Manual for running the BD Codes

January 18, 2026

Exercise II: Viscosity of a Lennard–Jones Fluid Using the Green–Kubo Relation

In this exercise, the shear viscosity of a Lennard–Jones (LJ) fluid is computed using equilibrium molecular dynamics and the Green–Kubo formalism. The viscosity is obtained from the time integral of the autocorrelation function of the off-diagonal components of the pressure tensor.

The shear viscosity is defined as

$$\eta = \frac{V}{k_{\mathrm{B}}T} \int_0^\infty \langle P_{\alpha\beta}(0)\, P_{\alpha\beta}(t) \rangle \, dt, \tag{1}$$

where $P_{\alpha\beta} \in \{P_{xy}, P_{xz}, P_{yz}\}$, V is the system volume, and T is the temperature.

All simulations for this exercise are performed using the folder `Exercise_2` and the LAMMPS input file `viscosity_input.in`.

Simulation setup

The LAMMPS input file `viscosity_input.in` is organized into logical sections that control system definition, equilibration, stress autocorrelation sampling, and viscosity evaluation. Each section is described in detail below.

1. **Simulation Units and General Settings**

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

The simulation is performed in reduced Lennard–Jones (LJ) units, where $\sigma = \epsilon = m = k_{\mathrm{B}} = 1$. Periodic boundary conditions are applied in all three spatial directions to model an infinite bulk fluid and eliminate surface effects. The atomic style is sufficient since particles have no internal degrees of freedom.

2. **User-Defined Thermodynamic and Numerical Parameters**

```
variable T      equal 1.0
variable Tinit  equal 1.2
variable dt     equal 0.00001
```

The variable T defines the target production temperature, while T_{init} is chosen slightly higher to accelerate equilibration. A very small timestep is initially used to safely relax the dense LJ system and avoid numerical instabilities during early equilibration.

3. **Correlation Sampling Parameters**

```
variable p      equal 400
variable s      equal 5
variable d      equal ${p}*${s}
```

These parameters control the sampling of the stress autocorrelation function:

- s: interval (in timesteps) between successive samples,
- p: number of correlation points,
- d: total correlation window length.

Together, they determine the maximum integration time of the Green–Kubo integral and must be chosen long enough for the stress correlations to decay to zero.

4. **System Setup and Density Control**

```
region          box block 0 20 0 20 0 20
create_box      1 box
lattice         fcc 0.6
create_atoms    1 box
mass            1 1.0
```

Particles are initialized on an FCC lattice to minimize initial overlaps and large repulsive forces at the start of the simulation. In LAMMPS, the command `lattice fcc <value>` specifies a reduced lattice density ρ^*, which is a dimensionless quantity. According to the LAMMPS manual, the actual number density ρ is related to the reduced density ρ^* by

$$\text{factor}^d = \frac{\rho}{\rho^*},$$

where d is the dimensionality of the system and $\rho = N/V$ with V being the volume of the lattice.

In Lennard–Jones reduced units, where $\sigma = 1$, the scaling factor (`factor`) depends on the lattice spacing generated by the `lattice` command. For the present system configuration, using `lattice fcc 0.6` corresponds to a lattice density $\rho = 0.6\rho^*$, which yields an actual number density of approximately $\rho \approx 0.6$ for the chosen simulation box and particle size ($\sigma = 1$, which has $\rho^* \approx 1.0$).

By adjusting the lattice density parameter or the simulation box dimensions, students can:

- reduce the number of particles to lower the computational cost, or
- increase the system size to improve statistical averaging of the Green-Kubo viscosity.

Students must therefore verify and, if necessary, rescale the lattice parameters by adjusting the line `lattice fcc 0.6` or box size so that the measured density $\rho = N/V$ matches the desired value.

Note: **In some cases, the specified lattice spacing does not fit an integer number of unit cells within the simulation box under periodic boundary conditions. When this mismatch occurs, atoms generated near opposite faces of the box may overlap periodically, leading to extremely large interatomic forces and eventual simulation failure (e.g., numerical instability or core dump). To avoid this issue, students are advised to adjust the simulation box dimensions so that they are commensurate with the chosen lattice spacing. This ensures that the lattice tiles the periodic box exactly and prevents unphysical atomic overlaps at the boundaries. The effective lattice constant used by LAMMPS is printed in the simulation log file and should be checked carefully before running long equilibration or production simulations.**

5. **Interatomic Interactions**

```
pair_style      lj/cut 3.5
pair_modify     shift yes
pair_coeff      1 1 1.0 1.0 3.5
```

Particles interact via the Lennard–Jones potential with a cutoff of 3.5σ. The potential is shifted to zero at the cutoff to remove discontinuities in energy, which is essential for accurate force and stress calculations required in Green–Kubo analysis.

6. **Time Integration and Thermodynamic Output**

```
timestep        ${dt}
thermo          ${d}
thermo_style custom step temp press ke
```

Thermodynamic quantities are printed at intervals consistent with the stress correlation sampling window. This allows monitoring of temperature and pressure stability throughout the simulation.

7. **Equilibration Phase**

```
velocity all create ${Tinit} 12345 dist gaussian
velocity all scale ${T}

fix thermostat all langevin ${T} ${T} 0.1 34300
fix integrator all nve

run 100000
```

The system is equilibrated using a Langevin thermostat to rapidly reach the desired temperature. The stochastic thermostat is suitable during equilibration but must be removed before production to avoid altering equilibrium stress fluctuations.

8. **Transition to Production Conditions**

```
variable dt equal 0.001
timestep ${dt}

run 50000
unfix thermostat
unfix integrator
```

After initial relaxation, the timestep is increased to improve computational efficiency. All equilibration fixes are then removed in preparation for the production run.

9. **Production Run Setup (NVE Ensemble)**

```
velocity all scale ${T}
fix integrator all nve
reset_timestep 0
```

The production phase is performed in the microcanonical ensemble. This is a crucial requirement for Green–Kubo calculations, as thermostats can artificially damp stress fluctuations and bias transport coefficients.

10. **Stress Tensor Components and Autocorrelation**

```
variable pxy equal pxy
variable pxz equal pxz
variable pyz equal pyz

fix SS all ave/correlate ${s} ${p} ${d} \
v_pxy v_pxz v_pyz type auto file S0S.dat ave running
```

The off-diagonal components of the pressure tensor represent microscopic momentum flux. Their time autocorrelation functions form the core input to the Green–Kubo viscosity relation.

11. **Green–Kubo Integration and Viscosity Evaluation**

```
variable V equal vol
variable scale equal ${V}/(${kB}*${T})*${s}*${dt}

variable v11 equal trap(f_SS[3])*${scale}
variable v22 equal trap(f_SS[4])*${scale}
variable v33 equal trap(f_SS[5])*${scale}
```

The stress autocorrelation functions are numerically integrated using the trapezoidal rule. Viscosities computed from P_{xy}, P_{xz}, and P_{yz} are averaged to improve statistical accuracy.

12. **Final Output and Density Verification**

```
variable eta equal (v_v11+v_v22+v_v33)/3.0
variable ndens equal count(all)/vol

print "Average viscosity: ${eta} (LJ units) @ ${T}
   ...LJ, density ${ndens}"
```

The final viscosity is reported in reduced LJ units together with the measured number density. Students should repeat the calculation for $\rho = 0.3, 0.4, 0.5, 0.6$ and analyze the dependence of viscosity on density.

Post-Processing: Viscosity vs. Density Plot

After performing Green-Kubo calculations for different densities ($\rho = 0.3, 0.4, 0.5, 0.6$), students can collate the results into a two-column text file named `viscosity_vs_density.txt`, where the first column is density and the second column is the corresponding viscosity.

This given python `plotter.py` script performs the following tasks:

- Loads the computed viscosity data for different densities.
- Plots viscosity (η) against density (ρ) using markers and connecting lines.
- Generates a high-resolution PNG file suitable for reports or presentations.

Expected Physical Trends

Students should observe that:

- Viscosity increases monotonically with density.
- Dense LJ fluids exhibit stronger momentum transport due to increased collision frequency.

This exercise demonstrates how macroscopic transport coefficients emerge from microscopic equilibrium fluctuations.
