

LAMMPS Tutorial - Brownian Dynamics Simulation

Introduction to BD and LAMMPS

February 5, 2026

Outline

1. Understanding Brownian Dynamics Simulation
2. Performing Your First Brownian Dynamics (BD) Simulation
3. Homework: Calculation of thermodynamic properties
4. Beyond Brownian Dynamics: Capturing Motion at the Scale of Relaxation Time
5. Continuum Models

Understanding Brownian Dynamics Simulation

What is Brownian Dynamics?

Key idea

Brownian Dynamics (BD) models the motion of particles in a solvent **without explicitly simulating solvent molecules.**

- Thermal motion arises from **random collisions**
- Solvent effects are replaced by:
 - Friction (dissipation)
 - Random forces (noise)
- Ideal for:
 - Soft matter
 - Polymers
 - Biomolecules

Brownian Dynamics: Kicking each other without realizing the same.

Passive Motion

Passive particles do not possess self-propulsion. Their motion arises from thermal fluctuations of the surrounding fluid and viscous dissipation.

Langevin Equation

The dynamics of a passive Brownian particle of mass m is governed by

$$m \frac{d\mathbf{v}}{dt} = -\gamma \mathbf{v} + \mathbf{F}_{\text{ext}} + \boldsymbol{\xi}(t),$$

where

- γ is the friction coefficient,
- \mathbf{F}_{ext} is an external deterministic force,
- $\boldsymbol{\xi}(t)$ is Gaussian white noise with

$$\langle \boldsymbol{\xi}(t) \rangle = 0, \quad \langle \boldsymbol{\xi}(t) \boldsymbol{\xi}(t') \rangle = 2\gamma k_B T \delta(t - t').$$

Brownian Dynamics Limit

Brownian dynamics corresponds to the **overdamped limit**:

$$m \frac{d\mathbf{v}}{dt} \ll \gamma \mathbf{v}$$

or equivalently,

$$\text{Re} \ll 1, \quad t \gg \tau,$$

where

$$\tau = \frac{m}{\gamma}$$

is the **momentum relaxation time**, and

$$\text{Re} = \frac{\rho v L}{\eta}$$

is the **Reynolds number**.

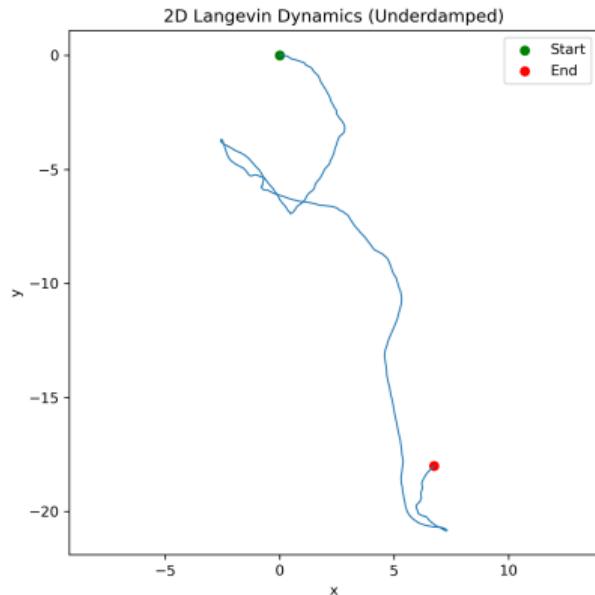
Brownian Dynamics Limit

In this regime, inertia is negligible and the Langevin equation reduces to

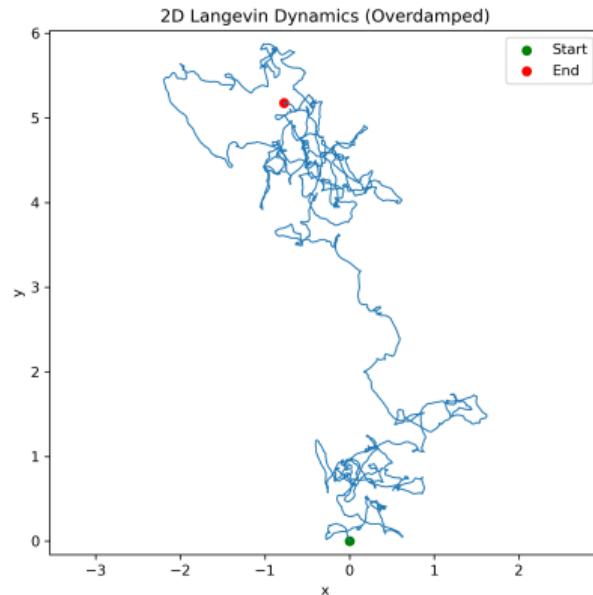
$$\gamma \frac{d\mathbf{r}}{dt} = \mathbf{F}_{\text{ext}} + \boldsymbol{\xi}(t),$$

where $\boldsymbol{\xi}(t)$ represents thermal noise, also known as Gaussian white noise.

Passive Motion and Brownian Dynamics

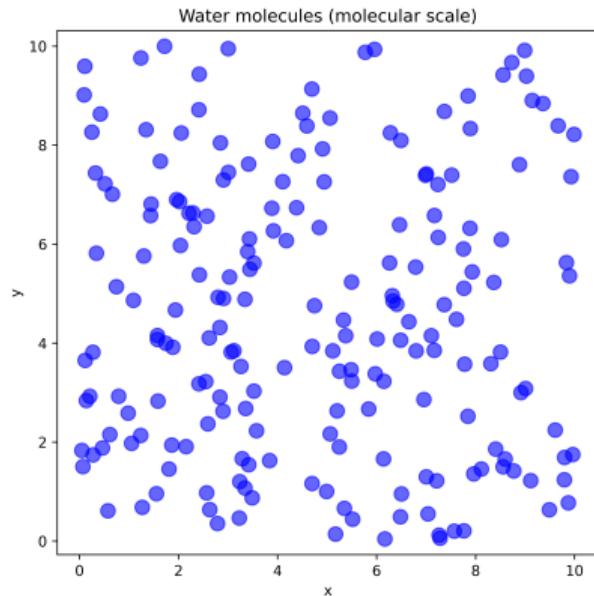


Underdamped motion trajectory.

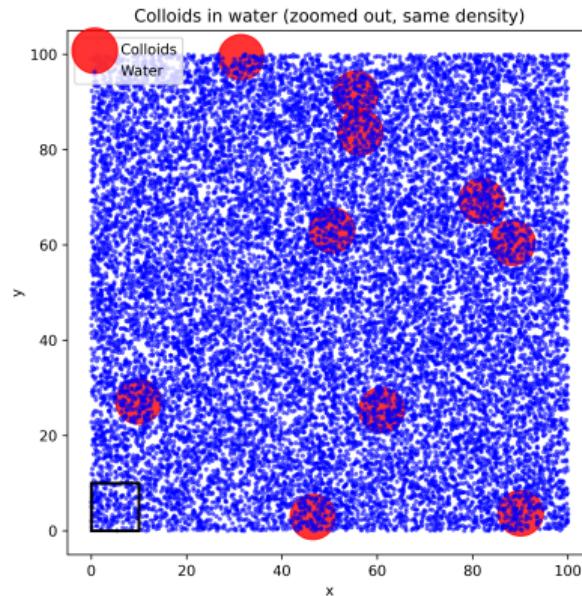


Overdamped motion trajectory.

Passive Motion and Brownian Dynamics



At smaller length scale.



At larger length scale.

What is Brownian Dynamics?

Key idea

Brownian dynamics can be extended to the active matter system. Majority of the real life system falls in this category.

The basic idea can be summarized just in two numbers.

- **Peclet number:** Measures the ratio of the energy due to the active inertial forces to the random kicks in the system. It depends on the degree of propulsion and mass of the particle. Related to the Brownian motion.
- **Reynold number:** It measures the ratio of the inertial forces to the forces due to the viscosity. Related to the Hydrodynamics.

Physical meaning

The Péclet number quantifies the competition between **directed motion** and **thermal diffusion**.

$$\text{Pe} = \frac{vL}{D}$$

- v – characteristic self-propulsion or drift velocity
- L – characteristic length scale (e.g., particle size)
- D – translational diffusion coefficient
- $\text{Pe} \ll 1$: motion dominated by Brownian diffusion
- $\text{Pe} \gg 1$: motion dominated by active or driven propulsion

Physical meaning

The Reynolds number compares **inertial forces** to **viscous forces** in a fluid.

$$\text{Re} = \frac{\rho v L}{\eta}$$

- ρ — fluid density
- v — characteristic velocity
- L — characteristic length scale
- η — dynamic viscosity
- $\text{Re} \ll 1$: overdamped, Stokes flow (typical for colloids, bacteria), Pedestrian dynamics.
- $\text{Re} \gg 1$: inertia-dominated, turbulent or inertial flow, e.g., swimmers, airplanes, cyclists, go-kart dynamics.

Performing Your First Brownian Dynamics (BD) Simulation

Simulating Patchy Particle Model System

Objective

We will create a patchy particle system, define its interactions, and simulate its Brownian dynamics in LAMMPS. This exercise demonstrates how directional interactions (patches) affect the motion and assembly of particles.

Folder Location

All relevant files are in the folder: [**patchy_particle_model**](#).

Step 1: Creating the Simulation Box

Defining the Box

- Box dimensions: $30 \times 30 \times 30$ LJ units.
- Periodic boundary conditions in all directions.
- Atom style: hybrid (`sphere molecular`) to allow for both core and patch atoms.

```
region box block 0 30.0 0 30.0 0 30.0  
create_box 2 box
```

Step 2: Loading Molecular Template and Creating Atoms

Molecule Template

The molecule template (`patchy_molecule.mol`) defines a central core (type 1) surrounded by patches (type 2), forming a rigid, directional particle.

Creating Molecules

- Create 1000 molecules randomly inside the box.
- Avoid overlap using `overlap` and `maxtry` options.

```
molecule patchy_part patchy_molecule.mol
create_atoms 0 random 1000 87910 NULL mol patchy_part 454756 overlap 1.5
....maxtry 50
```

Patchy Particle Molecule Template: File Contents

Molecule Template (`patchy_molecule.mol`)

The patchy particle is represented as a rigid molecule with one core atom and four peripheral patch atoms arranged tetrahedrally.

patchy_molecule.mol

5 atoms

Coords

1 1.0 1.0 1.0 # must be specified for all the atoms

...

Types

1 1 # must be specified for all the atoms

...

Explanation of Atom Types and Functionality

- Atom 1: Central core (type 1), massive, interacts via Lennard-Jones potential.
- Atoms 2-5: Peripheral patches (type 4), nearly massless, interact via directional cosine-squared potential.
- Tetrahedral geometry ensures patches create specific bonding orientations.
- In LAMMPS, the template is loaded using the `molecule` command, and multiple molecules are placed randomly in the simulation box.
- The design allows us to observe directional bonding and cluster formation in a Brownian dynamics simulation model using a relatively much simple yet powerful approach.

Step 3: Defining Particle Interactions

Pair Potentials

- Core-core: Lennard-Jones (LJ) potential.
- Patch-patch: Cosine squared (Gaussian-like) potential for directional binding.
- Core-patch: no interaction (**none**) to prevent artificial bonding.

```
pair_coeff 1 1 lj/cut 0.01 1.3 2.0
pair_coeff 1 2 none
pair_coeff 2 2 cosine/squared 8 0.3 0.35
```

Step 4: Grouping Particles and Optimization

Groups

- **core**: central atoms
- **patch**: peripheral patch atoms
- **rigid_molecule**: all atoms in the molecule

Neighbor List Optimization

Exclude intra-molecular neighbors from force calculations to speed up simulation:

```
neigh_modify exclude molecule/intra rigid_molecule every 1 delay 0
                                         ..check no
```

Step 5: Thermostat and Integration

Langevin Thermostat

Applied to core particles to simulate Brownian dynamics at constant temperature:

```
fix thermo_stat core langevin 1.0 1.0 0.1 428984 omega yes
```

Integration of Molecules

Use rigid-body integration for molecules:

```
fix rigid_thermo rigid_molecule rigid/small molecule
```

Step 6: Data Collection and Thermodynamic Quantities

Dump Particle Positions

```
dump 1 all custom 10 simulation_data.lammpstrj id type x y z mol
```

Compute Temperature and Kinetic Energy (Core)

```
compute kinetic_core core ke
fix kinetic_output core ave/time 100 1 100 c_kinetic_core
...file kinetic.dat mode scalar
compute temp_core core temp/sphere fix temp_output core
...ave/time 100 1 100 c_temp_core file temperature.dat mode scalar
```

Step 7: Running the Simulation

Timestep and Runtime

- Time step: 0.005 LJ unit

```
timestep = 0.005
```

Terminal Thermo Output

```
thermo 100
thermo_style custom step temp ke pe press c_kinetic_core c_temp_core
```

Run Command

```
run 100000
```

Step 8: Analysis of Results

Visualizing the System

- Load `simulation_data.lammpstrj` in OVITO or any visualization software.
- Observe motion of patchy particles and clustering due to directional interactions.

Thermodynamic Observables

- Check temperature and kinetic energy in logs.
- Copy paste the potential energy section in a .txt file and run the `gunplot` command to plot the result and see the kinetics and equilibration of the system.

Homework: Calculation of thermodynamic properties

Computing the liquid-gas phase diagram for the given LJ-fluid

Use the folder named [**Home_work/liquid_vapor_coexistence**](#) and follow the instructions given in the PDF file.

Beyond Brownian Dynamics: Capturing Motion at the Scale of Relaxation Time

From Brownian Dynamics to Hydrodynamics

Motivation

While Brownian dynamics accurately captures overdamped motion at long times, many transport properties—such as viscosity—require resolving momentum transport and velocity correlations over finite relaxation times.

What Changes?

- Momentum is no longer slaved instantaneously to thermal noise.
- Velocity gradients develop under external driving.
- Transport coefficients emerge from stress–velocity relations.

Objective: Viscosity from Non-Equilibrium Shearing

Objective

We compute the shear viscosity of a Lennard-Jones fluid using a non-equilibrium molecular dynamics (NEMD) approach.

Key Idea

- Impose a steady shear flow by moving confining walls.
- Measure the resulting velocity gradient.
- Extract viscosity from the momentum flux (pressure tensor).

Source Files

All relevant input scripts and analysis tools are available at:

Use the folder named **Hands_on/viscosity_through_shearing** and follow the instructions given in the PDF file.

Step 1: System Geometry and Confinement

Simulation Box

- A rectangular box extended along the shear-gradient direction.
- Periodic boundaries in all directions.
- Two solid walls define the shear geometry.

```
region box block -3 23 0 20 0 20  
create_box 3 box
```

Step 2: Defining Wall and Fluid Regions

Wall Construction

- Lower and upper regions act as rigid walls.
- Wall atoms are frozen during the simulation.

```
region lowersec block INF 0 INF INF INF INF  
region uppersec block 20 INF INF INF INF INF
```

Fluid Region

Atoms not belonging to wall regions constitute the sheared fluid.

Step 3: Interatomic Interactions

Lennard-Jones Fluid

- Standard LJ interactions are used for all atom pairs.
- Potential is shifted to ensure continuity at the cutoff.

```
pair_style lj/cut 3.0
pair_coeff * * 1.0 1.0 3.5
pair_modify shift yes
```

Step 4: Imposing Shear Flow

Driving the System

- The upper wall is assigned a constant velocity.
- The lower wall remains stationary.
- This creates a velocity gradient in the fluid.

```
velocity upper set 0.0 ${srate} 0.0 sum no  
fix frozenwall wall setforce 0.0 0.0 0.0
```

Step 5: Thermostatting Under Shear

Langevin Thermostat

- Applied only to the fluid atoms.
- Maintains temperature while allowing momentum transport.

```
fix thermostat fluid langevin 1.1 1.1 0.1 933888
fix integrator all nve
```

Streaming Velocity Removal

The thermostat is applied to thermal fluctuations only, excluding shear flow.

Step 6: Measuring Velocity Profiles

Velocity Gradient

- Fluid is binned along the shear-gradient direction.
- Average velocity in each bin is computed.

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

Step 7: Measuring Shear Stress

Momentum Flux

- The off-diagonal pressure tensor component P_{xy} represents momentum transport.
- Time averaging improves statistical accuracy.

```
variable pxy equal pxy
fix pxy_ave all ave/time 100 10 1000 v_pxy file pxy.dat
```

Step 8: Computing the Viscosity

Constitutive Relation

The shear viscosity is obtained from:

$$\eta = -\frac{\langle P_{xy} \rangle}{\partial v_y / \partial x}$$

Practical Implementation

- The velocity gradient is extracted from the velocity profile.
- The average shear stress is read from `pxy.dat`.
- A Python script performs fitting and analysis.

Step 9: Analysis and Visualization

Post-Processing

- Run the provided Python script to compute viscosity.
- Inspect velocity profiles and linear fits.
- Verify steady-state behavior of P_{xy} .

Physical Interpretation

This approach explicitly connects microscopic momentum transfer to macroscopic viscosity through non-equilibrium dynamics.

Continuum Models

Description

The Fokker-Planck equation describes the time evolution of the probability density function of a particle's position and velocity under stochastic forces, such as Brownian motion. It provides a continuum description of the dynamics, complementing discrete particle simulations like Langevin dynamics.

$$\frac{\partial P(\mathbf{r}, t)}{\partial t} = -\nabla \cdot (\mathbf{v}P) + D\nabla^2 P$$

where $P(\mathbf{r}, t)$ is the probability density, \mathbf{v} is the drift velocity, and D is the diffusion coefficient.

Objective

Observe the evolution of the density distribution in the continuum model using the Fokker-Planck equation.

1. Navigate to the folder `langevin_dynamics_vs_fokker_planck_equation`.
2. Run the Python script `evolution_focker_planck.py`.
3. Open the generated GIF file and observe the evolution of the probability distribution of particles under Brownian dynamics as predicted by the Fokker-Planck formalism.

Objective

Compare the discrete particle dynamics from Langevin simulations with the continuum Fokker-Planck model.

1. Run your LAMMPS input file **langevin.in**.
2. Load the output file **simulation_data.lammpstrj** in OVITO.
3. Apply **Add Modification → Histogram** and select **position_X**.
4. Set the histogram range between **0-20** for the x-axis.
5. Observe the histogram evolution and compare it with the Fokker-Planck probability distribution.

Key Takeaways

- Successfully set up a patchy particle system in LAMMPS to study self-assembly.
- Computed thermodynamic quantities using Brownian dynamics simulations.
- Studied viscosity in overdamped systems and non-equilibrium shearing.
- Visualized and compared discrete particle dynamics with continuum Fokker-Planck predictions.

Thank you!

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

----- May the force be with you ! -----

Welcome to Questions and Discussion Session

.... ????????

Feel free to ask, or send me an email on address vavarm@utu.fi or vikkivarma16@gmail.com