Molecular Dynamics: An Introduction

Lloyd Fosdick

January 19, 1995

High Performance Scientific Computing
University of Colorado at Boulder

The following are members of
the HPSC Group of the Department of Computer Science
at the University of Colorado at Boulder:

Lloyd D. Fosdick
Elizabeth R. Jessup
Carolyn J. C. Schauble
Gitta O. Domik

Contents

Trademark Notice

- Cray X-MP is a trademark of Cray Research, Inc.
- IBM is a trademark of IBM.
- MATLAB is a trademark of The MathWorks, Inc.

Molecular Dynamics:
A Introduction *

Lloyd D. Fosdick

January 19, 1995

1 Introduction

Molecular dynamics is concerned with simulating the motion of molecules to gain a deeper understanding of chemical reactions, fluid flow, phase transitions, droplet formation, and other physical phenomena that derive from molecular interactions. These studies include not only the motion of many molecules as in a fluid, but also the motion of a single large molecule consisting of hundreds or thousands of atoms, as in a protein. Much of this work uses simple classical Newtonian mechanics. This tutorial is concerned only with these classical systems and uses mathematical concepts that should be familiar to most upper division undergraduates in a physical science or engineering curriculum.

Computers are a critically important tool for these studies because there simply is no other way to trace the motion of a large number of interacting particles. The earliest of these computations were done in the 1950s by Berni Alder and Tom Wainwright at Lawrence Livermore National Laboratory. They studied the distribution of molecules in a liquid, using a model in which

*This work has been supported by the National Science Foundation under an Educational Infrastructure grant, CDA-9017953. It has been produced by the HPSC Group, Department of Computer Science, University of Colorado, Boulder, CO 80309. Please direct comments or queries to Elizabeth Jessup at this address or e-mail jessup@cs.colorado.edu.

the molecules are represented as *hard spheres* which interact like billiard balls [Alder & Wainright 59, Alder & Wainright 60]. Using the fastest computer at that time, an IBM 704, they were able to simulate the motions of 32 and 108 molecules in computations requiring 10 to 30 hours. Now it is possible to perform hard sphere computations on systems of over a million particles. A hard sphere model, with millions of molecules, has been used at NASA Ames Research Center by Leonardo Dagum to simulate hypersonic flow conditions encountered by flight vehicles at high altitudes [Dagum 88]. Computations using a more realistic molecular model known as *Lennard-Jones* have been performed at IBM Kingston by Lawrence Hannon, George Lie, and Enrico Climenti to study the flow of fluids [Hannon et al. 86]. In these computations the fluid was represented by $\sim 10^4$ interacting molecules. Even though this is miniscule compared with the number of molecules in a gram of water, the behavior of the flow was like that in a real fluid.

Another class of molecular dynamics computations is concerned with the internal motion of molecules especially proteins and nucleic acids, vital components of biological systems. The goal is to gain a better understanding of the function of these molecules in biochemical reactions. Interestingly, it has been found that quantum effects do not have much influence on the overall dynamics of proteins except at low temperatures. Thus classical mechanics is sufficient to model the motions, but still the computational power required for following the motion of a large molecule is enormous. For example, simulating the motion of a 1,500-atom molecule, a small protein, for a time interval of 10^{-10} seconds is a six hour computation on a Cray X-MP.

Martin Karplus and Andrew McCammon, in an interesting article "The Molecular Dynamics of Proteins" [Karplus & McCammon 86], describe a discovery concerning the molecule myoglobin that could only have been made through molecular dynamics. The interest in myoglobin comes about because it stores oxygen in biological systems. Whales, for example, are able to stay under water for long periods of time because of a large amount of myoglobin in their bodies. It was known that the oxygen molecule binds to a particular site in the myoglobin molecule but it was not understood how the binding could take place. X-ray crystallography work shows that large protein molecules tend to be folded up into compact three-dimensional structures, and in the case of myoglobin the oxygen sites were known to lie in the interior of such a structure. There did not seem to be any way that an oxygen atom could penetrate the structure to reach the binding site. Molecular dynamics provided the answer to this puzzle. The picture of a molecule provided by X-ray crystallography shows the average positions of the atoms in the molecule. In fact these atoms are in continuous motion, vibrating about positions of equilibrium. Molecular dynamics simulation of the internal motion of the molecule showed that a path to the site, wide enough for an oxygen atom, could open up for a short period of time.

Scientific visualization is particularly important for understanding the results of a molecular dynamics simulation. The millions of numbers representing a history of the positions and velocities of the particles is not a very revealing picture of the motion. How is one to recognize the formation of a vortex in this mass of data, or the nature of the bending and stretching of a large molecule? How is one to gain new insights? Pictures and animations enable the scientist to literally see the formation of vortices, to view protein bending, and thus to gain new insights into the details of these phenomena.

Computations that involve following the motion of a large number of interacting particles, whether the particles are atoms in a molecule, or molecules of a fluid or solid, or even particles in the discrete model of a vibrating string or membrane are similar in the following respect. They involve a long series of time steps, at each of which Newton's laws are used to determine the new positions and velocities from the old positions and velocities and the forces. The computations are quite simple but there are many of them. To achieve accuracy, the time steps must be quite small, and therefore many are required to simulate a significantly long *real* time interval. In computational chemistry the time step is typically about a femtosecond (10^{-15} seconds), and the total computation may represent $\sim 10^{-10}$ seconds which could cost about 100 hours of computer time. The amount of computation at each step can be quite extensive. In a system consisting of n particles the force computations at each step may involve $\mathcal{O}(n^2)$ operations. Thus it is easy to see that these computations can consume a large number of machine cycles. In addition, animation of the motion of the system can make large demands on memory.

The objective of this tutorial is to help you gain some understanding of the nature of these computations. We use the term *particle* to refer to the interacting objects: atoms or molecules. We concentrate on three different models: Hooke's law, Lennard-Jones, and hard sphere. In the Hooke's law model the force acts as if the particles were connected to their neighbors by springs. Lennard-Jones is a model with forces that are strongly repulsive at very short interparticle distances, attractive at larger distances, and extremely weak attractive at very large distances. In the hard sphere model, already mentioned, the particles interact as if they were billiard balls – they bounce off each other when they are a certain distance apart, otherwise they do not interact. Normally the Lennard-Jones model is used for 3-dimensional systems, but it is instructive to use it also for 1-dimensional and 2-dimensional systems.

After describing these models we discuss the equations of motion for each model, and then we consider numerical methods for solving these equations. Solving the equations of motion for the Hooke's Law model and the Lennard Jones model is intrinsically different from solving them for the hard sphere model. Solving the equations of motion for the hard sphere model requires solving some simple problems in geometry; in particular, we must determine when and where two spheres moving at constant velocity will collide. In one dimension this is extremely simple, but in two and three dimensions you need to draw on a knowledge of vector analysis. With Hooke's Law and Lennard Jones models we must solve a system of differential equations. For this we use two numerical methods: Euler's method and Verlet's method. We use Euler's method because it is the simplest of any method we could reasonably use and thus it provides the easiest introduction to some of the basic ideas of numerically solving the equations of motion. Verlet's method is only slightly more complex but more accurate. It is the simplest of the numerical methods used in serious molecular dynamics computations.

Finally, we consider the exact solution of the equations of motion for the Hooke's Law model. This is the only model of the three considered here that admits an exact solution. Exact solutions are important for us because they provide a means for testing the accuracy of our numerical methods. An understanding of this part of the tutorial requires some elementary knowledge of matrix eigenvalues and eigenvectors.

2 Models

Models of particle systems are characterized by the nature of the interactions between the particles. Generally it is assumed that the forces between the particles are conservative, two-body forces; that is, energy is conserved and the total force acting on a particle due to the other particles is the sum of the forces between pairs of particles. Thus the force acting on particle i is given by an expression of the form

$$f_i = \sum_{\substack{j=1 \\ j\neq i}}^{n} f_{i,j}, \tag{1}$$

where f_i is the total force on particle i due to the other particles, $f_{i,j}$ is the force on particle i due to particle j, and n is the number of particles in the system. Force is a vector quantity, so the sum in equation (1) is a vector sum. The order of the indices is important: the first index identifies the particle acted on, the second index identifies the particle causing the action. Newton's third law tells us that

$$f_{i,j} + f_{j,i} = 0. \tag{2}$$

There is an important relation between potential energy and force in a conservative system. If r is the position of a particle, $f(r)$ the force acting on it, and $\phi(r)$ its potential energy, then

$$f(r) = -\nabla\phi(r). \tag{3}$$

Thus we can describe a model in terms of the force or the potential energy. For example, if

$$\phi(r) = \|r\|^2, \quad \text{where } r = \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \tag{4}$$

then the three components of the force are

$$\begin{aligned} f_x &= -\frac{\partial\phi}{\partial x} = -2x, \\ f_y &= -\frac{\partial\phi}{\partial y} = -2y, \\ f_z &= -\frac{\partial\phi}{\partial z} = -2z. \end{aligned} \tag{5}$$

Since potential energy is a scalar quantity it is often more convenient to describe the model in terms of its potential energy function, ϕ.

At a point of minimum potential energy the partial derivatives of the potential energy are zero, and thus it is a point at which all of the forces are zero. Accordingly we call this point an *equilibrium point.*

We now consider three models, referred to as the *Hooke's law* model, *HL* for short, the *Lennard-Jones* model, *LJ* for short, and the *hard sphere* model, *HS* for short. Of these three, the LJ model comes closest to representing real molecular systems. On the other hand, the LJ model presents the most difficult computational challenge. The HL model is an approximation to the LJ model when the particles have low kinetic energy, thus remaining close to their equilibrium positions; similarly, the HS model is an approximation to the LJ model when the particles have high kinetic energy, or when attractive forces are very weak.

2.1 Hooke's Law model

In the HL model the potential energy of a particle is proportional to the square of its displacement from its equilibrium position. Figure 1 shows the potential energy function for a particle in a 1-dimensional system; and figure 2 shows the force on the particle that, according to equation (3), must be proportional to the displacement of the particle from its equilibrium position, and directed towards it. The equations for the potential energy, and force are:

$$\begin{aligned} \phi(x) &= \frac{k}{2}(x - x^{eq})^2 + \phi^{min}, \\ f(x) &= -k(x - x^{eq}), \end{aligned} \tag{6}$$

where k is a constant, sometimes referred to as the *force constant*; ϕ^{min} is a constant, the minimum potential energy; and x^{eq} is the equilibrium position of the particle. Notice that

$$f(x) = -\frac{d\phi(x)}{dx} \tag{7}$$

as required by equation (3). Thus the force is proportional to the displacement of the particle from its equilibrium position and it is directed towards the equilibrium position. The most familiar example of a system subject to a HL force is a small mass suspended by a spring: it moves up and down under the influence of a HL force imposed by the spring, figure 3.

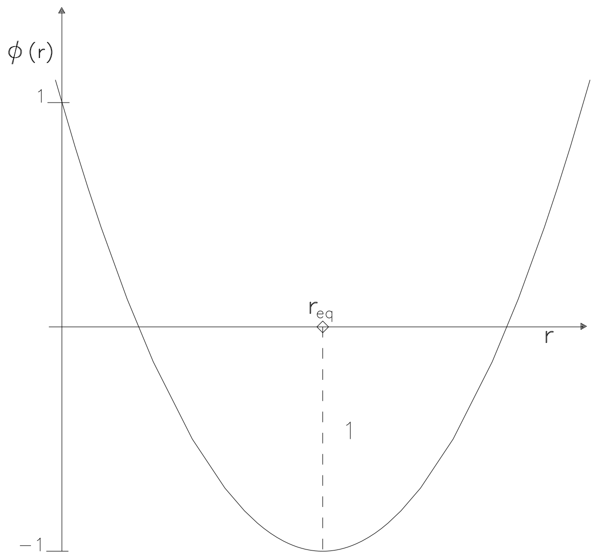

Figure 1: Potential energy of a particle in the HL model varies as the square of its displacement from equilibrium.

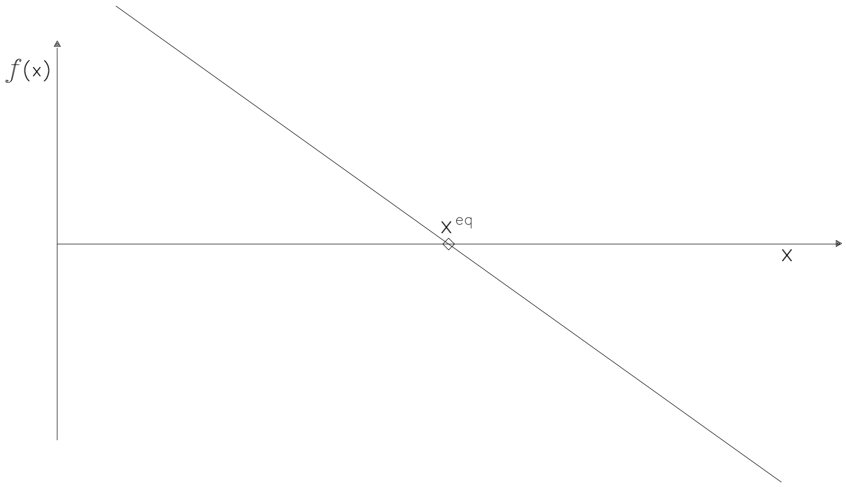

Figure 2: Force on a particle in the HL model is proportional to its displacement from equilibrium and in the direction of the equilibrium point.

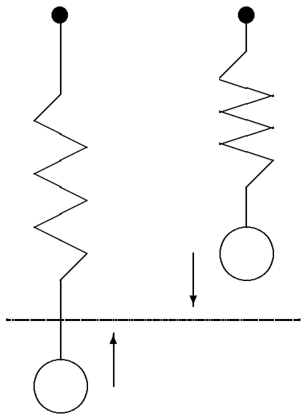

Figure 3: The motion of a small mass suspended from a spring typifies the motion of a particle subject to an HL force.

In two dimensions this model is described by

$$\phi(x,y) = \frac{k}{2}((x - x^{eq})^2 + (y - y^{eq})^2) + \phi^{min}, \tag{8}$$

which we may write more compactly as

$$\phi(r) = \frac{k}{2}\|r - r^{eq}\|^2 + \phi^{min}. \tag{9}$$

We now consider the more interesting case of HL models for systems of more than one particle, starting with the 2-particle case. It helps to think of a physical system of two masses connected by a spring as illustrated in figure 4. The force of the spring acts along a line joining the particles that we take to be the x axis. We assume that when the particles are separated by a distance d the spring is neither stretched or compressed, so the system is in equilibrium. When the distance between the particles is less than d the spring is compressed and the force acts to drive the particles apart; when the spring is stretched the force acts to bring the particles closer together. The potential energy function is

$$\phi(x_1, x_2) = \frac{k}{2}(x_1 - x_2 + d)^2 + \phi^{min}. \tag{10}$$

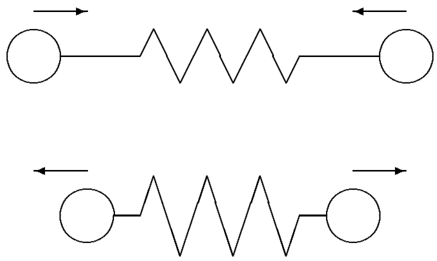

Figure 4: The motion of two small masses connected by a spring illustrates the nature of the motion of a 2-particle system in the HL model.

Figure 5: A 1-dimensional, 4-particle HL model. Motion is restricted to the x dimension and is oscillatory.

The forces can be obtained by taking the appropriate derivatives of the potential energy giving

$$\begin{aligned} f_1(x_1, x_2) &= -k(x_1 - x_2 + d), \\ f_2(x_1, x_2) &= k(x_1 - x_2 + d). \end{aligned} \tag{11}$$

A 1-dimensional, 4-particle system is shown in figure 5. An illustration of the motion of this system is shown in figure 6 The potential energy of this system is

$$\phi(x_1, x_2, x_3, x_4) = \frac{k}{2}((x_1 - x_2 + d)^2 + (x_2 - x_3 + d)^2 + (x_3 - x_4 + d)^2) + \phi^{min}. \tag{12}$$

This system is in equilibrium when the particles are ordered from left to right, each a distance d from its neighbors. The forces are

$$f_1(x_1, x_2) = -k(x_1 - x_2 + d), \tag{13}$$

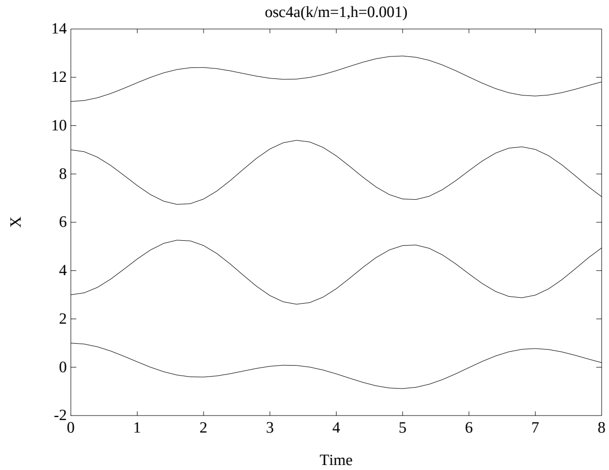

Figure 6: Illustration of the motion of a 4-particle system. The curves show the position of the 4 particles as functions of time. These results were for a system with $k/m = 1$, k is the force constant, and m is the mass of a particle. The equilibrium positions of the particles are 0, 4, 8, 12. The parameter h is the time step used in integrating the equations of motion.

$$\begin{aligned} f_2(x_1, x_2, x_3) &= -k(2x_2 - x_1 - x_3), \\ f_3(x_2, x_3, x_4) &= -k(2x_3 - x_2 - x_4), \\ f_4(x_3, x_4) &= -k(x_4 - x_3 - d). \end{aligned} \tag{14}$$

Extension of these equations to n-particle systems should be obvious.

The equations are simplified if we fix the equilibrium position of the first particle to be at the origin and we define new variables q_i as follows:

$$x_i = (i-1)d + q_i. \tag{15}$$

Thus q_i denotes the displacement from the equilibrium position of the i^{th} particle. In the new variables the potential energy is

$$\phi(q_1, q_2, q_3, q_4) = \frac{k}{2}((q_1 - q_2)^2 + (q_2 - q_3)^2 + (q_3 - q_4)^2) + \phi^{min}, \tag{16}$$

and the forces are

$$f_1(q_1, q_2) = -k(q_1 - q_2), \tag{17}$$

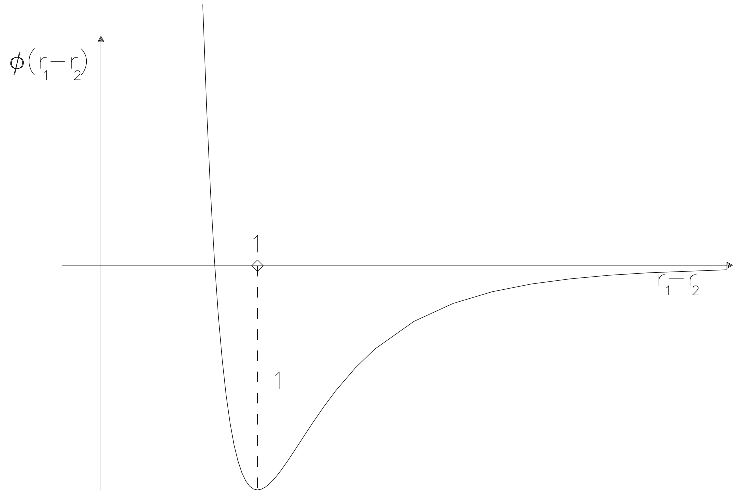

Figure 7: Potential energy of a particle in the LJ model.

$$\begin{aligned} f_2(q_1, q_2, q_3) &= -k(2q_2 - q_1 - q_3), \\ f_3(q_2, q_3, q_4) &= -k(2q_3 - q_2 - q_4), \\ f_4(q_3, q_4) &= -k(q_4 - q_3). \end{aligned}$$

2.2 Lennard-Jones model

We consider a three-dimensional system. The potential energy function for a pair of particles, 1 and 2, in the LJ model is given by

$$\phi(r_1, r_2) = \left(\frac{1}{\|r_1 - r_2\|^{12}} - \frac{2}{\|r_1 - r_2\|^6} \right). \tag{18}$$

The units have been chosen to locate the minimum of the potential energy at $\|r_1 - r_2\| = 1$, and the value of the minimum equal to -1. This potential function is illustrated in figure 7. Considering the slope of this function we see that the force is strongly repulsive at small distances, and is attractive

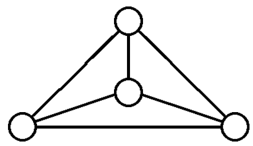

Figure 8: Minimum energy configuration for 4 particles in a 3-dimensional LJ model: particles are located at the corners of a regular tetrahedron.

at large distances, becoming extremely weak at very large distances. The crossover between the repulsive region and the attractive region occurs where $\|r_1 - r_2\| = 1$, the point of minimum potential energy. Note that at this point the force of the interaction is zero since the derivative of ϕ is zero.

The forces on particle 1 can be determined from the basic formula, equation (3):

$$\begin{aligned} f_{1,x} &= 12\left(\frac{1}{\|r_1 - r_2\|^{14}} - \frac{1}{\|r_1 - r_2\|^{8}}\right)(x_1 - x_2), \\ f_{1,y} &= 12\left(\frac{1}{\|r_1 - r_2\|^{14}} - \frac{1}{\|r_1 - r_2\|^{8}}\right)(y_1 - y_2), \\ f_{1,z} &= 12\left(\frac{1}{\|r_1 - r_2\|^{14}} - \frac{1}{\|r_1 - r_2\|^{8}}\right)(z_1 - z_2). \end{aligned} \tag{19}$$

In a many-particle LJ system the force on each particle is determined by summing over the pairwise interactions, using the above formulas. Since the force between widely separated pairs is very weak it is sometimes neglected: a cutoff distance is chosen, and the force between particles separated by more than the cutoff is ignored.

The equilibrium configuration is not easily determined. If there are only four particles, then the particles are at the corners of a regular tetrahedron, as shown in figure 8. But what about larger systems?

Particle coordinates for equilibrium configurations of 5 and 6 particles are shown in table 1[1]. Pictures of these configurations are shown in figures 9 and 10.

[1]These results were obtained by Elizabeth Eskow using minimization software developed by Robert Schnabel.

5 Particles		
x	y	z
0.751308	0.888623	1.293299
0.906831	1.164906	0.343358
1.619112	0.680194	0.846874
0.418779	1.771369	0.967718
1.398694	1.623815	1.085234

6 Particles		
x	y	z
5.951021	6.316232	5.356796
6.260952	5.532089	4.827508
6.271016	5.496511	5.822353
5.294356	5.769662	4.845783
5.614350	4.949941	5.311340
5.304420	5.734084	5.840627

Table 1: x-, y-, z-coordinates of particles in a 5-particle LJ model, and a 6-particle LJ model.

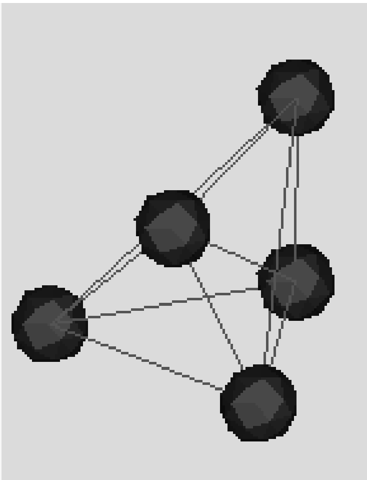

Figure 9: Minimum energy configuration for 5 particles.

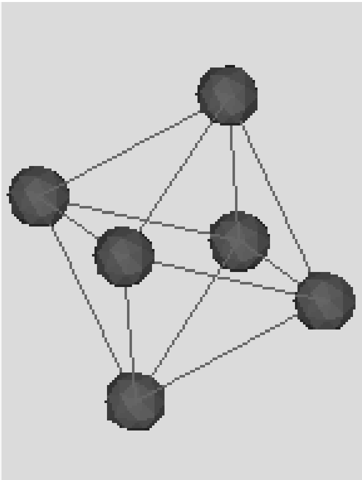

Figure 10: Minimum energy configuration for 6 particles.

2.3 Hard sphere model

This model is best visualized as a collection of hard, perfectly elastic, balls — like ball bearings, or billiard balls. The interactions between particles are like collisions between these balls. Two-dimensional and one-dimensional versions of this model, as well as the three-dimensional model, are studied: in two dimensions it is called the *hard disk model*, and in one dimension it is called the *hard rod model*.

The potential energy for a pair of particles is

$$\phi(r_1, r_2) = \begin{cases} 0, & \|r_1 - r_2\| > \sigma \\ \infty, & \|r_1 - r_2\| \leq \sigma \end{cases} \tag{20}$$

Thus there is no force acting on the particles except at the instant when they are a distance σ apart. At that point an instantaneous force is applied, causing a change in velocities. We discuss this further in the next section when we consider the equations of motion. For the present it is sufficient to think of the collision as if it were between two billiard balls of radius σ.

The HS model can be viewed as an approximation to the LJ model with high-velocity particles. When particles in the LJ model are moving at high velocities the effect of the attractive force is quite small. The particles move at high speed in straight lines (approximately) until they get close enough

Atom	Length (cm.)	Time (sec.)
He	2.87×10^{-8}	3.38×10^{-14}
Ne	3.10×10^{-8}	6.80×10^{-14}
Ar	3.83×10^{-8}	1.54×10^{-13}
Kr	4.11×10^{-8}	1.95×10^{-13}
Xe	4.43×10^{-8}	2.42×10^{-13}

Table 2: Length and time scale factors for the LJ model.

for the repulsive force to come into play, at which time they collide as in the hard sphere model. The repulsive force rises so steeply in the LJ model it has almost the same effect as a collision between hard spheres of diameter slightly less than 1.

2.4 Units and the connection with real systems

The models all have potentials that depend only on distance; i.e. they are spherically symmetric. Therefore they serve best as models for systems composed of atoms of helium (He), neon (Ne), argon (Ar), Krypton (Kr), or Xenon (Xe). Our computations with the LJ model apply to any of these systems by appropriate choice of a scale factor for length and time. In table 2 we show the scale factors for length and time for these elements. The interpretation of the numbers in this table can be illustrated for the case of argon: the *length* entry means that a distance of 1 unit, $\|r\| = 1$, in the formula for the LJ potential equation (18) represents 3.83×10^{-8} cm.; similarly, the *time* entry means that a time unit of 1 in the equations of motion (next section) represents 1.54×10^{-13} seconds, assuming the mass we use in the equations of motion is 1.

3 Equations of motion

Newton's second law gives us the equation for the motion of a particle:

$$ma = f, \tag{21}$$

where m is the particle's mass, a is its acceleration, and f is the force acting on it. From this equation and a knowledge of the initial position and the initial velocity of the particle we can, in principle, determine its position and velocity at future times. In a system of interacting particles their motion is determined by solving many of these equations, one for each particle. The equations are interdependent because the force on a particle is a function of the position of some or all of the other particles. The solution of these equations is our major concern in the next section. In this section we look at the form of these equations for the different models in order to gain an understanding of the nature of the problems we are trying to solve.

3.1 One-dimensional systems.

The equations of motion for a 1-dimensional system of two interacting particles are:

$$\begin{aligned} m_1\ddot{x}_1 &= f_1, \\ m_2\ddot{x}_2 &= f_2, \end{aligned} \tag{22}$$

where the acceleration is represented by $\ddot{x}$; that is,

$$\ddot{x}_i = \frac{d^2x_i}{dt^2}. \tag{23}$$

We assume that all particles have the same mass: $m = m_1 = m_2$.

If we write the equations of motion in matrix form we have

$$m\begin{bmatrix} \ddot{x}_1 \\ \ddot{x}_2 \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}. \tag{24}$$

These equations can be written more compactly as

$$m\ddot{x} = f, \tag{25}$$

with the understanding that $\ddot{x}$ and f are the vectors in equation (24). We could express the equations of motion for a system of n particles by exactly the same simple equation, with $\ddot{x}$ denoting a vector of n accelerations, and

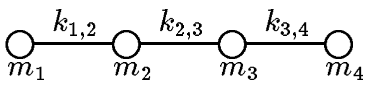

Figure 11: A 4-particle HL model with unequal masses and unequal force constants.

$f(x)$ denoting a vector of n forces. For example, the explicit matrix equation of a four-particle system, figure 5, is

$$m \begin{bmatrix} \ddot{x}_1 \\ \ddot{x}_2 \\ \ddot{x}_3 \\ \ddot{x}_4 \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix} \tag{26}$$

The equations of motion for four particles in the HL model are

$$\begin{aligned} m\ddot{q}_1 &= -k(q_1 - q_2), \\ m\ddot{q}_2 &= -k(2q_2 - q_1 - q_3), \\ m\ddot{q}_3 &= -k(2q_3 - q_2 - q_4), \\ m\ddot{q}_4 &= -k(q_4 - q_3), \end{aligned} \tag{27}$$

where the q_is were defined in the last section, equation (15).

We assumed that the particles are identical so that a common force constant, and common masses are used throughout. You might check your understanding of these equations by deriving the equations of motion for a non-homogeneous HL model consisting of four particles with unequal masses, and unequal force constants, as illustrated in figure 11.

The equations of motion for two particles with LJ forces, acting in just one dimension, are

$$\begin{aligned} m\ddot{x}_1 &= 12\left(\frac{1}{(x_1 - x_2)^{13}} - \frac{1}{(x_1 - x_2)^7}\right), \\ m\ddot{x}_2 &= 12\left(\frac{1}{(x_2 - x_1)^{13}} - \frac{1}{(x_2 - x_1)^7}\right). \end{aligned} \tag{28}$$

In a 4-particle system they are

$$\begin{aligned}
m\ddot{x}_1 &= 12\sum_{j=2}^{4}\left(\frac{1}{(x_1-x_j)^{13}}-\frac{1}{(x_1-x_j)^{7}}\right), \qquad (29)\\
m\ddot{x}_2 &= 12\sum_{\substack{j=1\\ j\neq 2}}^{4}\left(\frac{1}{(x_2-x_j)^{13}}-\frac{1}{(x_2-x_j)^{7}}\right),\\
m\ddot{x}_3 &= 12\sum_{\substack{j=1\\ j\neq 3}}^{4}\left(\frac{1}{(x_3-x_j)^{13}}-\frac{1}{(x_3-x_j)^{7}}\right),\\
m\ddot{x}_4 &= 12\sum_{\substack{j=1\\ j\neq 4}}^{4}\left(\frac{1}{(x_4-x_j)^{13}}-\frac{1}{(x_4-x_j)^{7}}\right),
\end{aligned}$$

In the 2-particle example the particles are in equilibrium when they are unit distance apart, but the 4-particle case is a little different. When the particles are unit distance apart, then the force on each is *almost*, but not exactly, zero: although the force from particles unit distance away is zero, particles 2 and 3 units away exert a small attractive force. In fact the forces for unit separation are:

$$\begin{aligned}
f_1 &= +0.0978,\\
f_2 &= +0.0923,\\
f_3 &= -0.0923,\\
f_4 &= -0.0978.
\end{aligned}$$

Therefore if we placed the particles at locations 0, 1, 2, 3 along the x axis we would expect the first two to start moving to the right, and the second two to start moving to the left.

The equations of motion for the HS model must be expressed a little differently. Consider two particles moving along the x axis. Their positions are given by

$$x_1 = x_1^{(0)} + \dot{x}_1^{(0)}(t - t^{(0)}), \qquad (30)$$

$$x_2 = x_2^{(0)} + \dot{x}_2^{(0)}(t - t^{(0)}), \qquad (31)$$

where $x_i^{(0)}$ denotes position at time $t^{(0)}$, and the velocities, $\dot{x}_i$, are the velocities at $t^{(0)}$. If the particles are moving towards each other, then they will collide at some time, say $t^{(1)}$. At this instant they change their velocities and the new positions are given by

$$\begin{aligned} x_1 &= x_1^{(1)} + \dot{x_1}^{(1)}(t - t^{(1)}), && (32) \\ x_2 &= x_2^{(1)} + \dot{x_2}^{(1)}(t - t^{(1)}). && (33) \end{aligned}$$

Of course if the particles are moving away from each other, then there is no collision. Thus solving the equations of motion in this case amounts to determining the time of the next collision; moving the particles to their positions at that time; and then determining the new velocities of the colliding particles. This process is repeated over and over. Thus we compute the motion from collision to collision.

A 1-dimensional HS model like this is not very interesting because the particles gradually move farther and farther apart, going off to $+\infty$ or $-\infty$. But we can make it interesting if we put "walls" on left and right constraining the particles to remain in some interval. When a particle hits the wall then we can assume it bounces back, i.e. reverses its velocity. Or we can assume that the particles are confined to a circle, as if we joined left and right ends of an interval of the x axis — this kind of assumption is referred to as a "periodic boundary condition."

3.2 Two-dimensional systems

We now assume

$$r = \begin{bmatrix} x \\ y \end{bmatrix}. \tag{34}$$

The position of the i^{th} particle is denoted r_i. With this understanding we can write the equations of motion for an n-particle system exactly as before, equation (25), but with r in place of x:

$$m\ddot{r} = f. \tag{35}$$

Consider a 2-particle system with

$$r = \begin{bmatrix} r_1 \\ r_2 \end{bmatrix}, \tag{36}$$

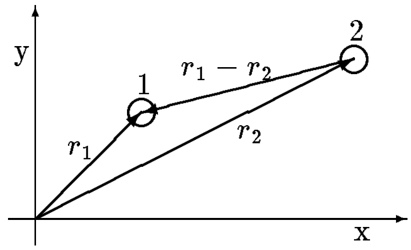

Figure 12: A 2-particle, 2-dimensional system.

$$= \begin{bmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \end{bmatrix}. \tag{37}$$

These coordinates are illustrated in figure 12. Thus we can write the equations of motion as follows:

$$m \begin{bmatrix} \ddot{x}_1 \\ \ddot{y}_1 \\ \ddot{x}_2 \\ \ddot{y}_2 \end{bmatrix} = \begin{bmatrix} f_{1,x} \\ f_{1,y} \\ f_{2,x} \\ f_{2,y} \end{bmatrix} \tag{38}$$

where $f_{i,x}$ and $f_{i,y}$ are the x- and y-components of the force on particle i.

The forces are directed along the line through the centers of the two particles, as illustrated in figure 13.

Now we can write the equations of motion for 2-particle HL and LJ models using equation (38) and the forces given in the last section. The equations of motion for a 2-particle HL model are

$$m \begin{bmatrix} \ddot{x}_1 \\ \ddot{y}_1 \\ \ddot{x}_2 \\ \ddot{y}_2 \end{bmatrix} = k(1 - \frac{d}{\|r\|}) \begin{bmatrix} x_1 - x_2 \\ y_1 - y_2 \\ x_2 - x_1 \\ y_2 - y_1 \end{bmatrix}; \tag{39}$$

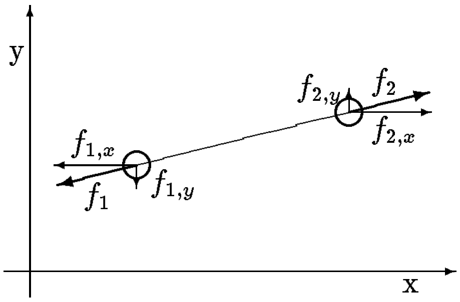

Figure 13: Force vectors in a 2-particle, 2-dimensional system.

and for a 2-particle LJ model they are

$$m \begin{bmatrix} \ddot{x}_1 \\ \ddot{y}_1 \\ \ddot{x}_2 \\ \ddot{y}_2 \end{bmatrix} = 12 \left(\frac{1}{\|r_1 - r_2\|^{14}} - \frac{1}{\|r_1 - r_2\|^{8}} \right) \begin{bmatrix} x_1 - x_2 \\ y_1 - y_2 \\ x_2 - x_1 \\ y_2 - y_1 \end{bmatrix}; \tag{40}$$

The equations for an n-particle system have the same form, the only difference being that when $n > 2$, a pairwise sum over interactions must be made to determine $f_{i,x}$,$f_{i,y}$; e.g., for the LJ model

$$f_{i,x} = 12 \sum_{\substack{j=1 \\ j \neq i}}^{n} \left(\frac{1}{\|r_i - r_j\|^{14}} - \frac{1}{\|r_i - r_j\|^{8}} \right) (x_i - x_j), \tag{41}$$

$$f_{i,y} = 12 \sum_{\substack{j=1 \\ j \neq i}}^{n} \left(\frac{1}{\|r_i - r_j\|^{14}} - \frac{1}{\|r_i - r_j\|^{8}} \right) (y_i - y_j).$$

Thus in an n-particle system the equations of motion are:

$$m \begin{bmatrix} \ddot{r}_1 \\ \ddot{r}_2 \\ \vdots \\ \ddot{r}_n \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{bmatrix}, \tag{42}$$

where $\ddot{r}_i$ and f_i are two-element column vectors:

$$\ddot{r}_i = \begin{bmatrix} \ddot{x}_i \\ \ddot{y}_i \end{bmatrix}, \quad f_i = \begin{bmatrix} f_{i,x} \\ f_{i,y} \end{bmatrix} \tag{43}$$

The equations of motion for a 2-dimensional HS system are the obvious extension of the 1-dimensional equations. We can write the equation for the i^{th} particle in vector form as follows;

$$r_i = r_i^{(1)} + \dot{r_i}^{(1)}(t - t^{(1)}) \tag{44}$$

The work of the computation is determining when the next collision will occur. Here there is an essential difference from the 1-dimensional computation. In 1-dimension only the particles on the left and right of a given particle are collision candidates; furthermore, these particles remain candidates for the entire calculation — in one dimension neighbors remain neighbors. Not so in two dimensions. Now the number of possibilities we must examine is much larger. Naively, we might consider every pair of particles but we can do better than this. One can, for example, divide space into bins of a certain size. If the bin size is chosen appropriately then for a given particle its collision candidates are the other particles in the same bin or in neighboring bins.

Another scheme uses a timetable of predicted collision times. Suppose that at some point we determine for every particle the time and partner for its next collision, *assuming no other collisions take place.* Thus we produce a timetable for collisions. The entry with the earliest collision time in the timetable is the next collision. Once we process that collision, we need to update the timetable. This updating process takes some work, but it may lead to less work overall than the naive approach. The updating process involves looking for a new collision partner for each of the two particles that just collided. When those collision partners have been found some other entries in the timetable may need to be updated, any that had one member of the colliding pair as a collision partner must be updated.

There is no need to consider 3-dimensional systems separately. The formulas and issues are the same as for the 2-dimensional systems just discussed. The only difference being that a z component must be added to the vectors.

4 Numerical solution of the equations of motion.

In general the equations of motion do not admit an analytic solution so they must be solved numerically. The numerical method is normally a *time* *stepping* algorithm; that is, the solution is generated incrementally in time starting from a set of initial conditions. This solution is simply a list of numbers, usually the particle positions and velocities at the time steps. The positions and velocities at any one time represent the *state* of the system at that time.

The process of generating the numerical solution is easy to describe in broad outline, though the details can be rather difficult. The broad outline is this. Starting from an initial state at a given time, the state at a slightly later time is computed, then the next state, and so on. The time interval between states is either constant or variable; in order to control the error, it must be kept small enough so that values of important variables undergo little change, but if the interval is too small then the computation becomes too slow. More complex algorithms usually accumulate less error per step and so bigger steps are possible, but they also take more time per step. Obviously these tradeoffs are an important consideration in designing a program for solving a molecular dynamics problem. Normally the state is not recorded at every time step, they are too close together. Instead, a different time interval that is a multiple of the time step is used for recording results. Important physical parameters such as energy, momentum, mean separation of particles, and so forth may be recorded during the computation or generated later from the state information.

Accuracy of the numerical solution is an important consideration. As a practical matter this usually must be estimated by indirect methods. Running the computation for different values of the time step is one indirect technique, and running with different numerical precision is another. Changing from single to double precision is relatively easy, and there are software tools for still higher precision [Bailey 91]. Special cases in which an exact solution is possible to test an algorithm are also used. The HL model is one of these special cases. It is possible to express the solution of the equations of motion for this system in terms of the eigenvalues and eigenvectors of a certain matrix. Although the eigenvalues and eigenvectors may have to be computed numerically, the error from computation is negligible compared with the error from a time stepping algorithm. Therefore the HL model admits, for all practical purposes, an exact solution against which numerical solutions can be compared.

The efficiency of these computations on vector and parallel computers depends on the models and the algorithms. The HL model, for example, can be run very efficiently on a vector computer but the HS model cannot. In most particle systems the neighbors of a particle change with time, thus the set of interacting pairs change and because of this an efficient decomposition of the computation for a parallel computer can be difficult to find.

For HL and LJ systems we use algorithms for solving second order differential equations. We consider two of these algorithms: Euler's method and Verlet's method. The HS system is essentially different since no differential equation needs to be solved. The work of the computation consists mainly in determining when the next collision occurs. For this we use only the naive algorithm mentioned: consideration of the more efficient algorithms would carry us beyond the scope of this tutorial.

4.1 Euler's method

A discussion of the basic ideas of Euler's method can be found in the tutorial [Fosdick 91] and any elementary numerical analysis book, for example [Conte & de Boor 80, Burden & Faires 85]. We describe it briefly in terms of the equations we are interested in solving. We begin with

$$m\ddot{x} = f(x), \tag{45}$$

with initial conditions

$$x(0) = x^{(0)}, \quad \dot{x}(0) = \dot{x}^{(0)}. \tag{46}$$

The first step is to write the second order differential equation, equation (45), as a pair of coupled first order differential equations:

$$\begin{aligned} m\dot{u} &= f(x), \\ \dot{x} &= u, \end{aligned} \tag{47}$$

where u is just a new name for $\dot{x}$. The next step is to use Euler's method on these equations to get the following formulas for generating the solution at times h, $2h$, $3h$, $\ldots$:

$$x(t+h) \approx x(t) + hu(t), \tag{48}$$

$$u(t+h) \approx u(t) + \frac{h}{m} f(x(t)). \tag{49}$$

The code segment for this computation is:

```
DO T = 1,NSTEP
    X(T+1) = X(T) + H*U(T)
    U(T+1) = U(T) + (H/M)*F(X(T))
END DO
```

The extension of these formulas to multiparticle, multidimensional problems is straightforward. The code segment for a 3-dimensional computation on an n-particle system, assuming all particles have the same mass, is:

```
DO T = 1,NSTEP
  CALL FORCE(X,Y,Z,FX,FY,FZ)
  DO I = 1,NPART
    X(I) = X(I) + H*U(I)
    Y(I) = Y(I) + H*V(I)
    Z(I) = Z(I) + H*W(I)
    U(I) = U(I) + (H/M)*FX(I)
    V(I) = V(I) + (H/M)*FY(I)
    W(I) = W(I) + (H/M)*FZ(I)
  END DO
  .
  .
  (Write positions and velocities)
  .
  .
END DO
```

The procedure `FORCE` evaluates the the forces on the particles from their current positions that are stored in the arrays `X, Y, Z` and returns the x-, y-, and z-components of the forces in the arrays `FX, FY, FZ`. The remark after the inner loop indicates a block of code that would write position and velocity information. Generally this information is not written at every time step because `H` is so small; rather it is written at a larger interval that is an

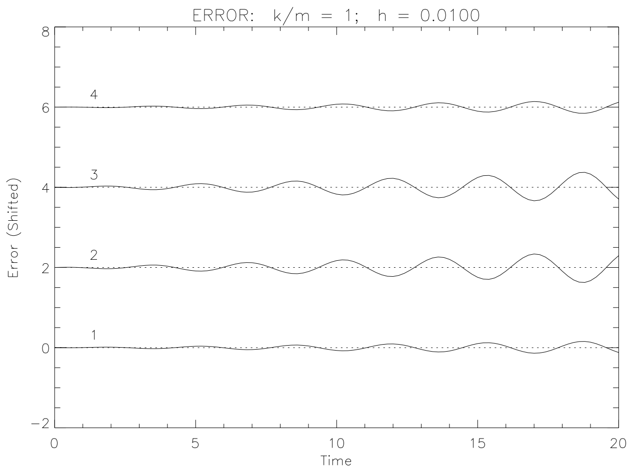

Figure 14: Error in numerical solution by Euler's method of 1-dimensional, 4-particle HL model: $h = 0.01$, $k/m = 1$. The curves have been shifted vertically: the point of zero error for a particular curve is its position at t=0.

integer multiple of `H`. Therefore, this block of code would include a test to determine if writing should take place at the current time.

If $f(x)$ is *well-behaved* it can be shown that the error in the computed solution is $\mathcal{O}(h)$. As an illustration of the error in solving the equations of motion with Euler's method we show the error as a function of time for a 1-dimensional, 4-particle HL system in figure 14. This result was obtained with a stepsize of $h = 0.01$. If we reduce the stepsize by a factor of ten, that is $h = 0.001$, then we get the results shown in figure 15. Comparison of the error for the two different values of h shows that the peaks in the error have been reduced by about a factor of 10 as we would expect because h is reduced by reduced by this factor and the error should be $\mathcal{O}(h)$.

While Euler's method has the virtue of simplicity it is far less accurate than other methods we might use. Verlet's method is still a relatively simple method but gives much better accuracy.

4.2 Verlet's method

The name we give to this method is commonly used in the molecular dynamics literature, but it is known in mathematics as *Störmer's method*. Actually

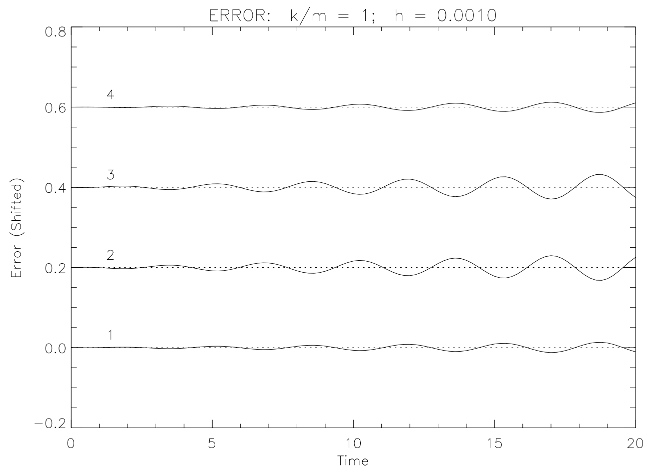

Figure 15: Error in numerical solution by Euler's method of 1-dimensional, 4-particle HL model: $h = 0.001$, $k/m = 1$. The curves have been shifted vertically: the point of zero error for a particular curve is its position at t=0.

there is a class of Störmer methods, of which this is the simplest. Henrici shows that the error in this method is $\mathcal{O}(h^2)$ [Henrici 62].

The basic idea is to approximate the second derivative with a finite difference,

$$\ddot{x} \approx \frac{x(t+h) - 2x(t) + x(t-h)}{h^2}. \tag{50}$$

The error in this approximation of the second derivative is $\mathcal{O}(h^2)$, as shown in [Fosdick 91]. If we use this approximation in equation (45) we obtain

$$x(t+h) \approx 2x(t) - x(t-h) + h^2 f(x(t)). \tag{51}$$

This, then, is the basis of the algorithm. We can generate the solution with the following code segment:

```
DO T = 1,NSTEP
  X(T+1) = 2*X(T) - X(T-1) + H**2*F(X(T))
END DO
```

There are important differences between this code segment and the corresponding code for Euler's method. Notice that two previous values of X are required at each time step, unlike Euler's method which required just one. Notice also that the force term has a factor H**2, not H as in Euler's method. And, finally, notice that the velocity does not appear.

Something special must be done to start the iteration because the initial values for the problem are usually position and velocity, not two position values. A Taylor series expansion can be used to compute $x(h)$ given $x(0)$ and $\dot{x}(0)$:

$$x(h) \approx x(0) + h\dot{x}(0) + \frac{h^2}{2} f(x(0)). \tag{52}$$

The fact that the force term is $\mathcal{O}(h^2)$ implies that we are adding a very small number to a much larger number at every step, resulting in a loss in accuracy. This can be mitigated by using a different form of the algorithm, called the *summed form*:

```
DO T = 1,NSTEP
  DX(T) = DX(T-1) + H*F(X(T))
  X(T+1) = X(T) + H*DX(T)
END DO
```

It is easy to verify that this is mathematically equivalent to the original algorithm: DX(T) is simply the name of (X(T+1) - X(T))/H). In other words, if all computations were exact (no roundoff error) then this code would produce the same result as the original. But real computations are not exact and the summed form gives a more accurate result.

The velocity can be computed from the position using a central difference approximation:

$$\dot{x}(t) \approx \frac{x(t+h) - x(t-h)}{2h}. \tag{53}$$

Alternatively, it can be computed within the algorithm as follows:

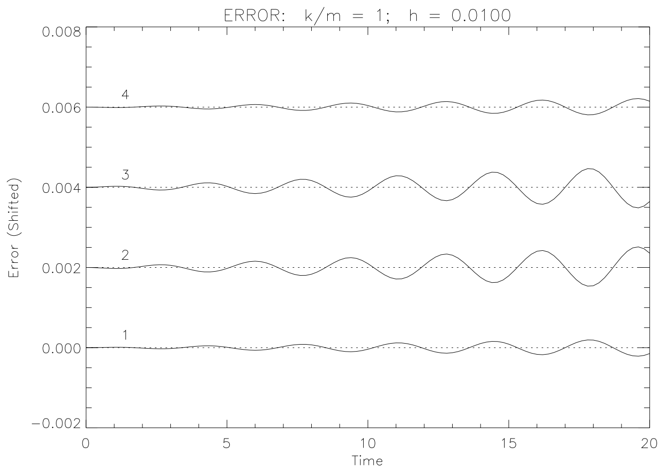

Figure 16: Error in numerical solution by Verlet's method of 1-dimensional, 4-particle HL model: $h = 0.01$, $k/m = 1$. The curves have been shifted vertically: the point of zero error for a particular curve is its position at t=0.

```
DO T = 1,NSTEP
  X(T+1) = X(T) + H*(U(T) + H*F(X(T))/2)
  U(T+1) = U(T) + H*(F(X(T+1))+F(X(T)))/2
END DO
```

An efficient implementation of this only requires one computation of the force at each time step, and saving it for use in the next time step. This form has the numerical accuracy of the summed form, but it requires more computations per time step. If velocities are not needed then the summed form should be used. If double precision arithmetic is used then the original unsummed form of the algorithm may give acceptable accuracy.

An idea of the difference in accuracy between Euler's method and Verlet's method is illustrated in figures 16 and 17, which correspond to figures 14 and 15: the same computation except Verlet's method was used. Comparison of the peaks in these error curves with the corresponding curves for Euler's method shows that Verlet's method is far more accurate than Euler's method: for $h = 0.01$ the error in Verlet's method is smaller than the error in Euler's

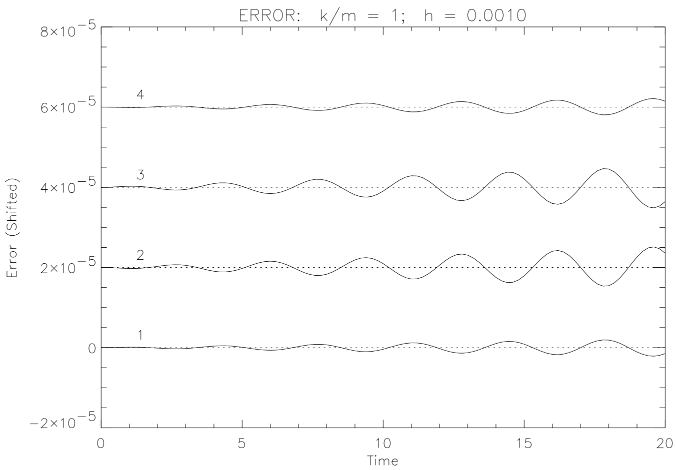

Figure 17: Error in numerical solution by Verlet's method of 1-dimensional, 4-particle HL model: $h = 0.001$, $k/m = 1$. The curves have been shifted vertically: the point of zero error for a particular curve is its position at t=0.

method by a factor of 10^{-3}. Comparison of the error in Verlet's method for $h = 0.01$ with that for $h = 0.001$ shows that the error is reduced by about a factor of 100, confirming the $\mathcal{O}(h^2)$ behavior of the error.

Extension of this algorithm to three dimensional, multiparticle systems should be evident from the discussion of Euler's method.

4.3 Hard sphere collisions

Determining the motion of a system of particles that are modeled as hard spheres does not require solving a differential equation, so the methods described above do not apply. Since hard sphere particles travel in straight lines with constant speed between collisions we only need to know when collisions occur and the velocities after collision in order to follow the motions of the particles. It sounds simple, but determining the sequence of collisions is a computationally intensive task. Bear in mind that we must determine which pair of particles will collide next after each collision, a task that must take into consideration all pairs of particles. We consider the easy case of a one dimensional system first.

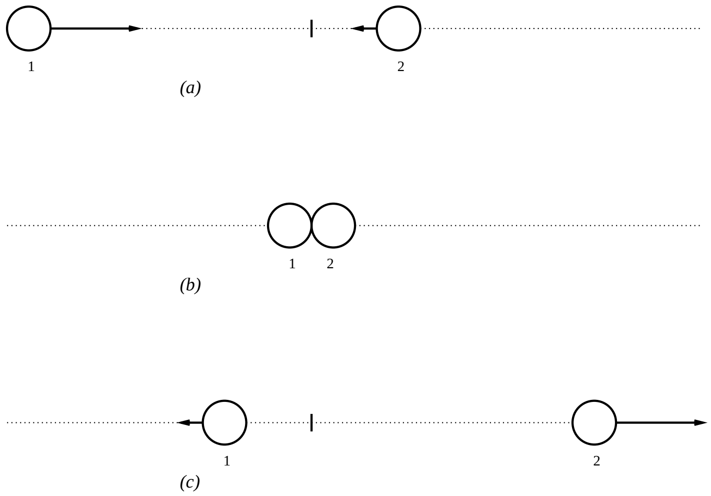

Figure 18: Collision of hard sphere particles in one dimension: (a) one time unit before collision; (b) instant of collision; (c) one time unit after collision. In this illustration particle 1 is travelling at four times the speed of particle 2 before the collision. At collision the particles exchange velocities so after the collision particle 2 is travelling at four times the speed of particle 1.

4.3.1 One-dimensional system

In one dimension the particles collide *head-on* as illustrated in figure 18. The conservation laws require that the total momentum and total energy do not change. Therefore we have:

$$\begin{aligned} u_{1,old} + u_{2,old} &= u_{1,new} + u_{2,new} \quad \textit{(conservation of momentum)}, \\ u_{1,old}^2 + u_{2,old}^2 &= u_{1,new}^2 + u_{2,new}^2 \quad \textit{(conservation of energy)}, \end{aligned} \tag{54}$$

where u denotes velocity, and it is assumed that both particles have the same mass. These equations can be solved easily for $u_{1,new}$ and $u_{2,new}$:

$$u_{1,new} = u_{2,old}, \quad u_{2,new} = u_{1,old} \tag{55}$$

Thus the particles simply exchange velocities when they collide.

Assume we have a system of hard-sphere particles ordered along the x-axis so that

$$x_1 < x_2 < \ldots < x_n. \tag{56}$$

Figure 19: A 6-particle, one-dimensional system of hard spheres.

as illustrated in figure 19 for $n = 6$. To determine the time of the next collision me must consider all pairs, executing a segment of code that looks like this:

```
COLLTIME = INFINITY
DO I = 1,N-1
  IF ((U(I+1) - U(I) .LT. 0)
    COLLTIMENEW = (X(I+1) - X(I))/(U(I) - U(I+1))
    IF(COLLTIMENEW .LT. COLLTIME)
      COLLPART = I
      COLLTIME = COLLTIMENEW
    END IF
  END IF
END DO
```

After execution of this segment we know that the time of the next collision is `COLLTIME` and that the collision partners are particles `I` and `I+1`, provided that `COLLTIME` $\neq$ `INFINITY`. The code segment we execute at a collision updates the positions of all particles, and the velocities of the colliding pair. It looks like this:

```
DO I = 1,N
  X(I) = X(I) + U(I)*COLLTIME
END DO
TMP = U(COLLPART)
U(COLLPART) = U(COLLPART + 1)
U(COLLPART + 1) = TMP
```

4.3.2 2,3-dimensional systems

In 2- and 3-dimensional systems collisions are not necessarily head on, they may be oblique, as illustrated in figure 20.

In an oblique collision the interaction or impact is along the line drawn between the centers of the particles at the instant of collision: there is no force exerted on the particles in the plane tangent to the two particles at the point of impact — our HS model assumes that the particles are perfectly smooth. An analysis like that used for the one dimensional case shows that in an oblique collision the particles exchange the components of their velocities *along the line between the centers of the particles*, no other velocity components are changed. Thus the effect of an oblique collision is as illustrated in figure 21.

The critical part of the computation for updating velocities is the determination of which particles will collide next. In its simplest form the steps in this computation are as follows. We let r_i and $\dot{r}_i$ denote the position vector and velocity vector of the i^{th} particle, and we let σ denote the diameter of a particle. It is convenient to define new position and velocity vectors

$$r_{i,j} = r_i - r_j, \quad \dot{r}_{i,j} = \dot{r}_i - \dot{r}_j \tag{57}$$

that represent the position and velocity of the i^{th} particle relative to the j^{th} particle. If we want to be explicit about the relative position at time t then we write $r_{i,j}(t)$. The test to determine if two particles collide can be broken into two parts: determine if they are approaching each other; if they are approaching each other then determine their distance of closest approach.

The particles are approaching each other if the component of their relative

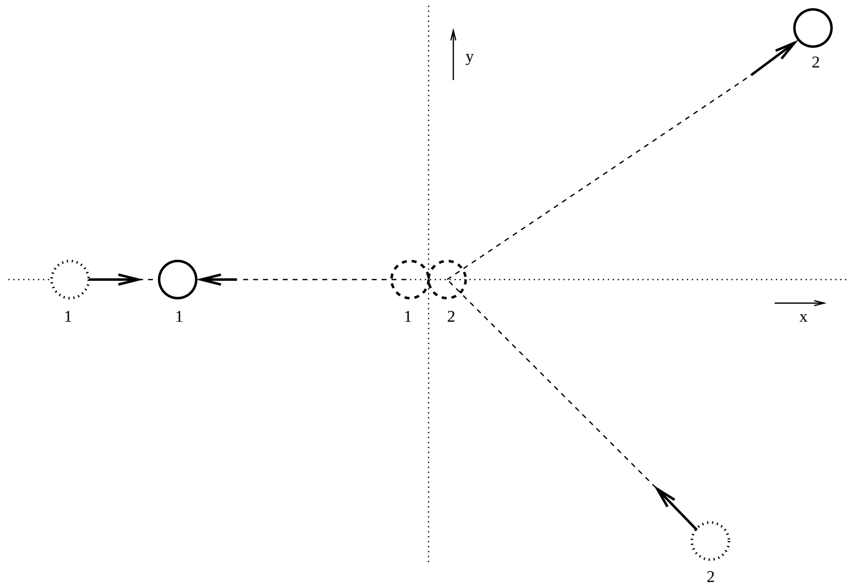

Figure 20: Collision of hard spheres in two dimensions: positions at one time unit before collision (dotted circles); at collision (dashed circle); and one time unit after collision (solid circles) are shown. Particles 1 and 2 are travelling at the same speed, s, before the collision. Particle 2 is travelling along a line that is 45° from the x-axis. At collision they exchange x-components of velocity; their y-components of velocity are unchanged. Therefore, after collision particle 1 is travelling in the x-direction at speed $s/\sqrt{2}$, while particle 2 is travelling upward and to the right, at an angle of $\arctan(1/\sqrt{2})$ to the x-axis, with a speed equal to $s\sqrt{3/2}$.

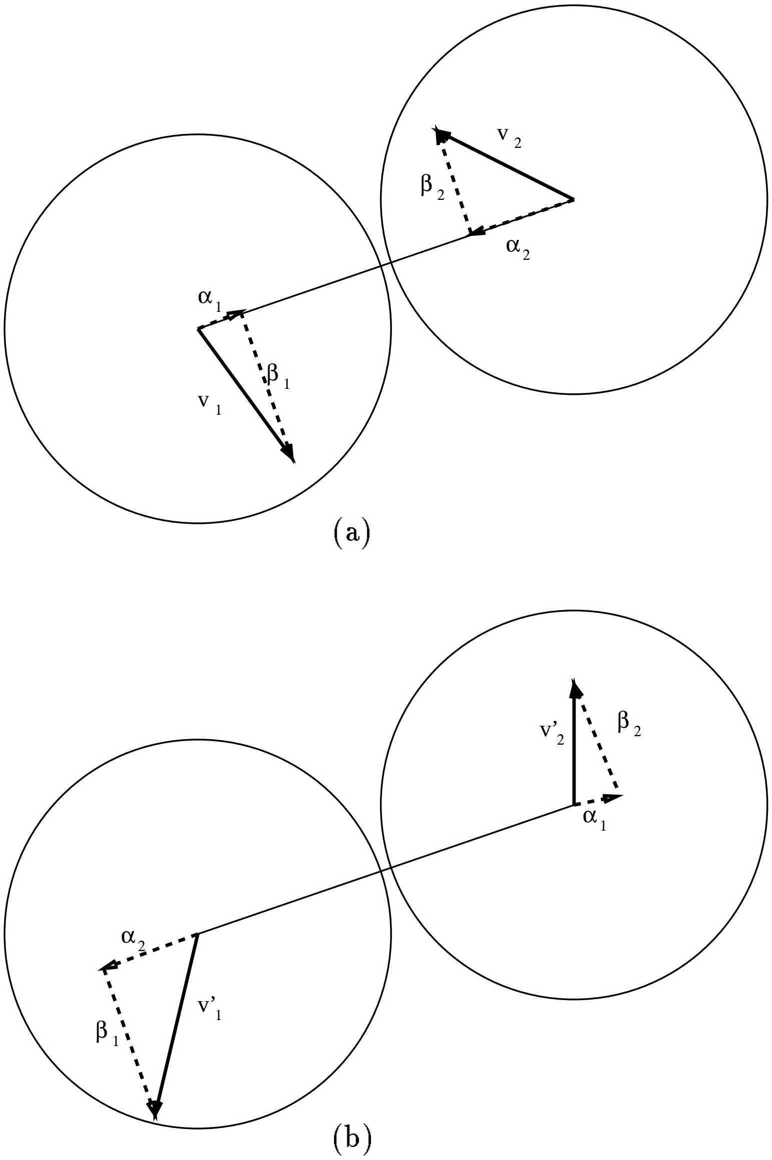

Figure 21: Detail of velocity components at instant before collision (a), and instant after collision (b). At collision the force is exerted only along the line joining the centers of the particles; no force is exerted in the plane perpendicular to this line because the particles are assumed to be *smooth*. Thus velocity components along this line are exchanged at the instant of collision, but velocity components in the plane perpendicular to this line are unchanged.

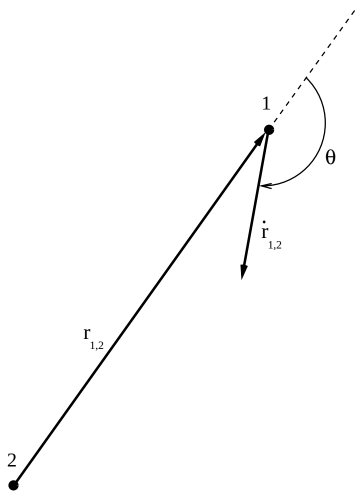

Figure 22: Illustration of necessary condition for a collision.

velocity in the direction of $r_{i,j}$ is negative; i.e.

$$r_{i,j} \cdot \dot{r}_{i,j} < 0, \tag{58}$$

where the product on the left is a scalar (dot) product: this notation is also used below to denote scalar product. The idea is illustrated in figure 22.

They actually collide if the following condition is true:

$$\|b_{i,j}\| \leq \sigma, \quad \|b_{i,j}\|^2 = \|r_{i,j}\|^2 - \left(\frac{r_{i,j} \cdot \dot{r}_{i,j}}{\|\dot{r}_{i,j}\|}\right)^2 \tag{59}$$

Some simple geometrical considerations, illustrated in figure 23 show that $b_{i,j}$ is the distance of closest approach if each particle had diameter zero.

If the particles do collide, then the time of the collision is given by the formula

$$t = -\frac{1}{\|\dot{r}_i - \dot{r}_j\|}\left(\frac{(r_i - r_j) \cdot (\dot{r}_i - \dot{r}_j)}{\|\dot{r}_i - \dot{r}_j\|} + (\sigma^2 - \|b_{i,j}\|^2)^{\frac{1}{2}}\right) \tag{60}$$

This equation can be understood by observing that the second factor on the right, namely

$$-\left(\frac{r_{i,j} \cdot \dot{r}_{i,j}}{\|\dot{r}_{i,j}\|} + (\sigma^2 - \|b_{i,j}\|^2)^{\frac{1}{2}}\right), \tag{61}$$

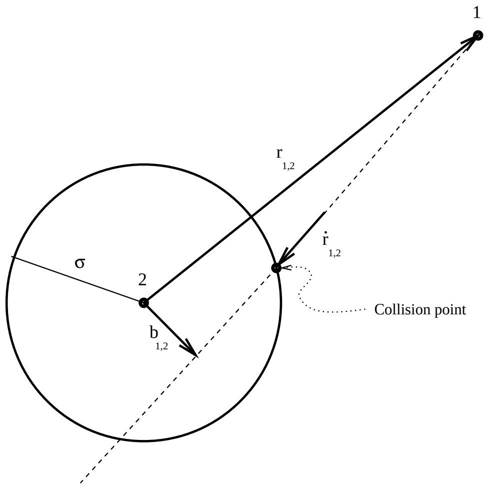

Figure 23: Illustration of collision parameters. Here $r_{1,2}$ is the position of the center of particle 1 relative to the center of particle 2; and $\dot{r}_{1,2}$ is the velocity of particle 1 relative to the velocity of particle 2. Thus you can think of particle 2 as fixed at the origin, and particle 1 at $r_{1,2}$ moving with velocity $\dot{r}_{1,2}$. For a collision to occur particle 1 must be moving towards the origin, and must come within a distance σ (the particle's diameter) of the origin. Particle 1 is moving towards the origin if the dot product $r_{1,2} \cdot \dot{r}_{1,2} < 0$ as in this picture. In this picture $\|b_{1,2}\|$ is the distance of closest approach by particle 1 to the origin: $b_{1,2}$ is a vector perpendicular to $\dot{r}_{1,2}$. The point at which the collision occurs is the point where the trajectory of particle 1 crosses the circle of radius σ around the origin, and the time of collision is the time it takes to move from $r_{1,2}$ to this point travelling at velocity $\dot{r}_{1,2}$.

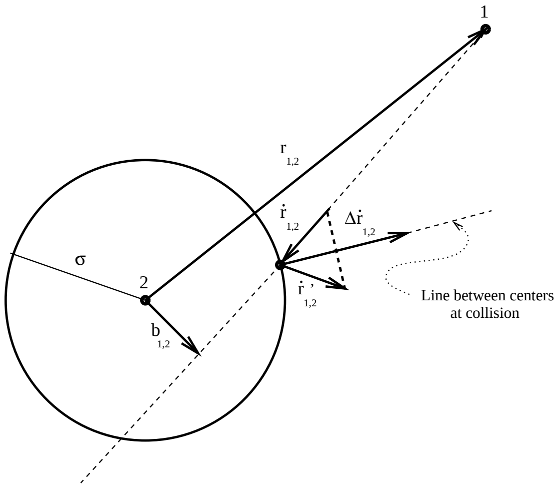

Figure 24: Change in velocity at collision of hard spheres. Parameters σ, $r_{1,2}$, $\dot{r}_{1,2}, b_{1,2}$ are as defined in figure 22; $\dot{r}'_{1,2}$ is the relative velocity after collision and $\Delta\dot{r}_{1,2}$ is the change in relative velocity due to the collision. Note that $\Delta\dot{r}_{1,2}$ is perpendicular to the line between centers at the time of collision.

is the distance to be traversed in the direction of $\dot{r}_{i,j}$ before collision: refer to figure 23 and do some elementary geometry. It is evident from these formulas that determining the time of the next collision is a good deal more complicated than in the 1-dimensional case.

When the particles do collide then the new velocities are easily computed. There is no change in velocity in the plane tangent to the spheres at the collision point; the velocity changes only in the direction of the line between the centers of the particle, as illustrated in figure 24.

The change in velocity takes place, as noted earlier, in the direction of the vector $(r_i - r_j)$ at the time of collision. In particular,

$$\begin{aligned} \dot{r}_{i,new} &= \dot{r}_{i,old} + \triangle\dot{r}_{i,old} \\ \dot{r}_{j,new} &= \dot{r}_{j,old} - \triangle\dot{r}_{i,old} \end{aligned} \tag{62}$$

where

$$\triangle\dot{r}_{i,old} = -\frac{(\dot{r}_{i,old} - \dot{r}_{j,old}) \cdot (r_{i,old} - r_{j,old})(r_{i,old} - r_{j,old})}{\|r_{i,old} - r_{j,old}\|^2}. \tag{63}$$

You should be able to verify that this formula gives the same result for the 1-dimensional case as obtained earlier, equation (55).

At this point you have all of the necessary formulas for constructing a program to solve the equations of motion by Euler's method or Verlet's method for any of the models — HL, LJ, or HS.

5 Exact solution of the equations of motion for HL model.

We noted earlier that the Hooke's Law model is special because, unlike the other models, it admits an exact solution. Here we outline the main ideas leading to the exact solution. In order to understand the material in this section, you need to know about eigenvalues and eigenvectors of matrices at an elementary level; for example, see [Fosdick 91, section 5.4].

Consider first the simple case of a single particle moving under the influence of an HL force, illustrated earlier in figure 3. The equation of motion for this particle is

$$m\ddot{q}(t) = -kq(t), \tag{64}$$

where m is the mass of the particle, k is the force constant, and $q(t)$ is the displacement of the particle from its equilibrium position at time t.

It is easy to verify that

$$q(t) = Q\cos(\omega t + \delta) \tag{65}$$

satisfies equation (64). If we substitute this expression for $q(t)$ into equation (64) we obtain

$$-mQ\omega^2\cos(\omega t + \delta) = -kQ\cos(\omega t + \delta). \tag{66}$$

The left side agrees with the right side provided that

$$\omega = \sqrt{\frac{k}{m}}. \tag{67}$$

Therefore

$$q(t) = Q\cos\left(\sqrt{k/m}\,t + \delta\right) \tag{68}$$

is a solution.

What about the parameters Q and δ appearing in this solution? We show here that they are determined by the initial conditions for the motion. Notice that when $t = 0$ we have

$$q(0) = Q\cos(\delta), \quad \dot{q}(0) = -\omega Q \sin(\delta). \tag{69}$$

Therefore, if we impose the initial conditions

$$q(0) = 1, \quad \dot{q}(0) = 0, \tag{70}$$

we easily find that

$$Q = 1, \quad \delta = 0. \tag{71}$$

Thus for the initial conditions given in equation (70) the solution to the equations of motion is

$$q(t) = \cos\left(\sqrt{k/m}\, t\right) \tag{72}$$

Thus the solution is a periodic function of time, with frequency $(2\pi)^{-1}\sqrt{k/m}$.

Now we turn to the more complicated case of a chain of four particles considered earlier and illustrated in figure 5. Following the one particle example above, we first "guess" a solution and then show that this guess satisfies the equations of motion provided certain parameters related to the frequency of the motion are satisfied. We will find that there are three distinct solutions, known as "modes". These modes correspond to the general solution, equation (68), for the one-particle case. The initial conditions determine a linear combination of the modes that is the particular solution, corresponding to equation (72).

The equations of motion for four particles in the HL model were given in equation (27). It is convenient to express them in matrix form:

$$m\ddot{q} = -kMq \tag{73}$$

where q is a vector and M is a matrix:

$$q = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix}, \quad M = \begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix}. \tag{74}$$

Of course q is a function of time but for simplicity we often write q rather than $q(t)$.

We proceed by analogy with the one-particle case just considered, guessing that the solution to these four equations has the form:

$$q = Q\cos(\omega t + \delta), \quad Q = \begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \\ Q_4 \end{bmatrix}. \tag{75}$$

This looks like the solution for the one particle case except q and Q are now vectors, q_i representing the displacement of the i^{th} particle from its equilibrium position:

$$q_i = Q_i \cos(\omega t + \delta). \tag{76}$$

We can verify that this guess is indeed a solution by the same process as before, that is by substituting it into the equations of motion, equation (73). Substitution and a little algebraic manipulation produces the result

$$MQ = \frac{m\omega^2}{k} Q. \tag{77}$$

You may recognize this as the usual form of a matrix eigenvalue equation where Q is an eigenvector and $m\omega^2/k$ is an eigenvalue of the matrix M; see [Fosdick 91, equation 14]. Thus our guessed solution satisfies the equations of motion provided that $m\omega^2/k$ and Q are an eigenvalue-eigenvector pair of the matrix M.

Since M is a 4×4 matrix it has four eigenvalues; these are

$$\lambda_0 = 0,\ \lambda_1 = 2 - \sqrt{2},\ \lambda_2 = 2,\ \lambda_3 = 2 + \sqrt{2} \tag{78}$$

and the corresponding eigenvectors are

$$\psi_0 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix},\ \psi_1 = \begin{bmatrix} 1 \\ \sqrt{2}-1 \\ -\sqrt{2}+1 \\ -1 \end{bmatrix},\ \psi_2 = \begin{bmatrix} 1 \\ -1 \\ -1 \\ 1 \end{bmatrix},\ \psi_3 = \begin{bmatrix} \sqrt{2}-1 \\ -1 \\ 1 \\ -\sqrt{2}+1 \end{bmatrix}. \tag{79}$$

An explanation of where these results came from would take us too far afield, but you can easily verify them using equation (77), or you could use MATLAB to compute the eigenvalues and eigenvectors numerically. Remember that an eigenvector can be normalized in various ways. Here we normalized them so that $\|\psi\|_\infty = 1$; on the other hand MATLAB normalizes them so that $\|\psi\|_2 = 1$.

Each eigenvalue-eigenvector pair represents a particular motion of the four-particle system called a "mode". Consider the mode represented by λ_3, ψ_3. The motion for this mode is given by

$$q = \begin{bmatrix} \sqrt{2}-1 \\ -1 \\ +1 \\ -\sqrt{2}+1 \end{bmatrix} \cos\left(\sqrt{(2+\sqrt{2})k/m}\, t + \delta\right). \tag{80}$$

Notice that each particle oscillates with a frequency

$$\frac{1}{2\pi}\sqrt{(2+\sqrt{2})k/m}.$$

Thus the frequency of the oscillation is proportional to the square root of the eigenvalue of the mode, and the relative amplitudes of the motion are determined by the eigenvector of the mode. figure 25 illustrates this mode. Since it has the largest eigenvalue, and therefore the highest frequency, we call it the "high-frequency mode".

The mode associated with λ_1 is called the "low frequency mode" since λ_1 is the smallest eigenvalue excepting zero (see below). The motion associated with this mode is illustrated in figure 26.

The eigenvalue $\lambda_0 = 0$ represents the state of no relative motion; i.e. no motion of the particles relative to each other. Notice that zero for the eigenvalue implies $\omega = 0$, which implies that the solution is independent of time. This situation would occur if we started all the particles in their equilibrium positions with no initial velocity: they would remain motionless. Here we are concerned only with states of relative motion; so we focus on the modes represented by the nonzero eigenvalues.

The importance of the modes comes from the fact that any motion of the four-particle system can be expressed as a linear combination of the modes. We illustrate this for a particular case. Consider the motion determined by initial conditions

$$q(0) = \begin{bmatrix} 1 \\ 0 \\ 0 \\ -1 \end{bmatrix}, \quad \dot{q}(0) = 0. \tag{81}$$

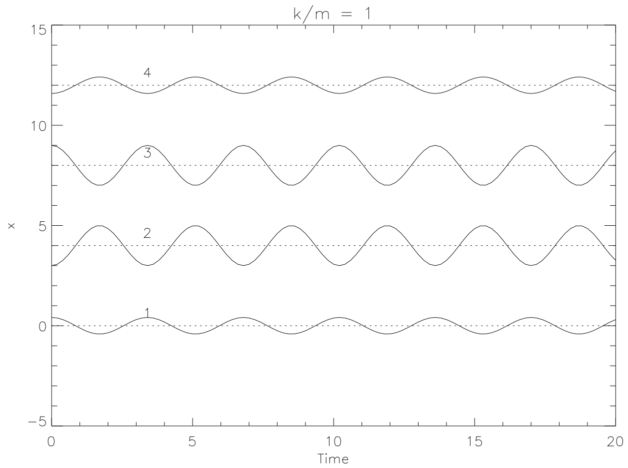

Figure 25: Illustration of the high-frequency mode of a four-particle system with $k/m = 1,\ \delta = 0$. Equilibrium positions of the four particles are assumed to be 0, 4, 8, 12.

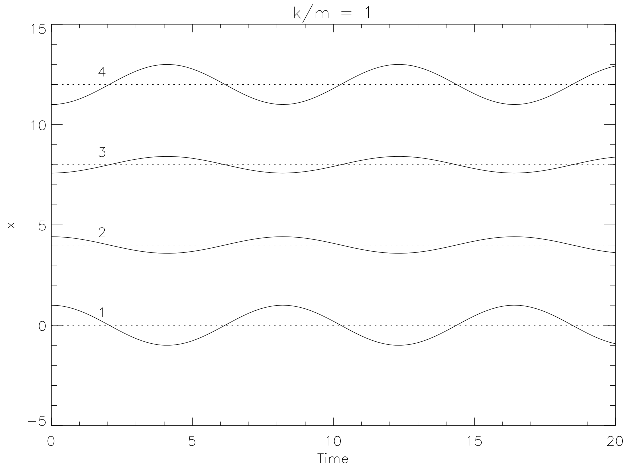

Figure 26: Illustration of the low-frequency mode of a four-particle system with $k/m = 1,\ \delta = 0$. Equilibrium positions of the four particles are assumed to be 0, 4, 8, 12.

We express the general solution as a linear combination of the modes:

$$q(t) = \sum_{i=1}^{3} c_i \psi_i \cos\left(\sqrt{(k/m)\lambda_i}\; t + \delta_i\right), \tag{82}$$

where c_1, c_2, c_3 are arbitrary constants. We must determine c_1, c_2, c_3 and $\delta_1, \delta_2, \delta_3$ so that $q(t)$ defined by equation (82) satisfies the initial conditions specified in equation (81).

To satisfy the condition on $q(0)$, set t equal to zero on the right side of equation (82) and substitute the result for $q(0)$ into the initial condition equation. This gives:

$$\sum_{i=1}^{3} c_i \cos(\delta_i)\psi_i = \begin{bmatrix} 1 \\ 0 \\ 0 \\ -1 \end{bmatrix} \tag{83}$$

Next differentiate both sides of equation (82) with respect to t to obtain the following expression for $\dot{q}(t)$

$$\dot{q}(t) = -\sum_{i=1}^{i=3} c_i \psi_i \sqrt{k/m\lambda_i} \sin\left(\sqrt{(k/m)\lambda_i}\; t + \delta_i\right). \tag{84}$$

Then set t equal to zero in this equation and substitute the result into the initial condition equation for $\dot{q}(0)$, giving the result

$$\sum_{i=1}^{3} c_i \sqrt{\frac{k}{m}\lambda_i} \sin(\delta_i)\psi_i = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{85}$$

The four equations in equation (83) together with the four equations in equation (85) give us a system of eight linear equations in six unknowns d_1, d_2, …, d_6 where

$$d_i = c_i \cos(\delta_i), \quad d_{i+3} = c_i \sin(\delta_i) \quad (i = 1, 2, 3). \tag{86}$$

It might seem that we cannot solve these equations because the number of equations exceeds, by 2, the number of unknowns. However two of the equations are redundant. Note that the sum of the components of each eigenvector is zero, and that the same is true for the sum of the components of $q(0)$ and $\dot{q}(0)$. Therefore there are at most six independent equations, not eight. These equations can be solved, for example with MATLAB, to obtain the result

$$\begin{aligned} d_1 &= -8.535533906e-01,\ d_2 = 0,\ d_3 = -3.535533906e-01, \\ &\quad d_4 = d_5 = d_6 = 0. \end{aligned} \tag{87}$$

From this is follows that

$$\begin{aligned} c_1 &= -8.535533906e-01,\ c_2 = 0,\ c_3 = -3.535533906e-01, \\ &\quad \delta_1 = \delta_2 = \delta_3 = 0. \end{aligned} \tag{88}$$

To confirm your understanding of this you might verify that these coefficients and phase angles produce a solution to the equations of motion that satisfies the initial conditions.

Similarly it is possible to obtain a solution to the equations of motion for any valid set of initial conditions (i.e. the sums of the components of $q(0)$ and of $\dot{q}(0)$ are both zero.)

The procedure for solving the problem with a chain of n atoms is the same; the matrix M has order n and has the form

$$M = \begin{bmatrix} 1 & -1 & 0 & 0 & \dots & 0 \\ -1 & 2 & -1 & 0 & \dots & 0 \\ 0 & -1 & 2 & -1 & \dots & 0 \\ \vdots & & & & & \vdots \\ 0 & 0 & \dots & -1 & 2 & -1 \\ 0 & 0 & \dots & & -1 & 1 \end{bmatrix} \tag{89}$$

The eigenvalues of M are given by

$$\lambda_p = 2\left(1 - \cos(\frac{p\pi}{n})\right) \tag{90}$$

and the eigenvectors are given by

$$\psi_j^{(p)} = A\ \cos\left(\frac{p\pi}{n}(j - \frac{1}{2})\right) \tag{91}$$

where $\psi_j^{(p)}$ is the j^{th} component of the p^{th} eigenvector. The index p takes values $0, 1, \ldots, n-1$; and the index j takes values $1, 2, \ldots, n$. The coefficient A, the normalization factor, is arbitrary. In the 4-particle example we chose it so that the element of maximum magnitude in $\psi^{(p)}$ has magnitude 1; i.e.,

$$\|\psi^{(p)}\|_\infty = 1. \tag{92}$$

It is worth noting that for every n there is one eigenvalue equal to zero. The trivial mode corresponding to this eigenvalue would be ignored, just as we did in the case for the 4-particle chain.

In order to check your understanding of the above discussion you might try to solve the following problems.

1. What is the high frequency mode for a 5-particle system?

2. What is the upper bound on the frequency for any one-dimensional chain, assuming $k/m = 1$?

3. What is the exact solution of the equations of motion for a 5-particle system with the following initial conditions?

$$q(0) = \begin{bmatrix} 1.0 \\ 0.5 \\ -0.5 \\ -0.5 \\ -0.5 \end{bmatrix}, \quad \dot{q}(0) = 0. \tag{93}$$

Assume $k/m = 1$.

4. What is the exact solution of the equations of motion for a 5-particle system with the following initial conditions?

$$q(0) = 0, \quad \dot{q}(0) = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ -1 \end{bmatrix}. \tag{94}$$

Assume $k/m = 1$.

5. Assume a two-dimensional HL model with 8 particles in each dimension. What is the matrix form of the equations of motion?

6. Assume a three-dimensional HL model with 8 particles in each dimension. What is the matrix form of the equations of motion?

References

[Alder & Wainright 59] ALDER, B. J. AND T. E. WAINRIGHT. [1959]. Studies in molecular dynamics i: General method. *J. Chem. Phys.*, 31:459–466.

[Alder & Wainright 60] ALDER, B. J. AND T. E. WAINRIGHT. [Nov 1960]. Studies in molecular dynamics ii: Behavior of a small number of elastic spheres. *J. Chem. Phys.*, 33(5):1439–1451.

[Bailey 91] BAILEY, DAVID H. [October 1991]. Mpfun: A portable high precision performance multiprecision package. RNR Technical Report RNR-90-022, NASA Ames Research Center, NASA Ames Research Center.

[Burden & Faires 85] BURDEN, RICHARD L. AND J. DOUGLAS FAIRES. [1985]. *Numerical Analysis.* Prindle, Weber & Schmidt, third edition. ISBN 0-87150-857-5.

[Conte & de Boor 80] CONTE, S. D. AND CARL de BOOR. [1980]. *Elementary Numerical Analysis: An Algorithmic Approach.* McGraw-HIll, third edition. ISBN 0-07-012447-7.

[Dagum 88] DAGUM, LEONARDO. [1988]. Implementation of a hypersonic rarefied flow particle simulation on the connection machine. RIACS Technical Report 88.46, RIACS, NASA Ames Research Center. Also published as conference paper in 1989.

[Fosdick 91] FOSDICK, LLOYD D. [1991]. A review of selected topics from numerical analysis. HPSC Course Notes.

[Hannon et al. 86] HANNON, L., G. C. LIE, AND E. CLIMENTI. [1986]. Molecular dynamics simulation of flow past a plate. *J. Sci. Computing*, 1(2):145–150.

[Henrici 62] HENRICI, PETER. [1962]. *Discrete Variable Methods in Ordinary Differential Equations.* Wiley.

[Karplus & McCammon 86] KARPLUS, MARTIN AND J. ANDREW MCCAMMON. [April 1986]. The dynamics of proteins. *Scientific American*, pages 42–51.