LAMMPS Tutorial - Brownian Dynamics Simulation

Introduction to BD and LAMMPS

February 4, 2025

UNIVERSITY OF TURKU

Outline

1. Understanding Brownian Dynamics Simulation
2. Studying viscosity of a fluid using shear flow
3. Performing simple thermodynamic calculations
4. Creating patchy particle system using LAMMPS
5. Caviate: Softcore potential within the context of DPD simulation

Understanding Brownian Dynamics Simulation

Going Beyond Soft-Core Potentials (I)

Soft-core potentials represent highly coarse-grained systems, but real physical systems follow Fermi statistics and therefore possess **hard-core repulsion**.

When the thermal energy scale satisfies

$$v \gg \lambda,$$

the system behaves classically. In this regime:

- Velocities follow Maxwell–Boltzmann distribution,
- Fluctuation–dissipation theorem applies,
- Equipartition theorem holds.

These assumptions greatly simplify computation.

Going Beyond Soft-Core Potentials (II)

The physical properties of such systems can be recovered by assigning **van der Waals interactions**, typically approximated by the **Lennard–Jones potential**.

- Using Lennard–Jones allows incorporation of both attraction and hard-core repulsion.
- Newton's equations of motion can then be applied to evolve the system.
- Bonded atoms (e.g, covalent) can be approximated by using harmonic spring potential.

This avoids the need for the computationally impractical **quantum unitary evolution**, such as solving the Schrödinger equation.

Summary: Molecular Dynamics and Beyond

Molecular Dynamics (MD) provides a classical framework for simulating particle motion using Newton's equations coupled with realistic interaction potentials such as:

- Lennard–Jones,
- Hard-sphere,
- Coulombic/Screened potentials.

Beyond basic MD, many specialized interaction models exist:

- **Patchy particles** — anisotropic interactions for self-assembly studies,
- **Dipolar or multipolar models**,
- **Bonded potentials** for polymers and biomolecules,
- **Coarse-grained force fields** (e.g., MARTINI).

These extend MD into richer regimes for studying complex soft-matter and biological systems.

Understanding BD: Recap to teaching session

Langevin equation for discrete models:

- **Langevin Equation:** Includes inertial effects to provide a microscopic discrete model for stochastic dynamics:

$$m\frac{d\mathbf{v}(\mathbf{r},t)}{dt} = -\gamma\mathbf{v}(\mathbf{r},t) + \boldsymbol{\xi}(t) - \nabla\Phi_i(\mathbf{r}).$$

Both the discrete (Langevin) and probabilistic whole-picture (Fokker-Planck) models stem from the foundational principles of Wiener processes. The Zwanzig-Mori formalism provides the theoretical core for deriving these equations.

Understanding BD: Recap to teaching session

Relation with the continuum model like Fokker-Planck equation:

- **Fokker-Planck Equation:** Describes the time evolution of the probability density function $P(x,t)$ in a continuous probabilistic framework:

$$\frac{\partial P(x,t)}{\partial t} = -\frac{\partial}{\partial x}\left[\mu(x)P(x,t)\right] + \frac{\partial^2}{\partial x^2}\left[D(x)P(x,t)\right].$$

Both the discrete (Langevin) and probabilistic whole-picture (Fokker-Planck) models stem from the foundational principles of Wiener processes. The Zwanzig-Mori formalism provides the theoretical core for deriving these equations.

Understanding BD: Recap

- A large molecule is considered as a single sphere with a Hard Core and some interaction site.
- Preferred for the directional and selective bonding, e.g, self assembly in biologically inspired materials.
- Highly used in the NVT ensemble modeling.
- It does not care about the hydrodynamics.
- However, it can be coupled with other hydrodynamic based methods like MPCD to simulate the micro-swimmers like sperm cells.

Homework: Let us Do and check the equivalence of both the formalism through Langevin Dynamics.

For the future reference, all the study materials can be found on the following github link

1. Download all the codes from the website:
   `https://github.com/vikkivarma16/lammps_tutorial`.
   Command line:

```
git clone https://github.com/vikkivarma16/lammps_tutorial
```

Homework: Let us Do and check the equivalence of both the formalism through Langevin Dynamics.

Observing the evolution of density distribution in continuum model:

1. Go to the folder **t2_Exercises_and_homework_t2_bd/t2_langevin_focker_planck**.
2. Run the Python code named **t2_evolution_focker_planck.py**.
3. Open the created Gif file and observe the evolution of the probability distribution of the particles going through Brownian dynamics, which is based on the Fokker-Planck formalism.

Homework: Let us Do and check the equivalence of both the formalism through Langevin Dynamics.

Observing the evolution of density distribution in discrete model:

1. Run you LAMMPS file named **`t2_langevin.in`**.
2. Load your output file **`simulation_data.lammpstrj`** into Ovito.
3. Select the **`add modification`** and **`histogram`**, and chose **`position_X`**.
4. Tick and set the range between $0 - 20$ for the **`x-axis`**.
5. Observe the evolution of the histogram and compare, it with the Fokker Planck probability distribution evolution.

Studying viscosity of a fluid using shear flow

Defining units, boundary conditions, and particle attributes

Step 1: LAMMPS Setup

- Use Lennard–Jones (LJ) units, appropriate for simple fluids like argon.
- Define a 3D periodic simulation cell.
- Use `atom_style atomic`, since particles are structureless LJ atoms.

LAMMPS Code

```
units           lj
dimension       3
boundary        p p p
atom_style      atomic
```

Defining pair interactions and physical variables

Step 2: Define the interaction model

- Use a Lennard–Jones interaction with a finite cutoff (3.0σ).
- Define physical variables: temperature, timestep, and shear parameters.
- These variables allow flexible modification of the simulation.

LAMMPS Code

```
pair_style      lj/cut 3.0
variable T      equal 1.0
variable dt     equal 0.0001
variable p      equal 400
variable s      equal 5
variable d      equal ${p}*${s}
variable srate  equal 3.5
```

Defining the simulation box and shear regions

Step 3: Create the box and divide it into regions

- Create an elongated simulation box to allow shear flow generation.
- Define two regions: upper and lower slabs (acting as moving walls).
- Create atoms on an FCC lattice for initialization.

LAMMPS Code

```
region box block -3 23 0 20 0 20
create_box 3 box
region lowersec block INF 0 INF INF INF INF
region uppersec block 20 INF INF INF INF INF
lattice fcc 0.3
create_atoms 1 box
mass * 1.0
```

Grouping regions for applying shear and constraints

Step 4: Assign walls and fluid groups

- Atoms in upper and lower regions are assigned as wall types.
- The remaining atoms constitute the fluid.
- Types 2 and 3 correspond to left (upper) and right (lower) wall atoms.

LAMMPS Code

```
group upper  region uppersec
group lower  region lowersec
group wall   union upper lower
group fluid  subtract all wall
set group upper type 2
set group lower type 3
```

Defining Lennard–Jones interaction parameters

Step 5: Specify LJ coefficients for wall and fluid atoms

- Use identical LJ interactions among all types.
- This keeps the system simple while enabling shear.

LAMMPS Code

```
pair_coeff 1 1 0.5 1.0 3.0
pair_coeff 2 2 0.5 1.0 3.0
pair_coeff 3 3 0.5 1.0 3.0
pair_coeff 1 2 1.0 1.0 3.0
pair_coeff 2 3 1.0 1.0 3.0
pair_coeff 1 3 1.0 1.0 3.0
```

Applying shear and initializing velocities

Step 6: Impose shear flow by wall motion

- Fluid atoms receive Maxwellian velocities at temperature T.
- Left wall moves upward; right wall moves downward.
- Wall atoms' forces are zeroed so they act as rigid boundaries.

LAMMPS Code

```
velocity fluid create ${T} 12345 mom yes rot yes
velocity left  set 0.0 ${srate} 0.0
velocity right set 0.0 -${srate} 0.0
fix frozenwall wall setforce 0.0 0.0 0.0
```

Thermostatting the fluid and removing shear bias

Step 7: Apply thermostat and compute unbiased temperature

- Use a Langevin thermostat only on the fluid.
- Remove flow velocity bias using `temp/partial`.
- Compute ramped temperature for shear flow stability.

LAMMPS Code

```
fix integrator all nve
fix thermostat fluid langevin 1.0 1.0 1.0 933888
compute bias fluid temp/partial 1 0 1
variable tsrate equal 2*${srate}
compute fluid_temp fluid temp/ramp vy 0 ${tsrate} x 0 20
```

Computing velocity profile across the channel

Step 8: Use chunk-averaging to compute shear profile

- Divide the box into bins along x.
- Compute the center-of-mass velocity of each bin.
- Save results to a text file for analysis.

LAMMPS Code

```
compute chunk_1 fluid chunk/atom bin/1d x lower 0.05
compute chunkvel fluid vcm/chunk chunk_1
fix 1 all ave/time 100 10 1000 c_chunkvel[*] file velocity_profile.txt \
mode vector
```

Running the production simulation

Step 9: Run long simulation for viscosity sampling

- Long trajectory (5M timesteps) is required for shear-viscosity averaging.
- Dump atomic positions for trajectory visualization.

LAMMPS Code

```
dump 1 all custom 5000 simulation_data.lammpstrj id type x y z
run 5000000
```

Performing simple thermodynamic calculations

Homework: Calculating the liquid-gas coexistence density:

Let us simulate LJ fluid and calculate the liquid-gas coexistence density following the instruction given in `Model 3` folder named **`T2E2_Instructions`**.

Creating patchy particle system using LAMMPS

Homework: Simulating patchy particles modeled as colloidal system with directional bonding:

Let us create a molecular template and load it to generated the configuration of molecules apply the integrator and simulate. `Model 3` folder named **`T2_molecular_simulation_Instructions`**.

Conclusion

Conclusion:

- You have visualize the relation between discrete model and continumm models.
- You have successfully set up a patchy particle system using LAMMPS to study the self assembly.
- You have calculated a thermodynamic quantity using Brownian dynamics simulation

Thank you!

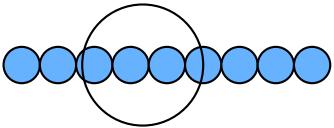
Questions and discussion welcome.

Caviate: Softcore potential within the context of DPD simulation

Understanding BD: A Look Back

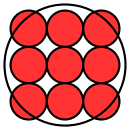
Advantage of not breaking the Symmetry at Mesoscopic Scale:

A sheet of water molecule



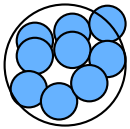
Isotropic thin film

Density dependent potential applied on Hard Core (BD)



Anisotropic minimum energy state with broken symmetry

Density dependent potential applied on Soft core (DPD)



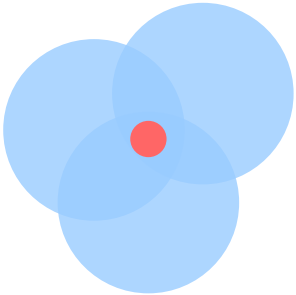
An isotropic fluid droplet

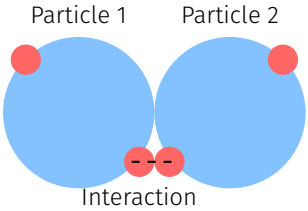
Understanding BD: A Look Back

Disadvantage of not breaking the Symmetry:

Let us consider a directional force defined by the potential

$$\chi_{ij} = \begin{cases} 1 & |\hat{\mathbf{r}}_{ij}.\hat{\mathbf{n}}_i| \geq cos(\Omega^\circ), \\ 0 & Otherwise \end{cases}$$





Patchy particle model
Treating directional forces

Soft-core potential
It does not serve the purpose

Hard-core potential
Serves the purpose

Understanding BD: A comparison between the Hard Core and Soft Core models

Hard Core Potential

1. Symmetry breaking requires nucleation and growth.
2. System can stuck in a meta-stable phase, due to kinetic stability.

Soft Core Potential

1. System generally does not get stuck in the metastable phase.
2. Metastability have purely thermodynamic nature.
3. Not suitable for the nucleation and growth kind of phenomenon, where kinetically driven metastability plays an important role.

Understanding BD: A Look Back

Application of Soft Core model:

It can be used to model a system, where surface energy plays an important role, e.g, surface tension.

It can be applied for the mass transfer through convection.

It can be applied for the heat transfer, where the particles have some internal degree of freedom.

Learn by Example

Congratulations you have been armed to tackle the real world problems!

——— May the force be with you ! ———