Lammps Tutorial

Rui Ma

26.05.2022

Outline

- General introduction of LAMMPS
- Core files in LAMMPS and their structure
- A visualization software
- An example

Short introduction to Lammps

- Acronym for **L**arge-scale **A**tomic/**M**olecular **M**assively **P**arallel **S**imulator.
- It is not the only package for Molecular Dynamics (MD) simulations
- Has big user community (google-able for problems)
- Primarily a molecular dynamics code, but can be used for energy minimization, monte carlo or dissipative particle dynamics

Short introduction to Lammps

- Can model systems from a few particles up to billions of particles
- Freely available, open-source code
- Current version is written in C++
- Can run on single-processor laptop/desktop, but optimized for clusters
- …

Conventions of Lammps

Cubic box & Coordinate system

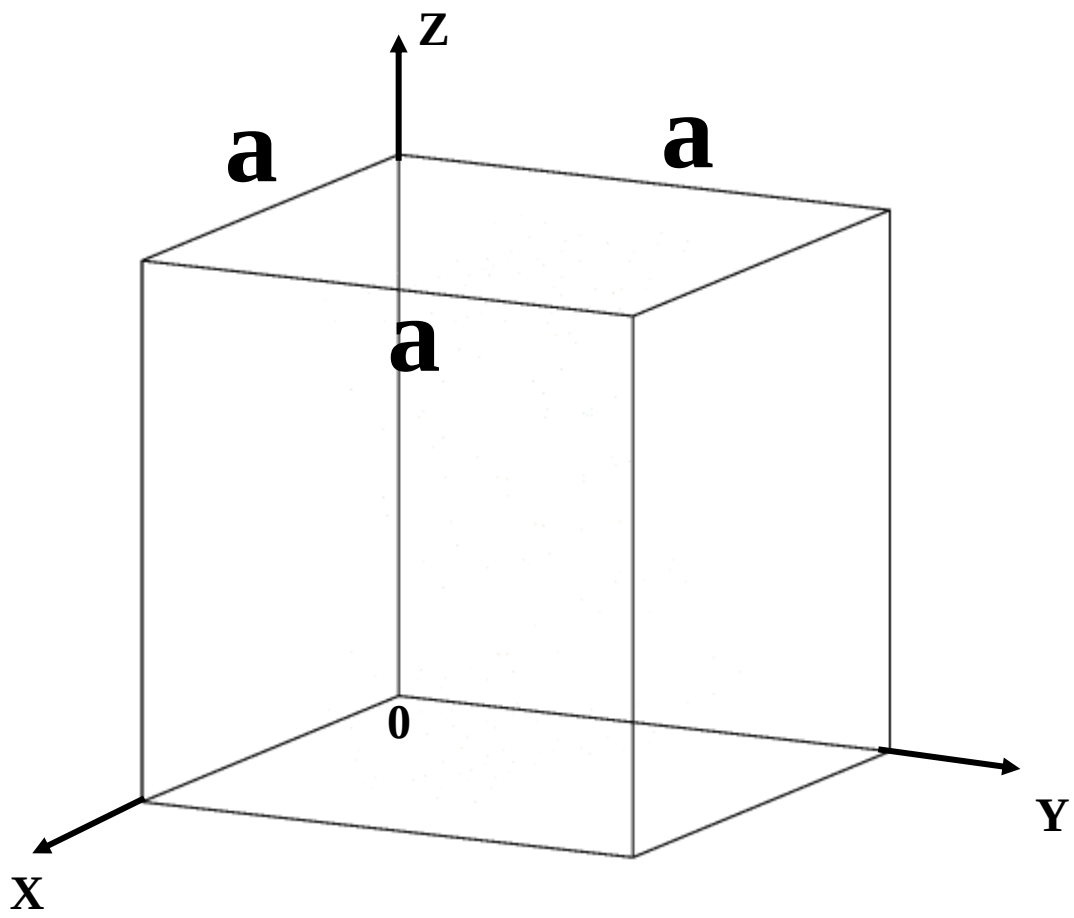
xlo, ylo, zlo can always be chosen as

being zero, so that:

xlo = 0, xhi = a

ylo = 0, yhi = a

zlo = 0, zhi = a



Conventions of Lammps

Cubic box & Coordinate system

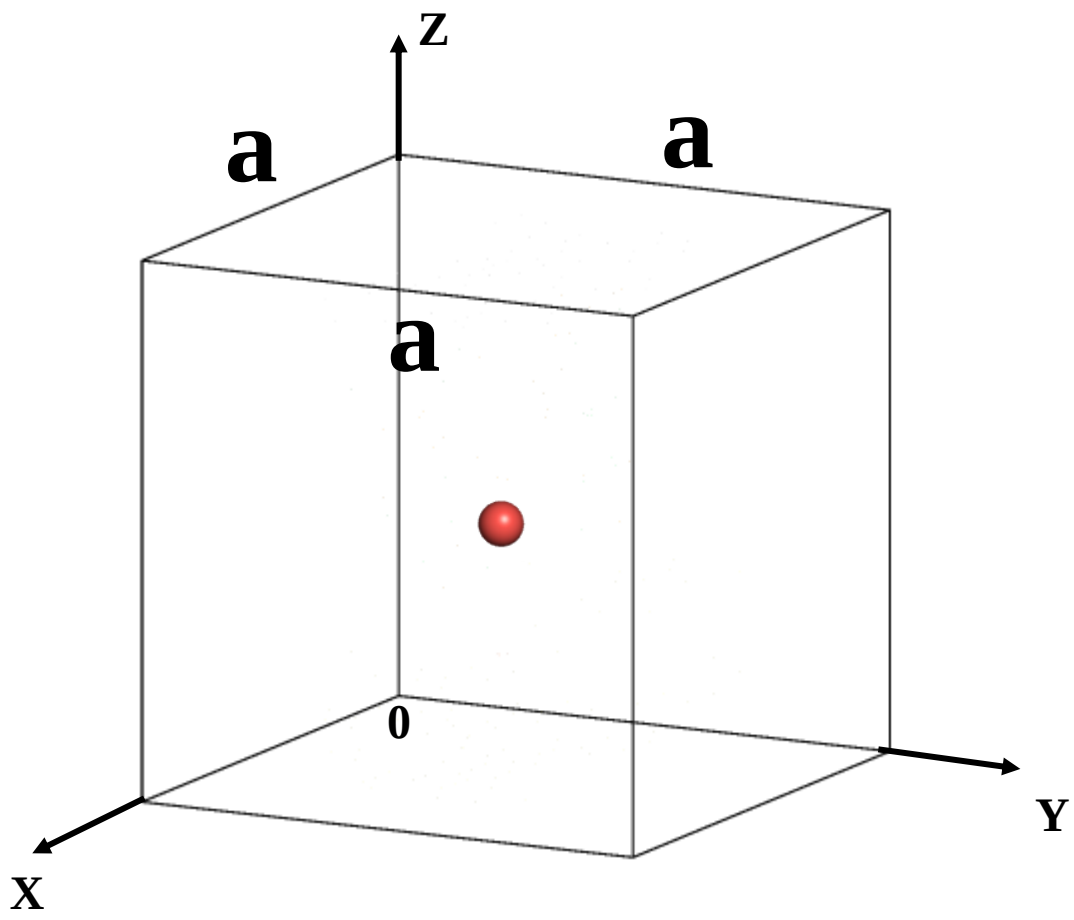
xlo, ylo, zlo can always be chosen as

being zero, so that:

xlo = 0, xhi = a

ylo = 0, yhi = a

zlo = 0, zhi = a

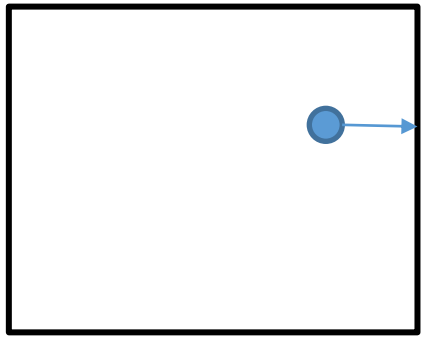


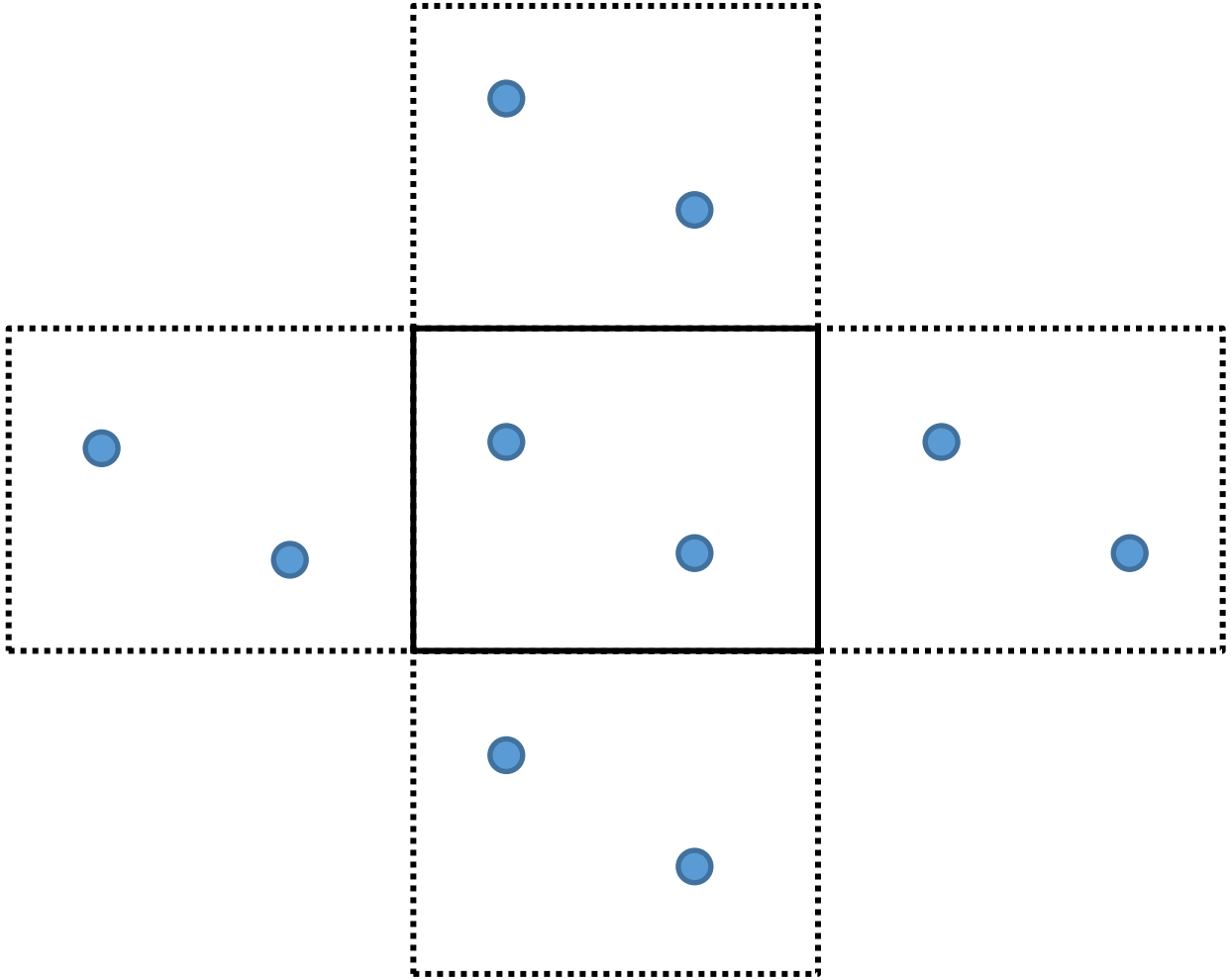
Conventions of Lammps

Periodic boundary conditions

P: periodic

S: non-periodic





Conventions of Lammps

Units

For style *real*, these are the units:

- mass = grams/mole
- distance = Angstroms
- time = femtoseconds
- energy = Kcal/mole
- velocity = Angstroms/femtosecond
- force = Kcal/mole-Angstrom
- torque = Kcal/mole
- temperature = Kelvin
- pressure = atmospheres
- dynamic viscosity = Poise
- charge = multiple of electron charge (1.0 is a proton)
- dipole = charge*Angstroms
- electric field = volts/Angstrom
- density = gram/cm^dim

For style *metal*, these are the units:

- mass = grams/mole
- distance = Angstroms
- time = picoseconds
- energy = eV
- velocity = Angstroms/picosecond
- force = eV/Angstrom
- torque = eV
- temperature = Kelvin
- pressure = bars
- dynamic viscosity = Poise
- charge = multiple of electron charge (1.0 is a proton)
- dipole = charge*Angstroms
- electric field = volts/Angstrom
- density = gram/cm^dim

General process of Lammps

Input files

File structure

- Each new simulation better be run in a new directory
- This directory should contain some required **input files** containing **information about the system** to simulate.
- The simulation will generate a list of output files containing the outcomes of the simulation

Input files

List of typical input files

- **structure.data-data file**: contains the size of the system, the number of atoms, the number of atom types, the mass of each type, the position of each atom (and potentially their velocity).
- **in.filename-inputfile**: contain the instructions of the tasks to perform and related parameters.
- **Ni.eam-potential file** (when necessary): information about the potential

Input files

Typical data file

```
# Polymer Chain data file

        20 atoms
        19 bonds

        1 atom types
        1 bond types

        0 20 xlo xhi
        0 20 ylo yhi
        0 20 zlo zhi

Masses

        1 1.00

Atoms  # bond ( atom-ID molecule-ID atom-type x y z)

        1   1    1   5.0    2.0    10.0
        ….

Bonds # bond-ID bond-type atomID1 atomID2)

        1   1   1   2
```

Input files

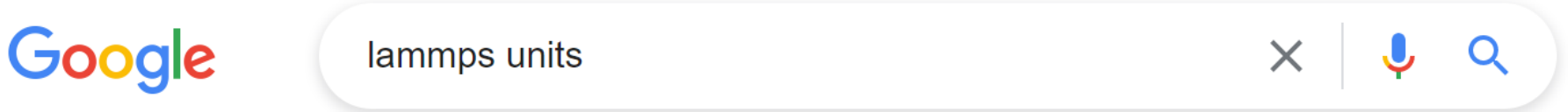
Typical input file

```
# Polymer Chain input file
units               lj
boundary     p p p
atom_style          bond
read_data           datafile

pair_style          lj/cut   1.122461
pair_modify         shift   yes
pair_coeff          1  1  1.0   1.0   1.122461
bond_style          fene
bond_coeff          * 30.0 1.5 1.0 1.0

dump                1 all atom 100 md.lammpstrj id type x y z
timestep            0.01
minimize       1.0e-8 1.0e-8 1000 10000

fix                 1 all nvt temp 1000 1000 100
run                 50000

write_data   100.data
```

Input files

https://docs.lammps.org/Manual.html

Typical input file



units command - LAMMPS documentation

This command sets the style of units used for a simulation. It determines the units of all quantities specified in the input script and data file, as well as ...

https://lammps.sandia.gov › doc › units

units command — LAMMPS documentation

This command sets the style of units used for a simulation. It determines the units of all quantities specified in the input script and data file, as well as ...

https://www.smcm.iqfr.csic.es › docs › lammps

units command

For style lj, all quantities are unitless. Without loss of generality, LAMMPS sets the fundamental quantities mass, sigma, epsilon, and the Boltzmann constant = ...

https://www.afs.enea.it › doc17 › html › units

units command — LAMMPS documentation - AFS ENEA

This command sets the style of units used for a simulation. It determines the units of all quantities specified in the input script and data file, as well as ...

» Commands » units command

Website Commands

Previous Next

units command

Syntax

```
units style
```

- style = *lj* or *real* or *metal* or *si* or *cgs* or *electron* or *micro* or *nano*

Examples

```
units metal
units lj
```

Description

This command sets the style of units used for a simulation. It determines the units of all quantities specified in the input script and data file, as well as quantities output to the screen, log file, and dump files. Typically, this command is used at the very beginning of an input script.

For all units except *lj*, LAMMPS uses physical constants from www.physics.nist.gov. For the definition of Kcal in real units, LAMMPS uses the thermochemical calorie = 4.184 J.

The choice you make for units simply sets some internal conversion factors within LAMMPS. This means that any simulation you perform for one choice of units can be duplicated with any other unit setting LAMMPS supports. In this context "duplicate" means the particles will have identical trajectories and all output generated by the simulation will be identical. This will be the case for some number of timesteps until round-off effects accumulate, since the conversion factors for two different unit systems are not identical to infinite precision.

To perform the same simulation in a different set of units you must change all the unit-based input parameters in your input script and other input files (data file, potential files, etc) correctly to the new units. And you must correctly convert all output from the new units to the old units when comparing to the original results. That is often not simple to do.

Potential or table files may have a `UNITS:` tag included in the first line indicating the unit style those files were created for. If the tag exists, its value will be compared to the chosen unit style and LAMMPS will stop with an error message if there is a mismatch. In some select cases and for specific combinations of unit styles, LAMMPS is capable of automatically converting potential parameters from a file. In those cases, a warning message signaling that an automatic conversion has happened is printed to the screen.

For style *lj*, all quantities are unitless. Without loss of generality, LAMMPS sets the fundamental quantities mass, σ, ϵ, and the Boltzmann constant $k_B = 1$. The masses, distances, energies you specify are multiples of these fundamental values. The formulas relating the reduced or unitless quantity (with an asterisk) to the same quantity with units is also given. Thus you can use the mass, σ, and ϵ values for a specific material and convert the results from a unitless LJ simulation into physical quantities. Please note that using these three properties as base, your unit of time has to conform to the relation $\epsilon = \frac{m\sigma^2}{\tau^2}$ since energy is a derived unit (in SI units you equivalently have the relation $1\mathrm{J} = 1\frac{\mathrm{kg \cdot m^2}}{\mathrm{s^2}}$).

Input files

Typical input file

```
# Polymer Chain input file
units               lj
boundary     p p p
atom_style          bond
read_data           datafile

pair_style          lj/cut   1.122461
pair_modify         shift   yes
pair_coeff          1  1  1.0   1.0   1.122461
bond_style          fene
bond_coeff          * 30.0 1.5 1.0 1.0

dump                1 all atom 100 md.lammpstrj id type x y z
timestep            0.01
minimize       1.0e-8 1.0e-8 1000 10000

fix                 1 all nvt temp 1000 1000 100
run                 50000

write_data   100.data
```

Output files

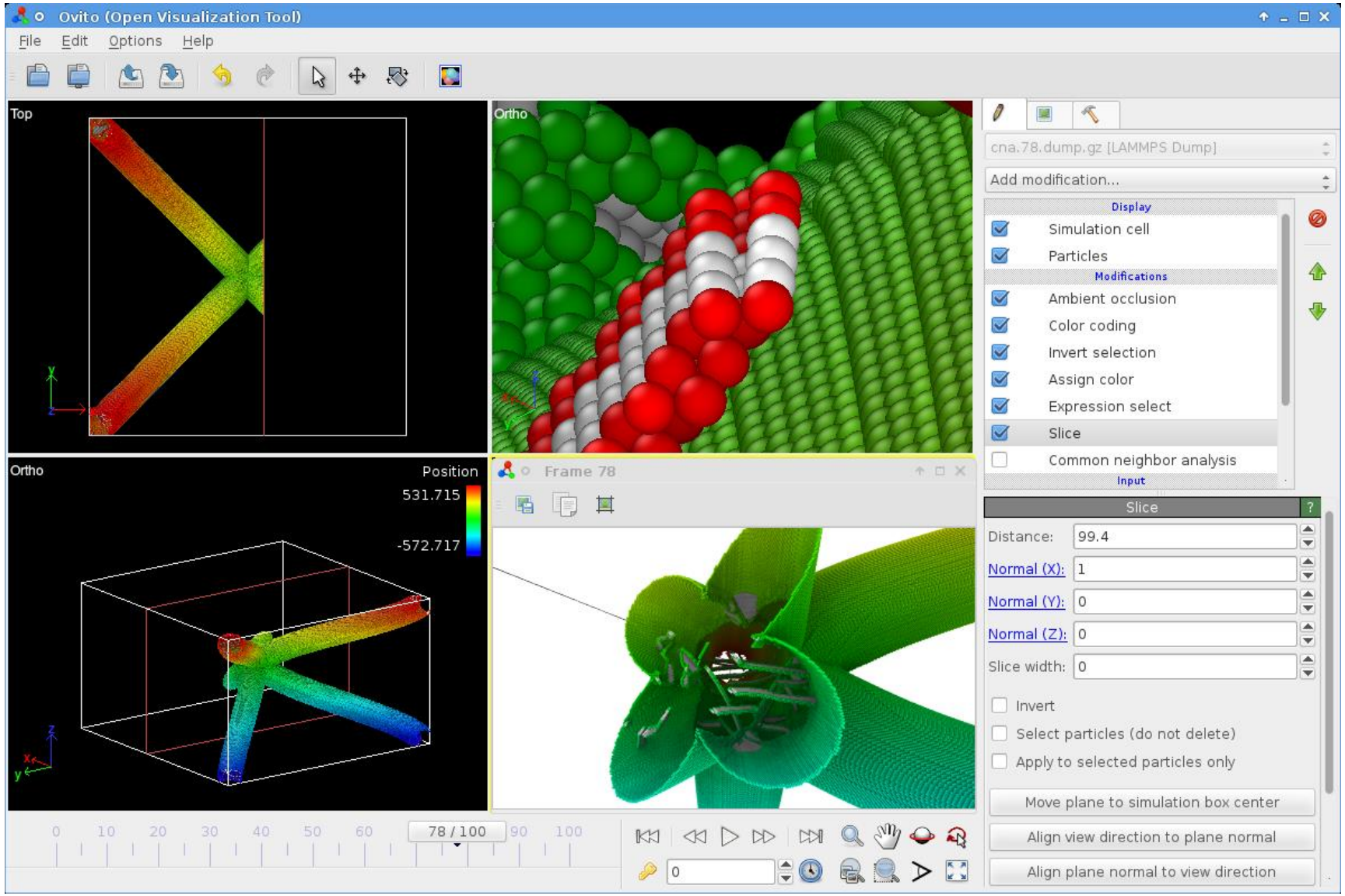
List of typical output files

- **finalstructure.dat-data file**: final structure information
- **md.lammpstrj-trajectory file**: for visualization and analysis.
- **Log.lammps-log file**: store all the actions performed by Lammps.: information about the potential

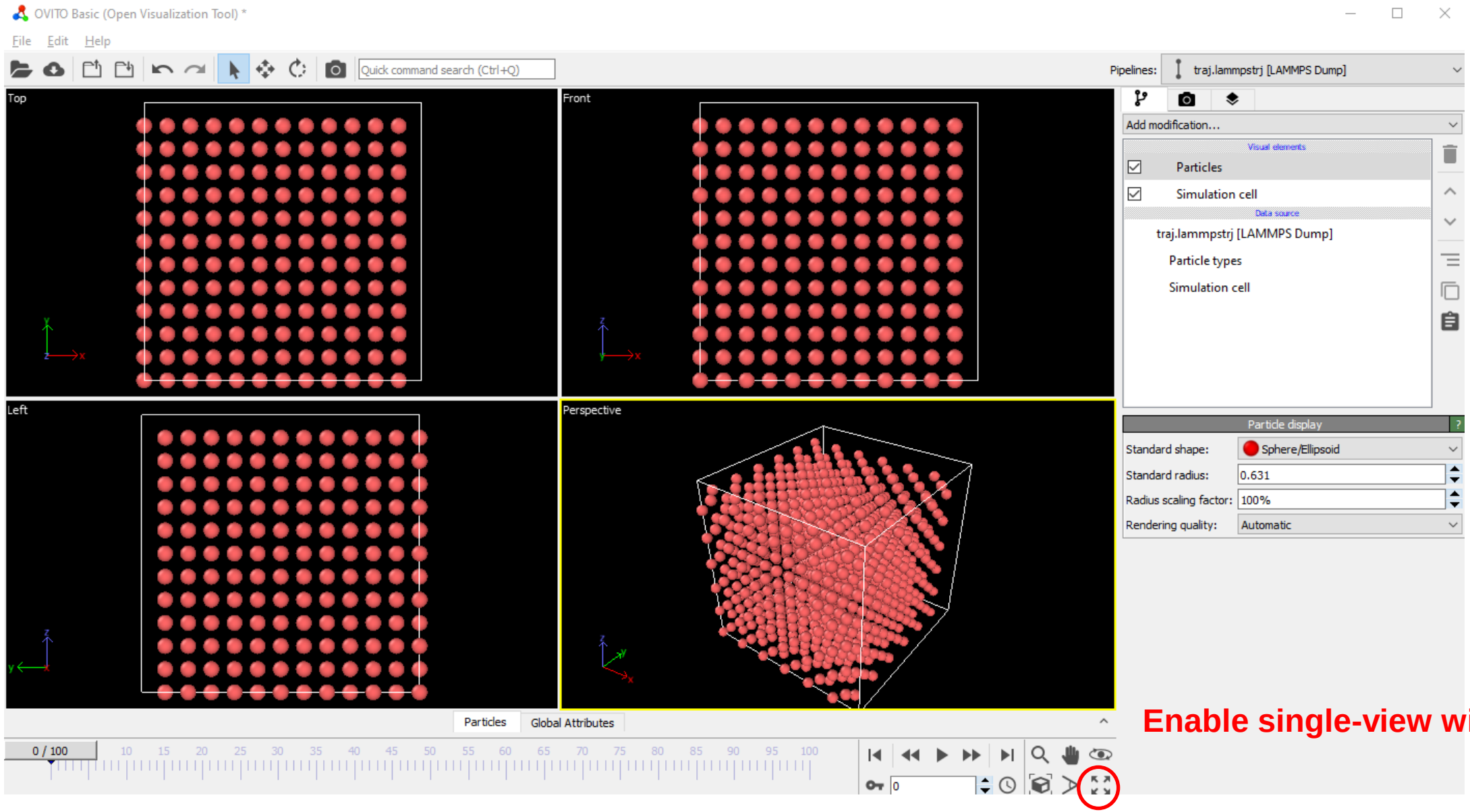
OVITO installation

OVITO
Open Visualization Tool

https://ovito.org

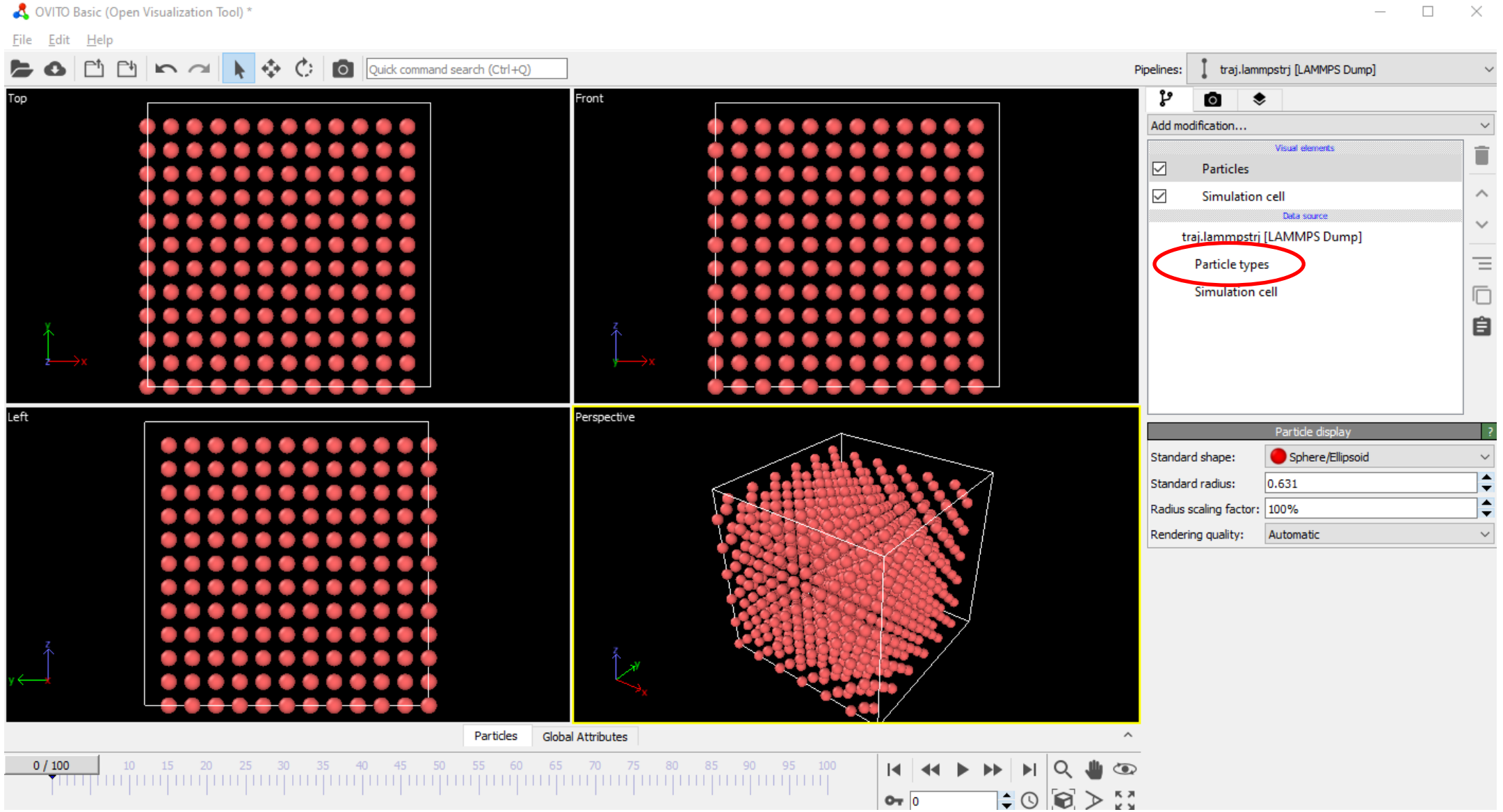


Visualize MD trajectory

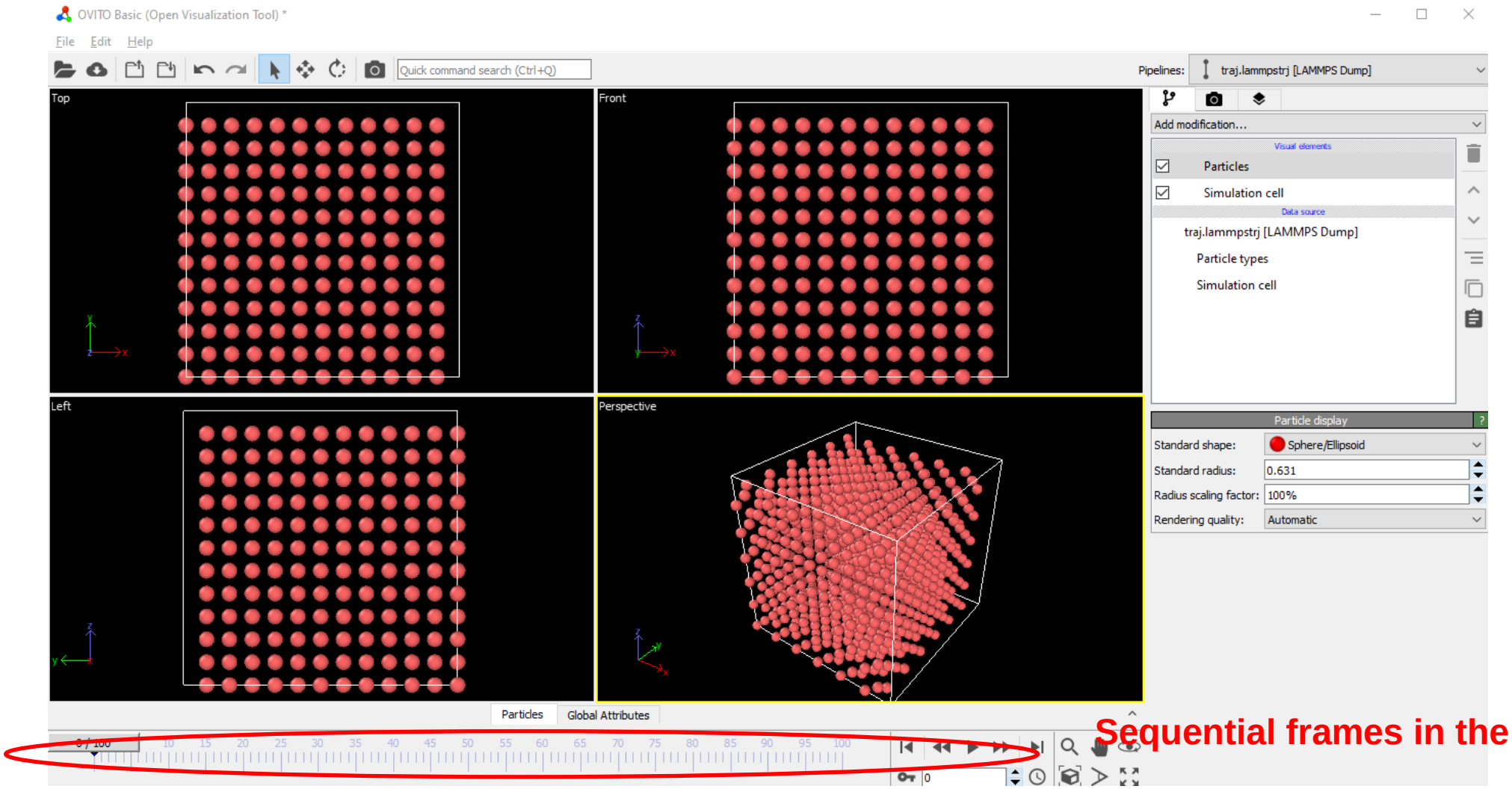


Enable single-view window

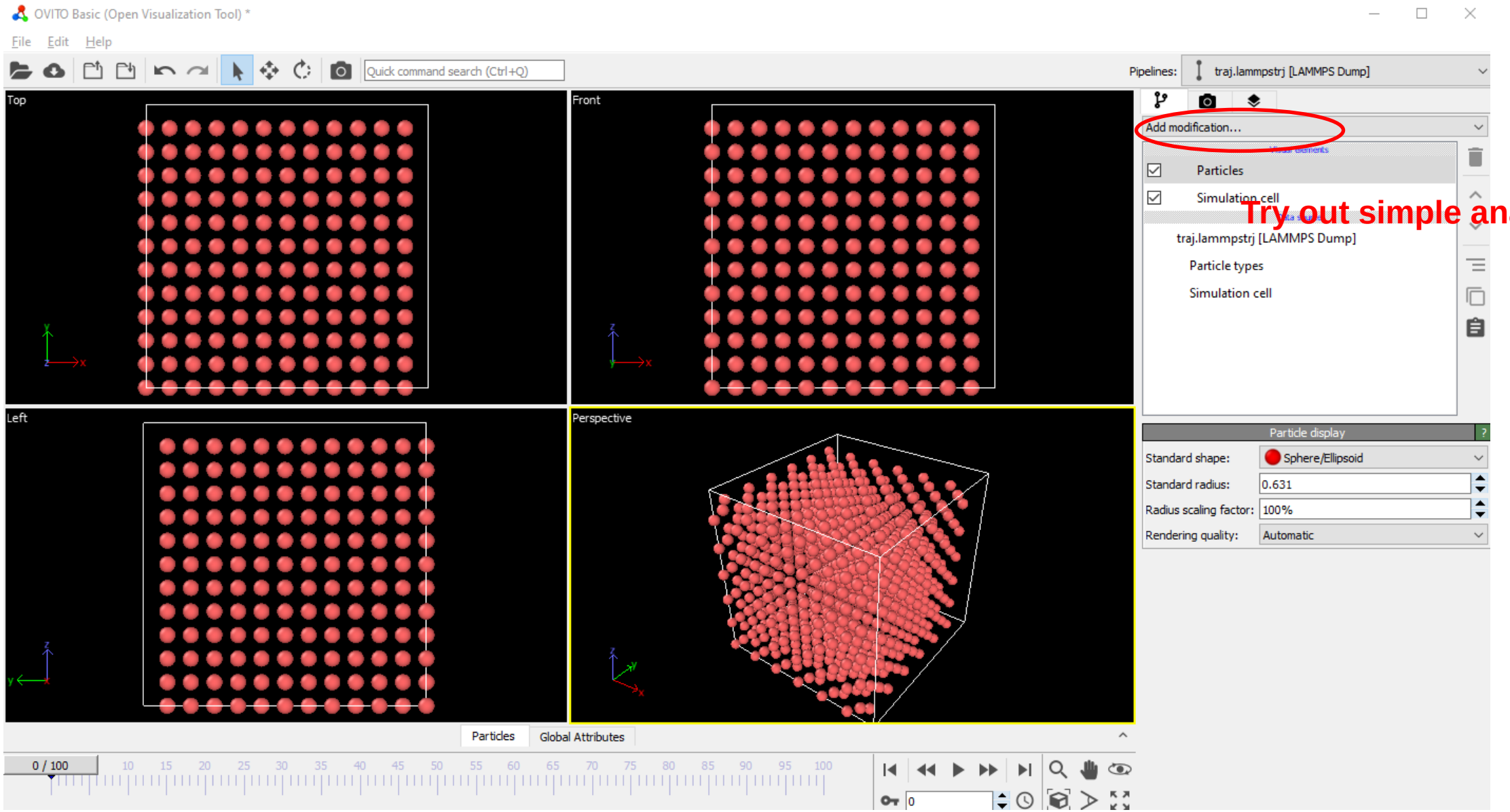
Visualize MD trajectory



Visualize MD trajectory

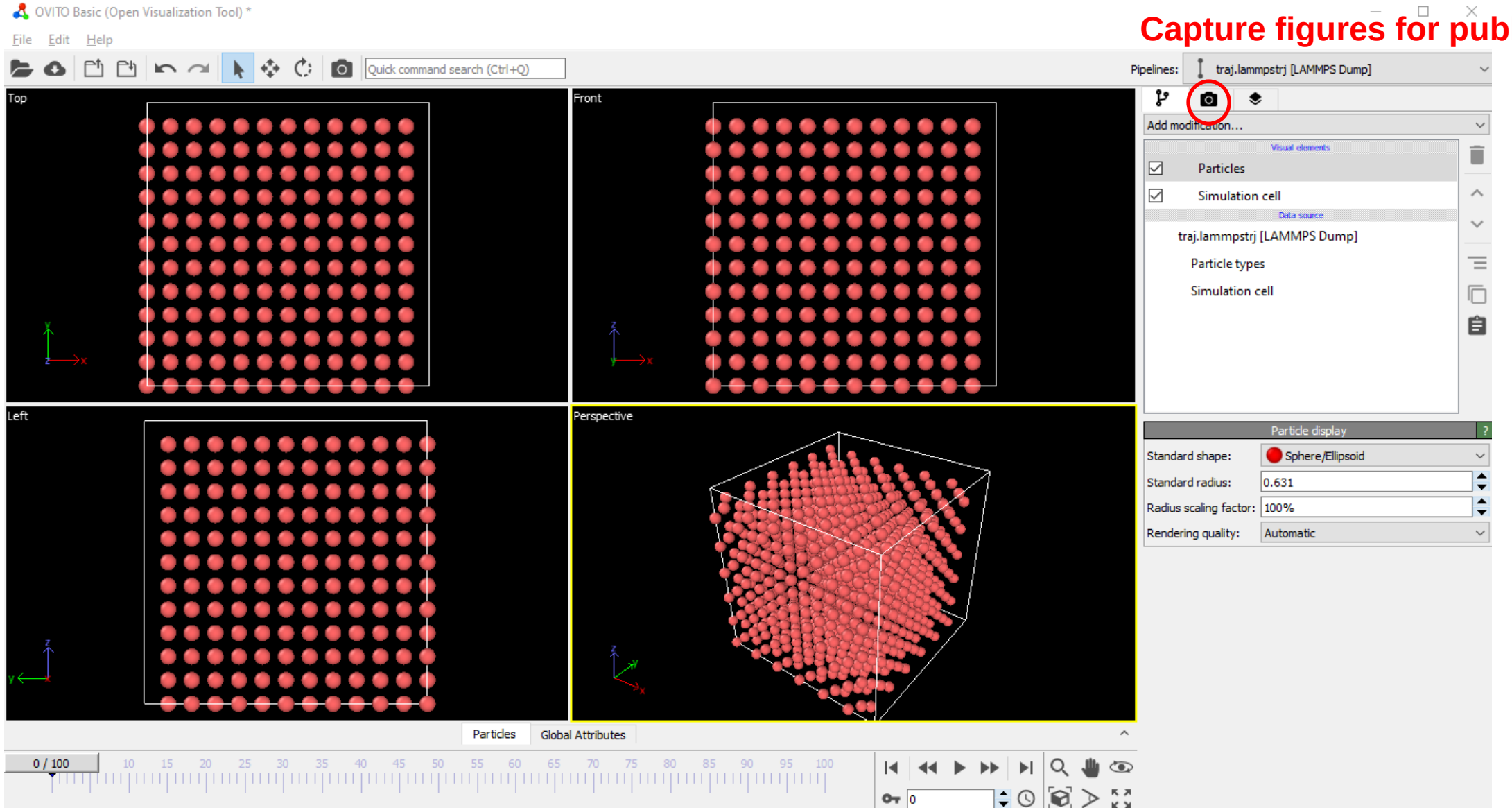


Visualize MD trajectory



Try out simple analysis

Visualize MD trajectory



Capture figures for publications

Lammps installation

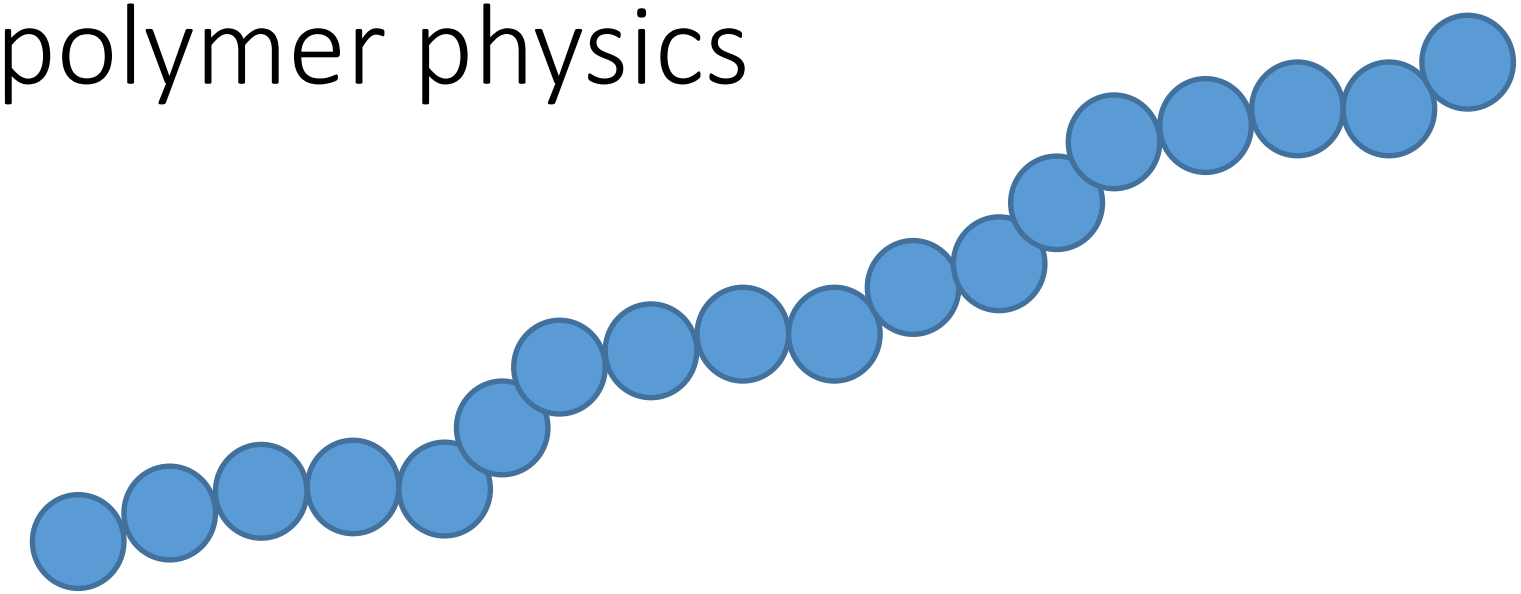
http://lammps.sandia.gov/download.html

How to run it: *lmp_serial –in inputFile*

or *lmp_serial < inputFile*

Example: building a fully flexible polymer chain

Brief polymer physics



The simplest model for polymer chains: fully flexible model

Polymer monomers by beads

Neighbor monomers bonded by free joints (covalent bonds)

Example: building a fully flexible polymer chain

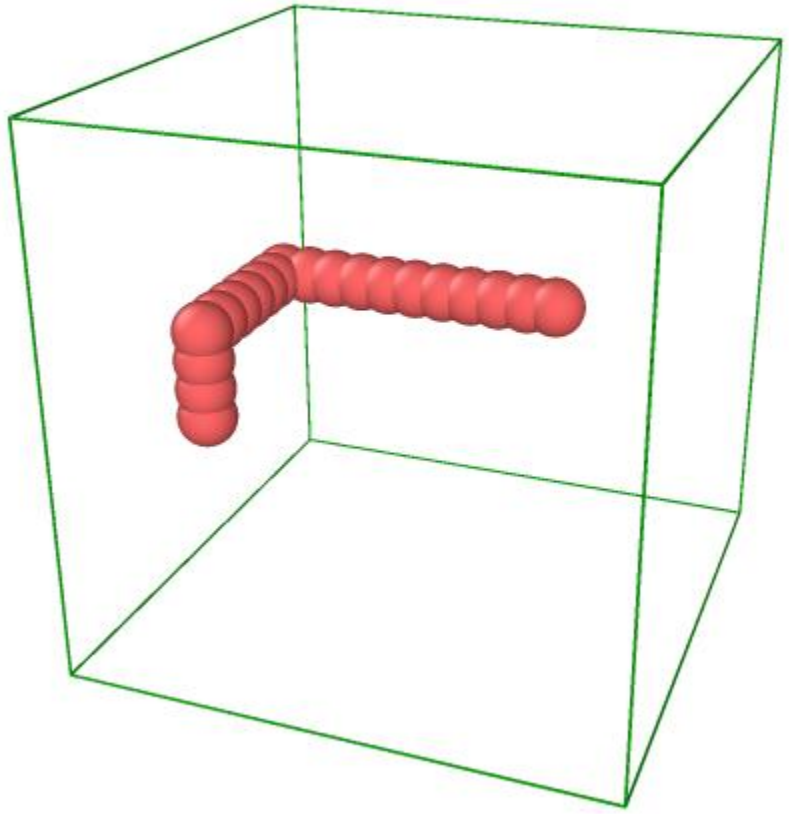
Coarse-graining:

One bead = one monomer

What is needed:

1. 3D coordinates of all beads

2. Neighbor beads pairs

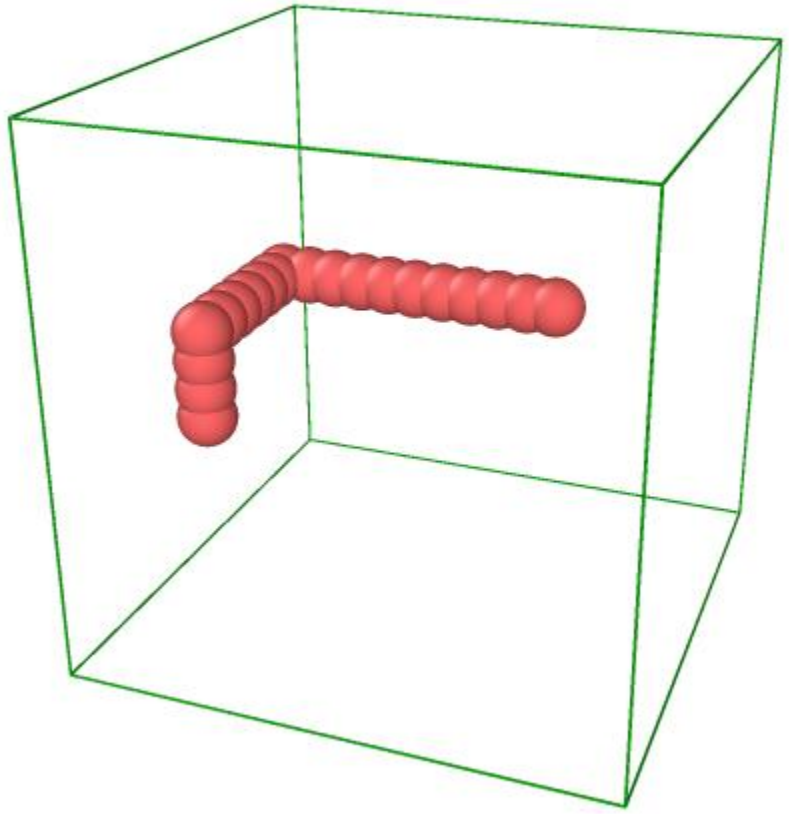


Example: building a fully flexible polymer chain

Two files for Lammps:

data file: structure information

input file: simulation control



Example: building a fully flexible polymer chain

data file:

Number of atoms

number of bonds

Number of atom & bond types

Masses of atoms

Atoms

bonds

Example

```
5 atoms
4 bonds

1 atom types
1 bond types

0 20 xlo xhi
0 20 ylo yhi
0 20 zlo zhi

Masses

1  1.00

Atoms

1   1   1   5   2   10
2   1   1   5   3   10
3   3   1   1   5   4   10
4   4   1   1   5   5   10
5   5   1   1   5   6   10

Bonds

1   1   1   2
....
```

Example: building a fully flexible polymer chain

input file:

Choose units system for the system

Read atomistic structure

Setup non-bonded and boned interactions

Choose time step and generate initial velocity

Choose simulation ensemble

Set simulation length and run

```
# Polymer Chain input file

units           lj
atom_style      bond

read_data       datafile

pair_style      lj/cut   1.122461
pair_modify     shift   yes
pair_coeff      1  1  1.0   1.0   1.122461

bond_style      fene
bond_coeff      * 30.0 1.5 1.0 1.0

dump            1 all atom 100 traj.lammpstrj

timestep        0.01
velocity        all  create  1.0  12345

fix             1 all nvt temp 100 100 100
run             5000
```

Example: building a fully flexible polymer chain

Print out system information

commands:

thermo_style custom step pe ke press vol etotal ….

http://lammps.sandia.gov/doc/thermo_style.html

thermo 50

http://lammps.sandia.gov/doc/thermo_style.html

Example: building a fully flexible polymer chain

Making bigger systems

replicate command in Lammps: *replicate 2 2 2*