared much more evenly between the two modes that are considered in Fig. 9.15. The behavior here certainly seems consistent with being ergodic, although a more thorough analysis (some of which is left for the exercises) is needed to confirm this.

One way to investigate the possibility of chaos at large values of β is to study the trajectories in the appropriate phase space. In the FPU problem, the normal mode amplitudes in (9.20) form a very useful phase space. However, since there are $N = 32$ masses, we have a $2N = 64$ dimensional phase space corresponding to the

[27]Note that for an FPU system with beta-type springs, the symmetry of the spring force does not allow energy to move from an odd to an even mode. That is, parity is conserved. So, in this example no energy is transferred to the even modes.

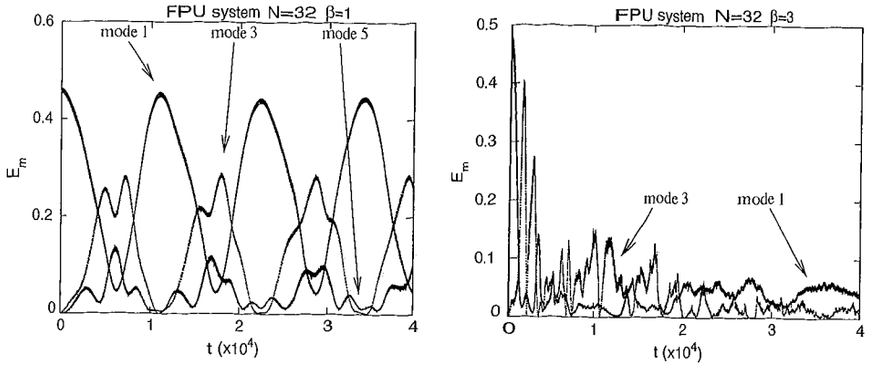

FIGURE 9.15: Energy as a function of time for a few of the lowest modes of an FPU system with $\beta = 1$ (left) and $\beta = 3$ (right). Each system was initially in mode 1 at $t = 0$, with $A = 10$.

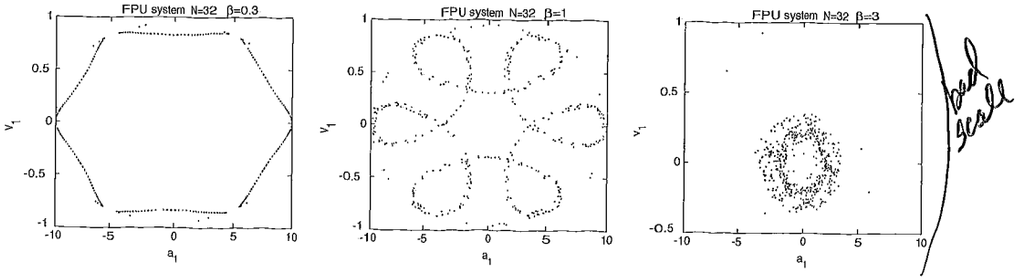

FIGURE 9.16: Phase space plots of the trajectory of an FPU system with beta-type springs. Here we have plotted the "velocity" of mode 1 (da_1/dt) as a function of a_1 whenever the system crosses the $a_3 = 0$ plane in phase space.

amplitudes and velocities of all of the normal modes. We must therefore consider a restricted portion of this space. It is useful to again think of our FPU system as moving on a trajectory in this phase space. We can then ask where the system is located when it crosses a particular plane in phase space. You may recall that this is precisely the approach we took in constructing phase space plots in our study of the Lorenz model in Chapter 3. In Fig. 9.16 we have have plotted the "velocity" of mode 1 (just da_1/dt) as a function of the amplitude of mode 1 (a_1), at those times when the trajectory crosses the plane $a_3 = 0$. When β is small the phase space plot is very regular. For $\beta = 0.3$ the phase plot approximates a hexagon, with only a few points lying off this curve. If one studies the trajectory carefully, one finds that these few points occurred at the very beginning of the simulation. Hence, over time the system settled down and the trajectory was "attracted" to the hexagon. This should remind you of the behavior we encountered in Chapter 3 in our studies of chaos; here the hexagon is acting very much like an attractor in phase space.

Somewhat similar behavior is found with $\beta = 1$. Recall that for this value of β we found very little sharing of energy between modes in Fig. 9.15. The corresponding phase space plot in Fig. 9.16 shows another regular trajectory, although is it more complex that the hexagon found when $\beta = 0.3$. However, when β is increased further things change dramatically. The phase plot with $\beta = 3$ has no trace of regularity, and confirms our suspicion that chaos is lurking very nearby.

We do not have space in this book to fully explore the FPU problem. Indeed, entire books have been written on the subject (see the end of chapter references). Even so, we can appreciate the central insight of the FPU problem -- that the interface between classical dynamics and statistical mechanics is much more complex than had been expected by Fermi, Pasta, and Ulam. In fact, this interface is *very* complex, and we will leave it to the references to take you deeper into this subject. Our results in this section do show a close connection to the behavior we encountered in our studies of chaos in Chapter 3. When the nonlinearity in the FPU problem is large (i.e., β is large) we have found chaotic behavior, while we know that the corresponding linear system is completely regular and non-chaotic. The surprise has come in the interface region of small nonlinearity. Here we found hints of both regular and chaotic behavior. The complications found in this interface region are, in a sense, associated with the route to chaos in this system.

EXERCISES

9.17. Repeat the FPU calculation in Fig. 9.14. Compute the total energy of the system, and confirm that it stays constant throughout the simulation. Thus in the FPU problem, chaotic-type states can be obtained even in the absence of external power supplied, in contrast to the case of the driven pendulum of Chapter 3.

9.18. Investigate an FPU system with alpha-type springs. You should find that the behavior is similar to that found with beta-type springs, with very little sharing of energy between modes when α is small. Here, "small" values are in the neighborhood of $\alpha = 0.25$ or less. Also, show that with alpha springs the energy can move between even and odd modes.

9.19. Carry out an FPU simulation similar to that in Fig. 9.14, but investigate the behavior when the system is initially placed in other modes, such as mode 2 and mode 3.

***9.20.** Investigate equipartition in an FPU system with large β, as in Fig. 9.15. Examine carefully how much energy actually flows to the higher modes, and try to develop a criteria for determining if the system is really ergodic.

9.21. Study other types of phase space plots for the FPU problem by constructing other projections of the trajectory in phase space. For example, you might make projections on the $a_1 = 0$ or $v_3 = 0$ planes.

***9.22.** Study the sensitivity to initial conditions in the FPU problem. Use this to determine if the system is really chaotic.

REFERENCES

[1] B. J. Alder, and T. E. Wainwright, "Studies in Molecular Dynamics I. General Method," J. Chem. Phys. **31**, 459 (1959). Pioneering application of molecular dynamics to a system of hard disks.

[2] D. Chandler, 1987, *Introduction to Modern Statistical Mechanics*, Oxford, New York, Section 7.4.

[3] E. Fermi, E., J. Pasta, and S. Ulam, 1955, "Studies of Nonlinear Problems I," Los Alamos Technical Report LA-1940. This is the paper that started the FPU problem. It never actually appeared in a scientific journal, as Fermi died before it could be fully written up and published. However, this report can be found in E. Fermi, 1965, *Collected Papers, Vol. II*, University of Chicago Press, Chicago.

[4] J. Ford, "The Fermi-Pasta-Ulam Problem: Paradox Turns Discovery," Phys. Reports **213**, 271 (1992). An advanced, but very readable account of the work that led to our current understanding of the FPU problem, and related issues in nonlinear dynamics.

[5] D. W. Heermann, 1990, *Computer Simulation Methods in Theoretical Physics, 2d ed.*, Springer-Verlag, New York. Contains a nice introduction to molecular dynamics and discusses how the method can be used to simulate a system that evolves according to the canonical ensemble.

[6] D. Marx and J. Hutter, 2000, "Ab initio molecular dynamics: Theory and Implementation," in *Modern Methods and Algorithms of Quantum Chemistry*, J. Grotendorst, ed., NIC Series, vol. 1, John von Neumann Institute for Computing.

[7] N. D. Mermin, "Crystalline Order in Two Dimensions," Phys. Rev. **176**, 250 (1968). Some intriguing exact results concerning the (non)stability of a hypothetical *infinitely* large solid.

[8] D. R. Nelson, 1983, "Defect-mediated Phase Transitions," in *Phase Transitions and Critical Phenomena, Vol. 7*, C. Domb and J. L. Lebowitz, eds., Academic Press, Orlando. Contains a detailed discussion of the theory of melting in two dimensions.

[9] R. K. Pathria, 1996, *Statistical Mechanics*, 2nd Ed., Butterworth Heineman, Oxford, Section 3.7.

[10] A. Rahman, "Correlations in the Motion of Atoms in Liquid Argon," Phys. Rev. **136**, A405 (1964). A very readable paper describing a molecular dynamics study of pair correlations in a Lennard-Jones liquid.

[11] F. Reif, 1965, *Fundamentals of Statistical and Thermal Physics*, McGraw-Hill, New York. Reviews statistical mechanics, including a discussion of the Maxwell-Boltzmann distributions.

[12] L. I. Schiff, 1968, *Quantum Mechanics, 3d ed.*, McGraw-Hill, New York. Contains a nice discussion of the origin of the Van der Waals interaction, which is a key ingredient of the Lennard-Jones potential.

[13] S. Toxvaerd, "Melting in a Two-Dimensional Lennard-Jones System," J. Chem. Phys. **69**, 4750 (1978). A careful study of melting in two dimensions using molecular dynamics.

[14] L. Verlet, "Computer 'Experiments' on Classical Fluids. I. Thermodynamical Properties of Lennard-Jones Molecules," Phys. Rev. **159**, 98 (1967). Description of the Verlet method and a nice description of some molecular dynamics simulations.

[15] T. P. Weissert, 1997, *The Genesis of Simulation in Dynamics: Pursuing the Fermi-Pasta-Ulam Problem,* Springer-Verlag, New York. A review of the Fermi-Pasta-Ulam problem and its role in the development the field of classical dynamics. Lots of interesting historical anecdotes.

CHAPTER 10

Quantum Mechanics

There are only a few problems in quantum mechanics that can be solved exactly, most notably the harmonic oscillator, a particle in a box, and the hydrogen atom. Nearly all other nontrivial quantum problems either have no known analytic solutions or can be attacked analytically only with extreme difficulty. This is why perturbation methods play such an important role in quantum theory and also why numerical methods are an attractive alternative. Indeed, an extremely wide variety of numerical methods have been developed for dealing with quantum problems. There are far too many such methods for us to even mention all of them in this chapter. Our goal instead will be to describe a few of the algorithms that have been developed with quantum mechanics in mind, and use them to treat several representative problems.[1] We also hope to find some enlightening overlap with problems and techniques we have encountered in earlier chapters.

This chapter opens with a brief review of the Schrödinger equation, and a few of the fundamental ingredients of quantum theory. We then consider several methods for treating time-independent problems. While we have dealt with partial differential equations in a number of previous settings, the problems associated with the Schrödinger equation involve extra complications associated with boundary conditions and require some extensions of the methods we have previously employed. We also describe how Monte Carlo methods can be teamed up with the variational principle to calculate wave functions and eigenvalues. In the second half of this chapter we turn to time-dependent problems and describe several numerical approaches. The goal, as usual, is to learn how to deal with problems that are difficult or impossible to treat analytically. However, we will find it useful to apply our numerical approaches to several exactly soluble problems, as this will allow us to test the methods and also illustrate some of the key themes of quantum theory.

10.1 TIME-INDEPENDENT SCHRÖDINGER EQUATION: SOME PRELIMINARIES

The time-independent Schrödinger equation for a particle in three dimensions is

$$-\frac{\hbar^2}{2m}\nabla^2\psi + V(\vec{r})\psi = E\psi\,, \tag{10.1}$$

where $\hbar$ is Planck's constant, m is the mass of the particle, V is the potential energy, E is the energy of the particle, and ψ is the wave function. As is well known with quantum theory, it is much easier to write this equation down than to really understand it. The key quantity here is the wave function, which has no direct classical counterpart. In the most general case, ψ is a complex function

[1] It is not our intention that this chapter be a substitute for a first course in quantum mechanics, although we will review a few aspects of the theory in the next several pages.