Manual for running the BD Codes

January 18, 2026

Exercise 1: Verification of the Einstein Diffusion Relation

This exercise aims to verify the validity of the Einstein diffusion equation for Brownian motion,

$$D = \frac{k_{\mathrm{B}}T}{\gamma}, \tag{1}$$

where D is the translational diffusion coefficient, k_{B} is the Boltzmann constant, T is the temperature, and γ is the friction coefficient.

The verification is performed using Langevin (Brownian) dynamics simulations in LAMMPS and post-processing of the mean squared displacement (MSD). All required input and analysis files are provided in the folder `Exercise_1`.

Simulation Setup

The LAMMPS input script for this exercise is `brownian_dynamics.in`. The key features of the simulation setup are summarized below.

1. **Simulation Units and System Definition**

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

The simulation is performed in reduced Lennard–Jones (LJ) units. The system is three-dimensional and employs periodic boundary conditions in all spatial directions, ensuring bulk-like behavior without surface effects.

2. **Creation of a Large Simulation Box**

```
region my_box block 0 30 0 30 0 30
create_box 1 my_box
```

A cubic simulation box of size $30 \times 30 \times 30$ is created. Only one particle type is defined in the system.

3. **Particle Initialization**

```
create_atoms 1 random 5000 87910 my_box
```

A total of 5000 particles are randomly distributed throughout the simulation box. This random initial configuration avoids artificial density correlations at the start of the simulation.

4. **Particle Properties and Interactions**

```
mass 1 1.0

pair_style lj/cut 2.5
pair_style none
```

All particles are assigned unit mass. Although a Lennard–Jones interaction is initially declared, it is explicitly switched off using `pair_style none`. This ensures that particles do not interact with each other and undergo ideal Brownian motion without caging or collective effects.

5. **Initial Velocity Distribution**

```
velocity all create 4.0 12345 mom yes rot no
```

Initial velocities are assigned according to a Maxwell–Boltzmann distribution corresponding to the specified temperature. The total linear momentum of the system is removed to prevent spurious drift of the center of mass.

6. **Langevin Thermostat for Brownian Dynamics**

```
fix my_thermo_stat all langevin 4 4 0.1 89080
```

The Langevin thermostat introduces viscous damping and stochastic forces that mimic the effect of an implicit solvent. The friction coefficient controls the strength of dissipation, while the random force maintains the target temperature.

7. **Time Integration Scheme**

```
fix integrator all nve
```

The `nve` integrator updates particle positions and velocities. Since temperature control is already provided by the Langevin thermostat, no additional thermostat is required.

8. **Time Step Specification**

```
timestep 0.005
```

The simulation time step determines the physical time advanced in each integration step. The chosen value ensures numerical stability for Langevin dynamics.

9. **Thermodynamic Output Settings**

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

Thermodynamic quantities such as temperature, kinetic energy, potential energy, and pressure are printed to the terminal every 100 time steps to monitor the system's evolution.

10. **Mean Squared Displacement (MSD) Calculation**

```
compute msd_all all msd
fix msd_output all ave/time 100 1 100 c_msd_all \
file msd.dat mode vector
```

The mean squared displacement is computed for all particles and written to the file `msd.dat`. This data is later used to extract the diffusion coefficient.

11. **Running the Simulation**

```
run 40000
```

The simulation is executed for 40,000 time steps, providing sufficient temporal data to observe the diffusive regime and reliably compute the diffusion coefficient.

Post-Processing and Diffusion Coefficient Extraction

After completing each simulation run, the mean squared displacement (MSD) data generated by LAMMPS are post-processed using the Python script **`msd_data_rectifier_and_plotter.py`**. This script converts the raw MSD output into a physically meaningful time series, extracts the diffusion coefficient, and visualizes the diffusive behavior.

Reading and Restructuring the MSD Data

The file `msd.dat`, produced by the LAMMPS command `compute msd` together with `fix ave/time`, does not store MSD data in a single-row-per-time-step format. Instead, MSD components are written sequentially over multiple lines.

The script performs the following operations:

- Reads the entire `msd.dat` file line by line.
- Ignores comment lines beginning with the character `#`.
- Identifies the simulation timestep and the corresponding MSD components ($\langle x^2 \rangle$, $\langle y^2 \rangle$, $\langle z^2 \rangle$, and total MSD).
- Reconstructs the MSD data into a structured format with one row per timestep.

The processed data are written to the file `msd_processed.txt` in the form

$$t \quad \langle x^2 \rangle \quad \langle y^2 \rangle \quad \langle z^2 \rangle \quad \langle r^2 \rangle,$$

where the simulation timestep is converted into physical time using the LAMMPS time step value.

Time Conversion and MSD Selection

The simulation time is computed using

$$t = N_{\text{step}} \times \Delta t,$$

where $\Delta t = 0.005$ is the integration time step. To avoid initial transient effects, the first data point is excluded from subsequent analysis.

Only the total MSD, $\langle r^2(t) \rangle = \langle x^2 \rangle + \langle y^2 \rangle + \langle z^2 \rangle$, is used for calculating the diffusion coefficient.

Diffusion Coefficient Calculation

The diffusion coefficient is extracted from the long-time behavior of the MSD using the Einstein relation

$$D = \frac{1}{2d} \frac{\Delta \langle r^2(t) \rangle}{\Delta t},$$

where $d = 3$ is the dimensionality of the system.

The script computes:

- The total change in MSD over the simulation time,
- The corresponding elapsed time,
- The diffusion coefficient D in Lennard–Jones units.

The computed diffusivity, total simulation time, and total MSD variation are printed to the terminal for quick verification.

Visualization of Diffusive Behavior

To verify the diffusive regime, the script generates a log-log plot of mean squared displacement versus time. In addition to the MSD data, a reference line corresponding to the theoretical diffusive scaling

$$\langle r^2(t) \rangle = 6Dt$$

is plotted for comparison.

The plot includes:

- Total MSD as a function of time,
- A linear reference line with slope proportional to the diffusion coefficient,
- Logarithmic scales on both axes to emphasize power-law behavior.

The final figure is saved as a high-resolution PNG file named `msd_vs_time.png` and must be included in the answer sheet.

Physical Interpretation

A linear increase of MSD with time in the log-log plot confirms normal diffusion. The extracted diffusion coefficient is then used in subsequent parameter studies to verify the Einstein diffusion relation and its dependence on temperature, friction coefficient, and particle mass.

Parameter Studies

To verify the Einstein diffusion relation, three independent parameter studies are performed.

1. **Dependence on Friction Coefficient**

   Run simulations for
   $$\gamma^{-1} = 0.1,\ 0.3,\ 0.6,\ 1.0.$$
   For each value, compute the diffusion coefficient D and record the results in `gamma_vs_d.txt`.

2. **Dependence on Temperature**

   Run simulations for
   $$k_{\mathrm{B}}T = 1,\ 2,\ 3,\ 7.$$
   Compute the diffusion coefficient for each temperature and store the data in `t_vs_d.txt`.

3. **Dependence on Particle Mass**

   Run simulations for particle masses
   $$m = 1,\ 2,\ 3,\ 6.$$
   Extract the corresponding diffusivities and store the results in `m_vs_d.txt`.

Final Plotting and Analysis

The script `plotter.py` is used to generate the final plots:

- D vs. γ^{-1}
- D vs. $k_{\mathrm{B}}T$
- D vs. m^{-1}

Each plot is saved as a separate PNG file and must be included in the answer sheet.

Expected Outcome

From the generated plots, students should observe that:

- D varies linearly with γ^{-1},
- D varies linearly with temperature T,
- D is independent of particle mass in the overdamped regime.

These results confirm the validity of the Einstein diffusion equation and demonstrate the consistency between Langevin dynamics simulations and theoretical predictions.
