Particle Simulation and Diffusion Experiment

Brian Burns
Advanced Physics Lab
Spring 2003
Dr. John Miller, Jr.

Abstract

A brief overview of molecular dynamics simulations is given, and a prototype particle simulation program is written in Matlab in order to demonstrate the measurement of various physical properties. The Maxwell-Boltzmann velocity distribution and Ideal Gas Law are verified, and heat conduction and convection are demonstrated, as well as a qualitative verification of liquid phase behavior. Finally, a real diffusion experiment is performed to verify Einstein's 1905 formula for the diffusion constant.

Background and Motivation

Molecular dynamic (MD) simulations are of growing importance in physical sciences, especially for their ability to act as nanoscale laboratories, allowing the measurement of properties impossible to do in a lab. Their ability to do this though, hinges on their accuracy, and the accuracy is limited by the amount of computer power and time available. Current computer power limits simulations to space and time scales on the order of micrometers and nanoseconds.

Particle interactions can be based on different potential functions. An ideal gas is modelled by treating the particles as billiard balls, with elastic collisions, a situation which corresponds to a potential function that is infinite up to the particle radius, then zero everywhere else. A more realistic potential function would have a minimum at some distance from the particle, and then increase sharply as distance from the particle decreases. The Lennard-Jones potential is a well-known example of such a function (described later). Such potential functions model the attractive force between atoms that results from their mutually induced dipole attraction, and the repulsive force that results when the atoms are brought too close together. Many other potential functions exist, all attempting to model the interatomic forces more accurately.

Two-particle potential functions are most accurate for dilute phases. If a third particle is brought in near the two particles, the induced dipoles would be distorted, so the two-particle potential function does not model the situation exactly. Hence for more accurate simulations, especially in the liquid phase, it is best to also include a three-body potential function in the calculations.

Most MD simulations model a box of particles with dimensions on the order of 10-100 angstroms, and implement periodic boundary conditions in order to reduce the effect the walls have on the properties of the collection of particles. This means that the box is treated as if it is surrounded by copies of itself, and these 'phantom' particles also contribute to the potential energy of the particles in the box. If a particle goes outside the bounds of the box it is simply translated to the other side.

Monte Carlo (MC) simulations take a different approach to simulating a collection of particles - they obtain averages over a random traversal of phase space rather than integrating the equations of motion for the particles.

For the most accuracy, ab initio simulations are used - the term means from first principles, and in such simulations the input is solely the atomic number of the atoms. Quantum mechanical calculations then determine the forces between particles.

There are various MD simulations available on the internet, including CHARMM, AMBER, and NAMD.

The standard test case is to model Argon as a gas, since it is a monatomic noble gas, and is reasonably well modelled by the Lennard-Jones potential.

Proposed Experiment and Methodology

There are many possible programming platforms available, and several were examined to determine the best one for the purpose of writing a simple MD simulation:

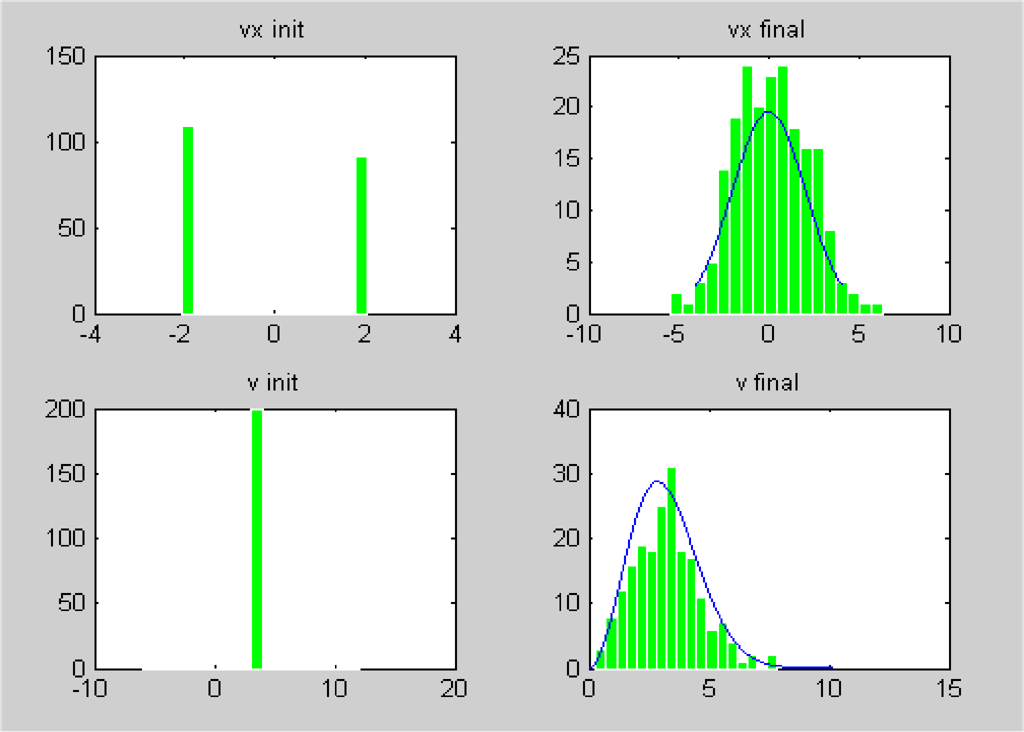
- Fortran – good for calculation-intensive programs, the standard for numerical computing, lots of mathematical libraries, parallel processing.
- C/C++ - apparently not as fast as Fortran, but depends on compiler used.
- Java – can actually compete with Fortran (at least IBM's Java compiler vs. the free g77 compiler).
- MatLab – simple language, interactive, good for prototyping, quick plotting.

Matlab was chosen for its simplicity, interactivity, and graphics capabilities. It is a good platform for developing a prototype in, though for speed the program would eventually need to be adapted to one of the other languages (Fortran, C++, or Java).

Internally, the program uses units of Angstroms, Picoseconds, and Daltons as its fundamental units, and others are calculated in terms of these (e.g. force is DA/ps^2).

Velocity Distribution Experiment

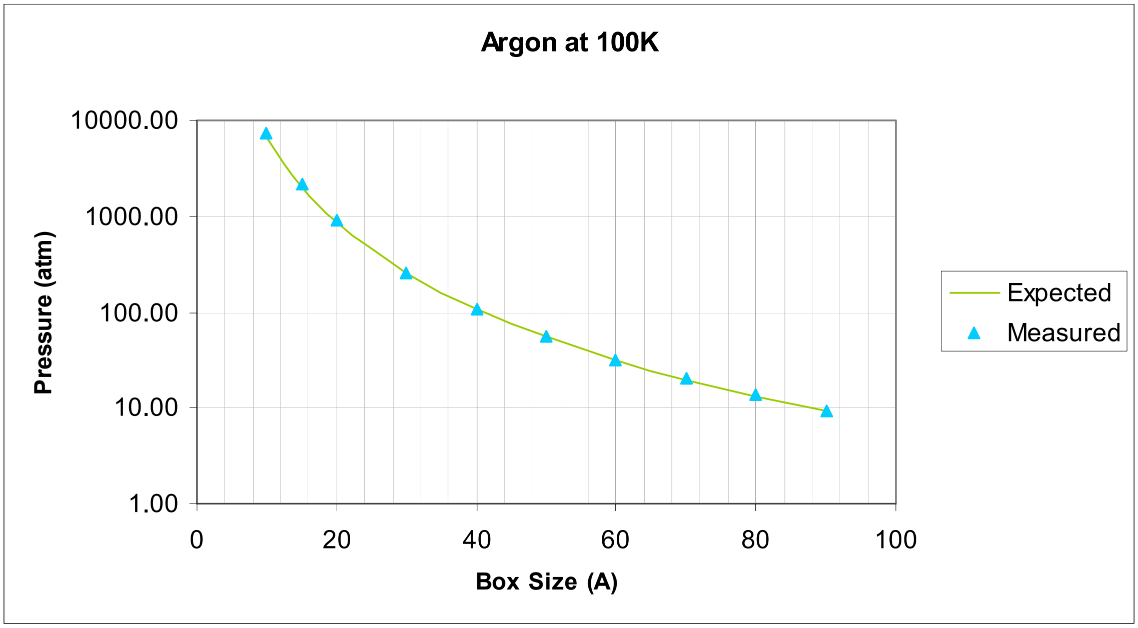
Starting with 200 particles randomly distributed in the box and having the same speed (but different directions), the simulation is run for 10 ps. The particles are treated as hard spheres (i.e. an ideal gas). The velocities along one dimension approach the Gaussian (normal) distribution, as expected by kinetic theory, and the net speeds approach the Maxwell-Boltzmann distribution (see figure below).



Ideal Gas Law Experiment

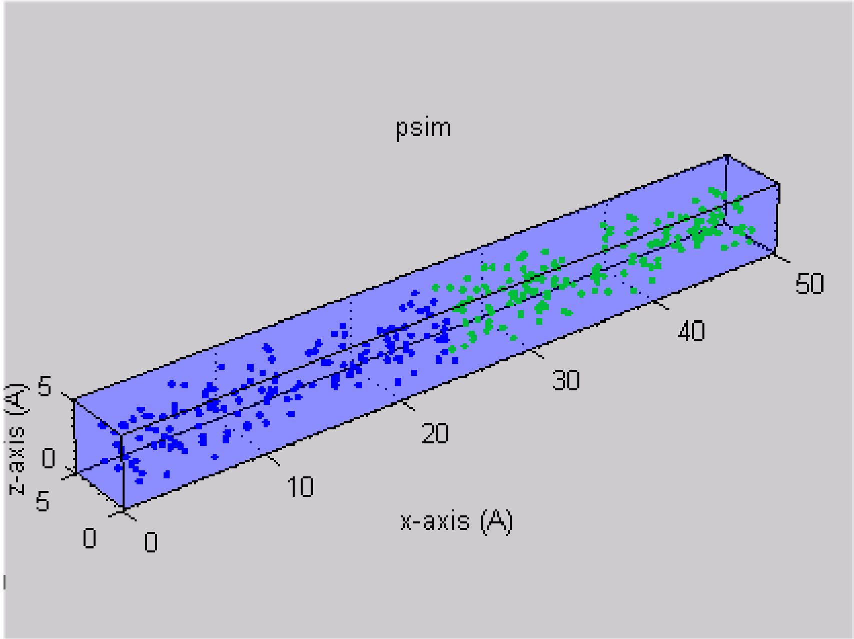
The pressure of the sample is obtained by collecting the change in momentum of the particles as they bounce off the walls, and dividing this by the total box area and the time interval. To verify that this aspect of the simulation is working correctly, we compare the measured pressure with that expected by the ideal gas law.

The size of the box is varied between 10 and 100 Angstroms on a side, and the measured pressure is plotted against P = NkT/V, where N = number of particles, k = Boltzmann's constant, T = temperature in Kelvins, V = volume. Particles are again treated as hard spheres.



Heat Flow Experiment

In this experiment, we start with particles in half of the box at 300K, and particles in the other half at 0K, and run for 10 ps. Particles are treated as hard spheres. Total number of particles is 300.



We get the temperature as a function of the x-axis, then fit the observed data to the solution of the heat equation pde:

$$u_t = k\, u_{xx}$$

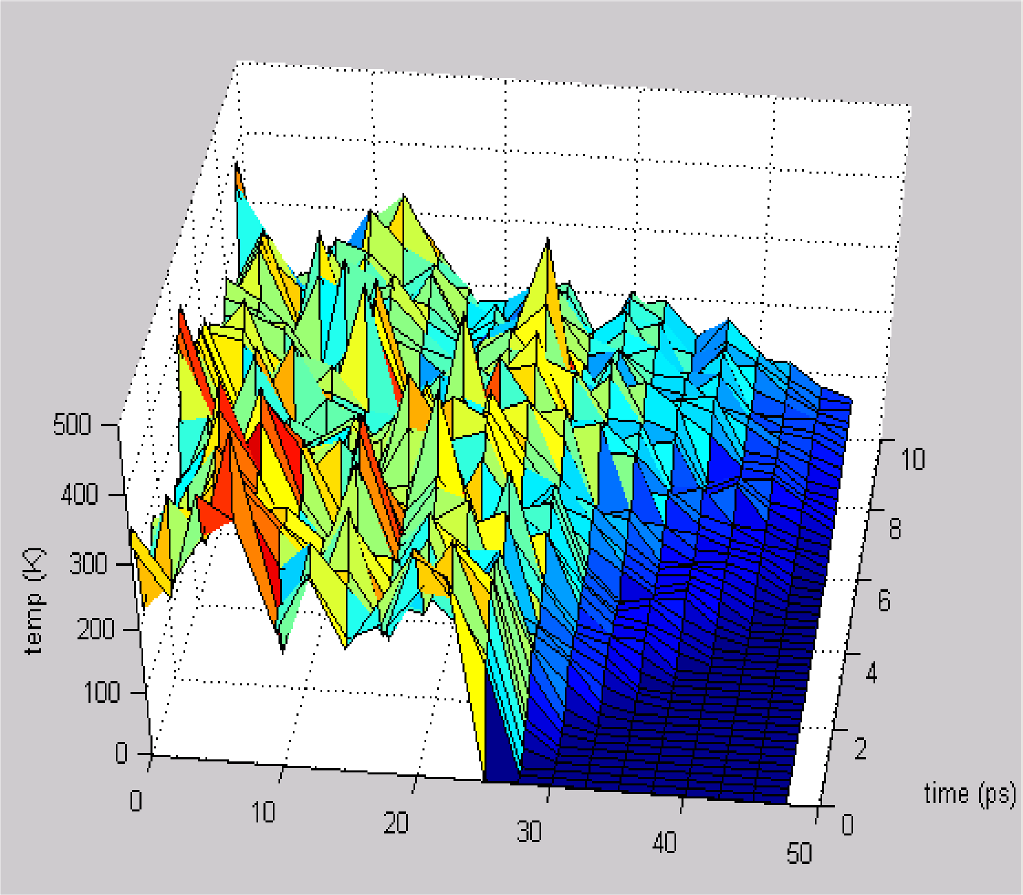
with **k = thermal diffusivity** = K_0 / cρ = 4.27e-5 cm^2/s, and

K_0 = thermal conductivity of Argon = 0.01772 W/m/K,
c = specific heat of Argon = 520 J/kg/K @ 300K,
ρ = mass density = nparticles * mass / volume = 7970 kg/m^3

Treating the box as infinite, the solution to the heat pde is

u(x,t,k) = T/2 * [1 + erf(x / sqrt(4kt))]

with T being the initial temperature on the hot side.



Minimizing the error between this ideal solution and the actual data (shown in figure above), we get a value for **k, the thermal diffusivity**:

8.1e-3 cm^2/s

which compares with the actual value of

4.3e-5 cm^2/s

The most notable source of error in this experiment is due to the convection, which is not included in the pde. Convection is the flow of heat carried by particles from one region to another. In this experiment, the hot particles diffuse into the cold particles, bringing their heat with them, while the pde only models the transfer of heat due to conduction, i.e. the transmission of heat vibrations only (for example as in a solid, where the particles are at fixed locations). A more accurate determination of k should include the diffusion of the particles as well.

Real Gas Experiment

Instead of treating the particles as hard spheres, we assign a Lennard-Jones potential between each pair, and determine the force as the negative gradient of this potential.

The Lennard-Jones potential between two atoms is given by

$$U(r) = 4\varepsilon[(\sigma/r)^{12}-(\sigma/r)^{6}]$$

where ε and σ are constants determined experimentally for a given atom. For Argon, these are

ε = 1.65e-21 Joules
σ = 3.4 Angstroms

We assume the pair potential is valid for multiple particles also, which is a good approximation.

Running the simulation at low temperatures, we can observe the particles clumping together as a liquid, with the particles essentially falling into each other's potential wells.

Other possible experiments

Time didn't allow for a full range of experiments to be performed, but some ideas include:

- **Diffusion** –measure concentrations and get diffusion coefficient D
- **Brownian Motion** – like diffusion for macroscale particles (1μm)
- **Osmosis** – semi-permeable membrane, should have higher pressure on side with solute.
- **Membrane Potential** – e.g. ions or protons on other side of membrane
- **Real Gas Law** – use Lennard-Jones potential, fit to Real Gas Law (van der Waals equation)
- **Speed of Sound** – measure speed of pressure wave transmission, compare with expected value
- **Joule-Thompson Effect** (cool gas by passing through constriction)
- **Heat Capacities** - allow rotational, vibrational kinetic energy, verify equipartition theory and heat capacities
- **Enthalpy, Entropy, Free Energy** – measure these
- **Water** - treat particles as fixed dipoles, compare properties
- **Liquid-Gas Equilibrium -** include gravity, see if liquid phase develops in bottom of box with gas at top

Diffusion Experiment

A real experiment was also performed:

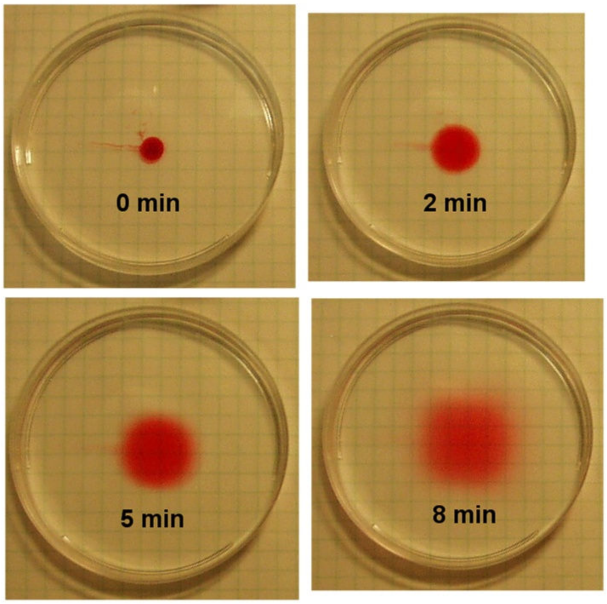
Procedure:

1. Fill petri dish with water that has been sitting out for half an hour and allowed to come to thermal equilibrium with surroundings.
2. Let the water sit in the petri dish for several minutes to allow turbulence to die down.
3. With a syringe, slowly inject a small amount of dye on the bottom of the dish, under the water.
4. Take digital pictures every few minutes - the digital time stamp automatically records the time.
5. Analyze pictures to determine approximate radius of dye over time.

From one of Einstein's 1905 papers, the diffusion constant for a small solute molecule in a solvent is given by

$$D = kT / 6\pi\eta a$$

Where η is the viscosity of the solvent, and a is the radius of the solute molecule.



The following table shows the results, with the measured diffusion constant obtained by $D = \lambda^2 / 2t$, with λ being the displacement.

Dye	**Approx. Radius (A)**	**D_{theo} (m^2/s)**	**D_{meas} (m^2/s)**
Blue	8	6.17e-10	5.87e-8
Green	8	6.17e-10	4.89e-8
Red	4	1.23e-09	1.52e-7
Liquid bluing	2	2.47e-09	2.84e-8

Note that the measured diffusion constant is higher than that predicted by theory - this may be due to the effect of the dye sinking and causing the dye to spread out farther than due to diffusion alone. A more detailed mathematical analysis would be necessary to confirm this.

Conclusion

This lab has just scratched the surface of MD simulations, which is a large topic in itself. Any future MD simulations should probably start with one of the existing simulations, which are freely available from the internet and include various integration methods and potential functions.

The goal of a universal nanoscale lab is a ways away yet – there are currently dozens of potential functions and different ways of modeling molecules and calculating interactions, but as computing power increases, perhaps some standards will begin to emerge which will allow setting up and easily simulating and calculating the properties of different nanoscale regions of interest. Doing so currently requires learning about and understanding MD simulations in some detail.

References

[1] J. M. Thijssen, *Computational Physics*, Cambridge University Press, Cambridge, 1999.
[2] E. Ermakova, 'Argon in condensed phase: Quantitative calculations of structural, thermodynamic, and transport properties from pure theory', *Journal of Chemical Physics* v102 n12, p 4942.
[3] P. Knoll and S. Mirzaei, Development of an interactive molecular dynamics simulation software package, *Review of Scientific Instruments* v74 n4, p. 2483.