

Writing a Monte Carlo code

Potential Energy

To verify that your function works, calculate the potential energy of the initial configuration and verify that you get -231.570016481.

Answer:

This is snapshot of my first potential energy after initialization. Thus my function works fine.

```
*****
0 -231.570016481
*****
```

Moves

write a function to update the coordinates of a single particle

Answer:

More details please refer to code uploaded on compass.

```
#def of updata_position
def update_position(pset,iat,sigma):
    # should return new position particle iat in pset
    # using the Gaussian distribution described above
    old_ptcl_pos = pset.pos(iat)
    delta_x      = np.random.normal(0.0,sigma,3)
    new_ptcl_pos = old_ptcl_pos + delta_x
    return new_ptcl_pos
# end def update_position
```

Metropolis Acceptance Criterion

1: For the Gaussian displacement move described above, write down expressions for $T(x' \rightarrow x)$ and $T(x \rightarrow x')$.

answer:

we have $x' = x + \eta$ and $x = x' - \eta$

$$T(x - x') = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-(x' - x)^2}{2\sigma^2}\right) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-\eta^2}{2\sigma^2}\right)$$

$$T(x' - x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-(x - x')^2}{2\sigma^2}\right) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-\eta^2}{2\sigma^2}\right)$$

2: Show that, for this move, $T(x \rightarrow x')/T(x' \rightarrow x) = 1$.

Answer:

according to the $T(x \rightarrow x')$ and $T(x' \rightarrow x)$ formulas we have above, it can be seen that $T(x \rightarrow x')/T(x' \rightarrow x) = 1$ is satisfied.

3: Put these results together and write down an explicit expression for $A(x \rightarrow x')$.

Answer:

$$A(x \rightarrow x') = \min\{1, \exp(-\beta(V(x') - V(x)))\} = \min\{1, \exp(-\beta * \Delta V)\}$$

Now that we have the pieces, let's write the Monte Carlo loop

Answer:

The loop can be viewed in code.

Monitoring Acceptance and Setting up Runs

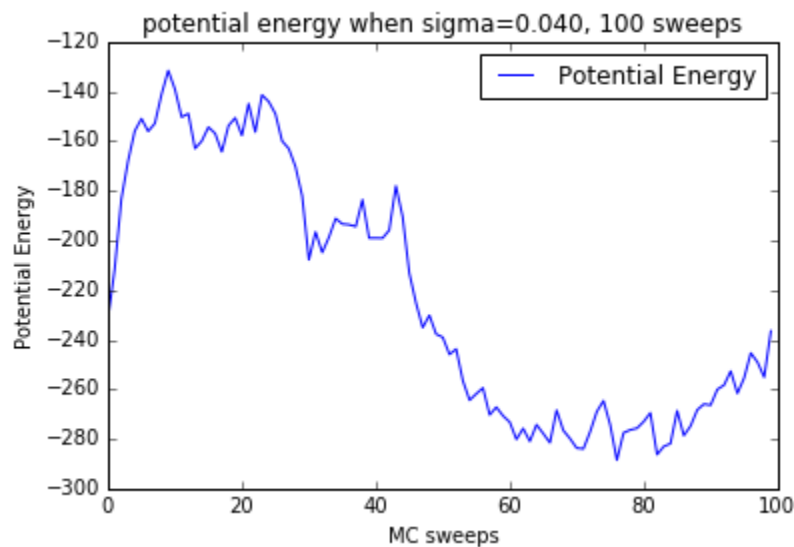
System parameters as below:

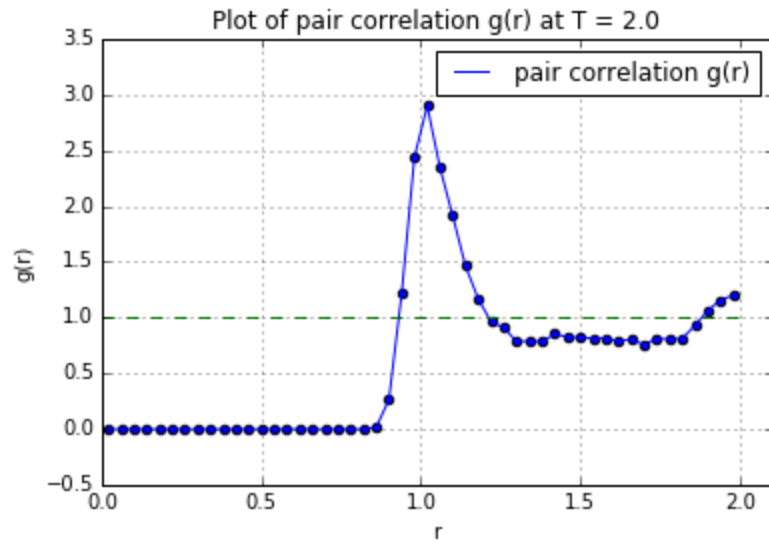
- Density = 1.0 (box length = 4.0)
- particle number = 64
- mass is 48.0
- Simulate the canonical ensemble at temperature $T=2.0$, or $\beta=1/T=0.5$.

1. Find a value of σ that gives you an acceptance ratio between 0.3 and 0.7. An acceptance ratio in this broad range is likely to give you decent sampling efficiency. Quote your acceptance ratio and your choice of σ . Using this value, perform a run of 100 sweeps. Produce a plot of the potential energy as a function of MC sweeps, and a plot of the pair correlation function for this run.

Answer:

Here I chose **sigma = 0.04** and **acceptance ratio = 0.554**. Run 100 sweeps, the potential energy and pair correlation function (average over the last 50 sweeps) are shown as below:

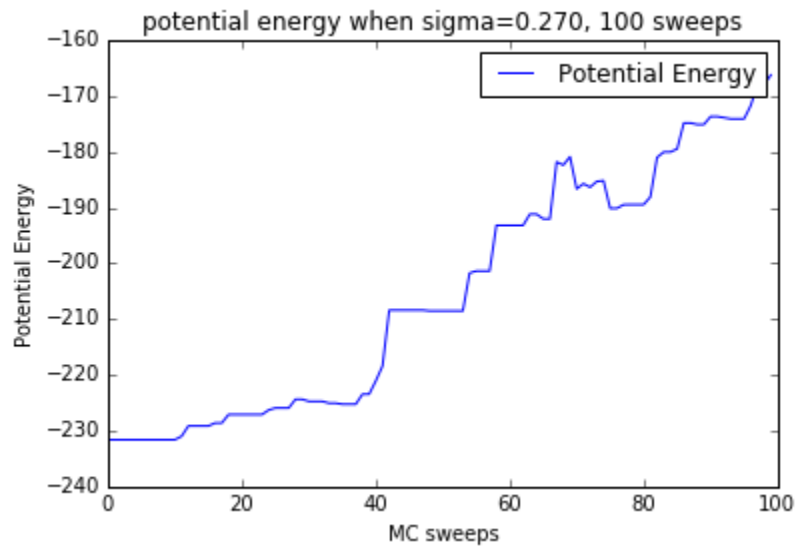


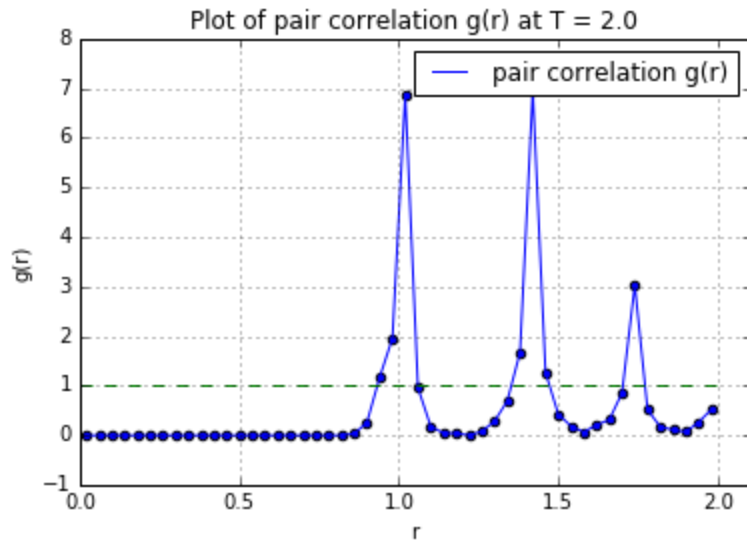


2. Find a value of σ that gives an acceptance ratio below 0.01; state the value of σ you are using. Run your simulation again for 100 sweeps and produce plots of the energy and pair correlation function.

Answer:

Here I chose **sigma = 0.27** and **acceptance ratio = 0.00828**. Run 100 sweeps, the potential energy and pair correlation function (average over the last 50 sweeps) are shown as below:

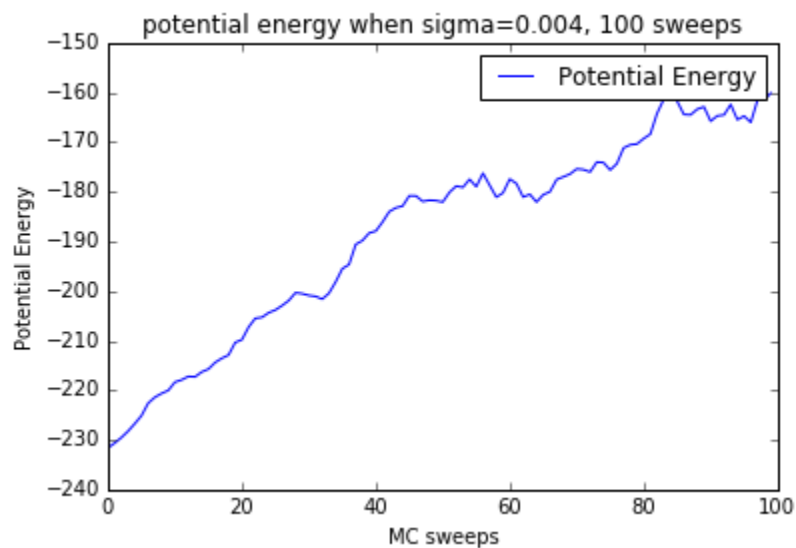


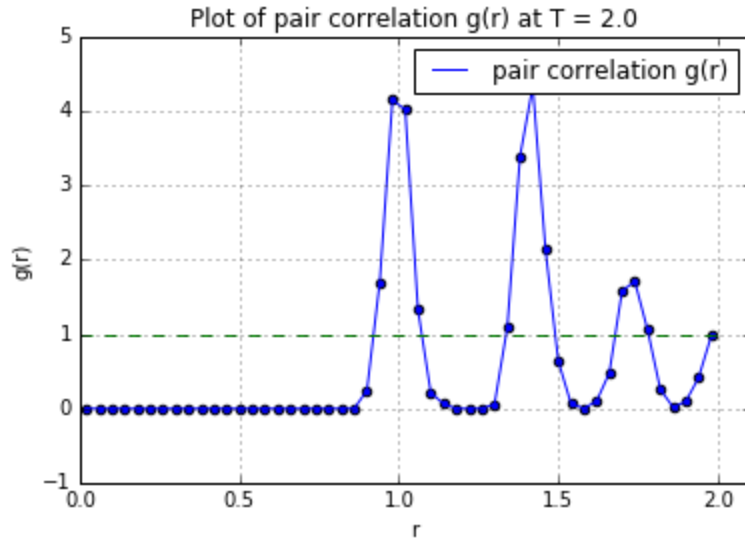


3. Find a value of σ that gives an acceptance ratio greater than 0.95; state the value of σ you are using. Run your simulation again for 100 sweeps and produce plots of the energy and pair correlation function.

Answer:

Here I chose **sigma = 0.004** and **acceptance ratio = 0.96375**. Run 100 sweeps, the potential energy and pair correlation function (average over the last 50 sweeps) are shown as below:





4. Comparing data from your three runs, discuss what happens to your simulation if your acceptance ratio is either effectively 0 or effectively 1. State specifically how this leads to an inefficient simulation in each case.

Answer:

When the acceptance is either nearly 0 or 1, from the curves of the potential energy we can see that the overall trend of potential energy is rising, while for acceptance ratio near 0.554 ($\sigma = 0.04$) the curve is rising first then went down to approach equilibrium. This is because when σ is too small, acceptance ratio nearly 1 but each step movement is still too small, resulting low efficiency. On the other hand, when σ is too big, meaning big movement step, but the acceptance ratio is just too small, again, the efficiency being low. When acceptance ratio is 0.3-0.7, (0.554 in my case), it is a good balance between movement step and acceptance ratio, thus leading to faster speed to equilibrium.

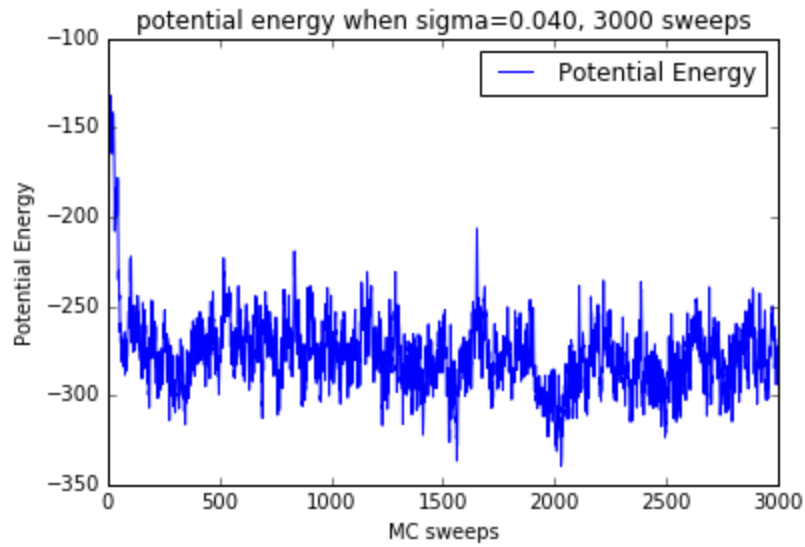
Because of not reaching equilibrium, the $g(r)$ have lots of spikes, needing more time to MC steps forward.

5. Run the above simulation conditions for a minimum of 2000 sweeps.

a. Submit a trace plot of the energy as a function of Monte Carlo "time", i.e. the number of MC sweeps made.

Answer:

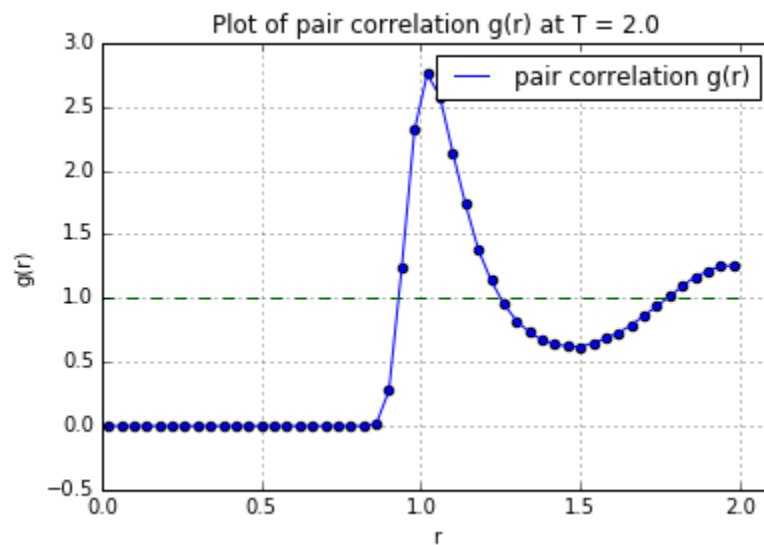
Chose **$\sigma = 0.04$** and run for 3000 sweeps, **accept ratio = 0.587**. The plot of energy is shown as below:

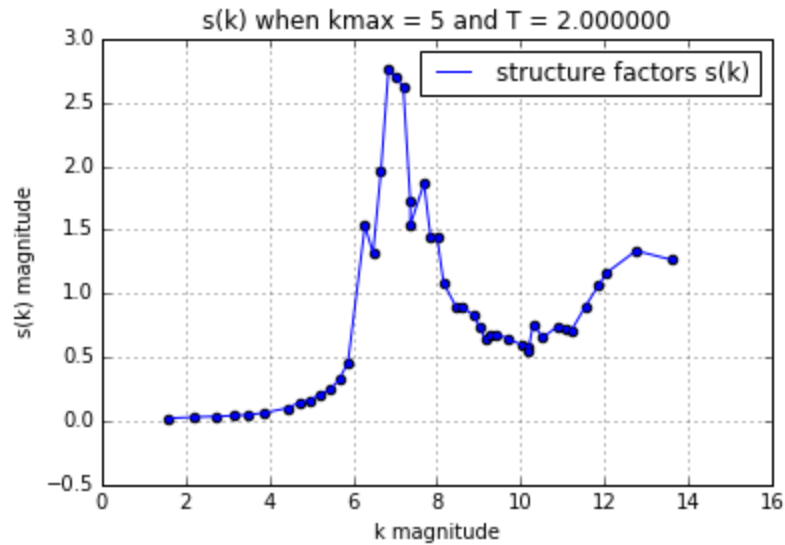


b. State the total number of Monte Carlo steps, the average energy, and the standard error for your simulation.

MC steps	Average potential energy	Standard error of average
2700(300-3000)	-279.07191494	2.47577599

c. Submit plots of the pair correlation function and structure factor.



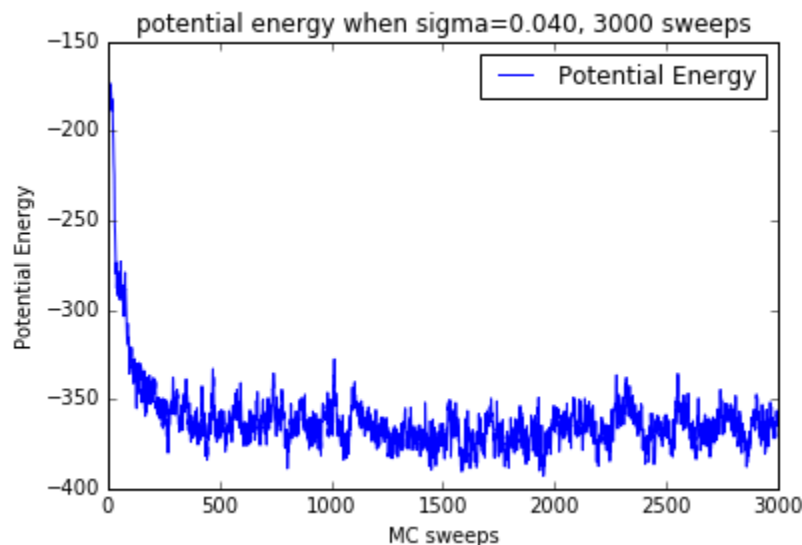


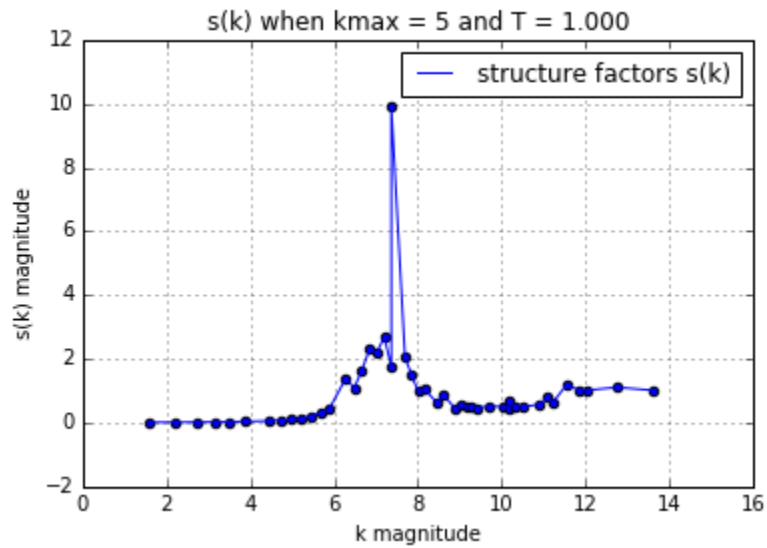
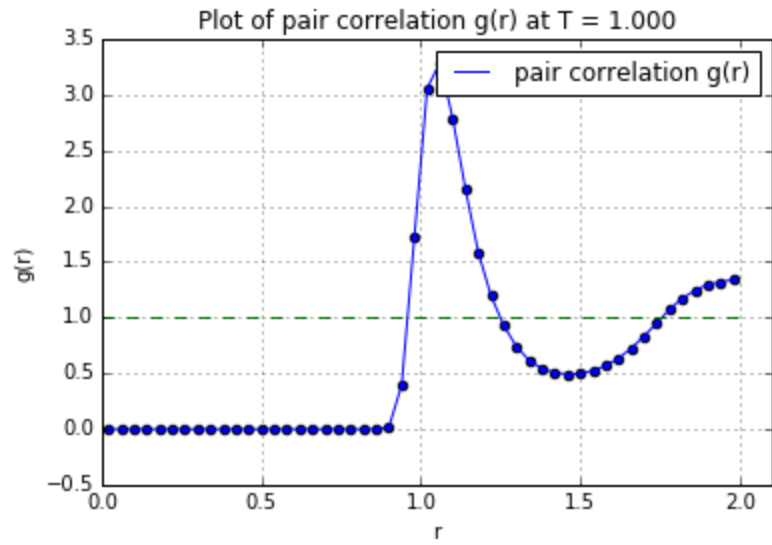
6. Now do two simulations, one at temperature $T=1.0$ and one at a temperature below $T=0.5$. Produce plots of the energy, structure factor, and pair correlation function at each temperature.

Answer:

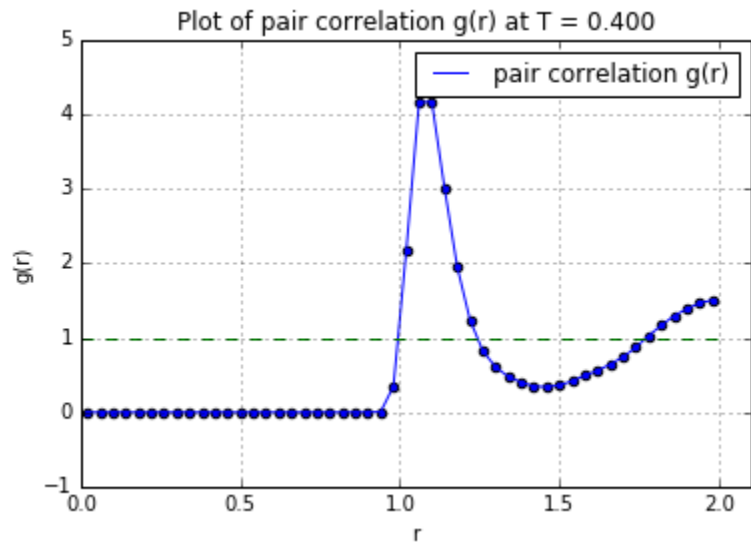
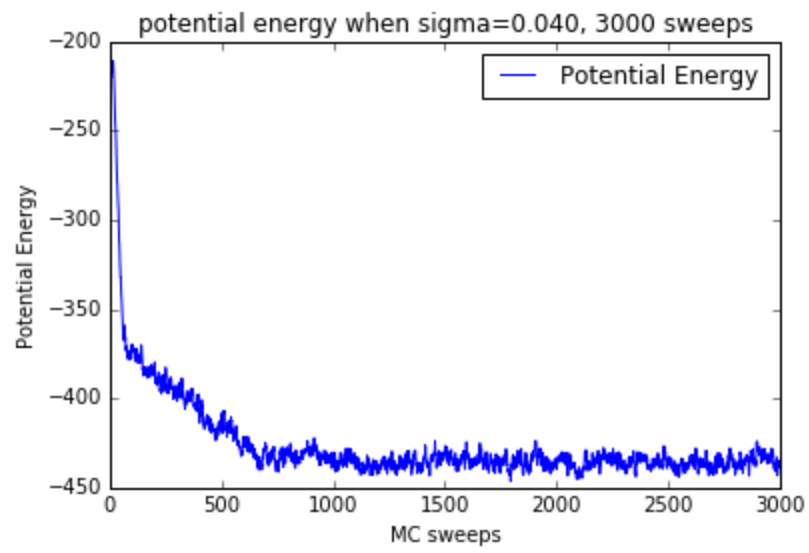
For another two temperatures. I used **sigma = 0.04** and run for 3000 sweeps, **accept ratio** is around **0.587**

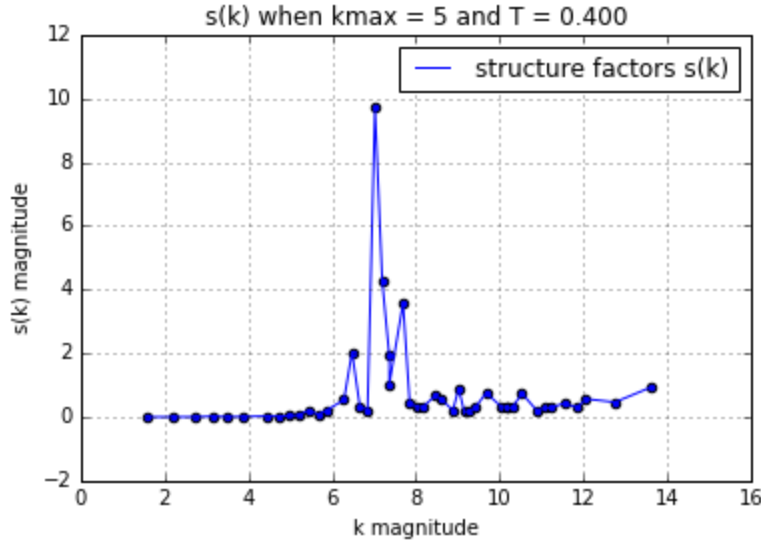
For temperature **T = 1.0**, plots are listed as below:





For temperature $T = 0.4$, plots are listed as below ($g(r)$ and $s(k)$ averaged from 1000-3000 sweeps):





7. Briefly compare these results with your data from the MD simulations in HW 3.

Answer:

Overall the trend of both $g(r)$ and $s(k)$ agree with MD results in HW3. When temperature is low, the system tends to be solid, resulting in larger first $s(k)$ peak value and more spikes. When temperature is high, the system may evolve into liquid or even gas state, with smaller first $s(k)$ peak and less spikes in whole profile.

MD algorithm is way complicated than MC, but can compute kinetics of system. MC is simpler, but faster in speed.

Detailed balance and smarter moves

Standard Metropolis Monte Carlo Acceptance Criterion

Show the standard Metropolis Monte Carlo acceptance criterion below satisfies the detailed balance equation.

Answer:

If we have:

$$A(x \rightarrow x') = \min\left\{1, \frac{\Pi(x')T(x' \rightarrow x)}{\Pi(x)T(x \rightarrow x')}\right\}$$

Then we also have:

$$A(x' \rightarrow x) = \min\left\{1, \frac{\Pi(x)T(x \rightarrow x')}{\Pi(x')T(x' \rightarrow x)}\right\}$$

if $\frac{\Pi(x)T(x \rightarrow x')}{\Pi(x')T(x' \rightarrow x)} > 1$, then $A(x \rightarrow x') = \frac{\Pi(x')T(x' \rightarrow x)}{\Pi(x)T(x \rightarrow x')}$ and $A(x' \rightarrow x) = 1$, then detailed balance is satisfied.

If $\frac{\Pi(x)T(x \rightarrow x')}{\Pi(x')T(x' \rightarrow x)} < 1$, then $A(x \rightarrow x') = 1$ and $A(x' \rightarrow x) = \frac{\Pi(x)T(x \rightarrow x')}{\Pi(x')T(x' \rightarrow x)}$, then detailed balance is also satisfied.

Force-Bias Type Move

1: For this move,

a: write an expression for $T(x \rightarrow x')$ and reverse probability $T(x' \rightarrow x)$.

Answer:

$$\begin{aligned}x' &= x + \eta + \frac{F(x)}{2m} \delta t^2 \\x &= x' + \eta + \frac{F(x')}{2m} \delta t^2 \\T(x \rightarrow x') &= \frac{1}{\sigma\sqrt{2\pi}} \exp \left\{ -\frac{(\eta + \frac{F(x)}{2m} \delta t^2)^2}{2\sigma^2} \right\} \\T(x' \rightarrow x) &= \frac{1}{\sigma\sqrt{2\pi}} \exp \left\{ -\frac{(\eta + \frac{F(x')}{2m} \delta t^2)^2}{2\sigma^2} \right\}\end{aligned}$$

b: combine these terms to write the acceptance criterion $A(x \rightarrow x')$.

$$\begin{aligned}A(x \rightarrow x') &= \min \left\{ 1, \frac{\Pi(x')T(x' \rightarrow x)}{\Pi(x)T(x \rightarrow x')} \right\} \\&= \min \left\{ 1, \exp \left[-\beta \Delta V + \frac{(\eta + \frac{F(x)}{2m} \delta t^2)^2 - (\eta + \frac{F(x')}{2m} \delta t^2)^2}{2\sigma^2} \right] \right\}\end{aligned}$$

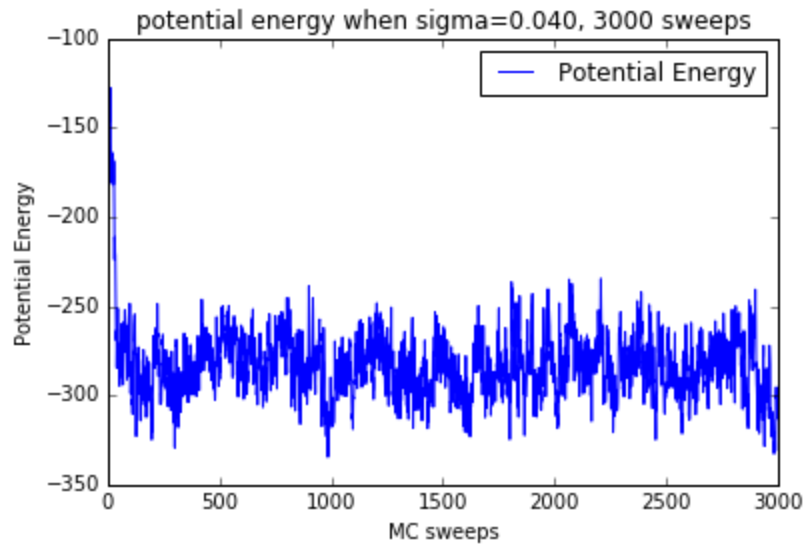
2. Again, measure the energy of your system using the same number of sweeps as you did before.

Answer:

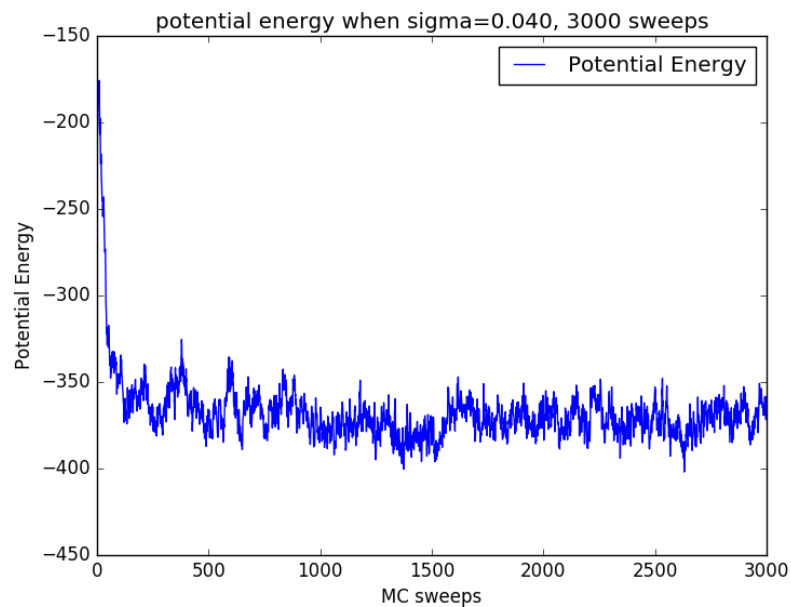
For this question, use $\sigma = 0.04$ and $T = 2.0$ and 1.0 for 3000 sweeps and show results as below ($g(r)$ and $s(k)$ will be averaged from 1000 to 3000 sweeps):

A: submit a trace of your potential energy vs. number of sweep.

For $T = 2.0$ trace of potential energy as below:



For $T = 1.0$ trace of potential energy as below:



B: report standard error of your simulation

temperature	Average potential energy	Standard error of mean
2.0	-284.766	1.56
1.0	-373.033	1.28

C: compare the standard error you get using this move with the standard error you get using the more naive move.

Answer:

It can be seen from the table that standard error of mean potential is lower for force biased method compared with that of the trivial Gaussian method.

Also, from the curves plot it is obvious that force biased curve is less changing with sweep going up after reaching equilibrium.

D: compare the total number of minutes it took to run.

Answer:

The naïve method is faster, 3000 seeps would take around 1000 seconds (17 min) to run. But, it takes around 2000 (33 min) seconds. The calculation of sampling probability $T(r \rightarrow r')$ is way complicated than that of the non-force-biased method, making this method slower.

3. Which move is better? Why?

Answer:

Overall the force biased method is better because the systems is moving in the right direction to the equilibrium, resulting in better potential properties and equilibrium. In fact, for force biased method, the system is evolving very fast to the equilibrium, and there is no need to run 3000 sweeps. From the plots above, we can see that 1500 sweeps, even 1000 sweeps is good enough for force biased method.

Your Own Move

Describe your move in detail and explain why you think it will be more efficient.

Answer:

The principle of more efficient method is to move the x in the direction of the energy lower direction, or the force direction. Apart from adding a component in the direction of the force direction, we can also multiply the magnitude of the eta by the x_{adjust} . It seems like we are trying to learn from MD.

For this new move, write an expression for the sampling probability $T(r \rightarrow r')$.

Answer:

$$x' = x + |\eta| * \frac{F(x)}{2mL} \delta t^2$$

$$x = x' + |\eta| * \frac{F(x')}{2mL} \delta t^2$$

$$T(x - x') = \frac{1}{\sigma\sqrt{2\pi}} \exp \left\{ -\frac{(|\eta| * \frac{F(x)}{2mL} \delta t^2)^2}{2\sigma^2} \right\}$$

NOTE: The L can be the box length or other characteristic length scale, added here in order to make $|\eta| * \frac{F(x)}{2mL} \delta t^2$ in the unit of position.

Write the reverse probability $T(r' \rightarrow r)$.

Answer:

$$T(x' - x) = \frac{1}{\sigma\sqrt{2\pi}} \exp \left\{ -\frac{(|\eta| * \frac{F(x')}{2mL} \delta t^2)^2}{2\sigma^2} \right\}$$

Write an expression for the acceptance probability $A(r \rightarrow r')$.

Answer:

$$\begin{aligned}
A(x \rightarrow x') &= \min\left\{1, \frac{\Pi(x')T(x' \rightarrow x)}{\Pi(x)T(x \rightarrow x')}\right\} \\
&= \min\left\{1, \exp\left[-\beta\Delta V + \frac{(|\eta| * \frac{F(x)}{2mL} \delta t^2)^2 - (|\eta| * \frac{F(x')}{2mL} \delta t^2)^2}{2\sigma^2}\right]\right\}
\end{aligned}$$

I think this algorithm will work though the speed may not be fast enough.