

LAMMPS Tutorial - Dissipative Particle Dynamics (DPD) Simulation

Introduction to DPD and LAMMPS

January 21, 2025

Outline

1. Understanding DPD Simulation

2. Calibration in DPD simulation

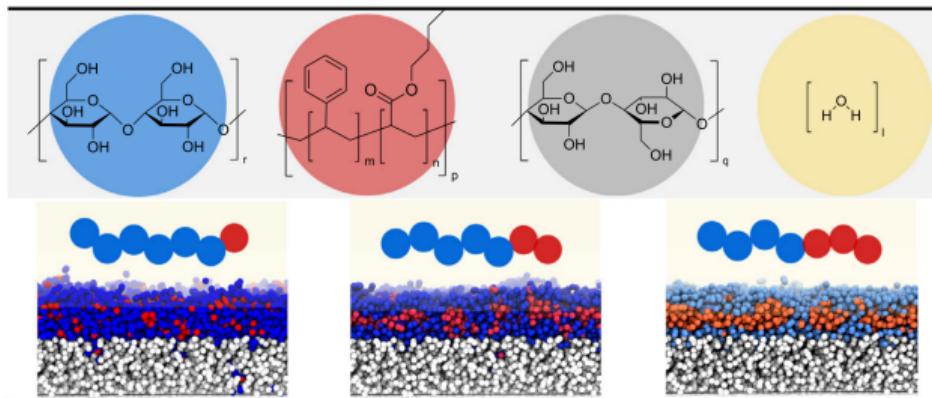
3. LAMMPS: An Introduction

Understanding DPD Simulation

Understanding DPD: What is DPD?

- DPD is a mesoscopic simulation technique used for soft matter systems.
- It represents molecules as clusters of particles, simplifying atomic details.
- Applications include simulations of polymers, colloids, and biological systems.

Understanding DPD: Why are you concerned with the formalism?



DPD modeling of polymers.

Source: Alberto Scacchi et al. (2023). “Controlling self-assembling co-polymer coatings of hydrophilic polysaccharide substrates via co-polymer block length ratio”. In: *Journal of Colloid and Interface Science* 640, pp. 809–819. ISSN: 0021-9797

Understanding DPD: Why are you concerned with the formalism?

In DPD, you are considering to simulate a large chunk of atoms or molecules as a single bead.

There is something compromised.

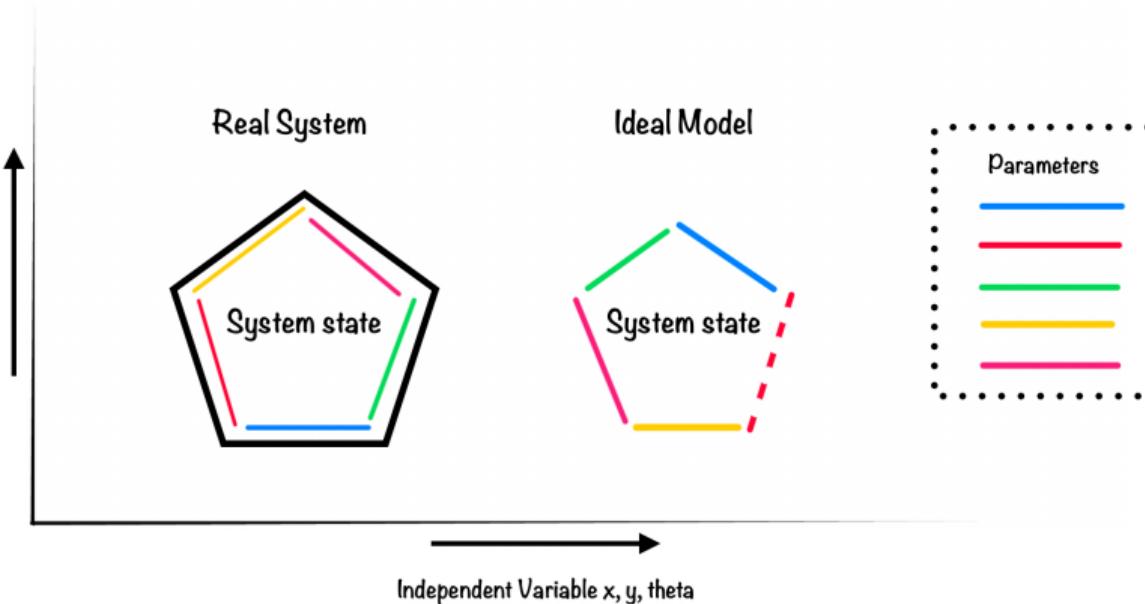
Understanding DPD: What is modeling?

In discrete modeling, the goal is to mimic the actual system with as much accuracy as possible.

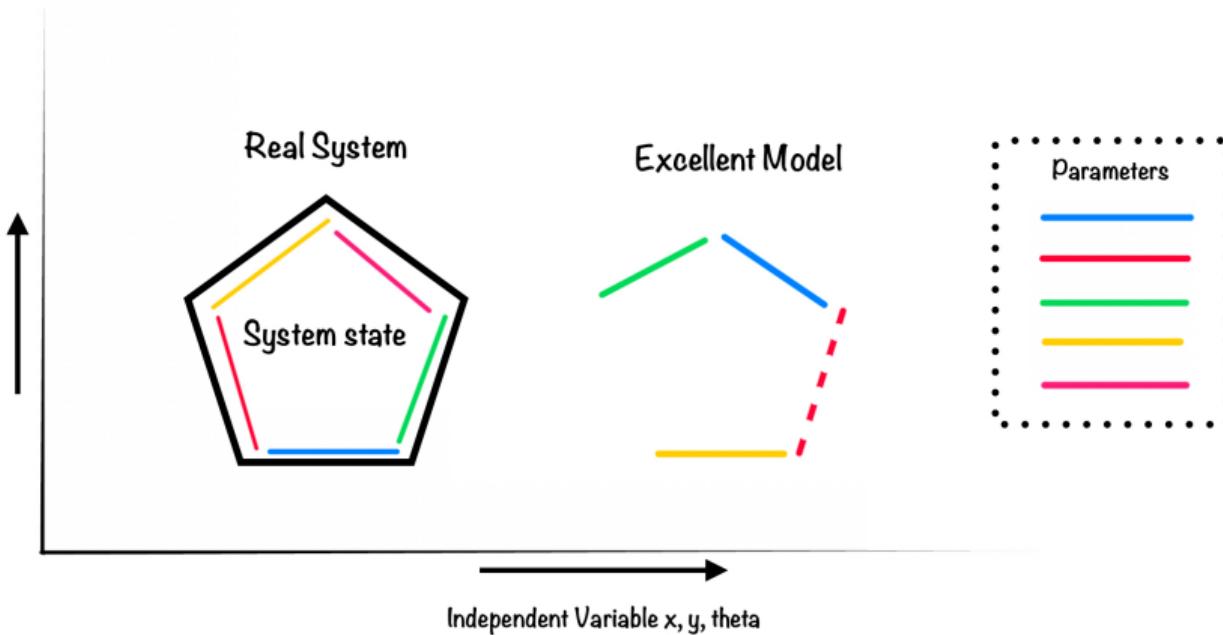
Bottlenecks:

1. Simulation time or computational cost
2. Unretrievable information
3. Complexity associated with the model and the calibration of the system based on the model

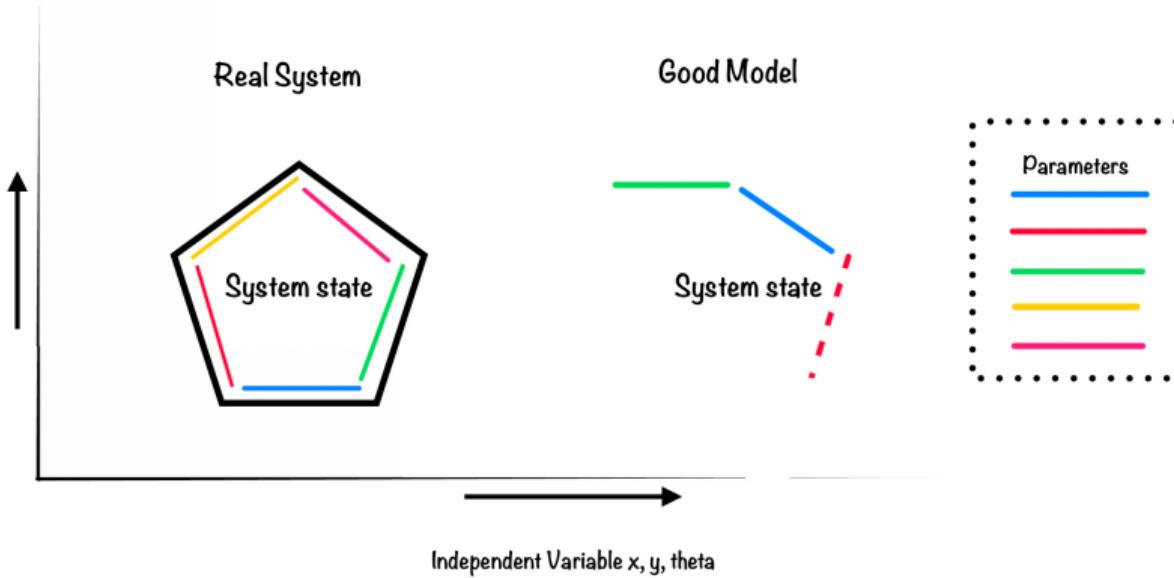
Understanding DPD: What is modeling?



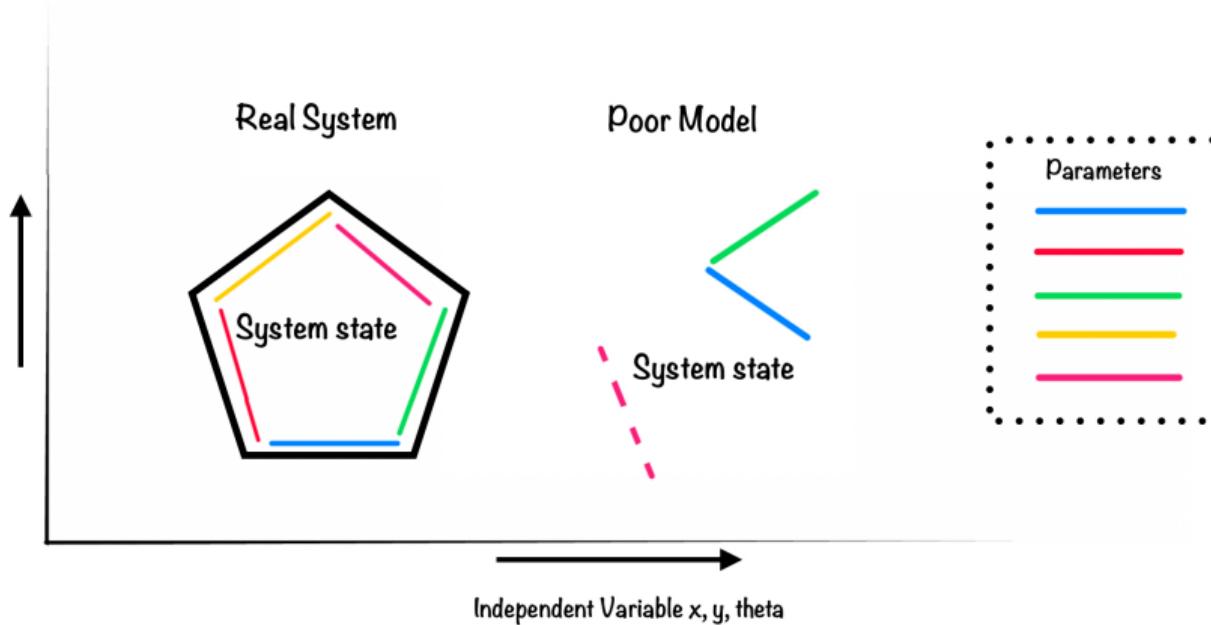
Understanding DPD: What is modeling?



Understanding DPD: What is modeling?



Understanding DPD: What is modeling?

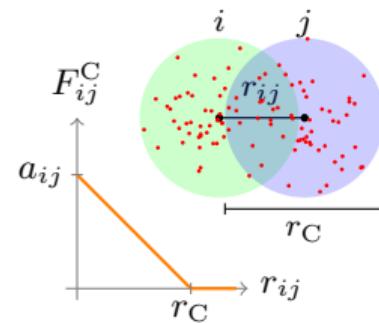


Understanding DPD: Key Aspects of Modeling in DPD

One of the main aspects of DPD governing the evolution of the system is,

The **conservative force** between beads i and j is **soft**-repulsive

$$\mathbf{F}_{ij}^C = a_{ij} \left(1 - \frac{r_{ij}}{r_C}\right) \hat{\mathbf{r}}_{ij}, \quad r < r_C, \quad (1)$$



a_{ij} is the repulsion strength, $r_{ij} = |\mathbf{r}_i - \mathbf{r}_j|$ their distance, and $\hat{\mathbf{r}}_{ij} = \mathbf{r}_{ij}/r_{ij}$ the force director.

Understanding DPD: Key Aspects of Modeling in DPD

- Is DPD a good model or a poor one?
- Let us run a DPD simulation, calibrate the system, and find out for ourselves.

Calibration in DPD simulation

Calibration in DPD: How to calibrate?

Calibration refers to the adjustment of variables to match parameters with desired values corresponding to the system of interest.

In atomistic simulation, the calibration process can be classified into two main parts:

- Matching **Thermodynamic parameters**, e.g., mechanical variables (isothermal compressibility), which should follow the equation of state and match experimental values.
- Matching **Dynamics**, e.g., diffusivity, viscosity, etc.

Calibration in DPD: What do we do in calibration?

We adjust simulation variables to achieve the desired values, then rescale or shift the independent variables to match the parameters of the system of interest.

The following points should be noted:

1. The most important aspect of calibration is the ability to **scale** independent variables, e.g., time, length, mass, and energy.
2. Your system must exhibit properties like **self-similarity** to enable scaling, which simplifies the calibration process.

Calibration in DPD: What do we do in calibration?

Units of independent variables are a key aspect of simulations and play a crucial role in system calibration.

1. **Time**, **Length**, **Mass**: These are related to the **dynamics** of the system, such as **diffusivity** and **mobility**.
2. **Energy**: This is associated with the **mechanics** of the system, including thermodynamic variables like **pressure** and **temperature**.

This will be discussed in detail during the tutorial on Brownian Dynamics simulation.

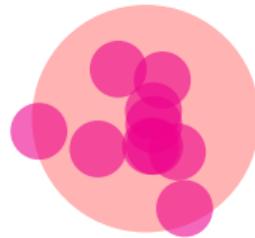
Calibration in DPD: Let us Do



DPD Water Molecule

Consider $1/N_m$ th of DPD Beads

$$R_s$$



DPD Bead

Consider N_m DPD Water
Molecule

$$R_{cm}$$



Real Water Molecule

1 real molecule with 30\AA^3 Unit
volume

$$R_w$$

Calibration in DPD: Relation between the DPD water and the DPD Beads

Displacement of the DPD bead is equivalent to the average of displacement of the N_m DPD water molecules.

Let us consider the $\mathbf{R}_1, \mathbf{R}_2, \mathbf{R}_3, \dots, \mathbf{R}_N$ as the displacements of the 1, 2, 3, ..., Nth DPD water molecules.

$$\mathbf{R}_{cm} = \left\langle \frac{\mathbf{R}_1 + \mathbf{R}_2 + \mathbf{R}_3 + \dots + \mathbf{R}_N}{N} \right\rangle$$

$$R_{cm}^2 = \left\langle \frac{(\mathbf{R}_1 + \mathbf{R}_2 + \mathbf{R}_3 + \dots + \mathbf{R}_N) \cdot (\mathbf{R}_1 + \mathbf{R}_2 + \mathbf{R}_3 + \dots + \mathbf{R}_N)}{N^2} \right\rangle$$

For a Brownian system exhibiting self-similarity without any loss, it can be written as:

$$R_{cm}^2 = \frac{R_{sdpd}^2}{N_m}$$

Calibration in DPD: Matching Length Scale

Relating Simulation and Physical Length Scales

- Unit length in your simulation box corresponds to the number of DPD water molecules:

$$N_{\text{dpd}} = \rho N_m$$

- Let us consider that the DPD water molecule represents a real molecule.
- The volume occupied by ρN_m DPD water molecules is:

$$V_{\text{water}} = \text{volume_per_molecule} \cdot \rho N_m$$

- Assuming a scaling factor, the DPD water volume becomes:

$$V_{\text{dpd-water}} = 30 \cdot \rho N_m \quad (\text{in } \text{\AA}^3 \text{ units}).$$

- Since the unit volume of the simulation box equals $V_{\text{dpd-water}}$, the corresponding unit length is:

$$r_{\text{sim}} = 3.107(\rho N_m)^{1/3} \quad (\text{in } \text{\AA} \text{ units}).$$

Calibration in DPD: Matching Time Scale

In simulating a molecular system, the time scale is calibrated based on the diffusivity of the DPD and water molecules. The equation for the diffusivity is given by:

$$D = \lim_{t \rightarrow \infty} \frac{\langle R^2 \rangle}{2dt}$$

where d is the dimensionality of the system. For a 3D system, this simplifies to:

$$D = \lim_{t \rightarrow \infty} \frac{\langle R^2 \rangle}{6t}$$

Now, consider the following relation between the diffusion of the water molecule and the DPD simulation:

$$\frac{\langle R_{\text{water}}^2 \rangle}{t_{\text{phy}}} = \frac{\langle R_{\text{DPD}}^2 \rangle}{t_{\text{sim}}}$$

Calibration in DPD: Matching Time Scale

However, in your simulation, you are obtaining R_{cm}^2 , the center-of-mass displacement (which is DPD bead: equivalent to N_m DPD-water molecules). By recalling the relation between the displacement of a DPD water molecule and the DPD beads, we have:

$$\frac{\langle R_{\text{water}}^2 \rangle}{t_{\text{phy}}} = \frac{N_m \langle R_{\text{cm}}^2 \rangle}{t_{\text{sim}}} = D_{\text{water}}$$

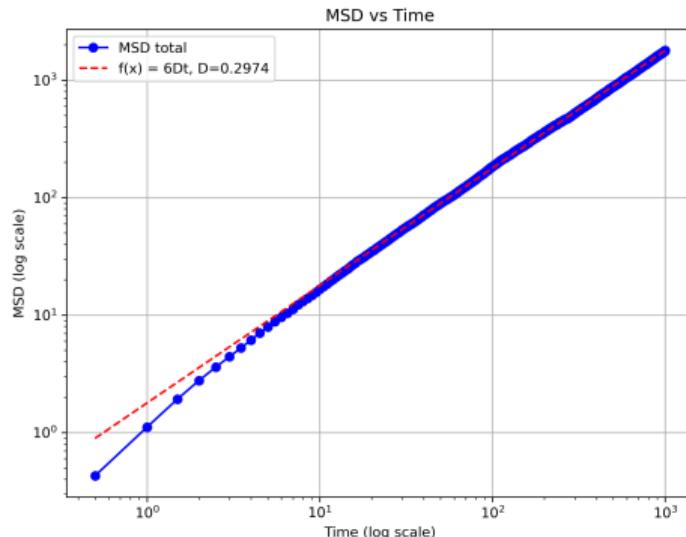
Therefore, the physical time scale t_{phy} is related to the simulation time scale t_{sim} by:

$$t_{\text{phy}} = \frac{N_m \cdot (30.0\rho N_m)^{2/3} \langle R_{\text{cm}}^2 \rangle}{6D_{\text{water}}} \quad (\text{in unit of } \text{\AA}^2/\text{s})$$

Finally, since $t_{\text{sim}} = t_{\text{phy}}$, the unit conversion is:

Unit
$$t_{\text{factor}} = \frac{t_{\text{phy}}}{t_{\text{sim}}}$$

Calibration in DPD: Now, its your turn



Mean squared displacement vs simulation time

$$D_{\text{DPD-beads}} = 0.2974 \text{ Unit } R_{\text{sim}}^2 / 6t_{\text{sim}}$$

Calibration in DPD: Lets DO

Please download the code from the github and then run in your system by using the command

https://github.com/vikkivarma16/lammmps_tutorial

```
lmp -in tutorial.in
```

Filename: tutorial.in

Calibration in DPD: For Time Scale

$$t_{\text{phy}} = \frac{10 \cdot (30.0 \cdot 3 \cdot 10)^{2/3} \cdot 1783.33}{6 \cdot 2.43 \cdot 10^{11}}$$

Let us consider that our DPD bead constitutes 10 DPD-water molecule:

$$D_{\text{water}} = 2.43 \cdot 10^{11} \text{\AA}^2/\text{s}$$

$$N_m = 10$$

$$t_{\text{phy}} = 1.1396 \cdot 10^{-6} \text{s}$$

$$R_{\text{sim}} = 1783.33$$

$$\text{Unit } t_{\text{sim}} = \frac{1.14 \cdot 10^{-6} \text{s}}{999.5}$$

$$t_{\text{sim}} = 999.5$$

$$t_{\text{sim}} = 1.14 \cdot 10^{-9} \text{s}$$

$$\rho = 3$$

Unit t_{sim} or $t_{\text{factor}} = 1.14 \text{ ns}$

Calibration in DPD: A look back

We have the following variables in our simulation considered so far,

Number Density	Number of DPD-water molecule per bead	Force coefficients and constants
ρ	N_m	A and γ

Can we put any value of ρ and N_m and calibrate the time and length scale accordingly, like we just did?

What else, we do need to calibrate?

Calibration in DPD: Thermodynamics

The virial equation of state can be expressed as:

$$\frac{P}{RT} = \frac{1}{V_m} + \frac{B(T)}{V_m^2} + \frac{C(T)}{V_m^3} + \dots$$

Alternatively, in terms of density $\rho = \frac{1}{V_m}$, it becomes:

$$\frac{P}{k_B T} = \rho + B(T)\rho^2 + C(T)\rho^3 + \dots$$

The isothermal compressibility κ_T is related to pressure as:

$$\kappa_T = -\frac{1}{V} \left(\frac{\partial V}{\partial P} \right)_T$$

In terms of density ρ , the isothermal incompressibility I_T is given by:

$$k_T^{-1} = \frac{1}{k_B T} \left(\frac{\partial P}{\partial \rho} \right)_T$$

Calibration in DPD: Thermodynamics

In terms of density ρ , the isothermal incompressibility I_T is given by:

$$k_T^{-1} = \frac{1}{k_B T} \left(\frac{\partial P}{\partial \rho} \right)_T$$

Note: Not to be confused by the compressibility factor Z . Which is given as $p/(\rho k_B T)$. Z tells about the deviation from the ideal gas behavior, where for the ideal gas behavior $Z = 1$. However, qualitatively both scales in the same way.

Calibration in DPD: Thermodynamics

The equation of state ($P-\rho$) relation is calibrated by finding the right conservative force coefficient A , which matches with the dimension-less quantity isothermal compressibility, which is given as,

$$k^{-1} = 16 \text{ for water.}$$

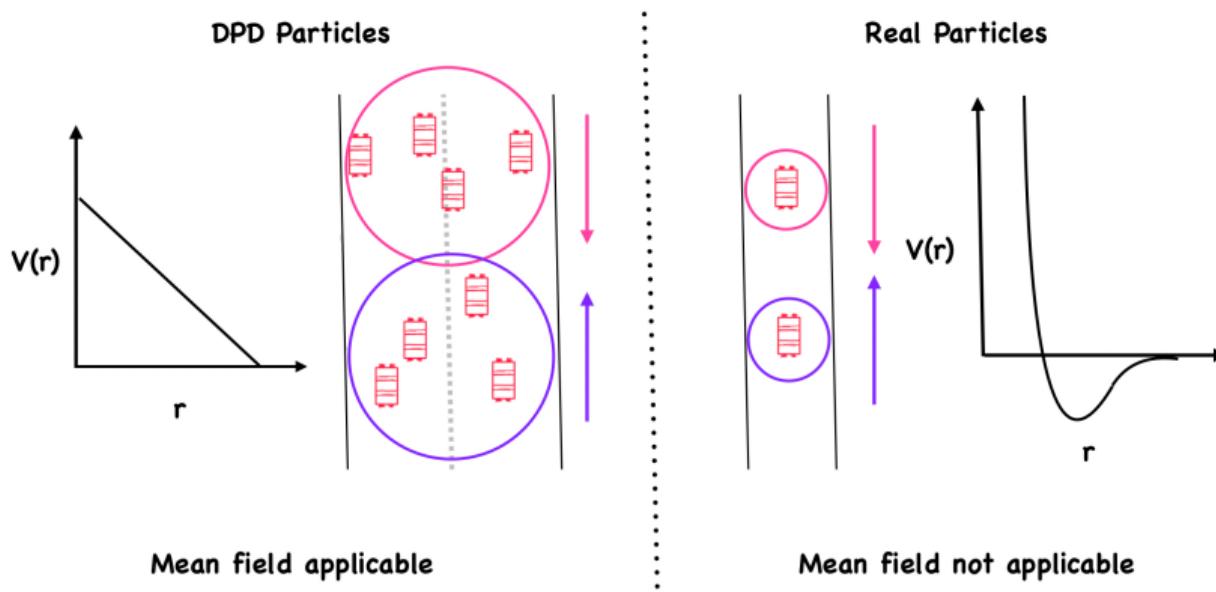
$$k_{\text{dpd}}^{-1} = \frac{1}{k_{\text{B}} T} \frac{\partial p}{\partial \rho} = \frac{1}{k_{\text{B}} T} \frac{\partial \rho_{\text{real}}}{\partial \rho} \frac{\partial p}{\partial \rho_{\text{real}}}$$

$$k_{\text{dpd}}^{-1} = N_m \cdot k_{\text{experimental}}^{-1}$$

$$\text{For } N_m = 10, k_{\text{dpd}}^{-1} = 160.$$

Calibration in DPD: A look back

The soft repulsive forces can be interpreted as something like this....



Calibration in DPD: A look back

The pressure p can be expressed as:

$$p = \rho k_B T + \frac{1}{3V} \left\langle \sum_{j>i} (\mathbf{r}_i - \mathbf{r}_j) \cdot \mathbf{f}_i \right\rangle$$

$$= \rho k_B T + \frac{1}{3V} \left\langle \sum_{j>i} (\mathbf{r}_i - \mathbf{r}_j) \cdot \mathbf{F}_{ij}^C \right\rangle$$

$$= \rho k_B T + \frac{2\pi}{3} \rho^2 \int_0^1 r f(r) g(r) r^2 dr,$$

$$= \rho k_B T + \frac{2\pi}{3} \rho^2 A \int_0^1 r w(r) g(r) r^2 dr,$$

$$p = \rho + \alpha A \rho^2, \text{ in } k_B T \text{ Unit}$$

Let us find out the validity of mean field expression

$$p = \rho + \alpha A \rho^2, \text{ in } k_{\text{B}} T \text{ Unit}$$

Let us open the manual for tutorial I and follow the instructions given in Sec. II.

What did you learn?

For the molecular simulation, you have to match, compressibility, and then the time and the length scale to match diffusivity, while choosing the appropriate DPD density (ρ) and the force constant (a).

Understanding DPD: Why are you concerned with the formalism? (from the previous slides)

In DPD, you are considering to simulate a large chunk of atoms or molecules as a single bead.

There is something compromised.

Where are we compromising?

For the molecular simulation, you have to match, compressibility, and then the time and the length scale to match diffusivity, while choosing the appropriate DPD density (ρ) and the force constant (a).

The consequences:

- In DPD there is no cage effect. So it has mobility much more greater than the actual fluid (almost 1000 times).
- Because of no hard core interaction, it has faster fluid response time and poor momentum transfer.
- Viscosity which measures the rate of momentum transfer becomes very small in compare to the actual fluid.
- To study fluid dynamics, we need to match viscosity rather than diffusivity.

Where are we compromising?

Poorer the mass transfer (Low Diffusivity) → Longer the fluid response time (Low Mobility) → Better the momentum transfer (High viscosity).

Better the mass transfer (High Diffusivity) → Shorter the fluid response time (High Mobility) → Poorer the momentum transfer (Low Viscosity).

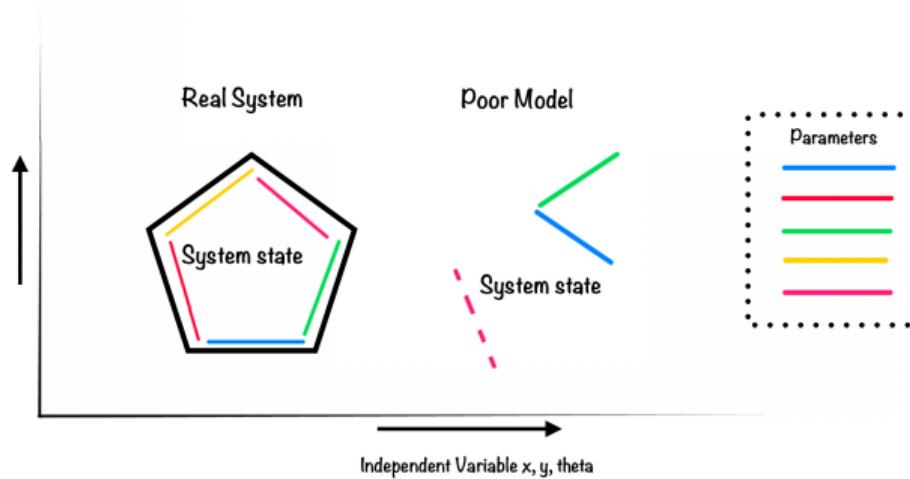
Where are we compromising?

The consequences:

- In DPD there is no cage effect. So it has mobility much more greater than the actual fluid (almost 1000 times).
- Because of no hard core interaction, it has poor fluid response time and poor momentum transfer.
- Viscosity which measures the rate of momentum transfer becomes very small in compare to the actual fluid.
- For the fluid response, and to study fluid dynamics, we need to match viscosity rather than diffusivity.

Understanding DPD: Where does DPD stand so far?

However, due to a significantly great achievable simulation time, conservation of momentum and scaling properties like self-similarity, makes DPD the best choice for many applications.



Calibration in DPD: Treating Binary mixture

The free energy per unit $k_B T$ is given by:

$$\frac{F}{k_B T} = \frac{\phi_A}{N_A} \ln \phi_A + \frac{\phi_B}{N_B} \ln \phi_B + \chi \phi_A \phi_B, \quad (17)$$

where ϕ_A and ϕ_B are the volume fractions of the *A* and *B* components, N_A and N_B are the number of segments per *A* and *B* molecule, respectively, and χ is the Flory-Huggins interaction parameter.

The volume fractions are subject to the condition:

$$\phi_A + \phi_B = 1,$$

which implies $\phi_B = 1 - \phi_A$, making ϕ_A the only degree of freedom.

The location of the minimum free energy for $x = N_A/(N_A + N_B)$ is determined by solving the following implicit equation:

$$\chi N_A = \frac{\ln [(1-x)/x]}{1-2x}. \quad (18)$$

Calibration in DPD: Treating Binary mixture

The volumetric free energy for a single-component system is given as:

$$\frac{f_V}{k_B T} = \rho \ln \rho - \rho + \frac{\alpha a \rho^2}{k_B T}, \quad (20)$$

For a two-component system of chains, it is expressed as:

$$\frac{f_V}{k_B T} = \frac{\rho_A}{N_A} \ln \rho_A + \frac{\rho_B}{N_B} \ln \rho_B - \frac{\rho_A}{N_A} - \frac{\rho_B}{N_B} + \frac{\alpha (a_{AA} \rho_A^2 + 2a_{AB} \rho_A \rho_B + a_{BB} \rho_B^2)}{k_B T}. \quad (21)$$

If we choose $a_{AA} = a_{BB}$ and assume $\rho_A + \rho_B$ is approximately constant, the volumetric free energy becomes:

$$\begin{aligned} \frac{f_V}{(\rho_A + \rho_B) k_B T} &\approx \frac{x}{N_A} \ln x + \frac{(1-x)}{N_B} \ln(1-x) + \chi x(1-x) + \text{constants}, \\ x &= \frac{\rho_A}{\rho_A + \rho_B}, \end{aligned} \quad (22)$$

Calibration in DPD: Treating Binary mixture

Following the mean field treatment, the Flory-Huggins interaction parameter χ is identified as:

$$\chi = \frac{2\alpha(a_{AB} - a_{AA})(\rho_A + \rho_B)}{k_B T} .$$

Let Us find out the relation between χ and ρ

Is χ linear in ρ ? If so, then in what regions?

Please open the DPD manual and follow the instructions given in Sec III to compute χ , at different ρ values.

- : ? : -

Some developmental works to do

- : Thank you so much for attending the tutorial session : -

Your feedback is greatly appreciated

Please follow the below section to develop better understanding for the LAMMPS.

LAMMPS: An Introduction

What is Lammps

LAMMPS : Large-Scale Atomic and Molecular Massively Parallel Simulator It is a package that simulates atomic and molecular systems using a classical approach, dealing with the physical quantities such as:

- Force
- Energy
- Harmonic Bonds
- Conservative Force Field
- Random Force
- Friction
- Coulomb Charge (usually not applicable in DPD)
- The emergent properties like, Viscosity, Stress, Strain and Pressure etc.

LAMMPS uses classical equations of motion and can handle conditions such as:

- Conservation of momentum
- Conservation of energy
- Conservation of the number of particles
- Changes in energy (e.g., through heating or cooling)
- Changes in momentum (e.g., due to external forces)
- Changes in the number of particles (e.g., through reactions)

From the Kingdom of Discrete Particle Models.
Belongs to the Province of Molecular Dynamics.

Where do we encounter these conditions while modeling a system using desecrate particle approach?

What kind of problem can be solved using LAMMPS equipped with these features?

Recommended Books for Concept Development (for various simulation techniques and fluid dynamics):

- *Understanding Molecular Simulation: From Algorithms to Applications* by Daan Frenkel and Berend Smit
- *Computer Simulation of Liquids* by M. P. Allen and D. J. Tildesley

Understanding the System

Example: What are the length and time scales of the problem you are aiming to study?

- **Ballistic motion:** Describes the motion of particles without any collisions.
- **Mean free path:** Average distance between particle collisions.
- **Brownian motion:** Random motion of particles suspended in fluid.
- **Viscous flow:** Fluid flow at the mesoscopic scale, governed by viscosity.
- **Turbulent flow:** Chaotic fluid motion at larger scales.
- **Continuum model:** Describes fluid behavior using continuous fields, without discrete particles.

Understanding the System

For the Theory and Physics Geeks: They aim to find the best model.

What Makes a Model the Best? A model that captures microscopic features and shows their collective effects at the macroscopic level.

Key Consideration: The model depends on the type of problem being studied.

Example: Simulating the motion of a bacterium swimming in water requires a model with these features:

- Conserves momentum at the local scale
- Keeps the temperature constant (*Total energy can change*)
- Maintains a constant number of particles
- Captures hydrodynamic effects (*Essential for simulating a swimmer*)

Looking for a Suitable Model

How to Check if your model Captures the Desired Effect?

- **Mathematical Proof:** Analytical constructions which demonstrates the model's behavior across different length scales.
- **Application Search:** Find practical applications where the model is relevant and match with the experiment quantitatively.

Example: Specify the condition for the movement
and the collision



Produces Navier-Stokes Equations
(Hydrodynamic Effect)

Looking for the suitable Model

If you find a suitable model then look what kind of quantities can be studied and calculated using the model.

Here it comes about your simulation method.

What are the available features in **LAMMPS**?

Literature survey can be helpful to find out the suitable model.

- Finding out the studies done previously on the similar kind of system.

Terminology in LAMMPS

Purpose:

Understanding terminology helps in addressing and using various models and features available in LAMMPS.

Origins of Terminology: These terms originate from the underlying theory behind the models implemented in LAMMPS.

Example: A system with the following features is called an **NVT Ensemble**:

- Conserves momentum at the local scale
- Keeps temperature constant (*Total energy may change*)
- Maintains a constant number of particles
- Captures hydrodynamic effects (*Critical for simulating swimmers*)

Terminology in LAMMPS

Important Concepts and Terms:

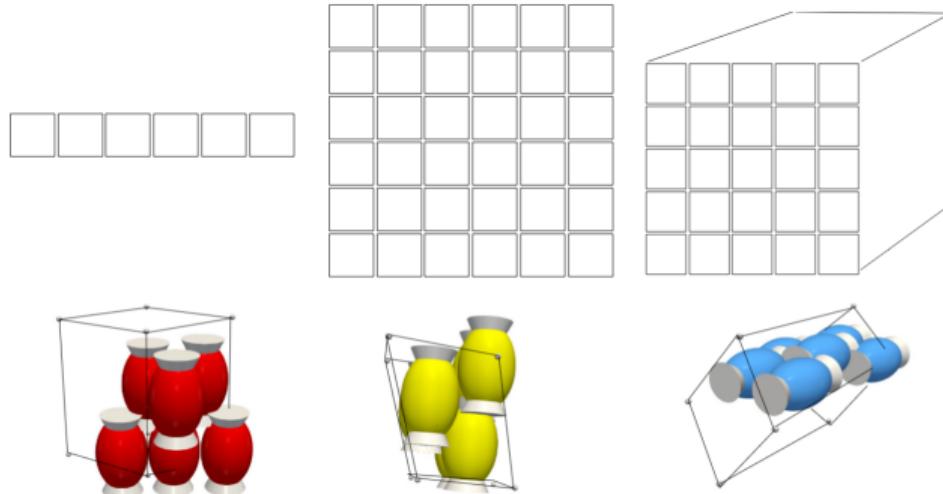
- Ensembles
- Dimension
- Periodic Boundary Conditions
- Particles
- Bonds
- Interaction Potential

Terminology in LAMMPS

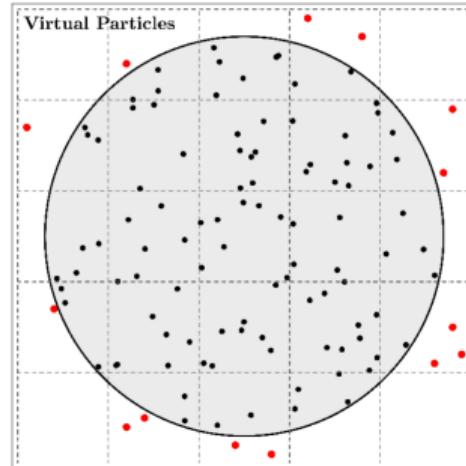
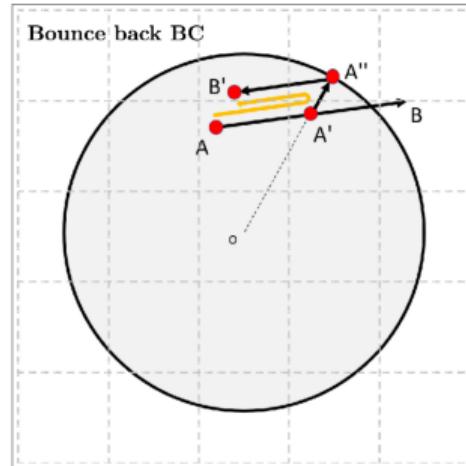
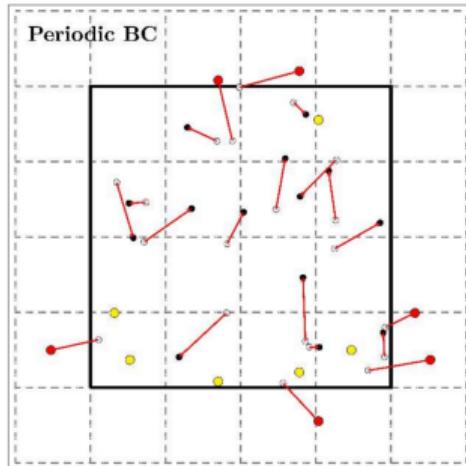
Ensembles: NVT, NpT, NhT

Terminology in LAMMPS

Dimension



Periodic Boundary Condition



Various periodic boundary conditions

Terminology in LAMMPS

Key Terms: Atom, Molecule, Rigid Body, Pair Potential, Sphere, Omega (all in its classical definition)

CAUTION:

- **Not Real Atoms or Molecules:** These terms refer to data structures representing particles, not actual atoms or molecules.
- **No Quantum Mechanics:** Calculations do not involve quantum fields, potentials, or quantum mechanics principles.
- **No Electron Clouds:** There is no modeling of electron clouds or quantum iterations.
- **No Iterative Solutions:** Unlike Quantum or Classical DFT methods, LAMMPS does not perform background numerical iterations.

LAMMPS provides a rich set of tools and options for simulations:

- **Simulation Box:** Create boxes of various shapes and sizes for different systems.
- **Particle Properties:** Flexibility to define molecules (bonded particles) and atoms (single particles) with a wide range of properties.
- **Ensemble Selection:** Run simulations in different ensembles such as NVT or NPT.
- **Customization Options:** Customize calculations to enhance efficiency and speed.
- **Built-in Functions:** Simulate specific systems like electrolytes, liquid crystals, and charged particles.

Note: Customization in LAMMPS allows for greater flexibility in modeling complex systems.

Solving Problems in LAMMPS

Solving problems in LAMMPS often follows these steps:

1. **Define the Problem:** Identify the problem and select a suitable model.
2. **Break Down the Problem:** Decompose the problem into components that can be input into LAMMPS.
3. **Run the Simulation:** Use LAMMPS to simulate and gather data.
4. **Analyze and Adjust:** Analyze the results, correct the code, and make necessary changes to the model.
5. **Iterate:** Repeat the process until the desired results are achieved.

Note: Iteration is a crucial part of refining simulations in LAMMPS.

Breaking the Problem Down:

Analyze the problem using a discrete particle model and the tools provided in LAMMPS.

- **System Requirements:** Use a system with constant volume, temperature, and number of particles (**NVT Ensemble**).
- **Defining Hard Spheres:** Specify a pair potential that ensures particles behave as hard spheres.
- **Implementation in LAMMPS: Example:** Apply a condition to prevent particles from being closer than $r_1 + r_2$, regardless of the vector direction $\mathbf{r}_1 - \mathbf{r}_2$.
- **Preparing NVT Simulation Box:** Initialize the box at various densities to study the freezing behavior.

...Parametrization of your problem

In other words you need to ensure about parameters which you are going to use in your model?

Which can be associated with attributes like:

1. **Particle's properties** (Intra-particle interaction)
2. **Interaction potential** or pair potential (Inter particle interaction)
3. **Thermodynamic parameters** like Temperature, Pressure, Density.
4. **System size** e.g, Size and shape of the simulation box, number of particles (**CAUTION: Always check...** if the properties of system being studied are system size independent before reporting for the evaluation in a journal or to your supervisor or project manager.)

Some Tricks of the Trade

Particle Properties: The properties depend on how much **degree of freedom** you want to introduce into your system.

Example: If the exchange of energy between the bond in a dimer and the surrounding fluid is not important, you can treat the dimer as a monomer (a single particle). This technique is known as **coarse-graining**.

Caution: While performing coarse-graining, do not overlook the importance of **entropy**!

Particle Specification: These attributes can be easily defined in LAMMPS:

- Number of atoms (minimum feature at least you expect from a package)
- Types of bonds among constituent particles
- Bond types: Normal radial bond, Angular bonds, Improper dihedral bonds, Dihedral bonds

Please Do Not Waste Resources

Be cautious: Using unnecessary features can consume resources without providing any benefit.

Example of Inefficient Usage:

If the exchange of energy between bonds and pair potentials plays no role, avoid using functions like:

`delete_bonds, create_bonds`

Better Approach:

Consider treating the system as a **rigid body** made of multiple particles without defining bonds among the constituent particles.

If No Alternative: Write your own code or develop a custom LAMMPS module by following the instructions on the official LAMMPS website.

Example LAMMPS Input Script

LAMMPS Input File with .in or .lammmps extension... run command: lmp -in input.in

```
# LAMMPS Simulation
# Section 1

units lj
atom_style atomic
dimension 3
boundary p p p
pair_style lj/cut 4.6
```

Example LAMMPS Input Script

go to the URL https://github.com/vikkivarma16/lammps_tutorial.git

please click on the **code** button

[Download ZIP](#)

Note: You can copy and paste the script into ChatGPT to learn the purpose of each command

Task I

Create two types of atoms.

The key line

Run command: lmp -in task_1.in

```
create_box 1 my_box
```

Hint: LAMMPS stores the information of different grids corresponding to different particle types, nearest neighbor, particle identity and bonds identity, all through the create box command. Making necessary allocation in the create box command is crucial. Otherwise your code will show an error.

Task II

Create two types of atoms in cylindrical regions, type 1 inside the cylinder and type two outside the cylinder.

The key line

Run command: lmp -in task_2.in

```
create_atoms 1 random 1000 341341 simulation_box
```

Hint: Play with the region defined in the create_atoms command.

Task III

Make any changes to create 5 atoms. Currently your script is able to create only 4 atoms.

The key line

Run command: lmp -in task_3.in

```
region my_box block 0 4.0 0 4.0 0 4.0  
or  
create_atoms 1 single 1.0 1.0 1.0
```

Hint: LAMMPS can not create atoms out side the defined region or box region. So either change the coordinate of the particles or change the range of the box region.

Task IV

Create a bond between atom id-1 and atom id-5.

The key line

Run command: lmp -in task_4.in

```
create_box 2 box bond/types 1 extra/bond/per/atom 3 ...
...extra/special/per/atom 3
```

Hint: LAMMPS stores the information of bonds identity, through the create box command. Making necessary allocation in the create box command is crucial. **extra/special/per/atom 3** command is there to take care about the next nearest neighbor where you can exclude the interactions, in case if you are making a rigid body or molecule and you don't want to include intra-particle interactions.

Task V

Change the polymer from linear to circular and visualize in OVIT. (use python to create ring coord)

The key line

Run command: lmp -in task_5.in

FILE: task_5_polymer.mol

Hint: Remember to change the co-ordinate of the constituent particles in the file, so that bond can be generated without creating any nonphysical situation. For example, in a ring when you connect last particle to the first, it should be located close to the first particle. First generating the co-ordinate of ring particle is necessary. Make sure that the distance between the particle is as same as the length of the harmonic spring defined in your bond coefficient command.

Task VI

Use the same polymer coord file and run the code. What do you observe, the polymer is rigid?

The key line

Run command: lmp -in task_6.in

```
molecule polymer task_5_polymer.mol
```

Hint: A rigid body integrator (`rigid/_small`) treat an object with the same molecular id as a rigid body.

Task VII

Why your DPD particles are not moving?

The key line

Run command: lmp -in task_7.in

```
fix integ all nve
```

Hint: In LAMMPS, particles are moved using integrators. NVE is the constant energy integrator, NVT is constant temperature integrator and so on. Without integrator your particle will not move therefore ensure to use the integrator.

Task VIII

What is wrong with this code, why are you not able to achieve the given temperature?

The key line

Run command: lmp -in task_8.in

```
fix vikki_thermo_state all nvt temp 4 4 0.1
```

Hint: You are using DPD which is by formalism thermostated. However, if you will apply NVT, then the integrator (NVT) have a thermostat included. So in effect, there are two thermostats being applied which can cause nonphysical situation. Therefore be aware with the integrators and fixes. Use right integrator and a single thermostat, if you want to use one. For example, with DPD, NVE integrator is applicable since it has no thermostat.

Task IX

Load the task_9_last_output.data file using python and calculate the velocity distribution and find out whether it is Gaussian or not?

However the script provided to you is wrong as there are two thermostats are being used at the same time (one from DPD and other from NVT), and you have to find if there is some discrepancies arising at the statistical level or it is just the wrong effective temperature due to NVT specifications.

The key line

Run command: lmp -in task_9.in

find the key line yourself

Task X

Use DPD to thermostat an LJ fluid?

The key line

Run command: lmp -in task_10.in

```
comm_modify vel yes #important to communicate velocity between the atoms,  
...along with the position, in dpd simulation to calculate forces  
pair_style hybrid/overlay lj/cut 3.2 dpd 4 3.2 3854262  
pair_coeff 1 1 lj/cut 1.0 1.0 2.5  
pair_coeff 1 1 dpd 10.0 4.5 1.5  
fix vikki_inet all nve
```

Congratulations you have completed all the tasks !

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

Key Takeaways

- DPD in LAMMPS is useful for mesoscale simulations.
- Choosing correct parameters are crucial.
- System size effect must be checked.

Questions?

Thank you!
Questions and discussion welcome.