22.103 Microscopic Theory of Transport (Fall 2003)
Lecture 25 (11/17/03)

Role of Atomistic Simulation in Transport - Basic Molecular Dynamics

References --

S. Yip, "Atomistic Modeling of Liquids", in *Encyclopedia of Advanced Materials*, D. Bloor, R. Brook, M. Flemings, S. Mahajan, eds. (Pergamon, 1994) and references therein.

With this lecture we begin a new topic in this course, that of atomistic simulation as a tool for the investigation of transport phenomena. We have seen how transport processes can be discussed in various ways, as transport coefficients, correlation functions, and in terms of phase-space distribution functions which satisfy transport equations. In the discussion of the Boltzmann equation we saw that one has to restrict himself to uncorrelated binary collisions in order to arrive at tractable approximations. This restriction to low-density gas means that we are not able to discuss transport phenomena that occur in liquids and solids, systems for which the Boltzmann equation cannot describe. In order to treat liquids and solids one can try to extend the Boltzmann approach based on particle collisions. This line of research has led to very complex mathematical problems involving correlated binary collisions and triple collisions. The mathematics become so heavy that physical insights become obscure. The alternative is to go to atomistic simulations where one studies particle trajectories numerically. This turns out to be a very powerful, though theoretically much less elegant approach.

We will develop the simulation approach to transport in the next few lectures. First we note that there are two major methods of atomistic simulation, molecular dynamics (MD) and Monte Carlo (MC). MD is very suited to the study of transport or dynamical processes, whereas MC also can be used, although not in as natural a manner as MD. For a comparison of these two methods, see the reference given above.

The point to emphasize again is that the basic information that we seek to obtain and use to build up our basic understanding lies in the particle trajectories $\{\underline{R}(t)\}$. Simulation provides the means to directly generate this data, using only the interatomic potential as the input. Simulation can be used for practically any system, whether it is a gas, liquid or solid. There are other significant features of simulation which are discussed below. We leave it to the individual student to decide how much details he wants to learn about simulation. We expect that some will want to know more than others. What every student should take away is the appreciation of what simulation can do, and some examples of what it has done already. We think it is safe to predict that the role of simulation in understanding transport will only become more and more important.

Basic Molecular Dynamics (MD)

This lecture is intended to be a concise, self-contained discussion of the basic elements of molecular dynamics (MD) simulation, serving as a read-me primer for the newcomer as well as a summary of essentials for the initiated student who is not yet an expert. We will develop the subject according to the following list of topics.

1. Defining the MD Method
2. The Pair Potential
3. Bookkeeping Matters
4. Properties Which Make MD Unique
5. Hands-On MD
6. Understanding Crystals and Liquids - An Example of an Application

1. Defining the MD Method

A working defintion of MD is: **The process by which one generates the atomic trajectories of a system of N particles by direct numerical integration of Newton's equations of motion (with appropriate specification of an interatomic potential and suitable initial and boundary conditions).**

For an even shorter definition, one can leave out the words in the parenthesis and consider them as detailed qualifications. What is meant by the few words in this statement? Consider a simulation model (system) to be N particles contained in a region of volume V at temperature T, shown schematically in Fig. 1.

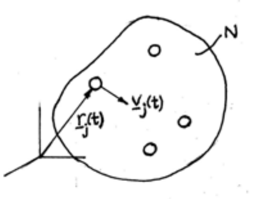

Fig.1. A model for molecular dynamics simulation, a system of N particles, each having an instantaneous position $\underline{r}_j(t)$ and velocity $\underline{v}_j(t)$, with all the particles interact with each other through a potential energy U. Simulation means letting the system evolve in time and recording the particle positions and velocities at small incremental time steps. How this is done is explained in this lecture.

The positions of the N particles are specified by a set of N vectors, $\{\underline{r}(t)\} = (\underline{r}_1(t), \underline{r}_2(t), ..., \underline{r}_N(t))$, with $\underline{r}_j(t)$ being the position of particle j at time t. Knowledge of $\{\underline{r}(t)\}$ at various time instants allows us to follow the particle **trajectories** as they move around. Our model system of particles has a certain energy E which is the sum of kinetic and potential energies of the particles, E = K + U, where K is the sum of individual kinetic energies

$$K = \frac{1}{2} m \sum_{j=1}^{N} \underline{v}_j^2 \qquad (1)$$

and U is a prescribed interatomic potential mentioned above, $U(\underline{r}_1, \underline{r}_2, ..., \underline{r}_N)$. In general U depends on the positions of all the particles in a complicated fashion. We will soon introduce a simplifying approximation, the assumption of two-body or pairwise interaction, which makes this most important quantity much easier to handle.

To find the atomic trajectories in our model we need to solve the equations that specify the particle positions; this is just what the Newton's equations of motion do, as everyone knows F = ma from simple mechanics. While we are all familiar with the equations of motion of a pendulum or a rolling body, the equations of motion for our N-particle model is more complicated because the equations for different particles are all coupled to each other through the potential energy U. We can see this readily when we write out the equations explicitly,

$$m\frac{d^2\underline{r}_j}{dt^2} = -\underline{\nabla}_{\underline{r}_j} U(\{\underline{r}\}), \qquad j = 1, ..., N \qquad (2)$$

That the motion of one particle depends on where the others are is not surprising, since the force acting on one particle changees whenever one of its neighbors moves. Eq.(2) looks deceptively simple, but it is as complicated as the famous N-body problem which we cannot solve exactly when N is greater than 2. It is a system of coupled second-order, non-linear ordinary differential equations. On the other hand, (2) can be solved numerically, which is what one does in molecular dynamics simulation.

When we say integrate (2) to obtain the atomic trajectories, we have in mind dividing a time interval of interest into many small segments, each being a time step of size Δt. Given the system conditions at some initial time t_o, $\{\underline{r}(t_o)\}$, integration means we advance the system successively by increments of Δt ,

$$\{\underline{r}(t_o)\} \to \{\underline{r}(t_o + \Delta t)\} \to \{\underline{r}(t_o + 2\Delta t)\} \to ...\{\underline{r}(t_o + N_t\Delta t)\} \qquad (3)$$

where N_t is the number of time steps making up the interval of integration.

How do we numerically integrate (2) for a given U? A simple way is to write a Taylor series expansion,

$$\underline{r}_j(t_o + \Delta t) = \underline{r}_j(t_o) + \underline{v}_j(t)\Delta t + \frac{1}{2}\underline{a}_j(t)(\Delta t)^2 + ... \qquad (4)$$

and a similar expansion for $\underline{r}_j(t_o - \Delta t)$. Adding the two expansions gives

$$\underline{r}_j(t_o + \Delta t) = -\underline{r}_j(t_o - \Delta t) + 2\underline{r}_j(t_o) + \underline{a}_j(t_o)(\Delta t)^2 + ... \qquad (5)$$

Notice that the left-hand side is what we want, namely, the position of particle j at the next time step Δt, whereas all the terms on the right-hand side are quantities evaluated at time t_o . We already know the positions at t_o and the time step before, so to use (5) we need the acceleration of particle j at time t_o. For this we can make use of (2) and substitute $\underline{F}_j(\{\underline{r}(t_o)\})/m$ in place of the acceleration, where F is just the right-hand side of (2). Thus through Eq.(5) one performs the integration of (2) in successive time increments, following the system evolution in discrete time steps. . Although there are more elaborate ways of doing the integration, this is basic idea of generating the atomic trajectories, the essence of MD. The particular procedure we have described is the Verlet central difference method. In the MD code given to the class, a more accurate method called the Gear Predictor-Corrector is used. Higher accuracy of integration allows one to take a larger value of Δt, desirable because one can cover a longer time interval. On the other hand, the tradeoff is that one needs more memory relative to the simpler method.

The time integrator is at the heart of MD simulation, with the sequence of positions and velocities (trajectories) being the raw output. A typical flow-chart for an MD code would look something like the following.

(a) → (b) → (c) → (d) → (e) → (f) → (g)

a = set particle positions
b = assign particle velocities
c = calculate force on each particle
d = move particles by timestep Δt
e = save current positions and velocities
f = if reach preset no. of timesteps, stop, otherwise go back to (c)
g = analyze data and print results

2. The Pair Potential

To make the simulation tractable, it is common to assume the interatomic potential U can be represented as the sum of two-body interactions,

$$U(\underline{r}_1,\ldots,\underline{r}_N) \cong \frac{1}{2}\sum_{i\neq j} V(r_{ij}) \qquad (6)$$

where r_{ij} is the separation distance between particles i and j. V is the pairwise interaction; it is a central force potential, being a function only of the separation distance between the two particles, $r_{ij} = \left|\underline{r}_i - \underline{r}_j\right|$. A very common two-body interaction energy used in atomistic simulations, known as the Lennard-Jones potential, is

$$V(r) = 4\varepsilon\left[(\sigma/r)^{12} - (\sigma/r)^{6}\right] \qquad (7)$$

where ε and σ are parameters of the potential. Like all pair potentials, this interaction energy rises sharply (with inverse power of 12) at close interatomic separations, has a minimum, and decays to zero at large separations. See Fig. 2 which also shows the behavior of the interatomic force,

$$F(r) = -\frac{dV(r)}{dr} \tag{8}$$

which is repulsive at short separations and attractive at large separations. We can understand the repulsion as arising from the overlap of the electron clouds, while the attraction is due to the interaction between the induced dipole in each atom. The value of 12 for the first exponent in V(r) has no special significance, as the repulsive term could just as well be replaced by an exponential. The second exponent results from quantum mechanical calculations (the so-called London dispersion force) and therefore is not arbitrary. Regardless of whether one uses (7) or some other interaction potentials, a short-range repulsion is necessary to give the system a certain size or **volume** (density), without which the particles will collapse onto each other. A longer range attraction is also necessary for the **cohesion** of the system, without which the particles will not stay together as they must in all condensed states of matter. Both are necessary for modeling the physical properties of solids and liquids which we know from everyday experience.

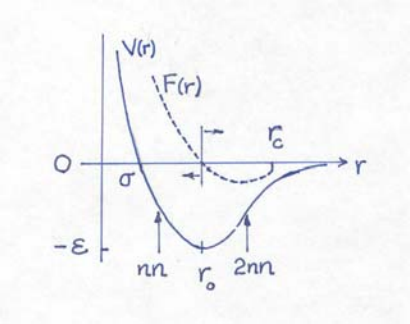

Fig. 2. The Lennard-Jones interatomic potential V(r). The potential vanishes at $r = \sigma$ and has a depth equal to ε .
Also shown is the corresponding force F(r) between the two particles (dashed curve) which vanishes at r_o. At separations less or greater than r_o the force is repulsive or attractive respectively. Arrow at nn and 2nn indicate typical separation distances of nearest and second nearest neighbors in a solid.

3. Bookkeeping Matters

Our simulation system is typically a cubical cell in which particles are placed either in a very regular manner, as in modeling a crystal lattice, or in some random manner, as in modeling a gas or liquid. The number of particles in the simulation cell is quite small. For the homework assignment only certain discrete values, 32, 108, 256, 500, 864, should be specified. These come about because the class code (which we call hailecode) is designed for a face-centered cubic lattice which has 4 atoms in each primitive cell. Thus, if our cube has s cells along each side, then the number of particles in the cube will be $4s^3$. The above numbers then correspond to cubes with 2, 3, 4, 5, and 6 cells along each side respectively.

Once we choose the number of particles we want to simulate, the next step is to choose what system density we want to study. Choosing the density is equivalent to choosing the system volume since density n = N/V, where N is the number of particles and V is the volume. Hailecode uses dimensionless reduced units. The reduced density DR has typical values around 0.9 - 1.2 for solids, and 0.6 - 0.85 for liquids. For reduced temperature TR we recommend values of 0.4 - 0.8 for solids, and 0.8 - 1.3 for liquids. Notice that assigning particle velocities in (b) above is tantamount to setting the system temperature.

For simulation of bulk systems (no free surfaces) it is conventional to use the **periodic boundary condition** (pbc). This means the cubical cell is surrounded by 26 identical image cells. For every particle in the simulation cell, there corresponds an image particle in each image cell. The 26 image particles move in exactly the same manner as the actual particle, so if the actual particle should happen to move out of the simulation, the image particle in the image cell opposite to the exit side will move in (and becomes the actual particle, or the particle in the simulation cell) just as the original particle moves out. The net effect is with pbc particles cannot be lost (destroyed) or gained (created). In other words, the particle number is conserved, and if the simulation cell volume is not allowed to change, the system density remains constant.

Since in the pair potential approximation, the particles interact two at a time, a procedure is needed to decide which pair to consider among the pairs between actual particles and between actual and image particles. The **minimum image convention** is a procedure where one takes the nearest neighbor to an actual particle, irregardless of whether this neighbor is an actual particle or an image particle. Another approximation which is useful to keep the computations to a manageable level is to introduce a **force cutoff** distance beyond which particle pairs simply do not see each other (see the force curve in Fig. 2). In order not to have a particle interact with its own image, it is necessary to ensure that the cutoff distance is less than half of the simulation cell dimension.

Another bookkeeping device often used in MD simulation is a **Neighbor List** which keeps track of who are the nearest, second nearest, ... neighbors of each particle. This is to save time from checking every particle in the system every time a force calculation is made. The List can be used for several time steps before updating. Each update is expensive since it involves NxN operations for an N-particle system. In low-temperature solids where the particles do not move very much, it is possible to do an entire simulation without or with only a few updating, whereas in simulation of liquids, updating every 5 or 10 steps is quite common. For further discussions of bookkeeping matters, the student should see the MD Primer of J. M. Haile (1980).

4. Properties Which Make MD Unique

There is a great deal that can be said about why MD is such a useful simulation technique. Perhaps the most important statement is that in this method (consider classical MD for the moment, as opposed quantum MD) one follows the atomic motions according to the principles of classical mechanics as formulated by Newton and Hamilton. Because of this, the results are physically as meaningful as the potential U that is used. One does not have to apologize for any approximation in treating the N-body problem. Whatever mechanical, thermodynamic, and statistical mechanical properties that a system of N particles should have, they are all still present in the simulation data. Of course how one extracts these properties from the output of the simulation – the atomic trajectories – determines how useful is the simulation. Before any conclusions can be drawn, one needs to consider how the various properties are to be obtained from the simulation data. We can regard MD simulation as an 'atomic video' of the particle motion (one which we can display as a movie). While there is a great deal of realistic details in the motions themselves, how to extract the information in a scientifically meaningful way is up to the viewer. It is to be expected that an experienced viewer can get much more useful information than an inexperienced one!

The above comments aside, we list here a number of basic reasons why MD simulation is so useful (or unique). These are meant to guide the thinking of the student and encourage the student to discover and appreciate the many interesting and thought-provoking aspects of this technique on your own.

(a) Unified study of all physical properties. Using MD one can obtain thermodynamic, structural, mechanical, dynamic and transport properties of a system of particles which can be a solid, liquid, or gas. One can even study chemical properties and reactions which are more difficult and will require using quantum MD.

(b) Several hundred particles are sufficient to simulate bulk matter. While this is not always true, it is rather surprising that one can get quite accurate thermodynamic properties such as equation of state in this way. This is an example that the law of large numbers takes over quickly when one can average over several hundred degrees of freedom.

(c) Direct link between potential model and physical properties. This is really useful from the standpoint of fundamental understanding of physical matter. It is also very relevant to the structure-property correlation paradigm in materials science.

(d) Complete control over input, initial and boundary conditions. This is what gives physical insight into complex system behavior. This is also what makes simulation so useful when combined with experiment and theory.

(e) Detailed atomic trajectories. This is what one can get from MD, or other atomistic simulation techniques, that experiment often cannot provide. This point alone makes it compelling for the experimentalist to have access to simulation.

We should not leave this discussion without reminding ourselves that there are significant limitations to MD. The two most important ones are:

(a) Need for sufficiently realistic interatomic potential functions U. This is a matter of what we really know fundamentally about the chemical binding of the system we want to study. Progress is being made in quantum and solid-state chemistry, and condensed-matter physics; these advances will make MD more and more useful in understanding and predicting the properties and behavior of physical systems.

(b) Computational-capability constraints. No computers will ever be big enough and fast enough. On the other hand, things will keep on improving as far as we can tell. Current limits on how big and how long are a billion atoms and about a microsecond in brute force simulation.

5 Hands-On MD

'Talk is cheap' when it comes to modeling and simulation. What is not so easy is to 'just do it'. In this spirit we want everyone to get your hands on an MD simulation code and just play with it.

There exists a very useful writeup – **A Primer on the Computer Simulation of Atomic Fluids by Molecular Dynamics, J. M. Haile (1980)**, which is effectively a User's Manual for the code. We will put a copy on reserve in the Reserve Library. See Dion if you want to order a personal copy.

5. Understanding Crystals and Liquids – An Example of an MD Application

There are many ways one can study the structure and dynamics of solids and liquids at the atomistic level using MD. In fact, this is one of main reasons why MD has become so well respected for what it can tell about the distribution of atoms and molecules in various states of matter, and the way they move about in response to thermal excitations or external stress such as pressure. We will encounter many examples of this kind of discussion, for now we will focus on two basic properties of matter, structure and motion at the atomic level. Fig. 3 shows the simulated trajectories of molecules as they move around in a typical solid, liquid, and gas. The most striking features that one can readily observe are that the atomic structure is highly ordered in the solid (crystalline) state, quite disordered (random but with some degree of local packing) in the liquid state, and very random in the gas state. The corresponding particle motions are small-amplitude vibrations about the lattice site, diffusive movements over a local region, and long free flights interrupted by a collision every now and then. The question we now ask is how to quantify this information in a way that it can be used to better understand the many physical properties of systems of atoms and molecules. In other words, if we can generate the data, how do we analyze them?

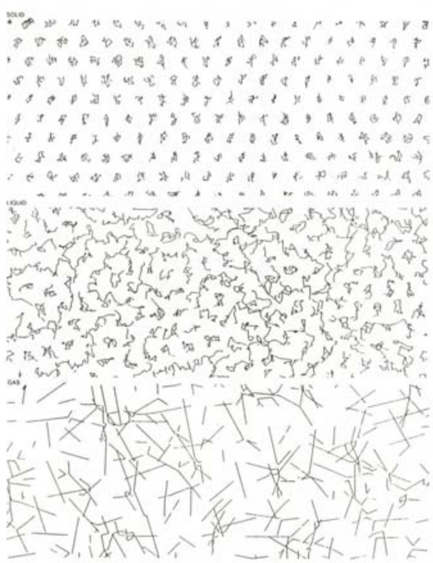

Fig. 3. Atomic trajectories of a two-dimensional crystal, liquid, and gas simulated by molecular dynamics [J. A. Barker and D. Henderson, Scientific American, Nov. 1981].

Imagine we are doing a simulation with the hailecode where we specify the following input:

NP = number of particles
NEQ = number of timesteps for equilibration
MAXKB = number of timesteps for the actual simulation run
TR = reduced temperature
DR = reduced density

The output of hailecode for this set of input parameters can be plotted in Matlab by following the above instructions. What you will get are three plots. Fig. 4 is a composite of three graphs showing the variation of pressure, potential energy, and temperature with time as the simulation evolves. This information is useful to make sure that the system is well equilibrated and that nothing strange is happening during the entire simulation. These graphs are like the meters on the wall of a reactor control room, showing how the pressure and temperature of the reactor are varying instantaneously during operation. Although important, they do not tell us anything about what is going on with the atoms *inside* the reactor. That is, they are macroscopic properties which are not sensitive to the details at the microscopic level. The only way you can tell whether you are dealing with a solid, liquid or gas is to examine the values of the equilibrium pressure, volume, and temperature, and use them to locate the system in the phase diagram. We will not do this here. Fig. 4, however, is useful for seeing that the system has reached equilibrium, as indicated by the convergence of the properties to steady-state values. The transients give us a feeling for how quickly the initial perturbations are damped out by the molecules interacting with each other.

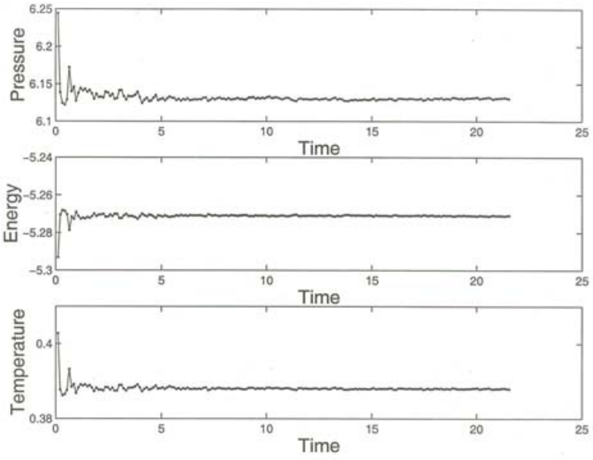

Fig. 4. Time variation of system pressure, energy, and temperature in an MD simulation of a solid. The initial behavior are transients which decay in time as the system reaches equilibrium.

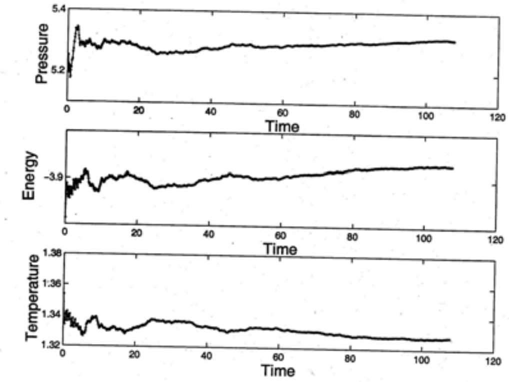

Fig. 5. Results of MD simulation similar to Fig. 4 except the system is now in a liquid state. Note the longer transients and the slower convergence to equilibrium.

The Radial Distribution Function

This quantity is defined as

$$g(r) = \rho(r) / \rho \qquad (9)$$

where $\rho(r)$ is the local number density. For the hailecode, DR is the dimensionless density $\rho\sigma^3$. By the way, the dimensionless temperature TR is just $k_B T / \varepsilon$, where k_B is the Boltzmann's constant. Recall that σ, ε are the two parameters of the Lennard-Jones potential model. Hailecode calculates g(r) according to the expression

$$g(r) = \frac{< N(r \pm \Delta r / 2) >}{\Omega(r \pm \Delta r / 2)\rho} \qquad (10)$$

where the numerator is the average number of particles in a spherical shell of radius r and thickness Δr, with the shell centered on one of the particles (any particle is as good as any other) in the system, and Ω in the denominator is the volume of this shell. A simple way to express the physical meaning of g(r) is: $g(r)2\pi r^2 dr$ = number of particles lying in a spherical shell of radius r and thickness dr given that a particle is at the origin.

What should g(r) look like if one plots it as a function of r? Two typical distributions are shown in Fig. 6. The function g(r) shows several peaks, which can be very sharp or quite broad depending on the state of the system. Physically a peak indicates a particularly favored separation distance for the neighbors to a given particle. The first peak corresponds to the nearest neighbor shell, the second peak to the second nearest neighbor shell, etc. Thus, g(r) is the function that reveals the atomic structure of the system being simulated.

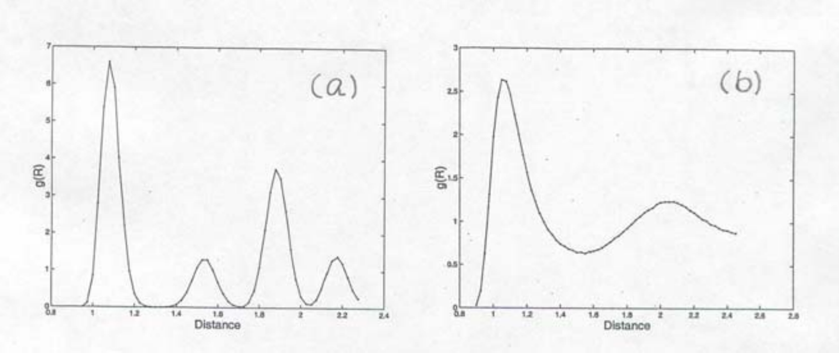

Fig. 6. Comparison of radial distribution functions simulated by molecular dynamics for a crystal (a) and a liquid (b) using the Lennard-Jones potential.

In the case of the output from hailecode we can even predict where the peaks should be located in the case of a low-temperature crystal. This is because the atoms are put into the simulation cell in the positions of a face-centered cubic lattice. It is known that in the primitive unit cell of fcc, one has 4 atoms in the cell. Sitting on any of the particles one can look around the surroundings in the lattice and count up the number of nearest neighbor, second nearest neighbor, third neighbor, ..., as well as the distances between the central particle and its various neighbors. The numbers should be 12, 6, 24, and 12 for the first four neighbors. Thus the four peaks in Fig. 6(a) correspond to these four neighbor rings. If one were to plot $2\pi r^2 g(r)$ instead of g(r), the areas under the four peaks will also match the number of neighbors. Looking now at the g(r) for a liquid, Fig. 6(b), we still see a prominent nearest neighbor peak which is considerably broadened by thermal motion at the higher temperature. We also see a broad second-neighbor at distances where the third and fourth neighbor peaks occur in the solid. In changing from solid to liquid, it appears that the second nearest peak has dissolved (implication is separation from the central particle at this distance is no longer favored in the liquid environment), while the third and fourth neighbor peaks merge to form what is now the second neighbor peak in the liquid.

The Square Displacement function $< \Delta r^2 >$

This quantity, which we have already encountered in the lecture on random walk, is defined as

$$< \Delta r^2 >= \frac{1}{N} \sum_i [\underline{r}_i(t) - \underline{r}_i(0)]^2 \qquad (11)$$

Here $\underline{r}_i(t)$ is the position of particle i at time t, so the square of the vector difference is the distance that particle i has moved during the time interval t. If we average over all the particles this then gives the mean square distance that the particles have moved on average during time t.

By definition $< \Delta r^2 >$ must start at zero at t = 0 and grows like t2 with a coefficient proportional to $k_B T / m$, where m is the particle mass. Recall from Fig. 3 that all the atoms in a solid are bound to some local position. So for a solid we expect $< \Delta r^2 >$ to grow to a characteristic value, which should be determined by how tightly are the atoms being to their lattice site, and fluctuate weakly about this level. In contrast, if the system were a liquid, then we expect all the atoms to be able to diffuse continuously through the liquid, as in Brownian motion. Diffusive motion manifest in $< \Delta r^2 >$ as a linear variation with t, which is the signature of classical diffusion. These simple features of $< \Delta r^2 >$ can be quite helpful when it comes to interpreting the simulation results.

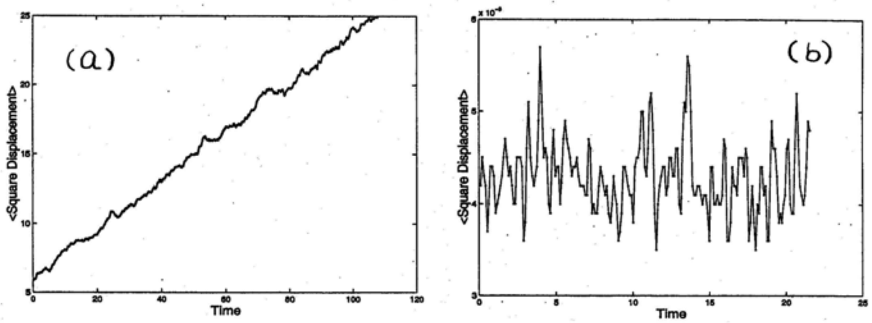

Fig. 7. Mean square displacement in a liquid (a) and a crsytal (b) as simulated by molecular dynamics. The temperature and density conditions are those in Fig. 6.

In closing, we have given some hints here as to how one can learn about the structure and dynamics of systems of particles by doing MD simulation. We hope the students will explore further on your own. You should play around with using different values for the input parameters. Consider the following suggestions.

NP = 32, 108, 256, 500, 864. Any one of the these values will work. Obviously a small system will run faster which means you get the results back right away if you use a 32-particle simulation cell as opposed to an 864-particle cell. The latter generally takes less than 5 minutes according to my experience.

NEQ = 1000. Use this default value. We can talk about doing something different later.

MAXKB = 2000 for a short run, 10000 for a longish run.

TR = 0.4 - 0.7 for crystal, 0.9 - 1.2 for liquid

DR = 0.9 - 1.2 for crystal, 0.6 - 0.9 for liquid

You should keep in mind that using more particles and running to longer times will give you smoother and higher-quality results. The cost is that the simulation will take longer. If you pick combinations of TR and DR values which are not in the part of the phase diagram that is clearly solid or liquid, then the system may have a hard time deciding which phase it should go into. Then you can get results that are not as clearcut as the behavior we havediscussed. There is much that you will learn by exploring on your own. Have fun and let us know how you are doing!