

Hands on session 8: Common Equilibrium Calculations

1. Common Equilibrium Calculation: The Diffusion Coefficient

The (late time) diffusion coefficient of a system of particles (atoms or colloids) in thermal equilibrium can be found in one of two ways: (i) from the mean-squared displacement of particles; and (ii) from the long-time integral of the velocity autocorrelation function. Our goal in this session is to use LAMMPS to calculate the diffusion coefficient for dilute “hard-sphere” colloids using one of these two methods. An input script template has been provided, but it is missing several key command lines for performing the desired simulation. Your task is to complete the input script, run it, analyze the results, and report back to the group your findings. Below are the steps to guide you through the process.

- a. Extra credit: verify Stokes Einstein ($D = kT/3\pi\eta d$) \rightarrow vary Temp
- b. Extra, extra credit (follow-up on after short course): verify scaling of late-time diffusion coefficient with volume fraction

2. The input script templates (choose “in.ms_d_template” or “in.vacf_template”)

Note there are several ways to perform an equilibrium simulation of colloids suspended in a fluid. Depending on the level of “resolution” one may choose an all-atom approach where atoms of both the colloid and solvent are resolved explicitly. Alternatively, one may choose a coarse-grained representation of the colloid and an explicit atomistic solvent. Here, we shall choose a coarse-grained representation of the colloid and a type of implicit, coarse-grained solvent. In particular, we consider only the hydrodynamic effects of the solvent to be relevant and choose a model which seeks to capture those forces applied to the colloids. Additionally, our choice of model will make certain simplifying assumptions about the nature of the hydrodynamic interactions to allow for more efficient numerical simulation.

3. Fill in pair style

- a. To simplify things, let’s consider mono-sized, “hard-sphere” or purely repulsive colloids
 - i. Fill in the pair colloid command
 1. Set the “diameter”, d ($= d1 = d2$) equal to 10σ
 2. Set “sigma” equal to 1.0
 3. Set “Hamaker” equal to $4\pi^2$
 4. To make the colloids hard-sphere-like, cut the potential off at the minimum. This is where $dU(r)/dr = 0$, or cutoff = $d + 30^{-1/6}\sigma$ (see doc pages).
 5. Note the use of non-dimensional (or LJ-type) units! In this case the Boltzmann constant, $k = 1.0$.
 - ii. Don’t forget the pair coeff command! (see doc pages)

4. Fill in thermostat and integrator

- a. Several ways to do particles in thermal equilibrium

- i. Run simulation in canonical ensemble: e.g., use Nose-Hoover thermostat (fix nvt)
 - ii. FLD (pair lubricate + pair brownian) + fix nve
 - 1. For this option, a pair overlay command is required to handle the multiple pair styles being applied to colloids
 - b. For simplicity, let's choose fix langevin + fix nve integrator
 - i. Fill in the fix langevin command
 - 1. Note: fix langevin applies a steady Stokes drag ($3\pi\mu d v_p$) to the colloids as a model of the hydrodynamic forces and a random force to model thermal fluctuations in the fluid.
 - 2. To use this command you need to specify the "damp" parameter and the "temperature" among others (see doc page)
 - 3. "damp" has units of time and happens to be the so-called momentum relaxation time, $\tau_B = m/3\pi\mu d$, where m is the mass of the colloid, μ is the shear viscosity of the fluid, d is the diameter of the colloid. Assume the viscosity $\mu = 1.01$ and $m = 345$.
 - 4. Set "temperature" to 1.0
 - ii. Note: Our simulation approach is sometimes called Brownian Dynamics except in using fix nve we do not ignore the colloid inertial terms as is often the case. At some other time you may be interested in reproducing these simulations without inertial terms and compare results. You can do this using FLD (pair lubricateU; see doc pages).
 - iii. Note: In the dilute limit, fix langevin is equivalent to Brownian FLD.
- 5. Fill in compute and thermo commands
 - a. For MSD route,
 - i. fill in the compute msd command
 - ii. Fill in the thermo style and thermo commands
 - 1. You'll want to use a custom thermo style to output the variable(s) defined for the compute msd.
 - 2. Don't forget to output other thermo data (e.g., temp) to check that the simulation is running correctly. In addition fill in the thermo command to output calculations every time step.
 - b. For the VACF route, we will have to postprocess the velocities to get the VACF and D
 - i. Fill in the dump command to output the velocities every time step to a file.
 - 1. The perl script postprocessing code (see, "calc_D-via-VACF.pl") expects a lammps formatted dump file with columns: "id", "type", "vx", "vy", "vz". This will require a custom dump style.
- 6. Fill in run command

- a. Careful! How long should you run to get reasonable results? For dilute colloids and the governing dynamics we are simulating, there is a characteristic timescale related to the time it takes for a colloid to “forget” its velocity at a previous time. This is often referred to as the momentum relaxation timescale. It is found as $\tau_B = m/3\pi\mu d$ (in LJ time units, τ -- see doc pages for units).
 - b. For times significantly less than τ_B , the colloids undergo ballistic motion, while for times much greater than this colloids undergo diffusive motion. As you may recall from physics 1, ballistic motion is characterized by kinematics, to leading order in Δt , of the form $\Delta x = v_0 \Delta t$, where v_0 is a constant. What does this mean we might expect the form of the mean-**squared** displacement to be in this regime?
 - c. You may also recall that for classical, Fickian diffusion the mean-squared displacement goes like $\langle \Delta x^2 \rangle \sim D \Delta t$, where D is the diffusion coefficient. If, as was stated above, the motion of the colloids is “diffusive” on timescales greater than τ_B , what does this suggest the length of our simulation should be? Remember, we are interested in calculating D from the slope of the MSD. How long should our simulation be in terms of number of timesteps (which is what LAMMPS expects in the run command)? Note, the time step we are using is $dt = 0.1\tau$, so $n_{timesteps} \gg \tau_B / dt$.
7. Now it’s time to get a number for D .
- a. For the msd route, you can plot the msd vs. t using gnuplot
 - i. In a terminal window, issue the “gnuplot” command to start the plotting program.
 - ii. Plot the msd vs. time by issuing the command “plot ‘log.lammps’ using 0.1*\$<timestep_column>:<computemsd_column>”; where <timestep_column> and <computemsd_column> should be replaced with the integer values of the columns in the lammps log output file that contain the timestep and msd values, respectively.
 - iii. Now you can “eyeball” D by fitting a line to the msd vs. time plot.
 1. In gnuplot, issue the command “replot <m>*x”; where <m> is the numerical value of the slope of the line. A good first choice for the slope would be $6D$ using the Stokes-Einstein relation, $D = kT/3\pi\mu d$. Did it work; how close were you? You can iterate this command to refine the value of D . What are possible sources of error?
 - b. For the VACF route, we will postprocess the velocity data
 - i. In a terminal window issue the command “perl calc_D-via-VACF.pl -h” to find out usage details
 1. The input file is the name of the file you dumped from lammps that contains the velocity data.
 2. Set the number of timesteps to 1000.

3. So the command will look something like “perl calc_D-via-VACF.pl <atoms.velocities> 1000”, depending on the name of velocity dump-file.
4. What D is reported? How does this compare to Stokes-Einstein, $D = kT/3\pi\mu d$? What are possible sources of error?
5. After running the perl script, you can view the VACF using gnuplot.
 - a. In a terminal window, issue the command “gnuplot”
 - b. In gnuplot, issue the command “plot ‘VACF.dat’ using 1:2”. Is this form of the VACF reasonable?