Tutorial: modeling polymers and using LAMMPS

JP Mahalik and Murugappan Muthukumar*

Department of Polymer Science and Engineering, University of Massachusetts Amherst, Massachusetts

E-mail: muthu@polysci.umass.edu

Langevin Dynamics

The 1D motion of a particle (much larger than solvent molecules) through a solvent medium can be described by the frictional force ($-\zeta v$):

$$m\frac{dv(t)}{dt} = -\zeta v(t) \tag{1}$$

where m, v, and t are the mass of the particle, velocity of the particle, and time respectively. ζ is the friction coefficient given by Stokes' law, $\zeta = 6\pi\eta a$, where η and a are the solvent viscosity and particle radius, respectively. With an initial velocity of the particle $v(0)$, its velocity decays to zero at longer times because the solution of Eq. 1 is $v(t) = v(0)exp(-\zeta t/m)$. It cannot be true because at thermal equilibrium $\langle v(t)^2\rangle = k_BT/m$. It remains in motion due to constant collision with solvent molecules. Therefore, we need to include contributions from the random force ($f(t)$) from the solvent in Eq. 1. This leads to Langevin equation:

$$m\frac{dv(t)}{dt} = -\zeta v(t) + f(t) \tag{2}$$

The first and second moment of this random force are given by:

$$\langle f(t) \rangle = 0 \tag{3}$$

$$\langle f(t) f(t') \rangle = 2B\delta(t-t') \tag{4}$$

The average of this random force is zero and the forces are correlated only within a small time window $t - t'$, beyond which they are uncorrelated. Eq. 2 is a linear, first-order, inhomogeneous differential equation, can be solved as:

$$v(t) = v(0)e^{-\frac{\zeta t}{m}} + \frac{1}{m}\int_0^t dt' e^{-\frac{\zeta(t-t')}{m}} f(t') \tag{5}$$

Let us get the mean square velocity $\langle v(t)v(t) \rangle$ so that it can be correlated to temperature.

$$v(t)^2 = v(0)^2 e^{-\frac{2\zeta t}{m}} + \frac{2v(0)}{m} e^{-\frac{\zeta t}{m}} \int_0^t dt' e^{-\frac{\zeta(t-t')}{m}} f(t) + \frac{1}{m^2}\int_0^t dt' e^{-\frac{\zeta(t-t')}{m}} f(t') \int_0^t dt'' e^{-\frac{\zeta(t-t'')}{m}} f(t'') \tag{6}$$

Taking average on both sides of Eq. 6 and using the properties of the random force simplification can be done. Using Eq. 3, second term can be eliminated and using the definition of the second moment (Eq. 4) of the random force, following can be obtained:

$$\langle v(t)^2 \rangle = v(0)^2 e^{-\frac{2\zeta t}{m}} + \frac{1}{m^2}\int_0^t dt' e^{-\frac{\zeta(t-t')}{m}} \int_0^t dt'' e^{-\frac{\zeta(t-t'')}{m}} 2B\delta(t'-t'') \tag{7}$$

Using the property of delta function integral $\int_{a-\epsilon}^{a+\epsilon} f(x)\delta(x-a) = f(a)$, the double integral can be simplified to a single integral

$$\langle v(t)^2 \rangle = v(0)^2 e^{-\frac{2\zeta t}{m}} + \frac{2B}{m^2}\int_0^t dt' e^{-\frac{2\zeta(t-t')}{m}} \tag{8}$$

Finally,

$$\langle v(t)^2 \rangle = v(0)^2 e^{-\frac{2\zeta t}{m}} + \frac{B}{\zeta m}\left(1 - e^{-\frac{2\zeta t}{m}}\right) \tag{9}$$

In the long time limit, the exponential terms drop out, hence $\langle v(t)^2 \rangle = B/(\zeta m)$, but $\langle v(t)^2 \rangle = k_B T/m$, which yields $B = \zeta k_B T$.

This is known as the *Fluctuation-dissipation theorem*, which relates the strength of the random noise term B to ζ the friction factor. The balance between the frictional dissipation term and the noise term maintains the thermal equilibrium in the system. Numerically, in LAMMPS the random force is proportional to $\sqrt{(\zeta k_B T/\Delta t)}$, where Δt is the timestep, and the magnitude and direction of the random force is obtained using uniform random distribution. Ideally it should be Gaussian distribution, due to computational efficiency of generating uniform distribution and the difference between the outcome from these distributions has been shown to be minimal, uniform distribution algorithm is implemented in LAMMPS.

The langevin dynamics is modified when there are interacting particles in the system

$$m\frac{dv(t)}{dt} = -\zeta v(t) + f(t) + \nabla U \tag{10}$$

where U is the interaction potential and ∇ is the gradient operator. U must be a smooth function in r.

Common Potentials for Polymers

Neutral Polymers

A polymer is represented as a bead spring model as shown in Fig. 1. For a flexible polymer, the most common potentials are short-range Lennard-Jones (LJ) potential between unconnected beads and the harmonic potentials between connected beads. The LJ potential in

LAMMPS is of the form (there are slight variants of this form in other references)

$$U_{LJ} = 4\epsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right] - 4\epsilon\left[\left(\frac{\sigma}{r_c}\right)^{12} - \left(\frac{\sigma}{r_c}\right)^{6}\right]; r \leq r_c \tag{11}$$

$$= 0; r > r_c \tag{12}$$

The harmonic potential is

$$U_{bond} = K_{bond}(r - r_o)^2 \tag{13}$$

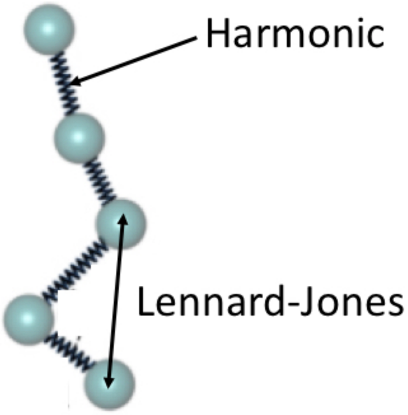

Figure 1: Bead spring model of a flexible polymer showing two most common potentials used.

Exercise-1

Plot U_{LJ} versus r for different r_c (Use $\sigma = 1$ nm and $\epsilon = 1$ k_BT). $r_c = 1.12$ for excluded volume interactions and higher values (typically 2-2.5) are used to model attractions. Most common mistake many people do is to not include the second term. Plot without the second term and notice the difference. The exclusion of the second term does not effect the dynamics but it affects the energy calculations.

Next, we try to determine the scaling behavior of neutral polymers. A C++ code for generation of the topology and parameter files for doing the exercise is provided and the steps for how to use those scripts in LAMMPS is also provided.

Exercise-2

I. Equilibrium criteria: Create a single chain with number of beads ($N = 60$). Other parameters: $\sigma = 1$, $\epsilon = 1$, $r_C = 1.12$, $damp = 1$(In LAMMPS ζ is written as $m/damp$), $K_{bond} = 5000$, $Temperature = 1.0$ (all in LJ units) and follow different energy contributions, temperature and the radius of gyration of the chain (R_g) as a function of time. Can you notice at what time the system has reached equilibrium?

II. For determining the dynamics of a system as a function of real time units, *damp* estimated from Stokes' law should be used, but the system equilibrates very slowly in real timescale. In many situations we might only be interested in getting to the equilibrium state and determine equilibrium properties of the system or we might be interested in relative timescale of two closely related systems. In those cases higher values of *damp* may be used. But too high value of *damp* may cause instability in the system (the fluctuations of different properties of the system may become too high). Run the same simulation by varying *damp*: 1, 10, 100, and 1000. Comment on the behavior of the system. How fast did the system reach equilibrium for different *damp* values? Comment on the fluctuations in the system properties: Energy, temperature and R_g for different *damp* values.

III. For a good solvent like condition, $R_g \sim N^{0.6}$ (modeled by using $r_c = 1.12$). We will perform langevin dynamics simulation on 4 different chain lengths ($N = 40, 80, 120, 200$) and determine their radius of gyration (R_g). Plot $log(R_g)$ versus $log(N)$ (R_g should be averaged over a large number of runs. Start by including only 10000 steps after equilibration and gradually keep on increasing the number of steps to be included for average R_g calculations. At what point the average R_g converge?

IV. We will plot the mean square displacement of the center of mass of $N = 120$ (averaged over five independent simulations) versus time (in log-log scale) and determine its slope. Is it unity? Now determine the mean square displacement of it's monomers (log log scale).

Expected values of the slope is 0.5 below τ_{Rouse} and 1 above τ_{Rouse},[1] where τ_{Rouse} is the Rouse time. Do you get this behavior for all the monomers or only for the central monomers? At the chain ends the slope is expected to be higher than 0.5 below τ_{Rouse}.

About the Code

In the folder $Neutral\_Flex$, the code name is createFile.cpp. To run the code you need g++ compiler. First compile the code using the command: g++ createFile.cpp. To run the code: ./a.out config.in 10 0. Here config.in contains the information about the system, such as the simulation box dimensions, N, etc. You only need to change N for the purpose of doing these exercises. The number 10 is a random number, you can give any integer and the number 0 is the step number. For the first run always use 0, it will create an in.complexation and data.complexation file. You need those two files, a lammps executable file (lmp) and a script (script.sh) to execute the simulation in Gibbs, or you can simply use these two files to run in your local machine (if you have already installed LAMMPS). The first step is system equilibration, in the next step run the C++ code again (./a.out config.in 10 1) to create an *in* file, the lammps executable must have created a binary file (restart.equil) that contains information about the topology of the polymer at the end of the simulation. Simply use the newly generated *in* file for running the next stage of the simulation. You can keep on running the simulation in stages by simply incrementing the number by 1, to generate a new *in* file and use the last step topology binary file generated by lammps .

Charged Polymers

The charged polymers can be modeled in two ways: (a) explicit ions through long-range Coulomb interactions (b) implicit ions through Debye-Hückel interactions.

$$U_C = \frac{1}{4\pi\epsilon_o\epsilon_r}\frac{q^2}{r} \tag{14}$$

$$U_{Debye} = \frac{1}{4\pi\epsilon_o\epsilon_r}\frac{q^2}{r}exp(-\kappa r) \tag{15}$$

Exercise-3

I. Plot U_C (in the units of k_BT) as a function of r, Use $\epsilon_r = 80$ and $q = 1e$. Compare with U_{LJ} versus r plot.

II. Get a feel of how salt effects the electrostatic interaction. Plot U_{Debye} as a function of r for variable κ. $\kappa^{-1}(nm) = 0.3/\sqrt{(I)}$, where I is the molarity of the salt and this equation is valid for unimolar salt in aqueous medium at 300K. Vary I between 1×10^{-5}M to 5 M and notice how the interaction energy changes, compare with the LJ interaction potential and get a feel for the relative strength of LJ interaction versus electrostatic interaction and the important role salt plays in the energetics.

III. We will model a real system based on sodium polystyrene sulfonate.[2] The mass of the monomer $m = 183$ g/mol, σ=0.25 nm (charge separation distance), $\eta = 8.9\times10^{-4}$ Pa.s, (Calculate $damp = m/(3\pi\eta\sigma)$), $\epsilon = 1\ k_BT$, $T = 300$K. Based on the LJ unit conversion table provided in LAMMPS, calculate all the derived units based on these fundamental units.

IV. We will reproduce some results from Ref.[2] This simulates an isolated polyelectrolyte chain equilibrium properties in the presence of its counterions. The free energy of the system determines how many monomers along the polymer chain will be neutralized by a counterion and how many will not, on an average. This exercise is for understanding some of the problems in modeling isolated polyelectrolyte chains and the effect of charge separation distance on the degree of adsorption of ions. Use *Charged_Flex* folder for generating the lammps scripts (Follow the same protocol as provided for neutral system). The only things you need to change in the input files are box size in config.in file and the QFACTOR_convert2LJ in the parameters.in file. First step is to estimate $1e$ in LJ units corresponding to $\sigma = 0.25$ nm and put it in front of QFACTOR_convert2LJ in the parameters.in file. Next, start with smaller box dimension (-30 to 30 in all dimension) in config.in file. Create the in and data files and run lammps. Compare the fraction of adsorbed ions and R_g with the Figure.1 of

Ou and Muthukumar paper[2] provided in the folder. Do you get the right value? Due to long range interactions, the periodic images can influence the properties of the chain. Therefore, we are not truly simulating an isolated chain if it can feel the periodic images. Increase the box dimension by 10 units and see whether you can get the expected result. If not then increase by 10 more units. At this stage you must be able to reproduce the result. Using this box dimension, simulate two other systems with $\sigma = 0.7$ nm and $\sigma = 1.0$nm. Can you reproduce the trend in Figure 1 of the attached reference? The figure is also presented here in Fig. 2

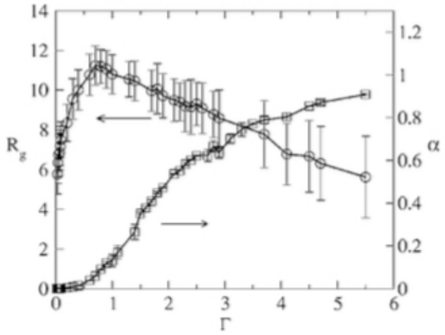

Figure 2: Effect of electrostatic strength on R_g and fraction of adsorbed ions of an isolated polyelectrolyte chain.[2]

References

(1) Muthukumar, M. *Polymer translocation*; CRC press, 2016.

(2) Ou, Z.; Muthukumar, M. Entropy and enthalpy of polyelectrolyte complexation: Langevin dynamics simulations. *The Journal of Chemical Physics* **2006**, *124*, 154902.