

HW3 Yubo Ouyang

Momentum

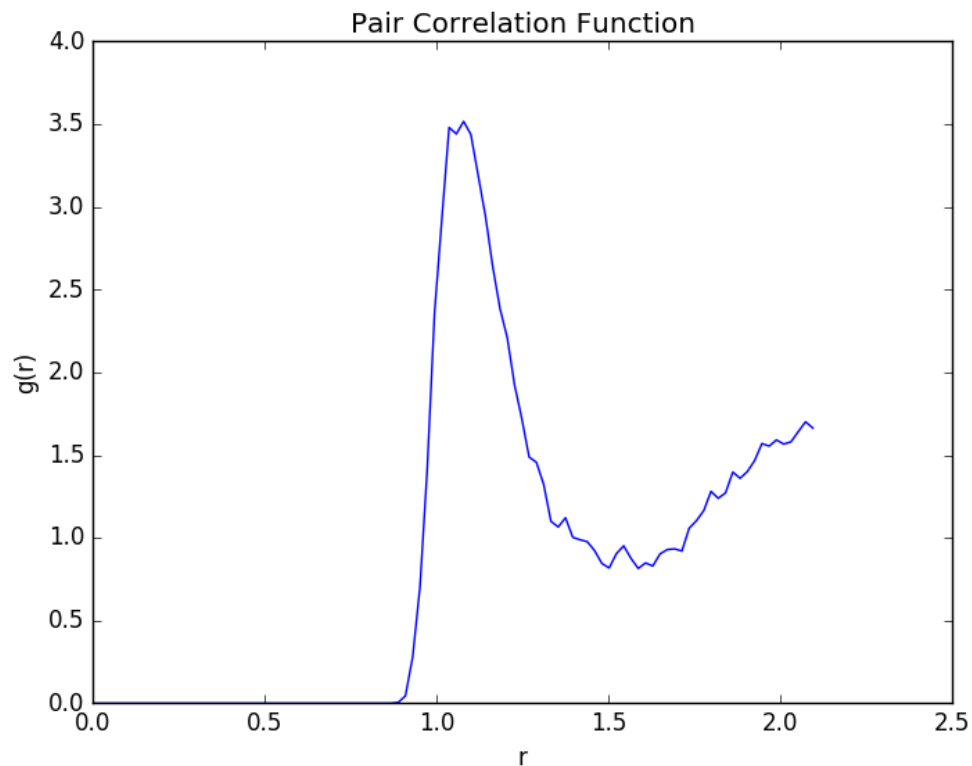
Q: Measure the total momentum of your system as it evolves in time. Is momentum conserved? Do you expect it to be conserved?

A: Yes, it is. It should be conserved (despite little fluctuations), since there is no external force.

Pair Correlation Function

Q: Implement the calculation of $g(r)$ in your code and submit a plot of $g(r)$ for the Lennard-Jones simulation conditions described above. Be sure to normalize your function correctly!

A:



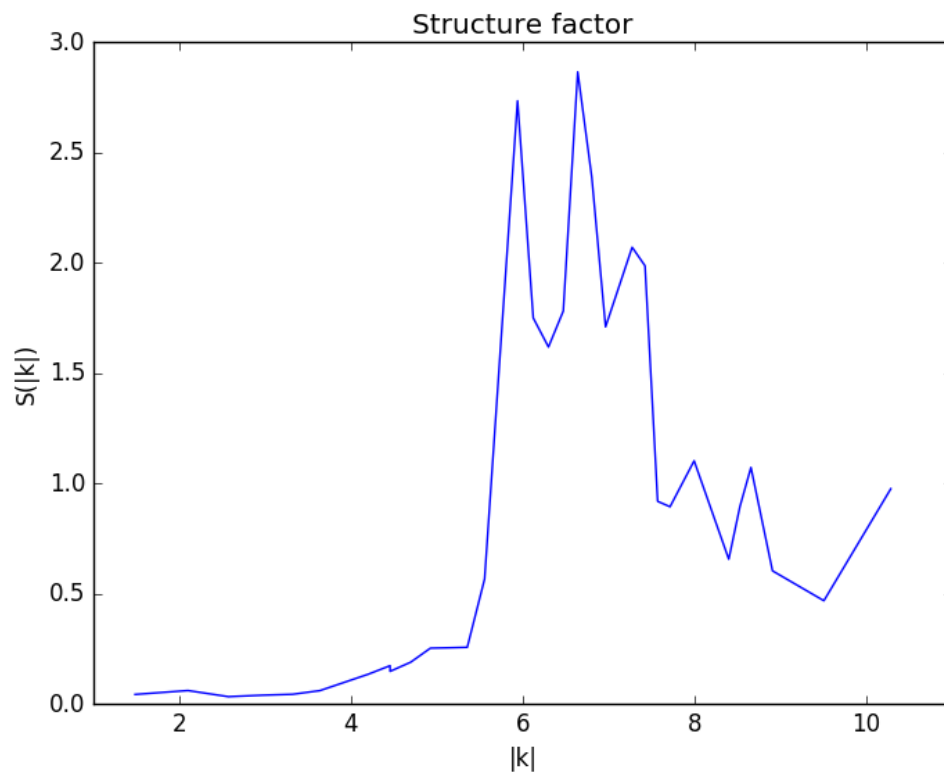
Comment: I do not know why, but all of my $g(r)$ consistently grew after $r = 1.5$ and reached $g(r) = 1.5$ on my computer. The weird thing is that when I ran my script on other's computer, the data was fine ($g(r)$ converged at 1). So I guessed this problem was probably due to my computer setting (my python version is 2.7).

Structure Factor

Q: Implement the calculation of $S(k)$ in your code and submit a plot of $S(k)$ from a run using the Lennard-Jones system parameters described above. Be sure to normalize your function correctly! Describe qualitatively how $S(k)$ and $g(r)$ should look for a liquid and a solid.

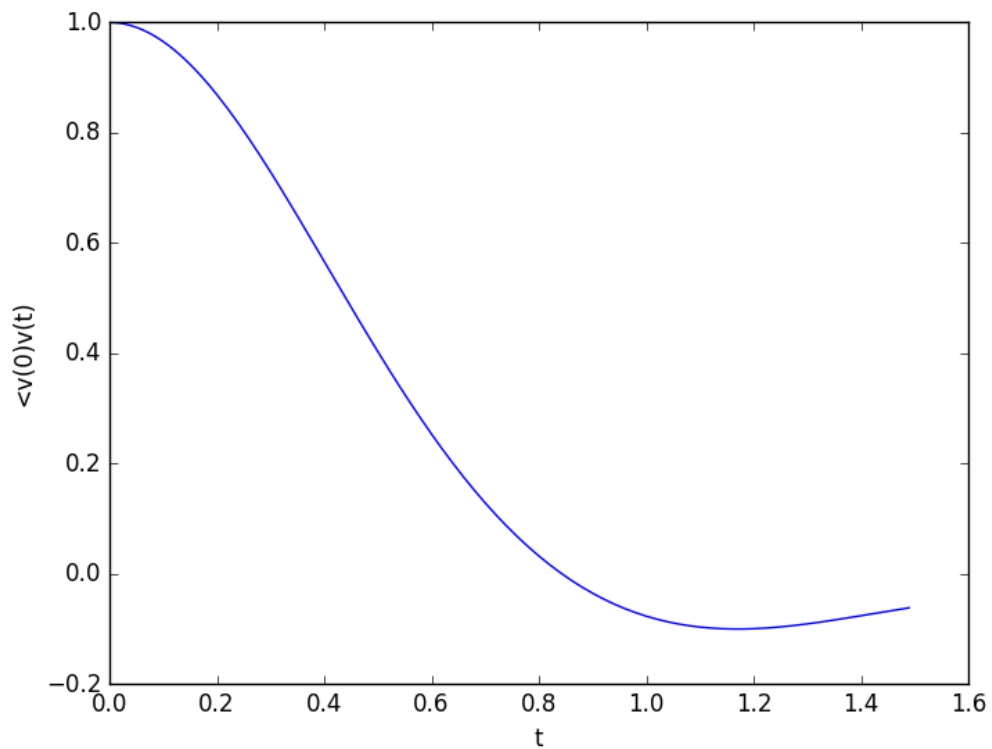
A: In solid state, $S(k)$ will have several very sharp peaks, while $g(r)$ will also have

several sharp peaks before converge to 1. In liquid state, the $S(k)$ peaks will become less sharp, and curve of $g(r)$ will become more flat.



Velocity-Velocity Correlation and Diffusion Constant

Q: . Using the simulation parameters described above, produce a plot of the velocity-velocity correlation as a function of time. Also report your value for the diffusion constant.

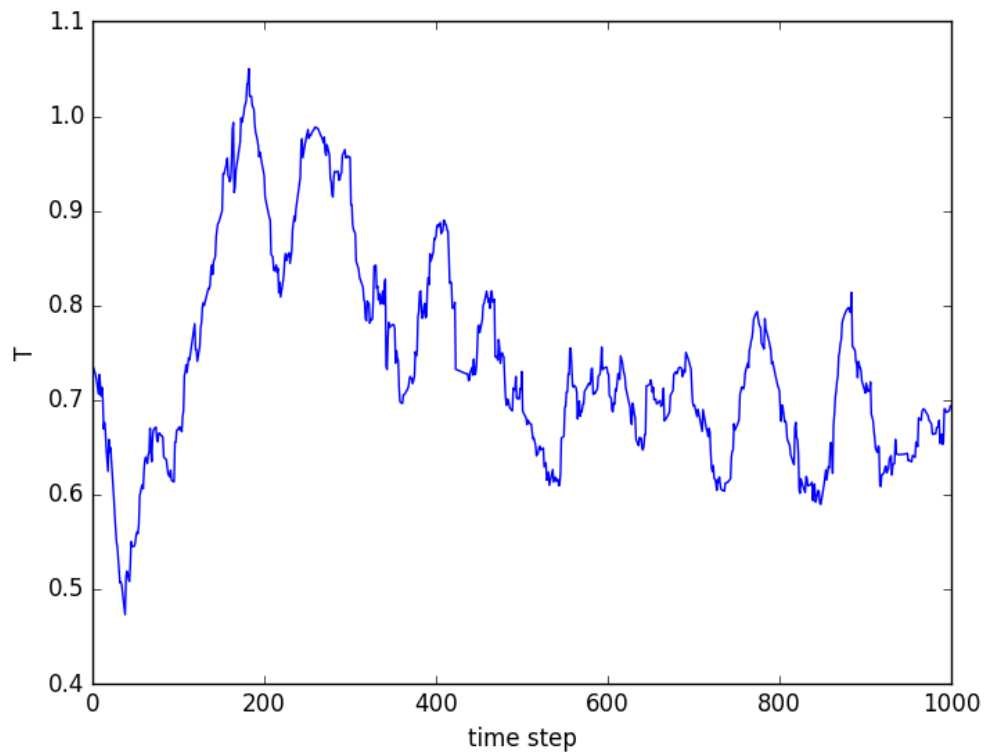


A: Diffusion constant was computed to be `0.399665949323`

Andersen Thermostat

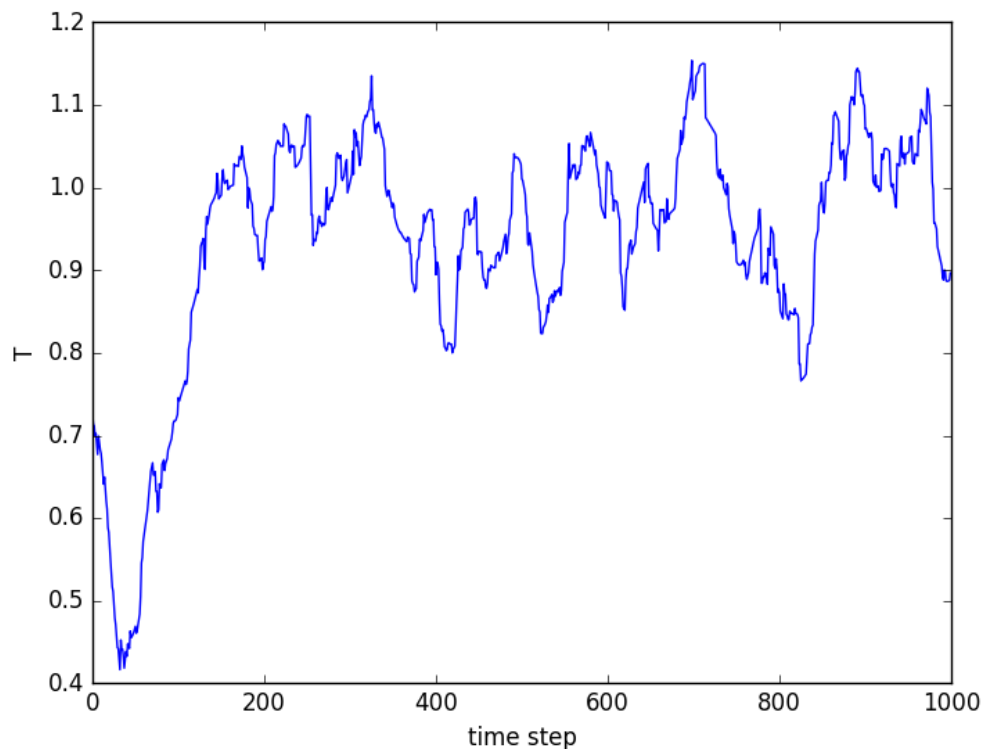
Q1: Implement this new technique in your code. To check that it is working, re-run the Lennard-Jones simulation described above but with your thermostat, and produce a plot of the instantaneous temperature as a function of simulation time.

A:



Q2: Now, increase the temperature to $T=1.0$, but still pass the value 0.728 as an argument to the "InitVelocity" function. Re-run and produce a plot of temperature vs. time for this simulation.

A.



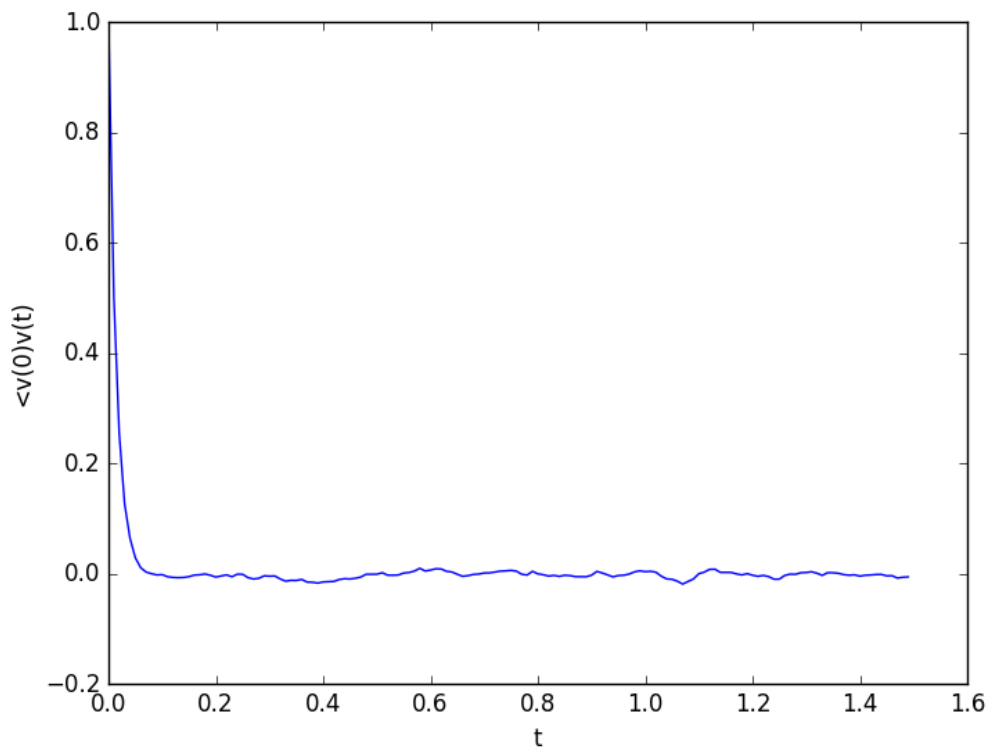
Q3: A limitation of the Andersen algorithm is that it uses randomized velocities, which corrupt the dynamics of the system. Static equilibrium properties are still accurate, but you will no longer be simulating the physical dynamics of your system.

To see this quantitatively, adjust the value of η so that your collision probability is around 50%. Produce a plot of the velocity-velocity correlation function with thermostating on and report your calculation of the diffusion constant. Compare these results with your data from the microcanonical simulation (above). Specifically, you should discuss what differences you expect to see in these observables and whether or not you do, in fact, see them.

A:

The diffusion constant here was **0.0166055482705**

And plot of vv-corr function is shown below:



This actually makes sense. Since Andersen thermostat randomly changed atoms' velocity, velocity from different steps would not have much correlation with each other. Thus, diffusion constant as an integral of vv-corr function will also be very small.

Calculating phase transitions

Q: Submit your estimate of the transition temperature along with data that support this claim. Minimally, you should include graphs of the pair correlation function and structure factor for temperatures above and below your estimated transition temperature.

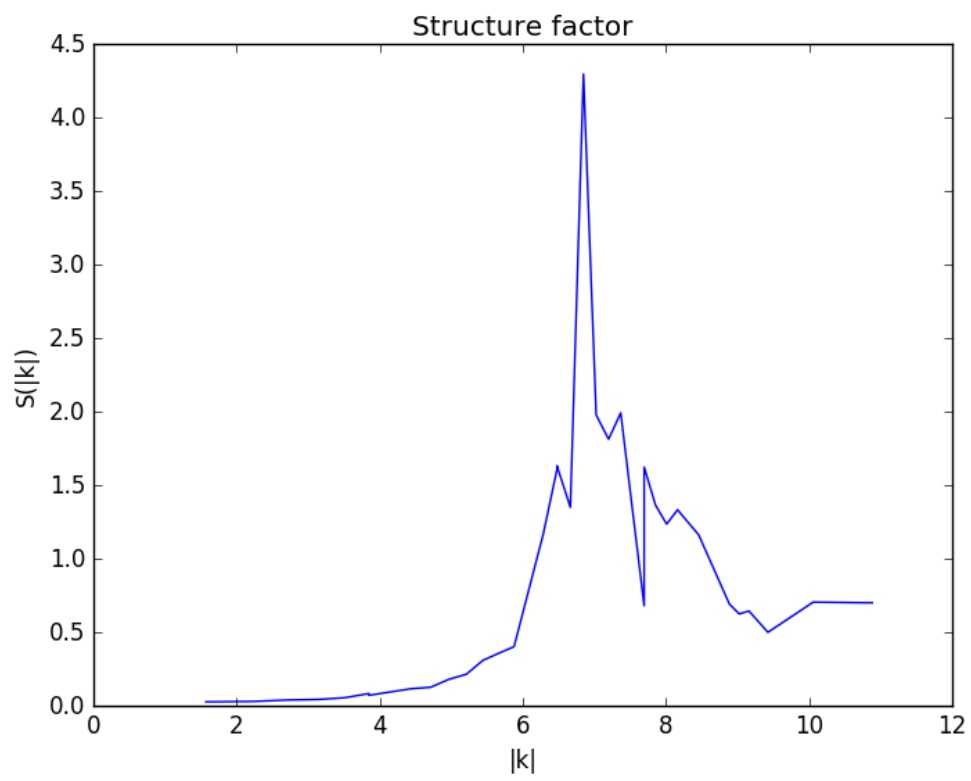
A:

Estimated transition temperature $T = 2.4$ (reduced units)

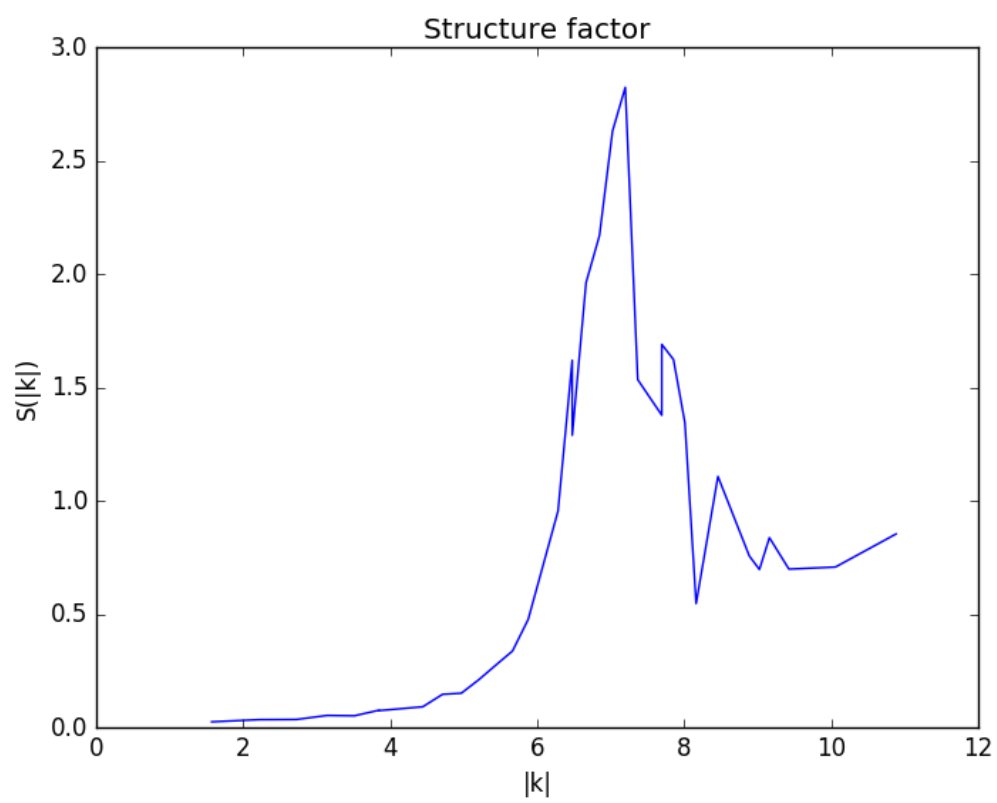
Comparing $S(k)$ from $T = 2.3$ to $T = 2.5$, it was clearly shown that peak of $S(k)$ at about $k = 7$ had a huge drop. For $T = 2.3$ and 2.5 , the value was about 4.25, 2.5, and value of $T = 2.4$ was in between. So transition temperature was estimated as 2.4. To make sure $T = 2.5$ was at the end of this huge drop, I also included $S(k)$ at $T = 2.7$. $g(r)$ plot was also included

Supported data:

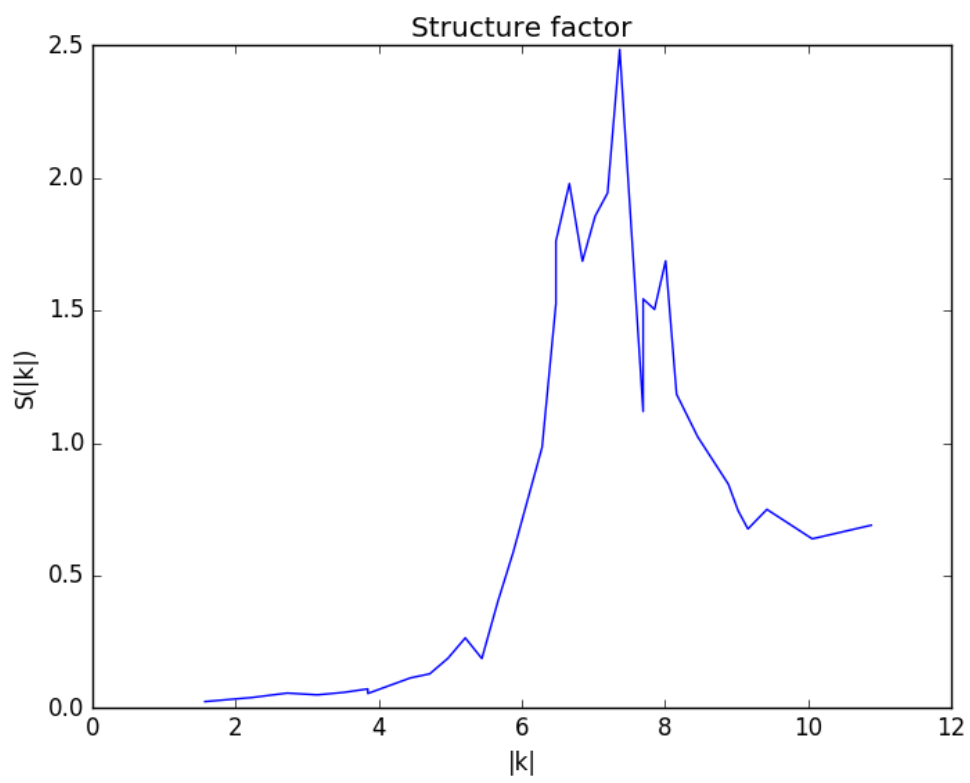
1. $S(k)$



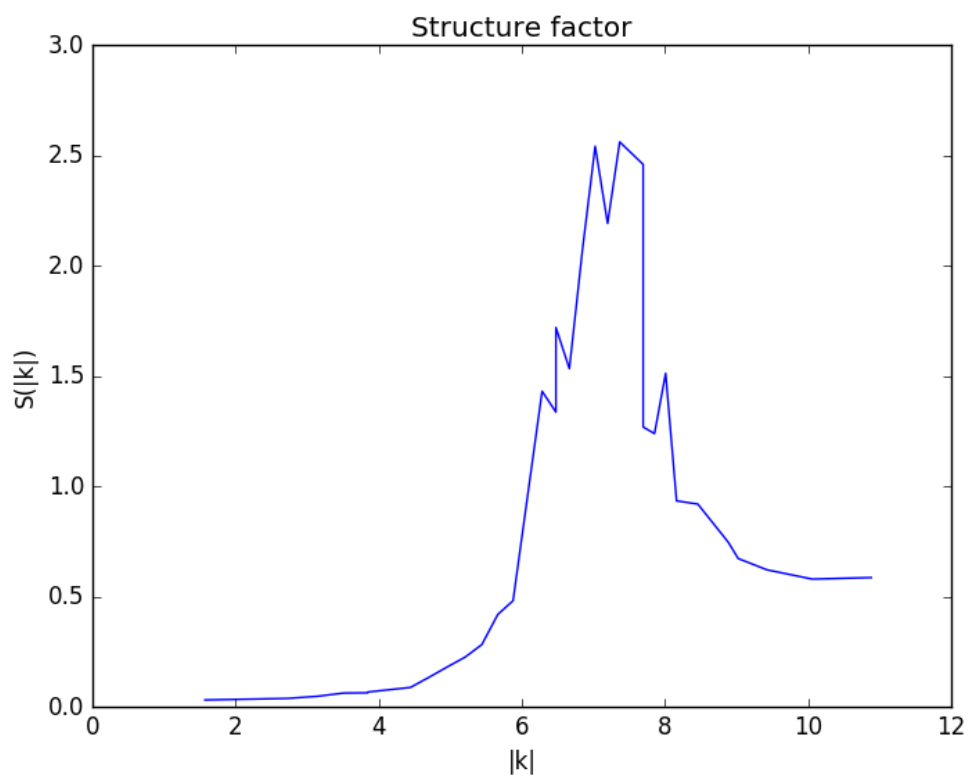
(1) $T = 2.3$



(2) $T = 2.4$

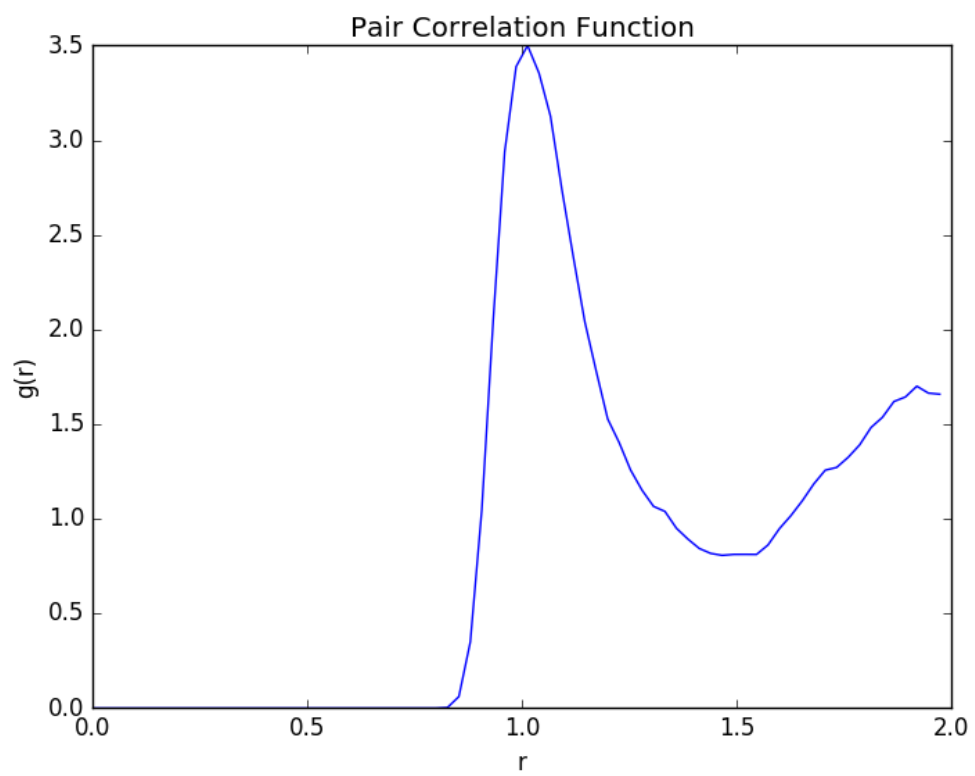


(3) $T = 2.5$

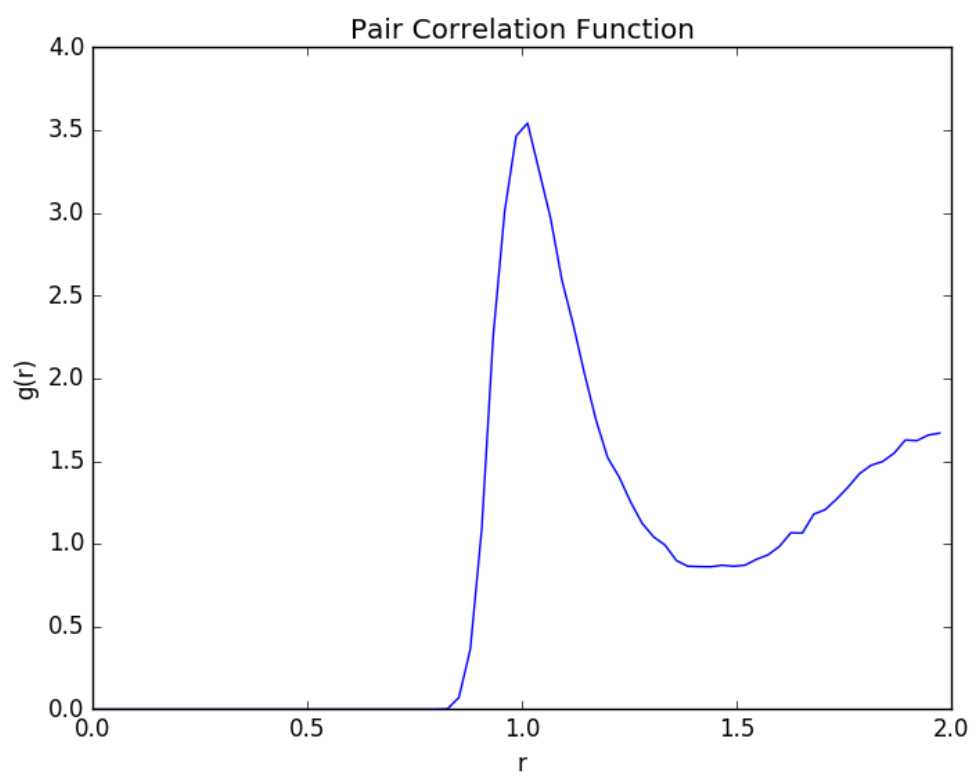


(4) $T = 2.7$

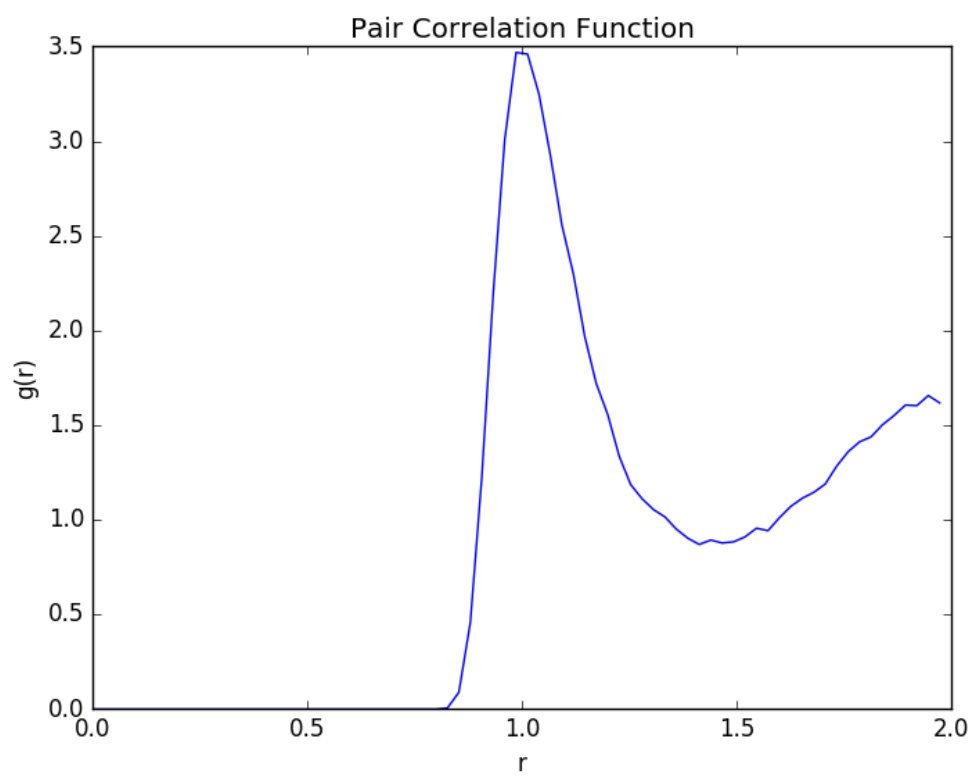
(2) $g(r)$:



(1) $T = 2.3$



(2) $T = 2.4$



(3) $T = 2.5$

