



LAMMPS Workshop

Physics Department
Sharif University of Technology
December 2009



LAMMPS input script

- LAMMPS read one line at a time and each command takes effect when it is read.
 - -----
 - timestep 0.5
 - run 100
 - run 100
 - -----
 - run 100
 - timestep 0.5
 - run 100



Rules

"&": The next line is concatenated to the previous line

"#" : comment.

"\$": indicate variables that are replaced with a text string

The line is broken into "words" separated by whitespace (tabs, spaces).

The first word is the command name:

- timestep 0.5
- run 100



Input script structure (In file)

- A LAMMPS input script typically has 4 parts:
 - Initialization
 - Atom definition
 - Settings
 - Run a simulation
- The last 2 parts can be repeated as many times as desired. I.e. run a simulation, change some settings, run some more, etc.



(I) Initialization

- Set parameters that need to be defined before atoms are created or read-in from a file.
- The relevant commands are:
 - units
 - dimension
 - newton
 - processors
 - boundary
 - atom_style
 - atom_modify.



(I) Initialization

- If force-field parameters appear in the files that will be read, these commands tell LAMMPS what kinds of force fields are being used:
 - pair_style
 - bond_style
 - angle_style
 - dihedral_style
 - improper_style



(2) Atom definition

- 3 ways to define atoms in LAMMPS:
 - Read them in from a data or restart files
 - create atoms on a lattice (with no molecular topology)
- Related commands:
 - read_data and read_restart
 - lattice, region, create_box, create_atoms



(3) Settings

- Force field coefficients:
 - pair_coeff
 - bond_coeff
 - angle_coeff
 - dihedral_coeff
 - improper_coeff
 - kspace_style
 - dielectric
 - special_bonds
-



(3) Settings

- Various simulation parameters:
 - Neighbor
 - neigh_modify
 - group
 - timestep
 - reset_timestep
 - run_style
 - min_style
 - min_modify.
-

(3) Settings

- **fix command**

- Forces and constraining***

- ***addforce***: add a force to each atom
 - ***Aveforce***: add an averaged force to each atom
 - ***Enforce2d***: zero out z-dimension velocity and force
 - ***Efield***: impose electric field on system
 - ***Indent***: impose force due to an indenter
 - ***Lineforce***: constrain atoms to move in a line
 - ***orient/fcc***: add grain boundary migration force
 - ***Planeforce***: constrain atoms to move in a plane
 - ***Poems***: constrain clusters of atoms to move as coupled rigid bodies
 - ***Setforce***: set the force on each atom

(3) Settings

- **fix command**

Forces and constraining

- ***reax/bonds***: write out ReaxFF bond information
recenter
- ***Rigid***: constrain one or more clusters of atoms to move as a rigid body
- ***Spring***: apply harmonic spring force to group of atoms
- ***spring/rg***: spring on radius of gyration of group of atoms
- ***spring/self***: spring from each atom to its origin

(3) Settings

- **fix command**

- Atoms and particles***

- ***Deposit***: add new atoms above a surface
 - ***Drag***: drag atoms towards a defined coordinate
 - ***Evaporate***: remove atoms from simulation periodically
 - ***Freeze***: freeze atoms in a granular simulation
 - ***Gravity***: add gravity to atoms in a granular simulation
 - ***Pour***: pour new atoms into a granular simulation domain
 - ***Tmd***: guide a group of atoms to a new configuration

- Box***

- ***box/relax***: relax box size during energy minimization
 - ***Deform***: change the simulation box size/shape

(3) Settings

- **fix command**

- Computations and out puts***

- **Com**: compute a center-of-mass
 - **Gyrat**: compute radius of gyration
 - **Rdf**: compute radial distribution functions
 - **Ms**: compute mean-squared displacement (i.e. diffusion coefficient)
 - **heat - add/subtract momentum**: conserving heat
 - **avthermal/conductivity**: Muller-Plathe kinetic energy exchange for thermal conductivity calculation
 - **e/atom**: compute per-atom time-averaged quantities
 - **ave/spatial**: output per-atom quantities by layer
 - **ave/time**: output time-averaged compute quantities
 - **Momentum**: zero the linear and/or angular momentum of a group of atoms
 - **Print**: print text and variables during a simulation
 - **coord/original**: store original coords of each atom
 - **dt/reset**: reset the timestep based on velocity, forces

(3) Settings

- **fix command**

Ensembles:

- ***Nph***: constant NPH time integration via Nose/Hoover
- ***Npt***: constant NPT time integration via Nose/Hoover
- ***npt/asphere***: NPT for spherical particles
- ***npt/sphere***: NPT for spherical particles
- ***Nve***: constant NVE time integration
- ***nve/asphere***: NVT for aspherical particles
- ***nve/limit***: NVE with limited step length
- ***nve/noforce***: NVE without forces (v only)
- ***nve/sphere***: NVT for spherical particles
- ***Nvt***: constant NVT time integration via Nose/Hoover
- ***nvt/asphere***: NVT for aspherical particles
- ***nvt/sllod***: NVT for NEMD with SLLOD equations
- ***nvt/sphere***: NVT for spherical particles

(3) Settings

- **fix command**

Pressure, Temperature, and Viscosity :

- ***press/berendsen***: pressure control by Berendsen barostat
- ***temp/berendsen***: temperature control by Berendsen thermostat
- ***temp/rescale***: temperature control by velocity rescaling
- ***Ttm***: two-temperature model for electronic/atomic coupling
- ***Langevin***: Langevin temperature control
- ***Viscosity***: Muller-Plathe momentum exchange for viscosity calculation
- ***Viscous***: viscous damping for granular simulations

(3) Settings

- **fix command**

Bonds and angles:

- ***bond/break***: break bonds on the fly
- ***bond/create***: create bonds on the fly
- ***bond/swap***: Monte Carlo bond swapping
- ***Shake***: SHAKE constraints on bonds and/or angles

Walls:

- ***wall/gran***: frictional wall(s) for granular simulations
- ***wall/lj126***: Lennard-Jones 12-6 wall
- ***wall/lj93***: Lennard-Jones 9-3 wall
- ***wall/reflect***: reflecting wall(s)
- ***Wiggle***: oscillate walls and frozen atoms



(3) Settings

- Various computations can be specified for execution during a simulation:
 - Compute
 - compute_modify
 - variable
- Output options
 - Thermo
 - dump
 - restart



(4) Run a simulation

- ***run*** : A molecular dynamics simulation
 - ***minimize*** : Energy minimization
-



LAMMPS vs. GROMACS

- GROMACS:
 - is easier to use and specified for biological molecules.
 - More water explicit models
 - More post processing utilities.
 - LAMMPS only support SHAKE constraint
- LAMPPS do more simulation tasks:
 - Implicit solvent
 - More systems: granular materials, coarse-grained mesoscale models, ellipsoidal particles



References

- <http://lammps.sandia.gov/doc/Manual.html>
-

- 
- Thank you for your attention.
-