A brief introduction to LAMMPS

Reusing this material

The basics

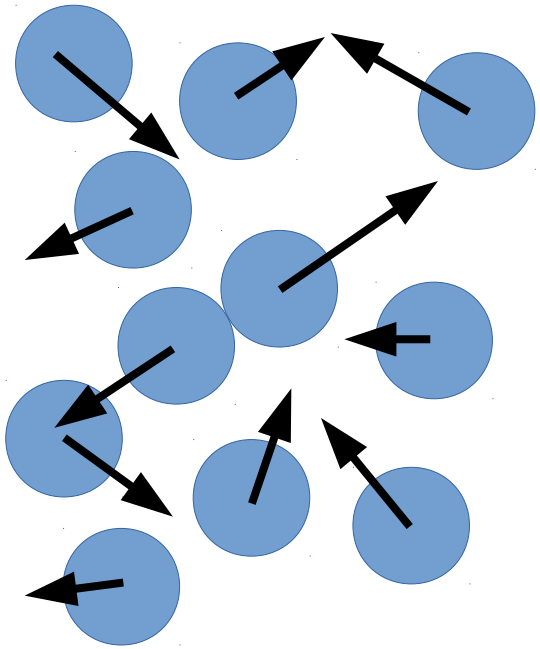
- LAMMPS – Large-scale Atomic/Molecular Massively Parallel Simulator
  - Developed by Sandia National Laboratories and by its wide user-base
  - Can be downloaded from:
    https://lammps.sandia.gov/download.html
- Everything we are covering today (and a lot of other info) can be found in the LAMMPS User Manual:
  https://lammps.sandia.gov/doc/Manual.html

Overview of this session

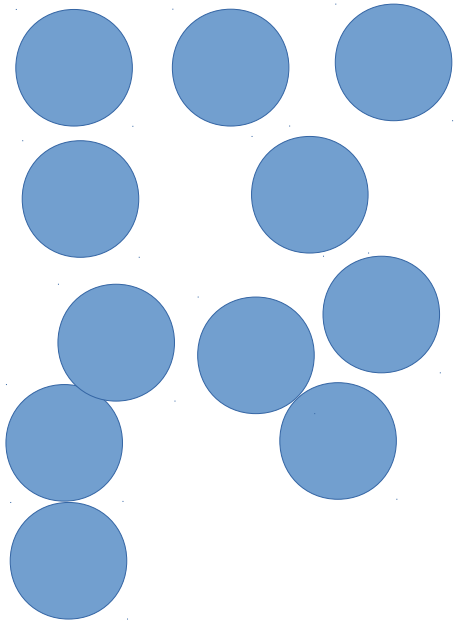
- Run first LAMMPS simulation on ARCHER
  - Briefly show how to run simulation locally
- Preparing a simulation:
  - How is simulation box prepared
  - How are particle interactions defined
  - Neighbour lists *etc*
- Coffee break
- Running a simulation
  - Fixes, computes, and variables
  - What can be output?
- Advanced options
- Questions & quick exercise

How molecular dynamics works

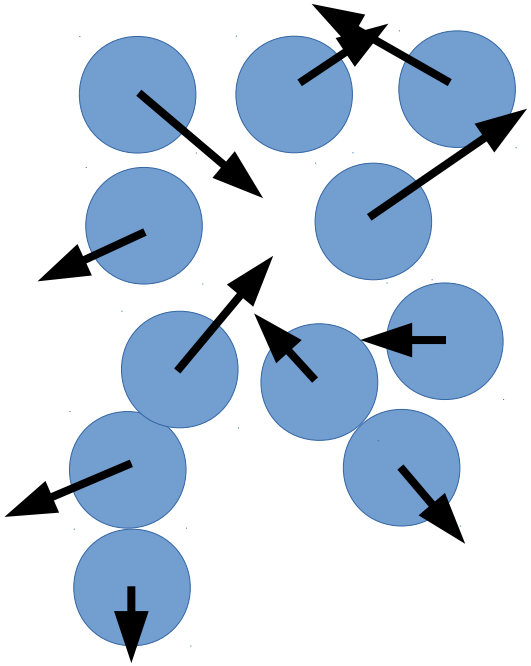
Start with all particle positions and velocities at time t



Update particle positions to time $t+\delta t$

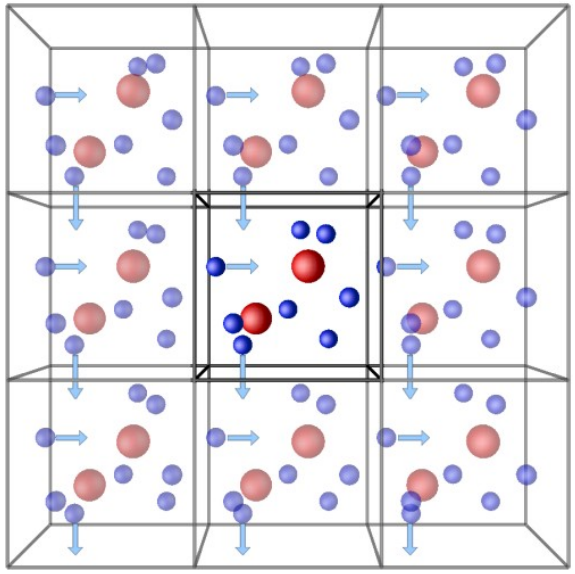


From new positions, find new potential energies, forces, and velocities at time $t+\delta t$

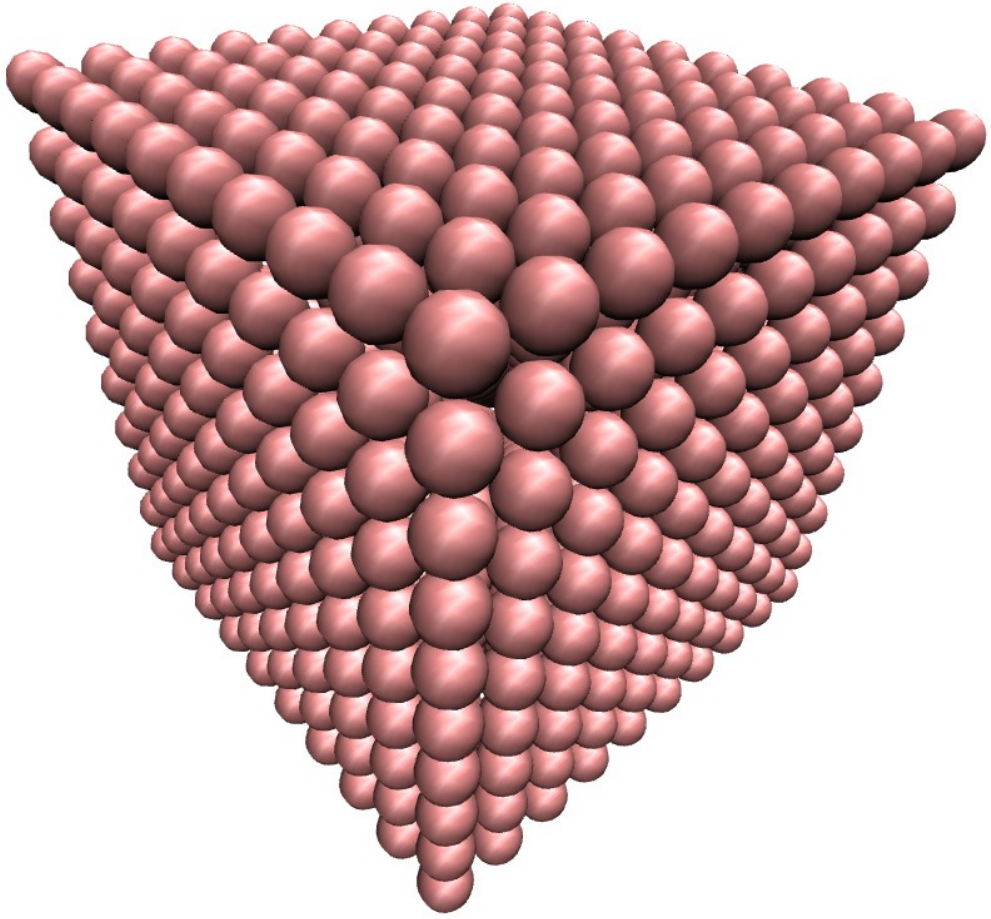


1) Simulation box setup

Periodic boundary conditions

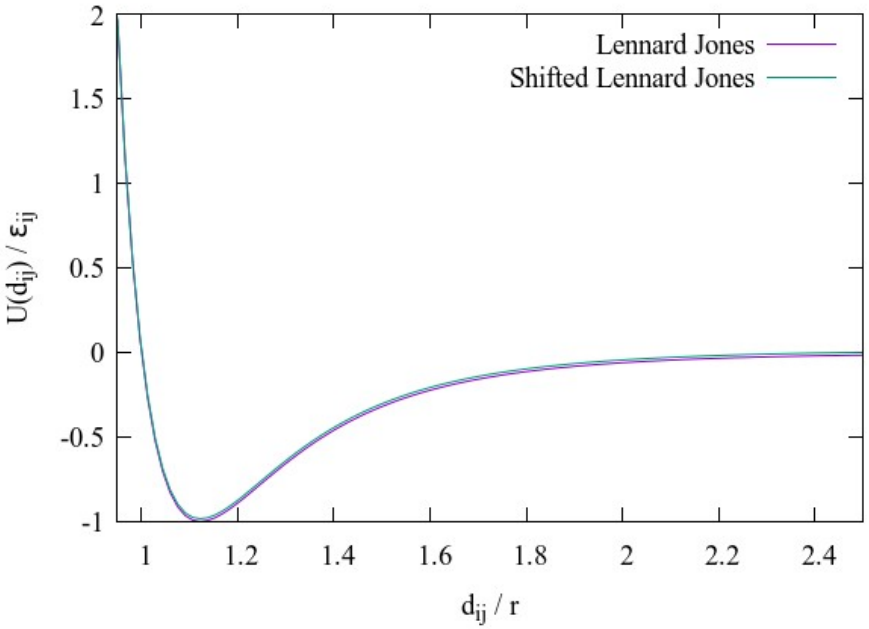


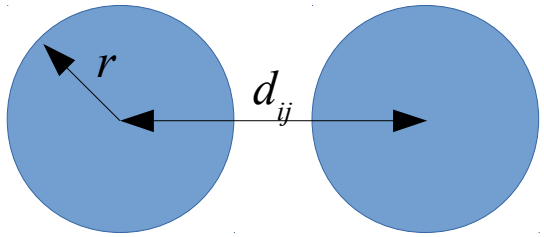
Starting simulation setup (ρ ~ 0.4)



2) Interparticle interactions

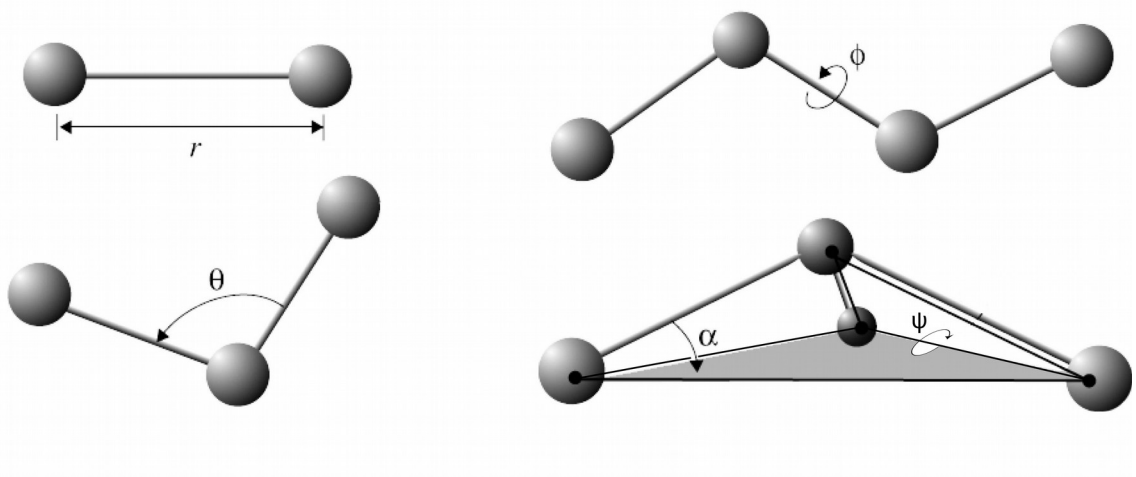
- There are 221 different pair styles in LAMMPS



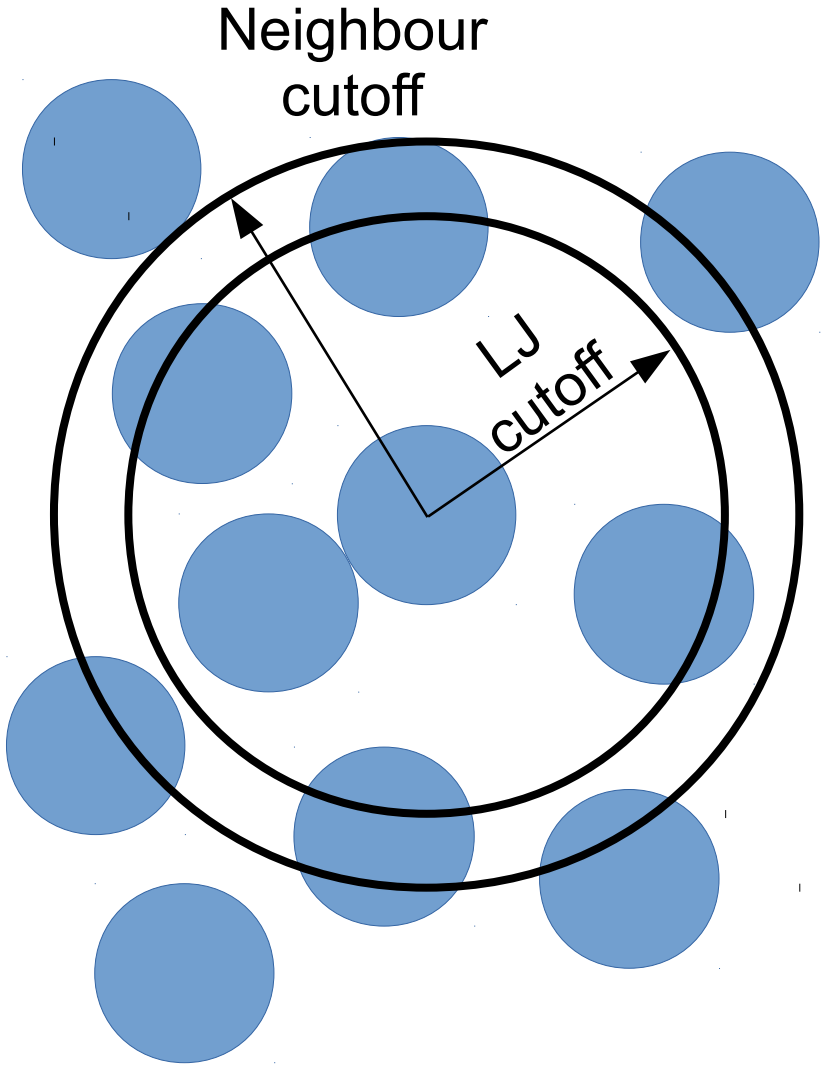


Not discussed here, but LAMMPS also has:

- 17 bond_style commands
- 22 angle_style commands
- 17 dihedral commands
- 13 improper_style commands
- Etc...



3) Neighbour lists



4) Fixes and computes

- 264 fixes
  - This sets general parameters
    - Temerature; pressure; gravity; viscosity; background flows; *etc.* defined by fixes
  - Our simple example has 2:
    - Fix NVT ensemble with temperature $T = 1.0$
    - Fix background flow to 0 (not needed)
- 138 computes
  - Where LAMMPS shines – these commands enable you to output ensemble or per-particle properties of interest

5) Simulation run and outputs

From known velocities and forces, update positions

$$\boldsymbol{r}(t+\delta t)=\boldsymbol{r}(t)+\dot{\boldsymbol{r}}(t)\,\delta t+\frac{1}{2M}\boldsymbol{F}(t)\,\delta t^2+\boldsymbol{O}(\delta t^3)$$

From known forces, update velocities

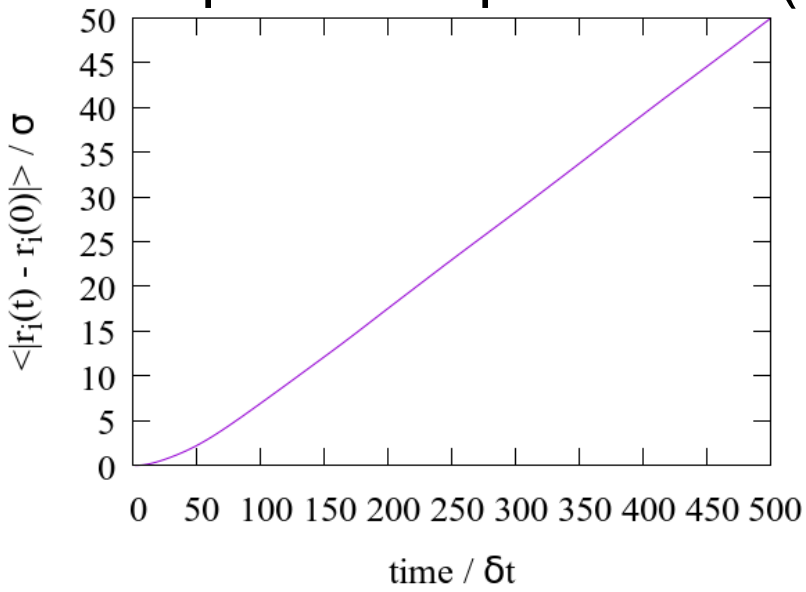
$$\dot{\boldsymbol{r}}(t+\delta t)=\dot{\boldsymbol{r}}(t)+\frac{\delta t}{2M}\left[\boldsymbol{F}(t)+\boldsymbol{F}(t+\delta t)\right]+\boldsymbol{O}(\delta t^3)$$

From known positions, update energies and forces

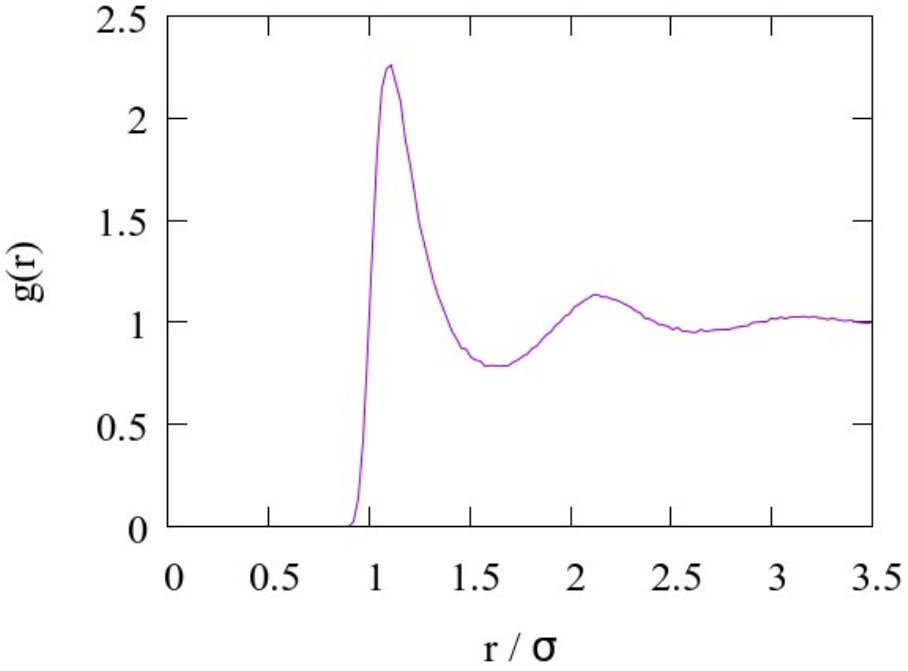
$$\boldsymbol{F}(t+\delta t)=\frac{-1}{|\boldsymbol{r}|}\nabla U\left[\boldsymbol{r}(t+\delta+t)\right]$$

6) Advanced inputs

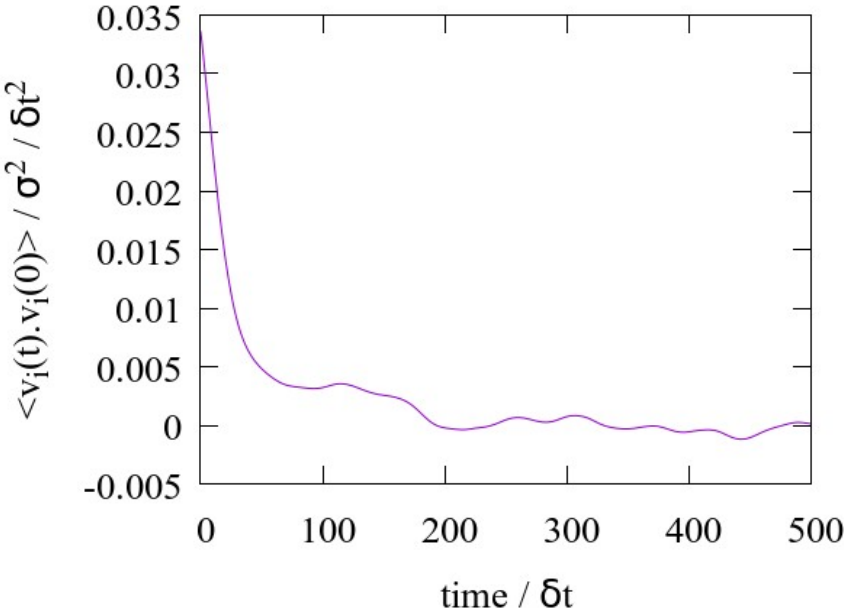
Mean-squared dispacement (MSD)



Radial distribution functions (RDFs)

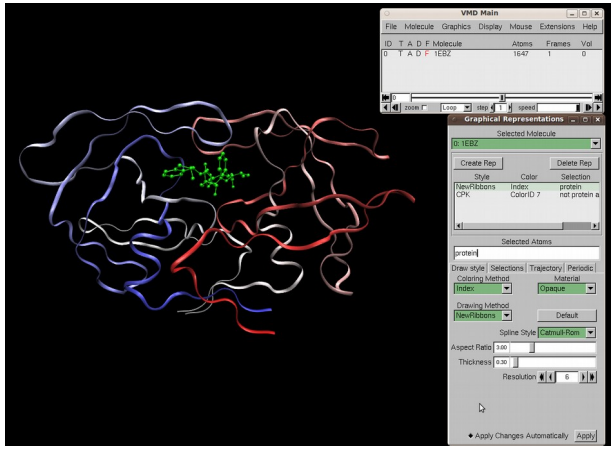


Velocity autocorrelation function (VACF)

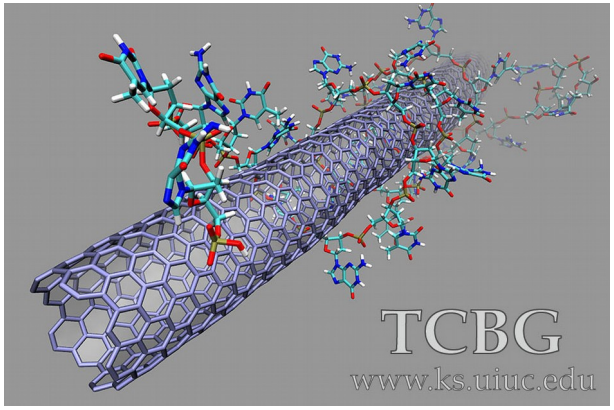


7) Useful third-party software

Visual molecular dynamics (VMD):
https://www.ks.uiuc.edu/Research/vmd/

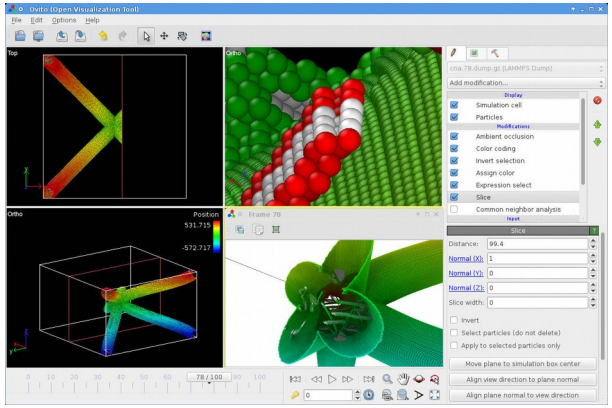


VMD GUI – Wikipedia



DNA binds to single-walled nanotube Xueqing Zou, 2013

Open Visualization Tool (OVITO):
https://www.ovito.org/



OVITO GUI – Wikipedia

Some Python pre- and post-processing packages:

- Pizza.py – https://pizza.sandia.gov/
- MDAnalysis – https://www.mdanalysis.org/
- PyLAT – https://github.com/MaginnGroup/PyLAT

8) Optional user exercise

- You have access to a LAMMPS input script called ‘in.lj_exercise’
  - This is the same script as before with two changes:
    - The starting density is defined by the variable ‘DENSITY’
    - A radial distribution function is output in the main simulation run
- For this exercise, you will look at how varying the system density affects the RDF
  - You should only vary the value of the variable ‘Density’
  - In particular, I recommend looking at the RDFs output at densities ρ = 0.05, 0.8, 1.4

Overview for next session

- Questions from this session
- Compiling LAMMPS on Archer (similar to local compilation)
  - Packages, libraries, etc.
- Smarter domain decomposition commands
- Delving deeper into LAMMPS source code