Diffusion Coefficient of the Lennard-Jones Fluid

Molecular Statistics

2014

1 Introduction

In this project we elaborate on the Molecular Dynamics (MD) simulation we have been working on for the past weeks. The main assignment for this project is to calculate the diffusion coefficient D of the Lennard-Jones Fluid.

1.1 Diffusion

Diffusion is the net movement of a substance (e.g., an atom, ion or molecule) from a region of high concentration to a region of low concentration. This is also referred to as the movement of a substance down a concentration gradient. [1] Since we are dealing with a homogeneous liquid, the concentration gradient is zero. What we are dealing with in this simulation is *self-diffusion*.

If we can calculate the two properties mean-square displacement (MSD) or the velocity auto-correlation function (VACF) we can calculate the self-diffusion coefficient D.

The relation between D and the MSD/VACF is as follows:

$$D = \lim_{t\to\infty} \frac{\text{MSD}(t)}{6t} \tag{1.1}$$

$$D = \frac{1}{3}\int_0^\infty \text{VACF}(t)\ \ dt \tag{1.2}$$

MSD is defined as:

$$\text{MSD}(t) = \langle |\mathbf{r}(t_0+t) - \mathbf{r}(t_0)|^2 \rangle = \frac{1}{N}\sum_{i=1}^{N} |\mathbf{r}_i(t_0+t) - \mathbf{r}_i(t_0)|^2 \tag{1.3}$$

where $\langle x \rangle$ indicates the expected (or average) value, N is the number of particles and $\mathbf{r}_i(t_0+t)$ is the position vector of particle i after time t has passed.

VACF is defined as:

$$\text{VACF}(t) = \langle \mathbf{V}(t_0+t)\cdot\mathbf{V}(t_0) \rangle = \frac{1}{N}\sum_{i=1}^{N} \mathbf{V}_i(t_0+t)\cdot\mathbf{V}_i(t_0) \tag{1.4}$$

where $\mathbf{V}_i(t_0+t)$ is the velocity vector for particle i after time t has passed and N is the number of particles.

[1] http://en.wikipedia.org/wiki/Diffusion

1.2 Usage of Correlation Function

Due to large variations in your simulation, the diffusion constant will be heavily dependent on your choice of t_0. Because of this we will average the MSD and VACF over several choices of t_0. That is redefining the MSD as

$$\text{MSD}(t) = \frac{1}{NM} \sum_{j=1}^{M} \sum_{i=1}^{N} |\mathbf{r}_i(t_j + t) - \mathbf{r}_i(t_j)|^2 \tag{1.5}$$

and the VACF as

$$\text{VACF}(t) = \frac{1}{NM} \sum_{j=1}^{M} \sum_{i=1}^{N} \mathbf{V}_i(t_j + t) \cdot \mathbf{V}_i(t_j) \tag{1.6}$$

with M being the number of t_0 values we average over.

When we run the actual simulation, we run it for at least n steps before we do any sampling, since the system needs to be at equilibrium. So all choices for the time origin t_0 must be after the initial burn-in to reach equilibrium.
In our case we will use each sample as a time origin.
You should calculate the average MSD for each choice of t and store it in a list `msd_list` where the index should represent how many time steps from the time origin the MSD is calculated. For example the list element

```
MSD_list[5]
```

contains the average MSD calculated 5 samples away from each time origin. For this choice of t you should average over all $n - 5$ choices of t_0, with n being your number of samples. This might be easier to understand by the below visualization:

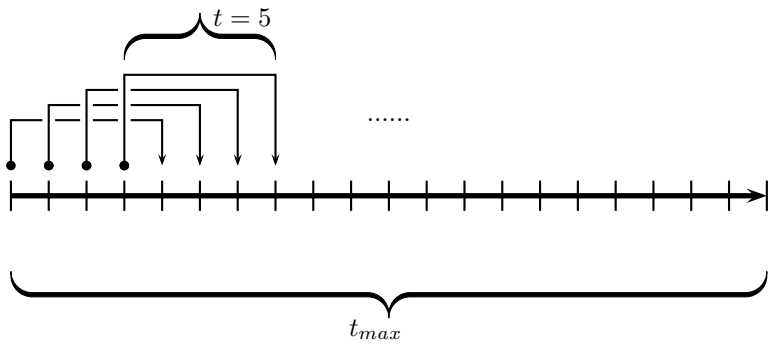


Figure 1.1: Illustrating calculation of `MSD_list[5]`, by calculating the MSD between each time origin (dots) and 5 steps ahead (arrow head).

To help you, we have written the Python code to do this:

```
t_max = n_samples/10
msd_list = np.zeros(t_max)
vacf_list = np.zeros(t_max)

for t in range(t_max):
    for t0 in range(n_samples - t):
        msd_list[t] += calculate_msd(R_list[t0+t], R_list[t0])
```

```
8           vacf_list[t] += calculate_vacf(V_list[t0+t], V_list[t0])
9
10      msd_list[t] /= (n_samples - t)
11      vacf_list[t] /= (n_samples -t)
```

Note: Remember to have a `time_list` with the same dimensions as `vacf_list` and `msd_list`, where the time for each sample is stored. *Hint!* use `np.arange(t_max)` and don't forget the time step dt.

1.3 Calculating the Diffusion Coeffecient

1.3.1 Limit of MSD

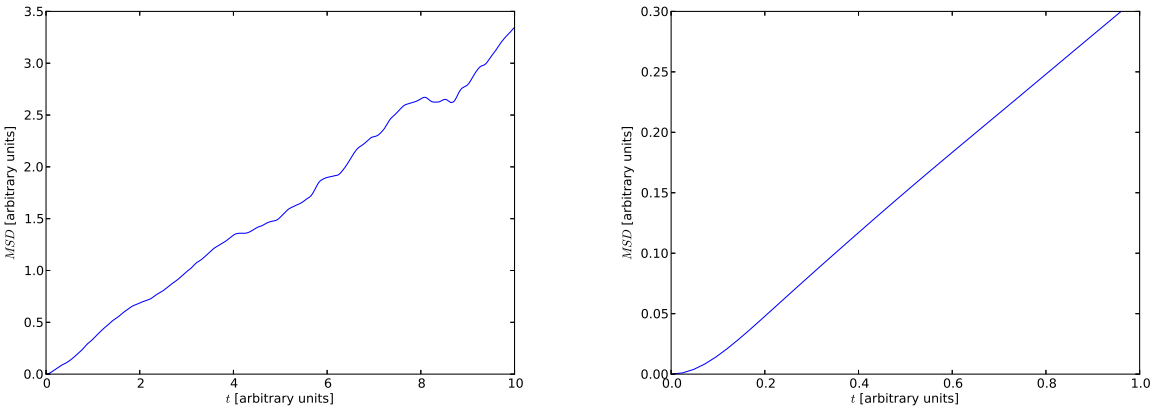
When we want to obtain the diffusion coefficient from MSD(t) we want to find the limit of

$$\lim_{t\to\infty} \frac{\text{MSD}(t)}{t} \tag{1.7}$$

by linear regression. If you plot `msd_list` as a function of the time t, the slope of that line is $6D$. Luckily there is a Numpy/Scipy package for linear regression. So `import scipy` and use it as follows:

```
1 import scipy
2 slope, intercept, r_value, p_value, std_err = scipy.stats.linregress(X, Y)
```

where `X` and `Y` are two lists. The returned value `slope` is then the limit.



(a) **(b)**

Figure 1.2: (a) Uncorrelated MSD vs time. (b) MSD after the correlation function has been done.

1.3.2 Integrating VACF

When we want to find the diffusion coefficient from VACF we need to solve the integral

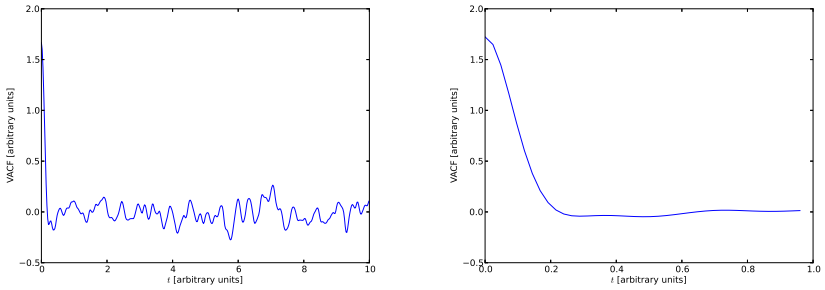
$$\int_0^\infty \text{VACF}(t) \;\; dt \tag{1.8}$$

by numerical integration using the trapezoidal rule. We have all the VACF in a list with index i describing each time step, then the integral I_i can be estimated as

$$I_i = \frac{1}{2}(\text{VACF}_{i+1} + \text{VACF}_i) \cdot (t_{i+1} - t_i) \tag{1.9}$$

The total integral is then the sum of all the individual integrals I_i:

$$\int_0^\infty \text{VACF} \quad dt = \sum_i I_i = \sum_i \frac{1}{2}(\text{VACF}_{i+1} + \text{VACF}_i) \cdot (t_{i+1} - t_i) \tag{1.10}$$



(a) **(b)**

Figure 1.3: (a) VACF vs. time using a single time origin. (b) Averaged VACF vs. time.

2 Assignment

The assignment is to create a python simulation that calculates the diffusion coefficient from MSD and VACF, and calculate the coefficient using different simulation variables.

2.1 Simulation Setup

First part is to setup the simulation to calculate D. To make it easy to get started we have split the setup into small manageable steps. An example for starting variables is

```
n_particles = 108
temperature = 0.5
rho = 0.7
equlib_steps = 10000
n_steps = 10000
sample_freq = 24
dt = 0.001
```

1. Define a sample frequency and save a list of position matrices `R` during the Velo-Verlet simulation after equilibrium has been reached.
2. Create a function that calculates the MSD between two coordinate matrices `R`.
3. Calculate MSD for each sampled `R` and plot MSD vs. simulation time. If you use the constants above it should look similar to figure 1.2a.
4. Implement the correlation function method for MSD and plot MSD vs. time. If you use the constants above it should look similar too figure 1.2b.
5. Use the correlated MSD list and the method for calculating the limit of MSD to calculate the diffusion coefficient.
6. Follow the same procedure for calculating the diffusion coefficient with the VACF method.

Hint: If you run your program with the settings suggested above, you will get $D \approx 0.05$.

2.2 Simulations

Now that you have a working simulation it is time to do some science. You need to do the following six simulations and conclude on the results (with plots).

- **Simulation 1**
  Check the convergence of the diffusion coefficient with simulation step number.
- **Simulation 2**
  Check the convergence of the diffusion coefficient with system size (particle numbers)
- **Simulation 3**
  Check MSD/VACF over time for given temperature at different densities.
- **Simulation 4**
  Check MSD/VACF over time for a given density ρ at different temperatures.
- **Simulation 5**
  Check the diffusion coefficient for a given temperature for different densities.
- **Simulation 6**
  Check the diffusion coefficient for a given density ρ for different temperatures.