

3.320 Atomistic Modeling of Materials Spring 2005

Problem set 4: Molecular Dynamics

Solution Set

Due date: 4/21/2005

In this problem set, we will examine some of the common features of molecular dynamics calculations (MD). First, we will examine the key features of the different integrators which form the basis of any MD engine. Our first exercise will focus on timestep selection as well as the stability and accuracy of a variety of different integration methods.

In this problem set, we will also look at Kr using MD with a Lennard-Jones energy model. In particular, we will be studying the melting behavior of Kr.

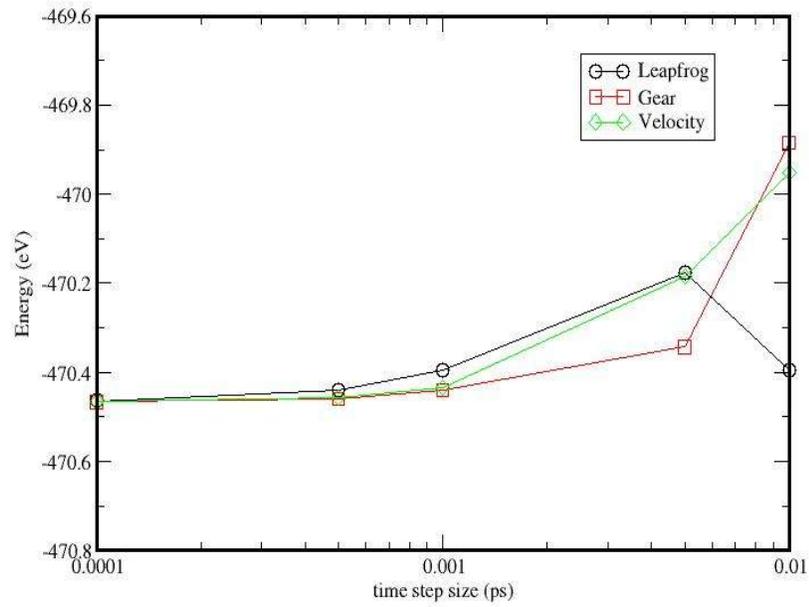
In addition, we will discuss the scalability of simulations with system size. We will also look at the fluctuation of temperature in a MD and how it changes with the system size.

Problem 1. Newton's equations of motion are simple, right? (20 pts)

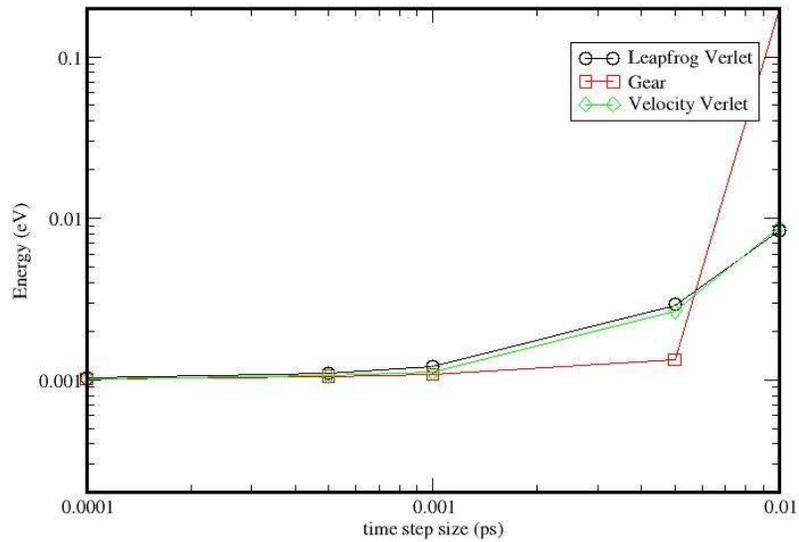
Newton's equations of motion are at the core of any classical MD calculation. While different ensembles may use different dynamics, all MD calculations require accurate and stable integrators in order to solve the equations of motion. In class, we discussed some Verlet algorithms and briefly mentioned some predictor-corrector algorithms. In this exercise, we will examine the accuracy and stability of different integrators as they apply to an EAM model of Ni.

A.) Using GULP's MD capability, examine the average total energy as a function of timestep size using the leapfrog Verlet integration method. Do all of these calculations at 300K. Plot your results. (2 pts)

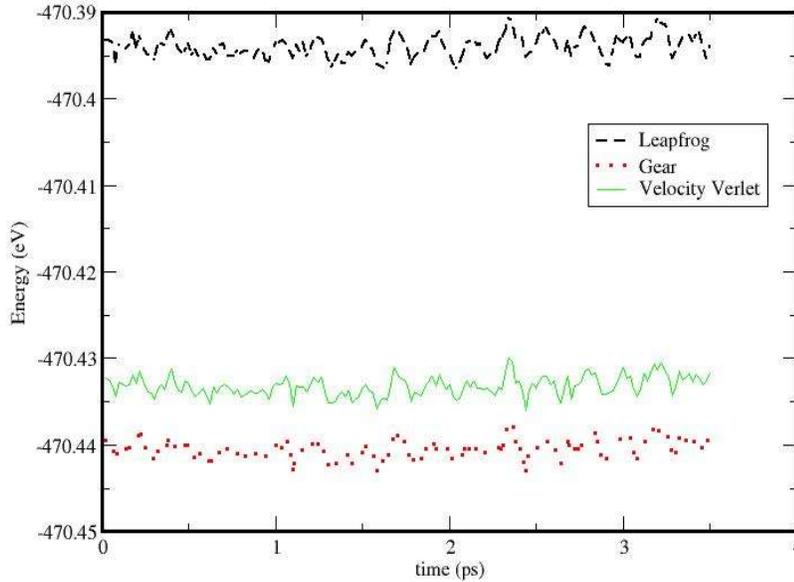
B.) On the same plot, show the average total energy as a function of timestep size using both the velocity Verlet and the Gear Predictor-Corrector algorithms. (3 pts)



C.) Now plot the standard deviation of the instantaneous total energy for the three different algorithms as a function of timestep size. (6 pts)



D.) For a timestep of 0.001 ps, plot the instantaneous energy of the three algorithms. (5 pts)



E.) Comment on your results with respect to which algorithm is most stable, which is most accurate, and which is most computationally expensive. (4 pts)

To examine which algorithm is most stable, we should look at both the deviations in total energy and the instantaneous energy. We see that the Verlet algorithms have roughly the same deviation in energy with respect to timestep size up until the algorithms break. The Gear algorithm actually outperforms the Verlet algorithms for small timestep sizes but gets much worse as the timestep increases. Leapfrog Verlet and velocity Verlet are equivalently stable while the Gear algorithm becomes unstable for large timestep sizes.

In terms of accuracy, the velocity Verlet and the Gear algorithm calculate the similar value of the total energy while the leapfrog Verlet is clearly calculating something different. The discrepancy is due to the fact that the leapfrog Verlet algorithm is always out of phase by $\frac{\delta t}{2}$ when calculating velocities.

In terms of computational expense, one might be surprised to find that all three algorithms are close to equivalent. At first one might think the Gear predictor-corrector would be the slowest algorithm by far since the algorithm requires and calculates more information. However, since the correction step is only applied once

and is based on a single value, the algorithm actually operates with the same complexity as the leapfrog and velocity Verlet. In reality, the leapfrog Verlet and the velocity Verlet are both 3 value predictor-corrector algorithms. The standard Verlet algorithm is actually the fastest but none of the codes we're using have chosen to implement it.

Problem 2. Melting temperature of bulk Kr (60 pts)

In this problem, we find the melting temperature of bulk Krypton by two methods, 1) direct annealing and 2) two-phase simulation method. We will use Lennard-Jones potential in this problem.

A.) Find the melting temperature of solid Kr by performing MD of the crystal at different temperatures. (45 pts) You have to find a suitable timestep for the simulations and system size. What ensemble do you use? Analyze your data with several techniques to determine the melting temperature (i.e. mean-square displacement, etc.) You should also mention other ways of determining melting which are conceptually easy but difficult without writing new analysis routines. Make sure to record all of your simulation parameters (such as time step, cell size, equilibration time, sampling time, ensemble used, and so on). Would this method overestimate or underestimate the melting temperature? Why? (45 pts)

After testing several timesteps in the range of 0.001 picoseconds to 1.0 picoseconds, we determine that timestep sizes greater than 0.01 ps are not stable and result in particle collisions. Energy drift with a timestep of 0.01ps is minimal.

Method 1: Direct visualization of average atomic coordinates.

This method using GDIS is not very reliable but does show when gross changes in structure occur.

The following pictures are from a 4x4x4 supercell under the NPE ensemble. These calculations used a total of 5000 MD steps. 1500 equilibration and 3500 production.

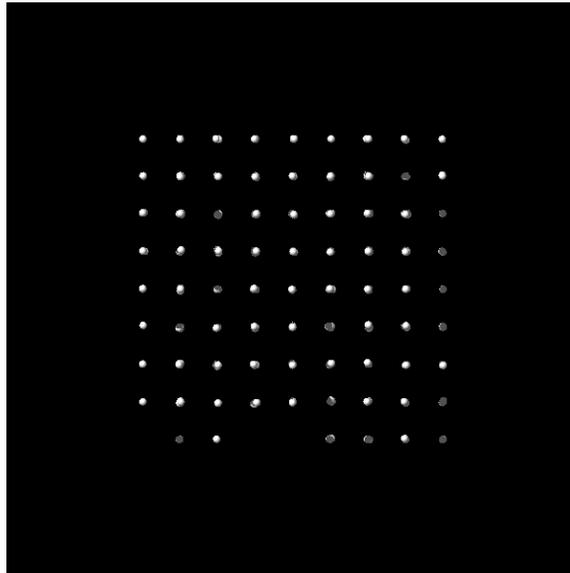


Figure 1. Kr at 10K, still very much solid.

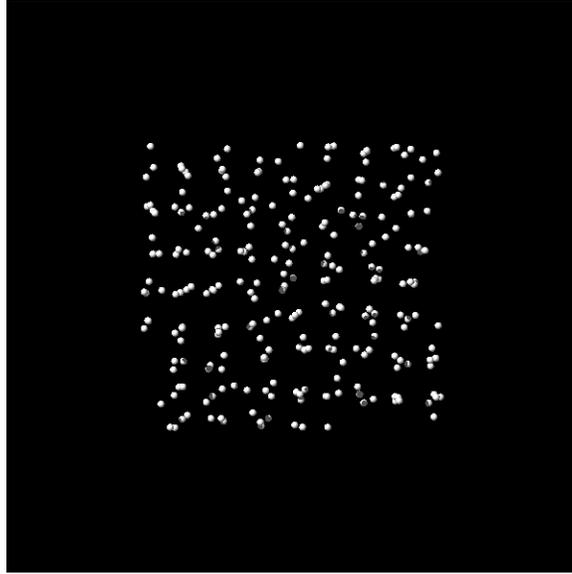
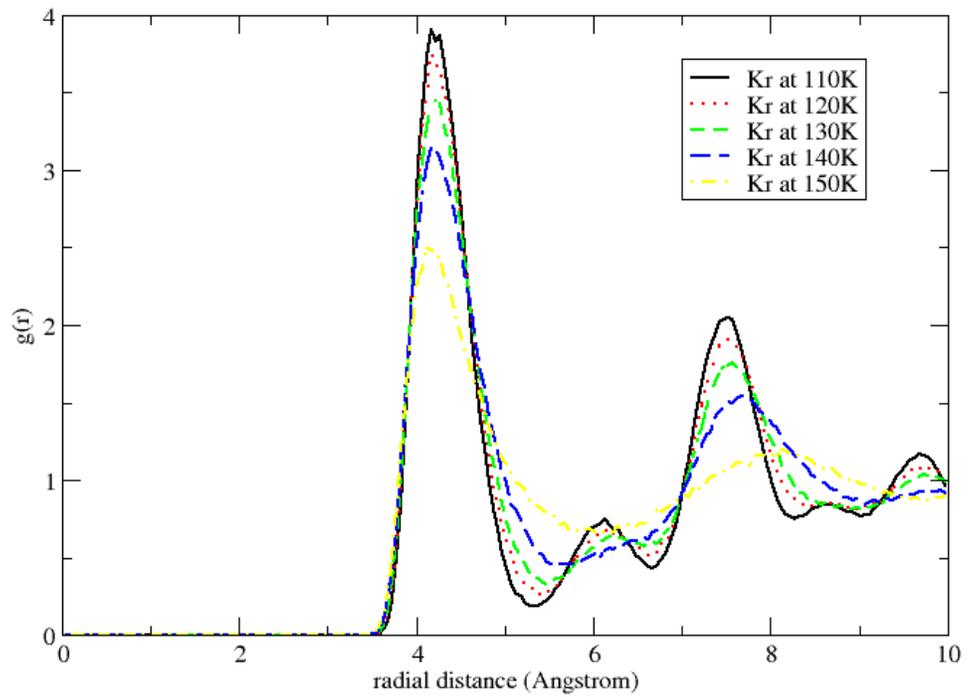


Figure 3. Kr at 140K. One might argue that it is still mostly solid but the beginnings of disorder are apparent.

Using the above mentioned parameters and this method, the melting temperature appears to be in the range of 130-150 K.

Method 2: Radial Distribution function analysis

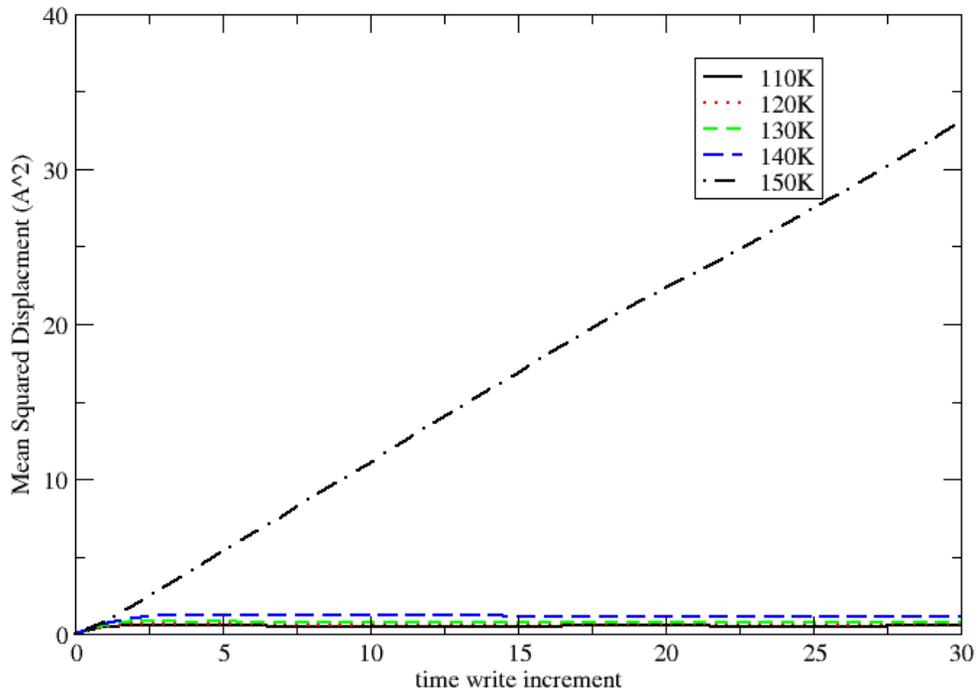
Again, use of this metric to detect melting tends to be a bit subjective. The disappearance of the second nearest neighbor peak is typically taken to be an indication of a liquid structure without long range order.



The second nearest neighbor peak appears to disappear somewhere in the range of 130 to 140K.

Method 3: Mean Squared displacements

Change of state from solid to liquid can be detected through large changes in the diffusivity. From the Einstein relation, we can extract the diffusion coefficient from the mean squared displacements.

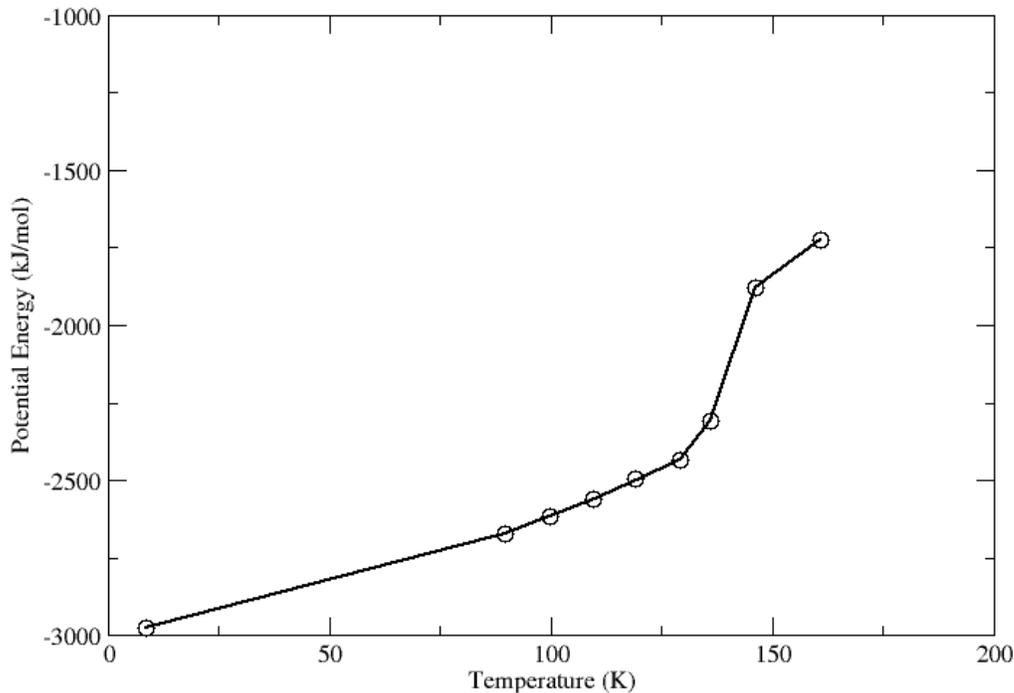


The behavior of the MSD curve clearly goes linear for a target temperature of 150K so the melting temperature is in the range of 140K to 150K.

Method 4: Caloric Curves

The total internal energy as a function of temperature has a discontinuity when the material changes from one state to another. The discontinuity is mainly seen in a plot of the potential energy vs. temperature.

A discontinuity is noted in the range between 140K and 150K.



Method 5: Stability of cell parameters

For constant pressure simulations, the cell vectors will fluctuate. For most materials the volume change upon transformation will be evident. You can extract this information using the moldyext utility, but this method wasn't mentioned in lecture.

Method 6: Bond angle distributions

An atom in an ordered arrangement has a characteristic distribution of bond angles. An order parameter can be constructed which tracks the number of atoms with locally fcc structures vs. the number of atoms with more randomized bond angle distributions. This method would require you to define an order parameter based on the bond angle histogram for each atom. The moldy code does not do this for you but translation and analysis of a restart or dump file is possible.

Method 7: Interface velocity interpolation

This is arguably one of the most accurate methods of calculating the melting temperature. Start with a system which is 50% solid and 50% liquid like. If you run an MD simulation at a temperature higher than melting, the interface will move into the solid phase. If you are at a temperature lower than melting, the interface will move into the liquid phase. The temperature at which the interface velocity is

zero is defined to be the equilibrium melting temperature. This method is mentioned in the web tutorial by Ercolessi which is linked to on the handout.

Method 8: Velocity autocorrelation functions

In much the same way as the mean squared displacements were used to track diffusion, the velocity autocorrelation function serves the same purpose. You can use the mdvaf utility to extract velocity autocorrelation functions.

Method 9: Solidification

Due to finite system sizes, melting is always dependent on a fluctuation which causes the nucleation of a new phase. For heating simulations, we will always overpredict the melting temperature. Similarly, we will always underpredict the melting temperature if we start from high temperature and start cooling. With cooling, we also have the possibility of a 2nd order phase transition to a glass instead of a crystal (much more common than actual nucleation of a crystalline phase).

Method 10: Free Energy calculations

While it was mentioned in the context of Monte Carlo calculations, one could also use a series of long simulation runs to calculate the free energy curves in different phases. In practice, this is a very difficult venture.

General:

Our calculated melting temperatures are generally higher than the experimental value. In most cases, the discrepancy is due to the difficulty in nucleating the disordered phase. Larger system sizes and longer production times will reduce the error in the calculation. Actually with production runs of 45000 after 5000 steps of equilibration with a 7x7x7 supercell, the value of the melting temperature using the NPE ensemble is between 110K and 120K, which was better than I expected. Students who are particularly curious will notice that we should have seen two phase transitions during this computational experiment since the liquid region of the Kr phase diagram is only about 4K wide. Our implementation of the NPE ensemble actually suppresses the liquid-gas phase transition.

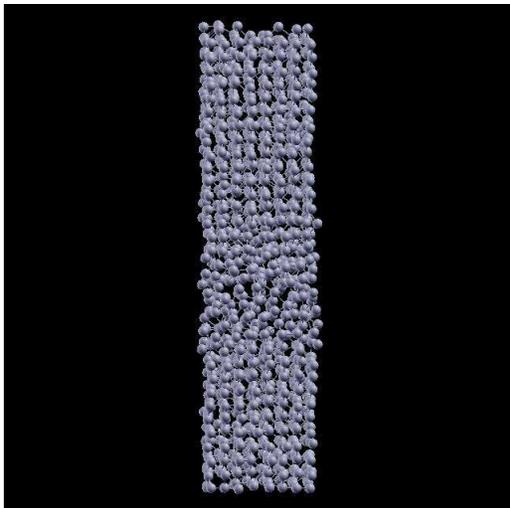
Note: The real purpose of this problem is to help you realize that MD is powerful because you retain all of the atomistic data throughout your calculation. Interpretation and presentation of this data is often as intensive and complex as the calculation itself.

Two phase simulation method: (15 pts)

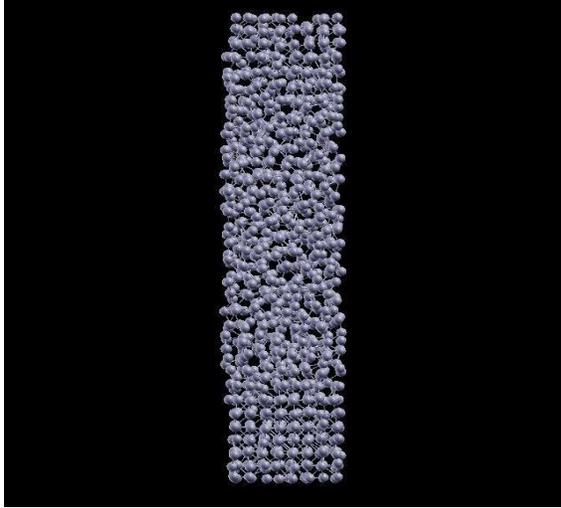
This method was first introduced by F. Ercolessi et. al. Surf. Sci. Lett. 258, L676 (1991). B.) Read the paper by Morris et. al. Phys. Rev. B 49, 5 (1993). Describes the method in the paper. Calculate the melting temperature of Kr using two phase simulation method. (15 pts.)

Following a method similar to that of the paper, it would be useful to test a range of starting temperatures. It is necessary to be in a constant energy ensemble to see the desired effect (the NVE ensemble was used). If your initial temperature is close enough to the melting temperature, out restriction on constant energy will cause the cell temperature to increase or decrease accordingly. This is due to the latent heat of melting/solidification. For the three pictures below, the initial and final temperatures are given, notice that for the initial temperature of 100 K to maintain constant energy the temperature of the cell increased to 119.5 K and similarly for the starting $T=130$ K the temperature had to decrease, thus if we can get the interface to stop within the cell (at equilibrium) we should have a good estimation of the melting temperature. You can see from the final plot that the the temperature converges with time when comparing the two starting temperatures. This should be a better estimation than the direct annealing method because we eliminate the need for nucleation, etc.

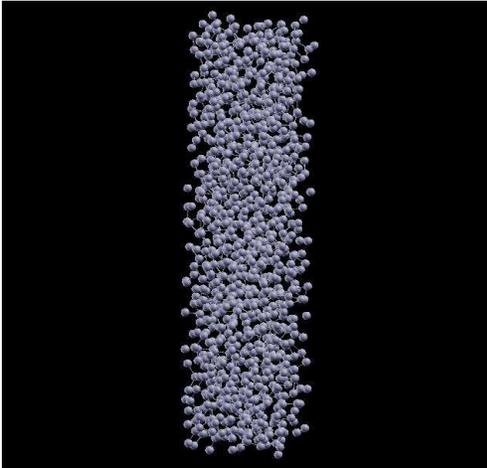
Initial T = 100 K, final T = 119.5 K.



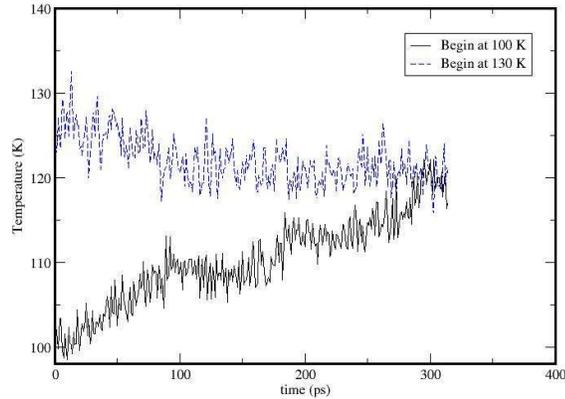
Initial T = 130 K, final T = 120.9 K.



Initial T = 140 K, final T = 138.4 K.



Convergence of Temp vs. time



Problem 3. Scalability of MD Systems (10 pts)

Consider a bulk material simulation using periodic boundary conditions and large supercells.

A.) For an fcc material, how many atoms are contained in a supercell consisting of $N \times N \times N$ unit cells? (1 pts)

There are $4 \times N^3$ atoms.

B.) How many force calculations are required for each timestep if all atoms are considered? (3 pts)

There are $4N^3 \times (4N^3 - 1)$ calculations required if all atoms are considered. Using Newton's Third Law, you can reduce this number of calculations by a factor of 2.

How many force calculations are required if a potential cutoff is implemented?

With a cutoff, there are $4N^3 \times M$ calculations required where M is the number of atoms which fall under the cutoff radius.

How many times must you calculate the distance between atoms?

You still have $4N^3 \times (4N^3 - 1)$ distance calculations. Again, Newton's Third Law allows you to reduce the number of calculations by a factor of 2.

C.) Rank the following force calculations in terms of the CPU time required: Hartree-Fock methods, Lennard-Jones pair potentials, EAM pair functional potentials, and Density Functional Theory. Explain your choices. (4 pts)

From most intensive to least intensive:

Hartree-Fock: This is an all electron calculation. These calculations are very accurate but as the number of electrons in your system increases, the scaling in computational complexity makes this approach impractical.

Density Functional Theory: Better than Hartree-Fock for solids since the effects of the core electrons can be approximated by an appropriate pseudopotential.

EAM pair functionals: Much faster than HF or DFT since we no longer have to self-consistently solve for the electrons.

LJ pair potentials: Even faster than EAM since all bonds are counted linearly.

D.) For simple forms of empirical potentials, most of the computation time is spent calculating the distance between atoms. Can you think of any ways of reducing the amount of time spent calculating distances? (2 pts)

The real solution to this problem is described in the Moldy manual. Aside from the application of Newton's third law which only buys you a factor of 2, the greatest time savings result from better handling of the distances in your simulation. One solution for solids is for each atom to keep a list of possible neighbors such that it only needs to search the distances of the neighbors in its list. This method is commonly referred to as the Verlet Neighborlist method. This method requires additional memory storage and the periodic recalculation of the list still requires $O(N^2)$ operations.

The actual method used by Moldy further breaks up the spatial domain such that the neighborlist is only calculated from a population of subcells in the simulation. Since the connections between the subcells is determined at the very beginning, recalculation of the neighborlist now only requires $O(NM)$ operations.

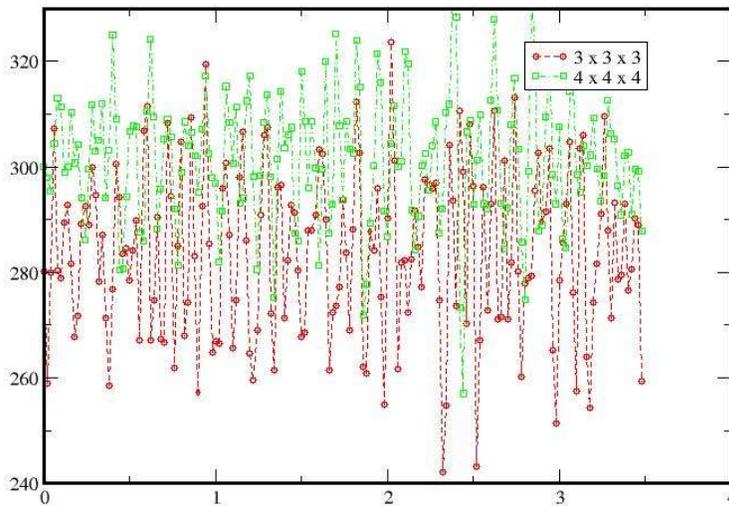
(This may seem like a lot of emphasis for something which is mostly a computer algorithm but these sort of tricks are absolutely necessary if you intend to study systems which are greater than a few 1000 atoms in size. Also, this problem points out that even the largest MD computations are only representative of very modest sizescales and timescales.)

Problem 4. Temperature fluctuation in a MD and how it changes with system size (10 pts)

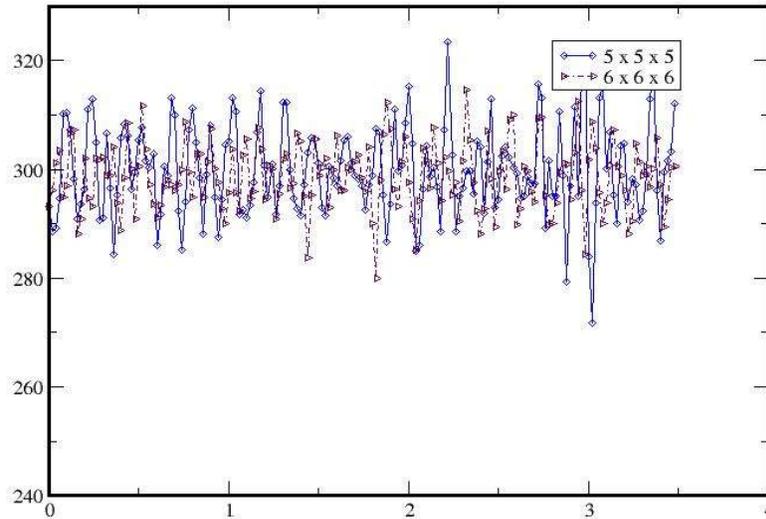
You may notice, when doing problem 1, the production run temperature fluctuates around its average value. In this question, we will explore the change in this fluctuation as we increase the system size.

A.) Go back to the output file in Problem 1 (Velocity verlet method, timestep = 0.001 ps), plot the instantaneous temperature of the production run as a function of time. (2 pts)

B.) Run the MD with bigger system sizes (cell = 4,5,6 etc) and plot the instantaneous temperature on the same graph. Remember to use the same integration method and timestep. (2 pts)



C.) Explain the changes in fluctuation of the temperature with the system size. (3 pts)



According to the data, the the fluctuation gets smaller when the system size increases. The temperature of the system is defined by the average kinetic energy of atoms. In a MD simulation, there is frequent kinetic and potential energy exchange. That's why the temperature fluctuates during the simulation. This fluctuation is more significant when the system is small. In larger system, the temperature change in one part of the system is uncorrelated with another part. When system size increases, the fluctuation in different part of the system cancel each other.

D.)What is the mathematical relation between the standard deviation of the temperature and the system size (ie. Number of atoms)? Verify this with your data in part C. (3 pts)

The relation is: $\sigma \propto 1/\sqrt{N}$.

Consider a speical case of ideal gas of N atoms, the kinetic energy of different atoms in the system is uncorelated. Let the variance of kinetic energy of one atom be

$$\sigma_0^2 = \langle K_0^2 \rangle - \langle K_0 \rangle^2$$

The average total kinetic energy is $\langle K_{total} \rangle = N\langle K_0 \rangle$ and the variance is

$$\sigma_{total}^2 = \langle K_{total}^2 \rangle - \langle K_{total} \rangle^2 = N(\langle K_0^2 \rangle - \langle K_0 \rangle^2)$$

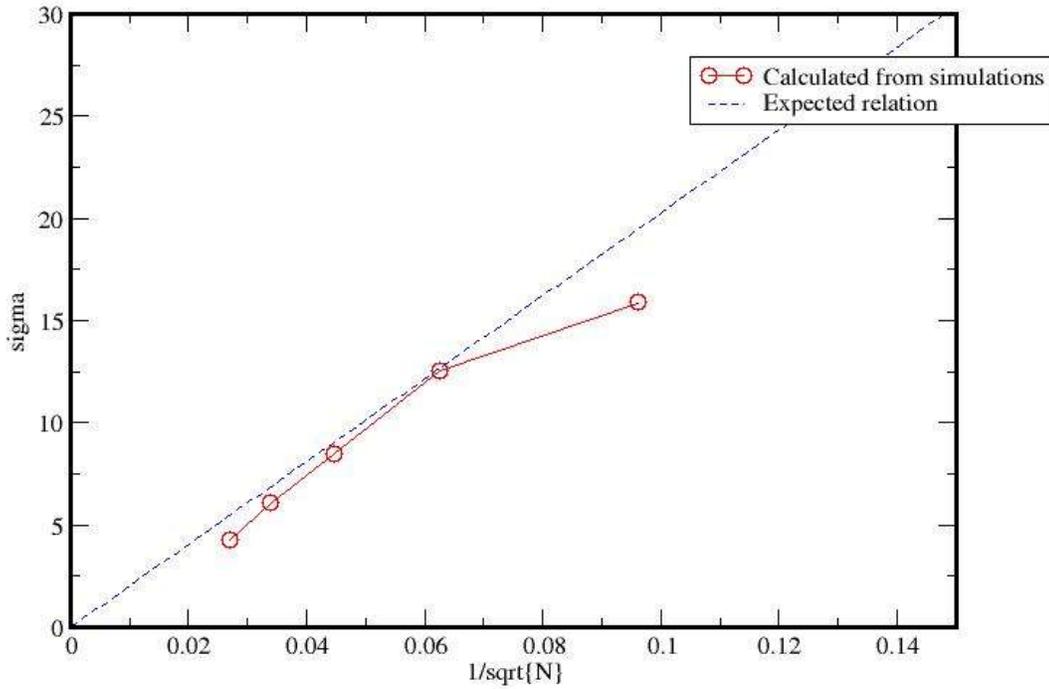
since the kinetic energies are uncorrelated. Therefore, the variance of temperature is calculated from variance of kinetic energy by

$$\sigma_T^2 / \langle T \rangle^2 = \sigma_{total}^2 / \langle K_{total} \rangle^2 = N(\langle K_0^2 \rangle - \langle K_0 \rangle^2) / N^2 \langle K_0 \rangle^2 \propto 1/N$$

Since $\langle T \rangle$ is independent of N , $\sigma_T \propto 1/\sqrt{N}$.

In condensed phase (solid or liquid), we cannot assume the kinetic energies of indivdual atoms are uncorrelated. However, if we consider a large system, we can seperate the large system into N_{cell} cells, each containing M atoms. If M is large

enough, the kinetic energy for each cell are uncorelated (if M is large enough) nad the above argument will still hold.



The data points fall approximately on a straight line, with a deviation for the smallest (3 x 3 x 3) simualtion cell.