Duke Center for Quantitative Living Systems

April 2025

Molecular Dynamics Simulations Demystified:
(Part II) Practical Aspects and Software

Instructor: **Jerelle Joseph**
email: jerellejoseph@princeton.edu

Contents

- Energy conservation
- Timesteps
  - Choosing correct timestep
  - Large timesteps
- Initializing
  - Positions
  - Velocities
- Introduction to Thermostats and Barostats
  - Measuring temperature
  - Measuring pressure
  - Adjusting Temp.: Thermostats
  - Adjusting pres.: Barostats
- MD Software

Energy Conservation & Timestep considerations

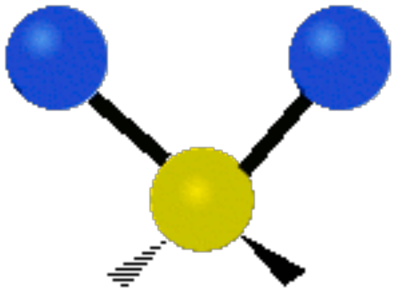
- In MD we are numerically integrating Newton's Equations of motion
- One practical consideration is how much to advance the positions and velocity by
- What timestep to use?

$$\mathbf{v}_i(t+\boxed{\delta t}) = \mathbf{v}_i(t) + \frac{\delta t}{2m_i}\left[\mathbf{f}_i(t) + \mathbf{f}_i(t+\delta t)\right] \qquad \text{Velocity Verlet algorithm}$$

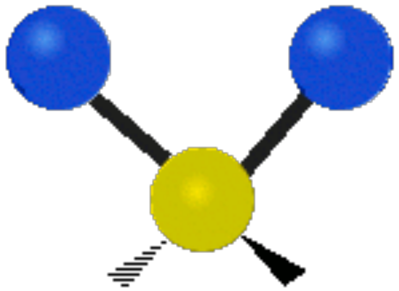
- In general, Newton's Equations of motion should obey conservation of energy
- Therefore, the timestep, should be chosen to numerically conserve this condition.
- "Numerically conserve" → fluctuations in total energy should be $< 1\%$ over the course of an NVE simulation
- Additionally, the timestep should be small enough to capture fluctuations in forces.

Choosing the correct timestep: Water example

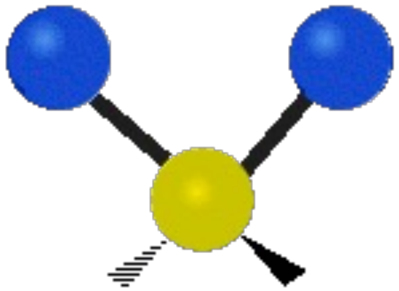
Note: Intramolecular motions are generally faster than intermolecular ones



Symmetric stretch:
$v_1 \sim 3700\ \text{cm}^{-1}$ ← fastest freq



Asymmetric stretch:
$v_2 \sim 3600\ \text{cm}^{-1}$



Bend: $v_3 \sim 1600\ \text{cm}^{-1}$

Rule-of-thumb: for numerical stability/accuracy, common integrators should employ timestep that is at least an order of magnitude smaller than the fastest frequency of motion needed to describe the system

We can work out the time related to the fastest frequency motion in our system.

$$\tau = \frac{1}{vc} \leftarrow \text{time}$$

$$\tau_1 = \frac{1}{3700\ \text{cm}^{-1} \times 3 \times 10^{10}\text{cms}^{-1}}$$

$$\tau_1 \sim 9 \times 10^{-15}\ \text{s}$$

$$\sim 9\text{fs}$$

$\Rightarrow$ to effectively capture v_1, we need $\delta t \leq 0.9\text{fs}$

Typical motions	timestep
bond vibrations	0.5 – 1.0fs
bending modes	2.0fs
translations	5 – 10fs

Choosing the correct timestep

Why don't we use the smallest δt possible?
total time to simulate $= n_{\text{step}} * \delta t$

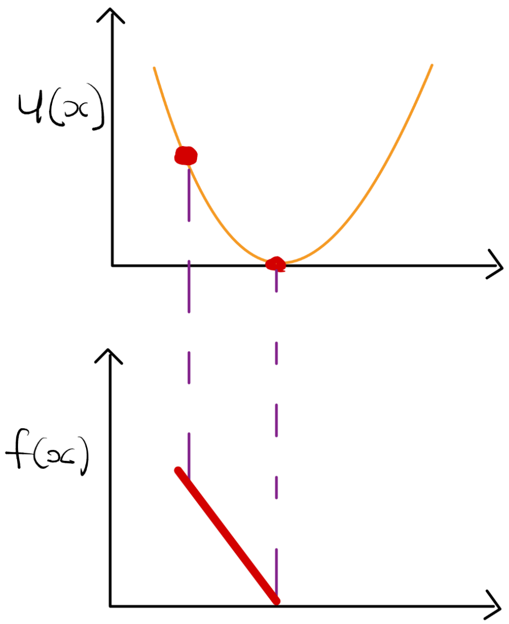
- If t is the target amount of time we need to simulate to capture the relevant phenomenon we are probing,
- as δt decreases, n_{step} needs to increase to achieve the same t
- This means that with smaller δt, our actual (wall clock time) of our simulation is longer
- In general, δt should be small enough to capture the fluctuations in forces
- SHAKE, RATTLE → these algorithms can be used to constrain certain motions and allow for larger δt

The Catastrophe of a large timestep

- Choose too large a timestep results in catastrophic explosions!
- Explosions usually occur due to lack of energy conservation
- Instead we have accumulation of energy due to an overestimation of particle displacements. **Let's see why**
- When integrating equations of motion, for each timestep δt, we are assuming that the force is constant over δt

$$\mathbf{r}(t + \delta t) = \mathbf{r}(t) + \mathbf{v}(t)\delta t + \frac{\mathbf{f}(t)}{2m}\delta t^2$$

Position Update Step: The position is updated using the current force, and this assumes that the force remains approximately constant throughout the timestep δt.



The Catastrophe of a large timestep

- But if δt is too large, then the force is no longer constant. Hence, we will be using the "wrong" force in our algorithm for large δt
- Therefore, we will not "sense" the curvature in the potential energy
- Consequently, we overproject particle displacements, and energies and forces get progressively worse.

Initialization of positions

- **Positions**: can obtain these from an experimental structure (e.g. for a protein), placing molecules on a grid with random rotations (e.g., for water molecules)

- In general, because our simulation algorithm relies on forces, we need to be careful to avoid particle overlaps that can lead to "explosions"

- However even with "good" a starting structure, there may still be large forces (e.g., due to force field paramters)

- There are several strategies that can be used to "massage" the initial structure and relax the system

  - Use energy minimization algorithms
  - Start with a Monte Carlo simulation
  - Impose artificial restraints/limits on initial displacements
  - Start with smaller timesteps

Initialization of velocities: Maxwell–Boltzmann Distribution

- **Velocities**: we often choose the initial velocities to obtain a given initial temperature

- The initial velocities of particles are usually drawn from a **Maxwell–Boltzmann distribution**, which describes the distribution of velocities for particles in an ideal gas at thermal equilibrium

$$P\left(v_{i,x}\right) = \sqrt{\frac{m_i}{2\pi k_B T}} \exp\left(-\frac{m_i v_{i,x}^2}{2k_B T}\right)$$

- The distribution depends on the temperature T and the mass m of the particles, and it ensures that the system's kinetic energy corresponds to the desired temperature

Initialization of velocities: Maxwell–Boltzmann Distribution

- The Maxwell–Boltzmann Distribution has the general form of a Gaussian distribution:

$$P\left(v_{i,x}\right) = \sqrt{\frac{m_i}{2\pi k_B T}} \exp\left(-\frac{m_i v_{i,x}^2}{2 k_B T}\right)$$

$$P\left(v_x\right) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{v_x^2}{2\sigma^2}\right),$$

where σ^2 is the variance, given by: $\sigma_{i,x}^2 = \frac{k_B T}{m_i}$

- The velocity distribution is Gaussian with zero mean $\langle v_{i,x} \rangle = 0$
- In practice, we draw random velocities from a Gaussian distribution (normal) with mean=0 and calculated standard deviation $\sigma_{i,x}$

Simulating under different conditions (ensembles): Thermostats and Barostats:

- Natively, MD should correspond to the NVE ensemble. Thus, additional algorithms are needed to approximate the target ensemble

- **Thermostats:** these are used to control the effective temperature in an MD simulation

- **Barostats:** these are used to control the pressure in a simulation

Simulating under different conditions (ensembles): Thermostats and Barostats:

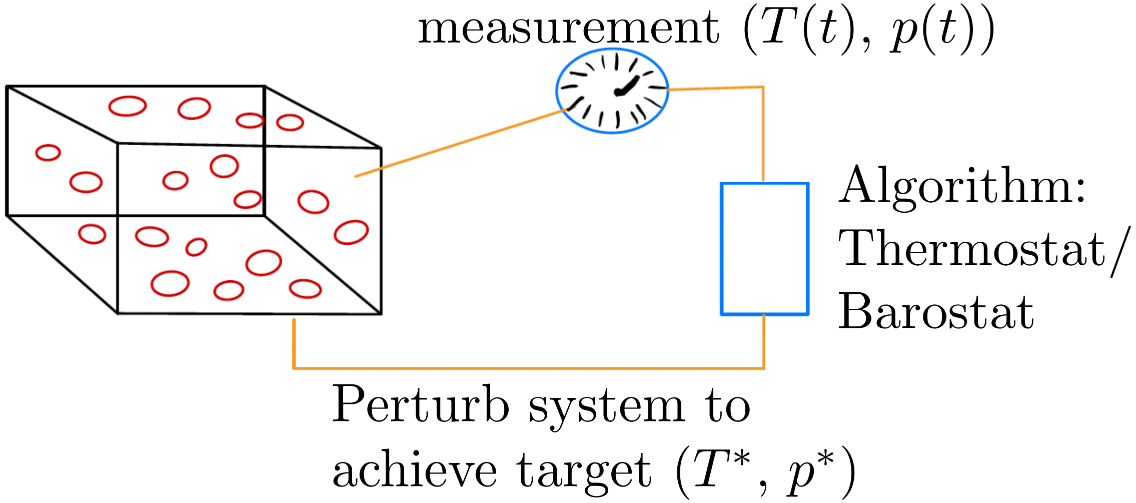
- Thermostats and barostats are like feedback controllers that adjust the system based on observation of variables that relate to the target temperature/pressure (T^*, p^*)

- Based on our measurement of the instantaneous temperature, $T(t)$, or pressure, $p(t)$, these will **perturb** the system so that it goes close to the prescribed target (T^*, p^*)

- T^*, p^* are chosen according to the desired ensemble

- Note T and p will fluctuate during the simulation

- However, the goal is: for $t \to \infty$, $\langle T(t) \rangle \to T^*$


measurement ($T(t)$, $p(t)$)
Algorithm: Thermostat/ Barostat
Perturb system to achieve target (T^*, p^*)

Measuring temperature in an MD simulation

- Temperature, T, is a thermodynamic quantity which we can compute from simulations using ensemble averages

- Here, the kinetic energy is used to estimate the temperature

- According to the equipartition principle, the average kinetic energy per degree of freedom is: $\frac{k_\mathrm{B}T}{2}$

- Hence, for an N-particle system in 3D, the average kinetic energy is given by

$$\langle K \rangle = 3N * \frac{k_\mathrm{B}T}{2}$$

- It follows that the average system temperature is: $\langle T \rangle = \frac{2\langle K \rangle}{k_\mathrm{B}3N}$

- And the instantaneous temperature is: $T = \frac{2k(t)}{k_\mathrm{B}3N}$ where, $k(t) = \frac{1}{2}\sum_i m_i v_i(t) v_i(t)$

- This is the most common way to estimate the temperature (i.e., from the kinetic energy)

Velocity rescaling: Simple/cheap algorithm for adjusting temp.

Velocity Rescaling
Scale velocities at some frequency (so after n timesteps) so that $k(t)$ yields $T(t) \to T^*$

Implementation

(i) $T(t) = \frac{2k(t)}{3Nk_B} = \frac{2}{3Nk_B}\frac{1}{2}\sum_i m_i v_i^2$

(ii) $\lambda = \sqrt{T^*/T(t)}$

(iii) ${v_i}' = \lambda v_i$

Result

$$k'(t) = \frac{1}{2}\sum_i m_i \lambda vi \lambda v_i = \lambda^2 k(t)$$

$$T'(t) = \lambda^2 T(t)$$

$$T'(t) = \frac{T^*}{T(t)} T(t) = T^*$$

Velocity rescaling: Simple/cheap algorithm for adjusting temp.

Velocity Rescaling
Scale velocities at some frequency (so after n timesteps) so that $k(t)$ yields $T(t) \rightarrow T^*$

Note, this is not technically correct from a thermodynamics perspective!

Why?
In the limit of rescaling velocities every step, results in a isokinetic ensemble and not the canonical (NVT) ensemble.
Isokinetic → Kinetic energy is the same value at event time step.

$$\rightarrow \sigma_K^2 = 0$$

Velocity rescaling:
Simple/cheap algorithm for adjusting temp.

Velocity Rescaling
Scale velocities at some frequency (so after n timesteps) so that $k(t)$ yields $T(t) \rightarrow T^*$

Isokinetic → Kinetic energy is the same value at event time step.

$$\rightarrow \sigma_K^2 = 0$$

BUT, at a given temperature, statistical mechanics predicts:

$$\begin{aligned}\sigma_k^2 &= \left\langle k^2 \right\rangle - \left\langle k \right\rangle^2 \\ &= \frac{3N\left(k_B T\right)^2}{2} \neq 0\end{aligned}$$

i.e., the canonical ensemble does in fact exhibit Kinetic energy fluctuations!

Thermostats: Adjusting temperature

- To implement a canonical constant-temperature simulation, we need to weakly couple the system to a fictitious heat bath that imposes the desired temperature

- Noting that T will fluctuate during the simulation. Hence, the goal is $\langle T(t) \rangle \rightarrow T^*$ for $t \rightarrow \infty$

- There is no uniquely "correct" way of doing this. Possible approaches can be broadly classified as stochastic (e.g., Andersen, Langevin) or deterministic (e.g., Nosé-Hoover thermostat).

Desirable features of thermostats:

- Generate the correct microstate distribution – i.e. preserve thermodynamics; meaning it does not mess up the canonical distribution of microstates, if we are in the NVT ensemble

- If possible, minimally disrupts the realistic dynamics. Important especially if we care about properties that depend on the momenta/velocities of our system (transport properties)

Andersen Thermostat: Stochastic algorithm

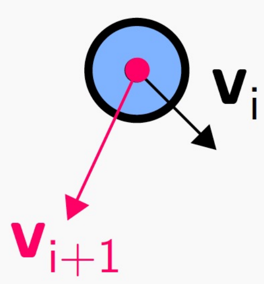
- Parameter, ν, defines the frequency with which the system collides with the heat bath.

- Probability of collision over the small interval of time δt

$$p = \nu\delta t$$

- ν is typically very small; in the region of 0.001 or 0.01 inverse units of δt

```
For each step:
        do MD stuff (i.e., update positions, velocities, forces)

        for each particle:
             get r = u (0,1) # r is a random number
             if r < p:
                sample new velocity for particle from Maxwell-Boltzmann Distribution
```



- In practical terms, the new velocity of particle i is drawn from a Gaussian distribution with mean $= 0$ and $\sigma^2 = \frac{k_B T}{m_i}$

Andersen Thermostat: Stochastic algorithm

- Parameter, ν, defines the frequency with which the system collides with the heat bath.

- Probability of collision over the small interval of time δt

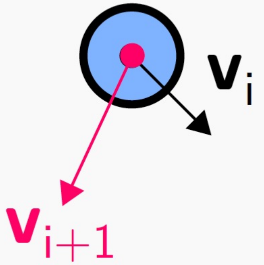
$$p = \nu \delta t$$

- ν is typically very small; in the region of 0.001 or 0.01 inverse units of δt

- The frequency ν determines the coupling strength between the bath and the particle system.

- This algorithm is guaranteed to generate a canonical distribution, but the reassignment of velocities may strongly affect the dynamics (notably the velocity auto-correlation function).

- **Verdict:** The Andersen thermostat is useful for sampling conformational space, but not so much for the computation of time-dependent properties.



Nosé–Hoover Thermostat: Deterministic algorithm

- One of the "better" thermostats
- **Deterministic** approach: additional terms are included in the equations of motion, which correspond to the heat bath
- i.e., there are additional variables that need to be integrated, which correspond to a fictitious bath that regulates the temperature
- **Verdict:** The Nosé–Hoover thermostat only mildly affects the dynamics of particles, so dynamic properties are relatively safe to compute with this thermostat

Thermostats: Which one to use?

- For **relaxing initial structures**:
  - use a cheap/simple thermostat
  - velocity rescaling, Berendsen thermostat
- For **thermodynamic properties** (i.e., to reproduce the canonical ensemble):
  - avoid velocity rescaling/Berendsen
  - use Andersen, Langevin thermostat
- For **dynamical properties**:
  - Safest approach is to first thermostat the system to the desired temperature and then simulate in NVE ensemble (i.e., no thermostat)
  - Next best approach is to use Nosé–Hoover or some algorithm that minimally perturbs the dynamics

Measuring pressure in an MD simulation

As with temperature, pressure is a thermodynamic quantity and so we can also compute it using an ensemble average.

The average pressure is given by:

$$\langle K \rangle = 3N * \frac{k_{\mathrm{B}}T}{2}$$

$$\langle p \rangle = \frac{1}{3V} \left\langle 2K + \sum_i \boldsymbol{f}_i \cdot \boldsymbol{r}_i \right\rangle$$

where V is the volume. The first term relates to the ideal gas and the second term is the virial term (arises due to interparticle forces).

And at any given point in time, we can estimate the instantaneous pressure as:

$$p = \frac{1}{3V(t)} \left[2k(t) + \sum_i \boldsymbol{f}_i(t) \boldsymbol{r}_i(t) \right]$$

Where $k(t) = \frac{1}{2} \sum_i m_i v_i(t) v_i(t)$

Common Barostats

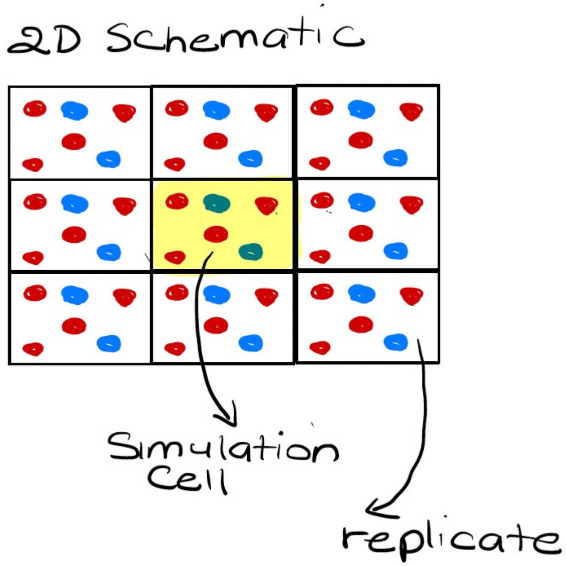
- Many of the approaches developed for adjusting temperature have also been reformulated to control the pressure
- Here, the primary goal is the adjust the pressure

Approach (* = best methods to use)	**Description**
Volume Rescaling	Scale volume based on estimated pressure $p(t)$ vs. target pressure
Berendsen Barostat	Weak coupling to pressure bath that leads to rescaling (does not generate true NPT ensemble)
Nosé-Hoover Barostat*	Employ additional EOM variables to tune the pressure (works isotopically; assumes uniform pressure in all directions)
Parrinello-Rahman Barostat**	Similar to Nosé-Hoover but applied independently to different unit cell vectors (anisotropy allowed)
MTTK*	Corrects Parrinello-Rahman for small systems
Monte Carlo*	Use MC to adjust the volume with correct probability distribution

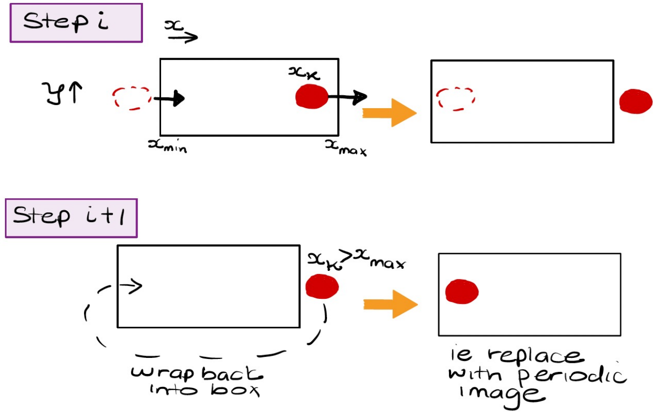
Extra algorithms/schemes that are important (not MD specific)

- **Periodic boundary conditions**: tile the space with exact replicas of main simulation cell to avoid surface/edge effects and better mimic bulk behavior
  - **Wrapping particles**: If particles leave the simulation cell at one end it reenters at the other end (think: of a 2D video game)
  - **Minimum image convention**: If simulation cell length is L particles further than $L/2$ away cannot interact with each other
- **Cutoff distances**: Used to truncate pairwise potentials; dictating up to what range of distances particles can interact with each other

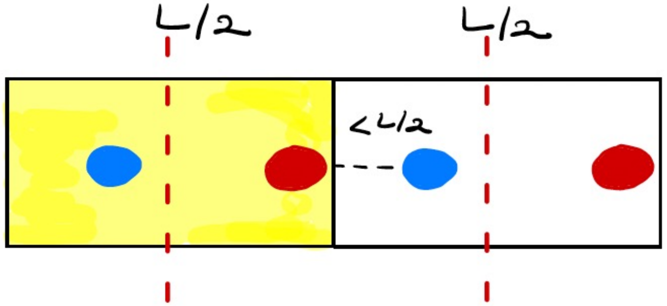
PBC



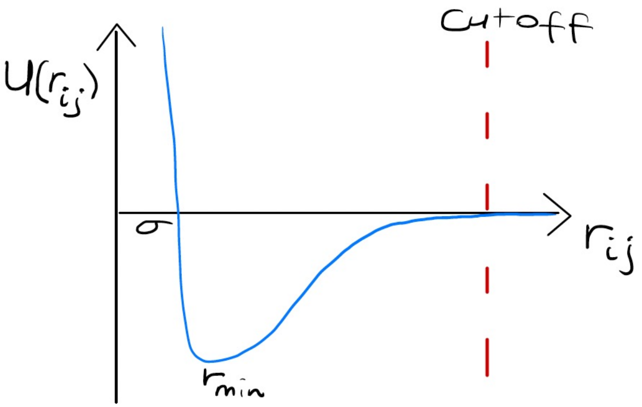
Wrapping



Minimum image convention



Cutoff distances



MD Simulation Software

Software	Language	Beginner Friendly	Best For	Online Support
LAMMPS	C++	Moderate	Large-scale atomic/molecular systems, materials science	Excellent (active community, extensive documentation)
GROMACS	C, C++	Moderate	Biomolecules (proteins, lipids, nucleic acids), high performance	Strong (active user base, tutorials)
OpenMM	Python, C++	Yes (Python API)	Biophysics, GPU-accelerated simulations	Good (active development, user forums)
HOOMD-blue	Python, C++	Moderate	Coarse-grained, soft matter, GPU-focused	Strong (active community, documentation)
NAMD	C++	Moderate	Biomolecules, scalable parallel simulations	Excellent (tutorials, forums)
AMBER	C, Fortran	Less friendly (complex setup)	Biomolecular simulations, force field development	Moderate (some community support)
CHARMM	Fortran, C	Less friendly	Biomolecular simulations, force fields	Limited (mainly institutional support)

AMBER is partially open source and CHARMM is not. All others are open source

Tips for getting started:

- Read through the documention
- Find a good beginner tutotial
- Run a simulation to reproduce existing results

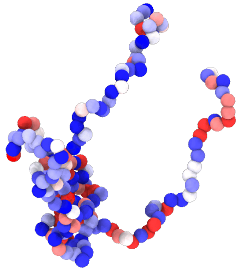
Complementary MD Software

- **Setting up the initial structures/system**
  - Packmol*
  - PyMol*
  - CHARMM*
  - Moltemplate (for LAMMPS users)
  - Avogadro (good for small molecules)
  - UCSF Chimera/ChimeraX (alternative to PyMol)
  - GROMACS Tools* (e.g., `pdb2gmx`, `editconf`, `genbox`)
- **Analysis of MD trajectories**
  - MDAnalysis*
  - MDTraj
  - CPPTRAJ (commonly used with AMBER, but supports other formats)
  - PLUMED (for enhanced sampling/free energy calculations, also has analysis features)
- **Visualization and Analysis**
  - Ovito*
  - VMD*
  - ChimeraX

* = recommendations

Example: Computing the radius of gyration of a disordered protein in LAMMPS

1. Configuration/structure file (number of particles, some force field info, cartesian coordinates of particles, molecule connectivity)



2. LAMMPS Parameter file (force field info*, conditions, instructions for writing output)

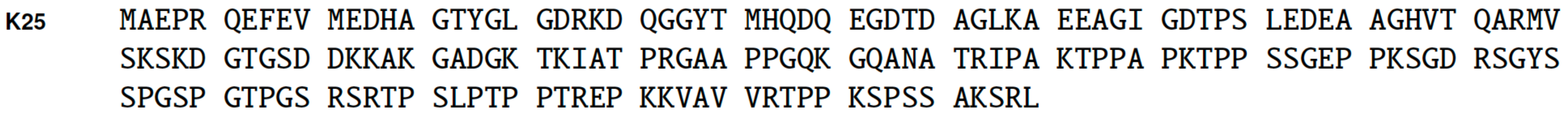
3. Run simulation and compute protein *R*g

** Could also be a separate file that is called within the LAMMPS parameter file*

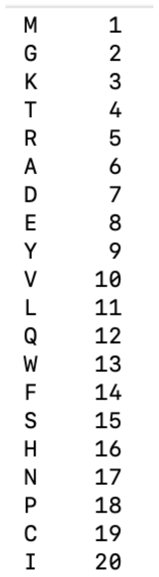
*R*g of K25 protein: Protein sequence

```
K25     MAEPR QEFEV MEDHA GTYGL GDRKD QGGYT MHQDQ EGDTD AGLKA EEAGI GDTPS LEDEA AGHVT QARMV
        SKSKD GTGSD DKKAK GADGK TKIAT PRGAA PPGQK GQANA TRIPA KTPPA PKTPP SSGEP PKSGD RSGYS
        SPGSP GTPGS RSRTP SLPTP PTREP KKVAV VRTPP KSPSS AKSRL
```

Particle types in Mpipi force field



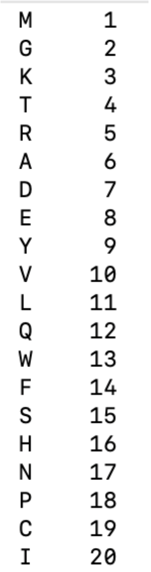
```
K25     MAEPR QEFEV MEDHA GTYGL GDRKD QGGYT MHQDQ EGDTD AGLKA EEAGI GDTPS LEDEA AGHVT QARMV
        SKSKD GTGSD DKKAK GADGK TKIAT PRGAA PPGQK GQANA TRIPA KTPPA PKTPP SSGEP PKSGD RSGYS
        SPGSP GTPGS RSRTP SLPTP PTREP KKVAV VRTPP KSPSS AKSRL
```



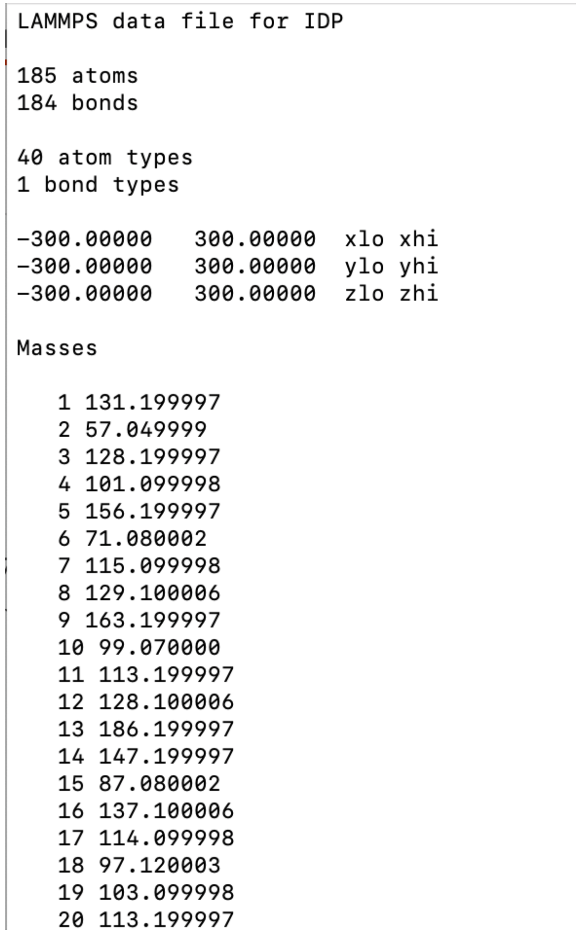
M	1
G	2
K	3
T	4
R	5
A	6
D	7
E	8
Y	9
V	10
L	11
Q	12
W	13
F	14
S	15
H	16
N	17
P	18
C	19
I	20

Configuration file

K25 MAEPR QEFEV MEDHA GTYGL GDRKD QGGYT MHQDQ EGDTD AGLKA EEAGI GDTPS LEDEA AGHVT QARMV
SKSKD GTGSD DKKAK GADGK TKIAT PRGAA PPGQK GQANA TRIPA KTPPA PKTPP SSGEP PKSGD RSGYS
SPGSP GTPGS RSRTP SLPTP PTREP KKVAV VRTPP KSPSS AKSRL



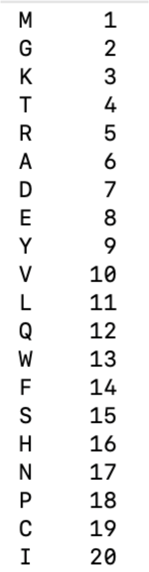
M	1
G	2
K	3
T	4
R	5
A	6
D	7
E	8
Y	9
V	10
L	11
Q	12
W	13
F	14
S	15
H	16
N	17
P	18
C	19
I	20



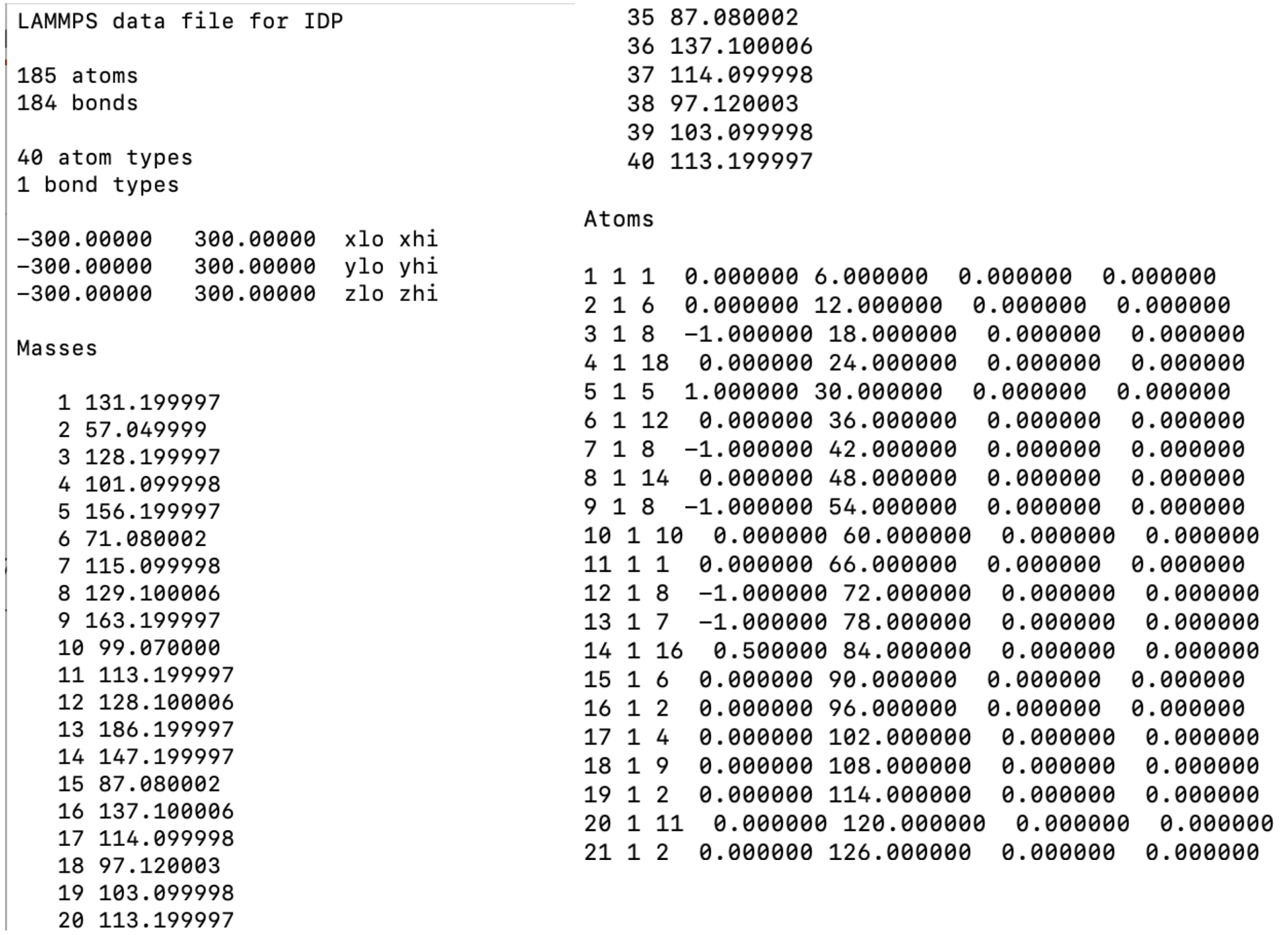
```
LAMMPS data file for IDP

185 atoms
184 bonds

40 atom types
1 bond types

-300.00000   300.00000  xlo xhi
-300.00000   300.00000  ylo yhi
-300.00000   300.00000  zlo zhi

Masses

   1 131.199997
   2 57.049999
   3 128.199997
   4 101.099998
   5 156.199997
   6 71.080002
   7 115.099998
   8 129.100006
   9 163.199997
   10 99.070000
   11 113.199997
   12 128.100006
   13 186.199997
   14 147.199997
   15 87.080002
   16 137.100006
   17 114.099998
   18 97.120003
   19 103.099998
   20 113.199997
```

Configuration file: Initial particle positions (“atoms”)

K25 MAEPR QEFEV MEDHA GTYGL GDRKD QGGYT MHQDQ EGDTD AGLKA EEAGI GDTPS LEDEA AGHVT QARMV
SKSKD GTGSD DKKAK GADGK TKIAT PRGAA PPGQK GQANA TRIPA KTPPA PKTPP SSGEP PKSGD RSGYS
SPGSP GTPGS RSRTP SLPTP PTREP KKVAV VRTPP KSPSS AKSRL



M	1
G	2
K	3
T	4
R	5
A	6
D	7
E	8
Y	9
V	10
L	11
Q	12
W	13
F	14
S	15
H	16
N	17
P	18
C	19
I	20

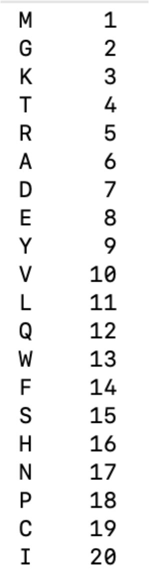


```
LAMMPS data file for IDP

185 atoms
184 bonds

40 atom types
1 bond types

-300.00000   300.00000  xlo xhi
-300.00000   300.00000  ylo yhi
-300.00000   300.00000  zlo zhi

Masses

   1 131.199997
   2 57.049999
   3 128.199997
   4 101.099998
   5 156.199997
   6 71.080002
   7 115.099998
   8 129.100006
   9 163.199997
   10 99.070000
   11 113.199997
   12 128.100006
   13 186.199997
   14 147.199997
   15 87.080002
   16 137.100006
   17 114.099998
   18 97.120003
   19 103.099998
   20 113.199997
```

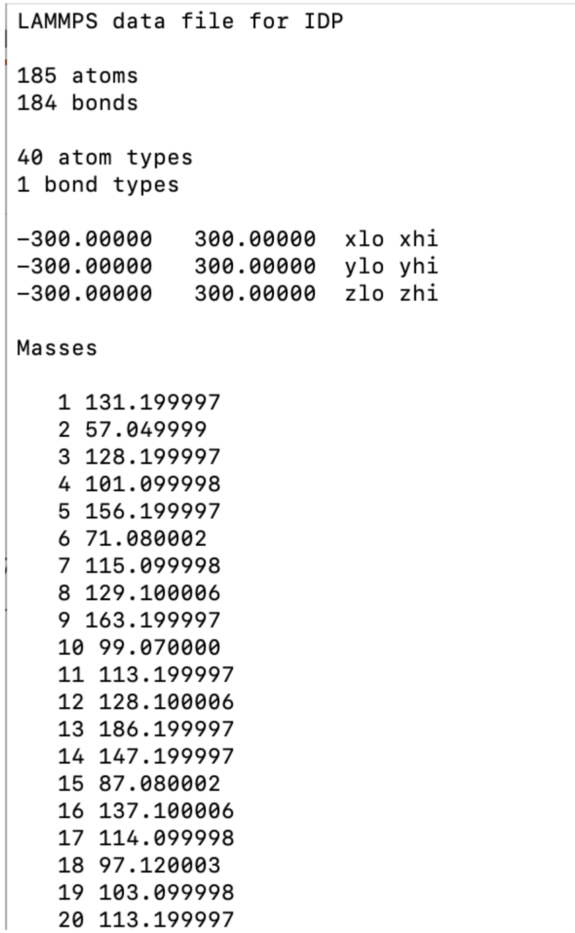
```
   35 87.080002
   36 137.100006
   37 114.099998
   38 97.120003
   39 103.099998
   40 113.199997

Atoms

1 1 1  0.000000 6.000000  0.000000  0.000000
2 1 6  0.000000 12.000000  0.000000  0.000000
3 1 8  -1.000000 18.000000  0.000000  0.000000
4 1 18  0.000000 24.000000  0.000000  0.000000
5 1 5  1.000000 30.000000  0.000000  0.000000
6 1 12  0.000000 36.000000  0.000000  0.000000
7 1 8  -1.000000 42.000000  0.000000  0.000000
8 1 14  0.000000 48.000000  0.000000  0.000000
9 1 8  -1.000000 54.000000  0.000000  0.000000
10 1 10  0.000000 60.000000  0.000000  0.000000
11 1 1  0.000000 66.000000  0.000000  0.000000
12 1 8  -1.000000 72.000000  0.000000  0.000000
13 1 7  -1.000000 78.000000  0.000000  0.000000
14 1 16  0.500000 84.000000  0.000000  0.000000
15 1 6  0.000000 90.000000  0.000000  0.000000
16 1 2  0.000000 96.000000  0.000000  0.000000
17 1 4  0.000000 102.000000  0.000000  0.000000
18 1 9  0.000000 108.000000  0.000000  0.000000
19 1 2  0.000000 114.000000  0.000000  0.000000
20 1 11  0.000000 120.000000  0.000000  0.000000
21 1 2  0.000000 126.000000  0.000000  0.000000
```

Configuration file: Connectivity (bonds)

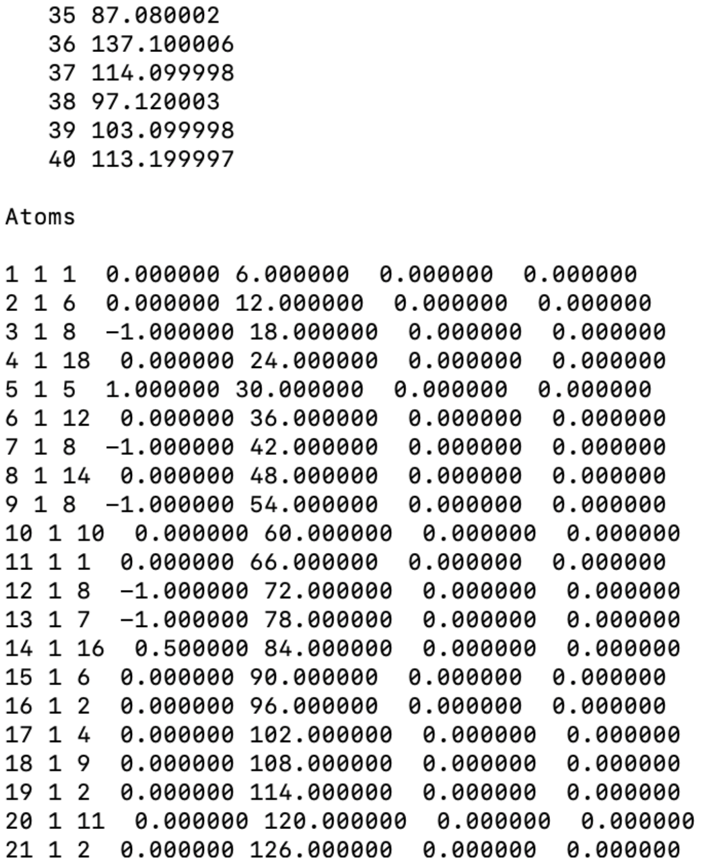
K25
```
MAEPR QEFEV MEDHA GTYGL GDRKD QGGYT MHQDQ EGDTD AGLKA EEAGI GDTPS LEDEA AGHVT QARMV
SKSKD GTGSD DKKAK GADGK TKIAT PRGAA PPGQK GQANA TRIPA KTPPA PKTPP SSGEP PKSGD RSGYS
SPGSP GTPGS RSRTP SLPTP PTREP KKVAV VRTPP KSPSS AKSRL
```



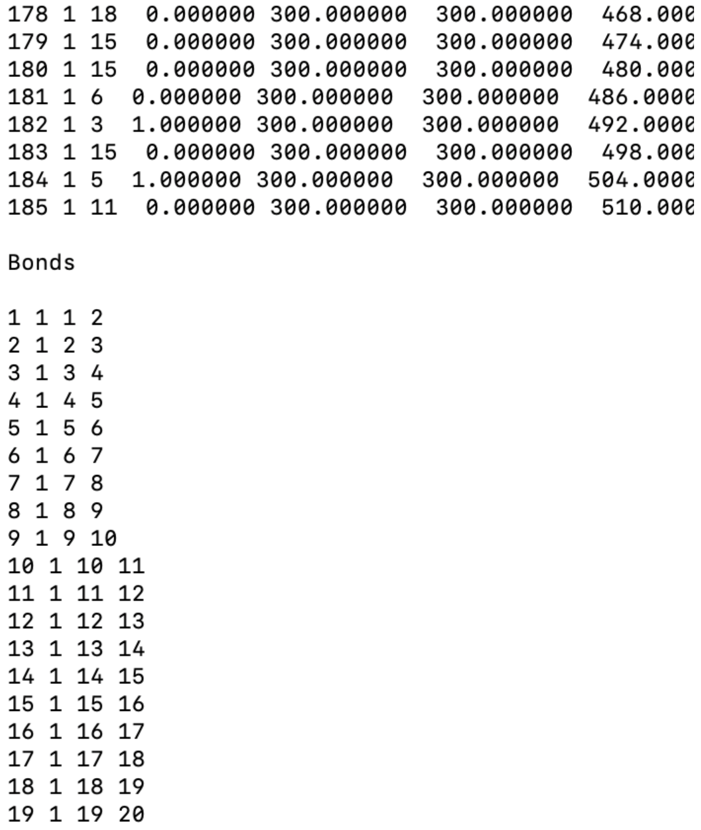
```
M    1
G    2
K    3
T    4
R    5
A    6
D    7
E    8
Y    9
V   10
L   11
Q   12
W   13
F   14
S   15
H   16
N   17
P   18
C   19
I   20
```



```
LAMMPS data file for IDP

185 atoms
184 bonds

40 atom types
1 bond types

-300.00000   300.00000  xlo xhi
-300.00000   300.00000  ylo yhi
-300.00000   300.00000  zlo zhi

Masses

   1 131.199997
   2 57.049999
   3 128.199997
   4 101.099998
   5 156.199997
   6 71.080002
   7 115.099998
   8 129.100006
   9 163.199997
   10 99.070000
   11 113.199997
   12 128.100006
   13 186.199997
   14 147.199997
   15 87.080002
   16 137.100006
   17 114.099998
   18 97.120003
   19 103.099998
   20 113.199997
```



```
   35 87.080002
   36 137.100006
   37 114.099998
   38 97.120003
   39 103.099998
   40 113.199997

Atoms

1 1 1  0.000000 6.000000  0.000000  0.000000
2 1 6  0.000000 12.000000  0.000000  0.000000
3 1 8  -1.000000 18.000000  0.000000  0.000000
4 1 18  0.000000 24.000000  0.000000  0.000000
5 1 5  1.000000 30.000000  0.000000  0.000000
6 1 12  0.000000 36.000000  0.000000  0.000000
7 1 8  -1.000000 42.000000  0.000000  0.000000
8 1 14  0.000000 48.000000  0.000000  0.000000
9 1 8  -1.000000 54.000000  0.000000  0.000000
10 1 10  0.000000 60.000000  0.000000  0.000000
11 1 1  0.000000 66.000000  0.000000  0.000000
12 1 8  -1.000000 72.000000  0.000000  0.000000
13 1 7  -1.000000 78.000000  0.000000  0.000000
14 1 16  0.500000 84.000000  0.000000  0.000000
15 1 6  0.000000 90.000000  0.000000  0.000000
16 1 2  0.000000 96.000000  0.000000  0.000000
17 1 4  0.000000 102.000000  0.000000  0.000000
18 1 9  0.000000 108.000000  0.000000  0.000000
19 1 2  0.000000 114.000000  0.000000  0.000000
20 1 11  0.000000 120.000000  0.000000  0.000000
21 1 2  0.000000 126.000000  0.000000  0.000000
```



```
178 1 18  0.000000 300.000000  300.000000  468.000
179 1 15  0.000000 300.000000  300.000000  474.000
180 1 15  0.000000 300.000000  300.000000  480.000
181 1 6  0.000000 300.000000  300.000000  486.0000
182 1 3  1.000000 300.000000  300.000000  492.0000
183 1 15  0.000000 300.000000  300.000000  498.000
184 1 5  1.000000 300.000000  300.000000  504.0000
185 1 11  0.000000 300.000000  300.000000  510.000

Bonds

1 1 1 2
2 1 2 3
3 1 3 4
4 1 4 5
5 1 5 6
6 1 6 7
7 1 7 8
8 1 8 9
9 1 9 10
10 1 10 11
11 1 11 12
12 1 12 13
13 1 13 14
14 1 14 15
15 1 15 16
16 1 16 17
17 1 17 18
18 1 18 19
19 1 19 20
```

LAMMPS parameter file: Force field information

K25

```
MAEPR QEFEV MEDHA GTYGL GDRKD QGGYT MHQDQ EGDTD AGLKA EEAGI GDTPS LEDEA AGHVT QARMV
SKSKD GTGSD DKKAK GADGK TKIAT PRGAA PPGQK GQANA TRIPA KTPPA PKTPP SSGEP PKSGD RSGYS
SPGSP GTPGS RSRTP SLPTP PTREP KKVAV VRTPP KSPSS AKSRL
```

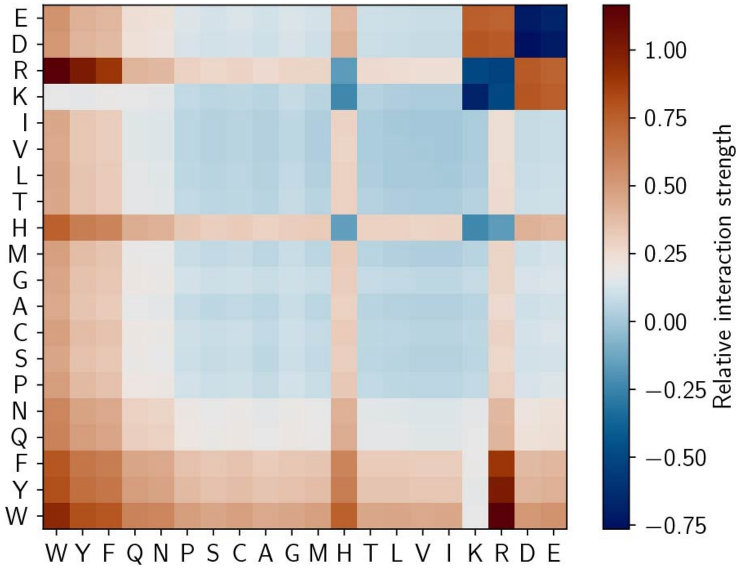
```
bond_coeff          1     9.600     3.81
pair_style  hybrid/overlay wf/cut 25.0 coul/debye 0.131 0.0

pair_coeff 1 1 wf/cut 0.039564 6.467950871367303 1 2 19.403852614101908
pair_coeff 1 2 wf/cut 0.068017 5.576178403526399 1 2 16.728535210579196
pair_coeff 1 3 wf/cut 0.032686 6.5677835361737795 1 2 19.70335060852134
pair_coeff 1 4 wf/cut 0.035161 6.178228033595178 1 2 18.534684100785533
pair_coeff 1 5 wf/cut 0.171515 6.643964892255349 1 2 19.93189467676605
pair_coeff 1 6 wf/cut 0.044522 5.868502140125925 1 2 17.605506420377775
pair_coeff 1 7 wf/cut 0.073172 6.14145960937047 1 2 18.424378828111408
pair_coeff 1 8 wf/cut 0.077577 6.318176525841784 1 2 18.95452957752535
pair_coeff 1 9 wf/cut 0.229375 6.590037435016079 1 2 19.770112305048237
pair_coeff 1 10 wf/cut 0.022571 6.320571545407576 1 2 18.96171463622273
pair_coeff 1 11 wf/cut 0.025281 6.483443734640188 1 2 19.450331203920562
pair_coeff 1 12 wf/cut 0.120006 6.36518785664648 1 2 19.09556356993944
pair_coeff 1 13 wf/cut 0.294931 6.755728450520449 1 2 20.267185351561345
pair_coeff 1 14 wf/cut 0.215603 6.538208522101147 1 2 19.61462556630344
pair_coeff 1 15 wf/cut 0.050582 5.938940625510648 1 2 17.816821876531947
pair_coeff 1 16 wf/cut 0.203491 6.392507281968151 1 2 19.177521845904455
pair_coeff 1 17 wf/cut 0.116706 6.188018494610175 1 2 18.564055483830522
pair_coeff 1 18 wf/cut 0.059120 6.1344640513130955 1 2 18.403392153939286
pair_coeff 1 19 wf/cut 0.054437 6.094294827619396 1 2 18.282884482858186
pair_coeff 1 20 wf/cut 0.019979 6.496393035701747 1 2 19.48917910710524
pair_coeff 1 21 wf/cut 0.033511 6.472967463905089 1 2 19.418902391715267
pair_coeff 1 22 wf/cut 0.053464 5.579884038976499 1 2 16.7396521169295
pair_coeff 1 23 wf/cut 0.028840 6.572908139010548 1 2 19.718724417031645
pair_coeff 1 24 wf/cut 0.030487 6.183085528083141 1 2 18.549256584249424
pair_coeff 1 25 wf/cut 0.126057 6.646517494086263 1 2 19.93955248225879
pair_coeff 1 26 wf/cut 0.037087 5.873266644604083 1 2 17.61979993381225
pair_coeff 1 27 wf/cut 0.057192 6.145621669676131 1 2 18.43686500902839
pair_coeff 1 28 wf/cut 0.060220 6.322430858089529 1 2 18.967292574268587
pair_coeff 1 29 wf/cut 0.166559 6.592040039094211 1 2 19.776120117282634
pair_coeff 1 30 wf/cut 0.022157 6.322886509244067 1 2 18.9686595277322
pair_coeff 1 31 wf/cut 0.023649 6.486360198534797 1 2 19.45908059560439
pair_coeff 1 32 wf/cut 0.089698 6.368425539955892 1 2 19.105276619867674
pair_coeff 1 33 wf/cut 0.212282 6.7573688868401405 1 2 20.27210666052042
pair_coeff 1 34 wf/cut 0.156918 6.5402914181442595 1 2 19.620874254432778
pair_coeff 1 35 wf/cut 0.041217 5.943583554201671 1 2 17.830750662605013
pair_coeff 1 36 wf/cut 0.148384 6.394666170151445 1 2 19.183998510454334
pair_coeff 1 37 wf/cut 0.087498 6.191219964195186 1 2 18.57365989258556
pair_coeff 1 38 wf/cut 0.047275 6.139108925912469 1 2 18.417326777737408
pair_coeff 1 39 wf/cut 0.043970 6.098946985727069 1 2 18.296840957181207
pair_coeff 1 40 wf/cut 0.019979 6.496393035701747 1 2 19.48917910710524
pair_coeff 2 2 wf/cut 0.096470 4.695110240398406 1 2 14.085330721195216
pair_coeff 2 3 wf/cut 0.061139 5.671338290828199 1 2 17.014014872484598
```

LAMMPS parameter file: Force field information

K25

```
MAEPR QEFEV MEDHA GTYGL GDRKD QGGYT MHQDQ EGDTD AGLKA EEAGI GDTPS LEDEA AGHVT QARMV
SKSKD GTGSD DKKAK GADGK TKIAT PRGAA PPGQK GQANA TRIPA KTPPA PKTPP SSGEP PKSGD RSGYS
SPGSP GTPGS RSRTP SLPTP PTREP KKVAV VRTPP KSPSS AKSRL
```

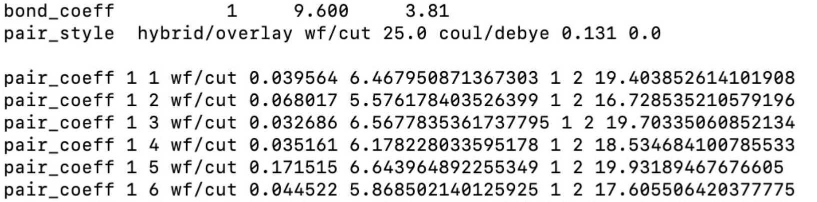
```
bond_coeff          1     9.600     3.81
pair_style  hybrid/overlay wf/cut 25.0 coul/debye 0.131 0.0

pair_coeff 1 1 wf/cut 0.039564 6.467950871367303 1 2 19.403852614101908
pair_coeff 1 2 wf/cut 0.068017 5.576178403526399 1 2 16.728535210579196
pair_coeff 1 3 wf/cut 0.032686 6.5677835361737795 1 2 19.70335060852134
pair_coeff 1 4 wf/cut 0.035161 6.178228033595178 1 2 18.534684100785533
pair_coeff 1 5 wf/cut 0.171515 6.643964892255349 1 2 19.93189467676605
pair_coeff 1 6 wf/cut 0.044522 5.868502140125925 1 2 17.605506420377775
```



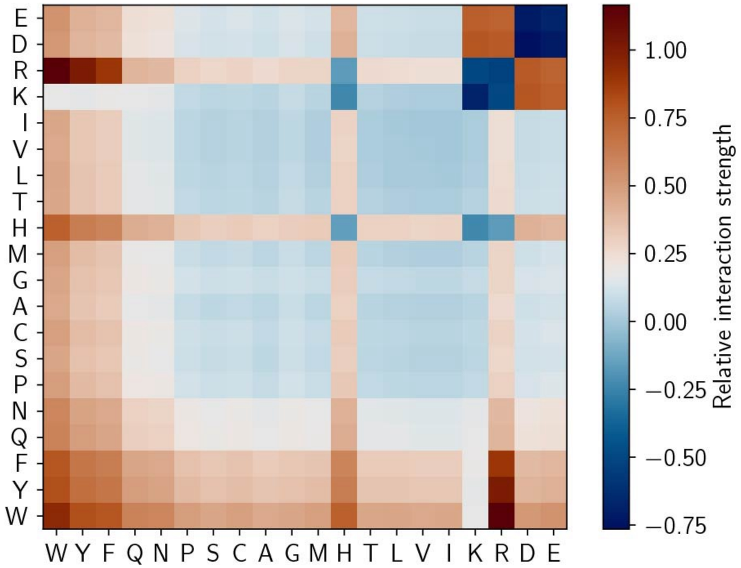
```
pair_coeff 1 34 wf/cut 0.156918 6.5402914181442595 1 2 19.620874254432778
pair_coeff 1 35 wf/cut 0.041217 5.943583554201671 1 2 17.830750662605013
pair_coeff 1 36 wf/cut 0.148384 6.394666170151445 1 2 19.183998510454334
pair_coeff 1 37 wf/cut 0.087498 6.191219964195186 1 2 18.57365989258556
pair_coeff 1 38 wf/cut 0.047275 6.139108925912469 1 2 18.417326777737408
pair_coeff 1 39 wf/cut 0.043970 6.098946985727069 1 2 18.296840957181207
pair_coeff 1 40 wf/cut 0.019979 6.496393035701747 1 2 19.48917910710524
pair_coeff 2 2 wf/cut 0.096470 4.695110240398406 1 2 14.085330721195216
pair_coeff 2 3 wf/cut 0.061139 5.671338290828199 1 2 17.014014872484598
```

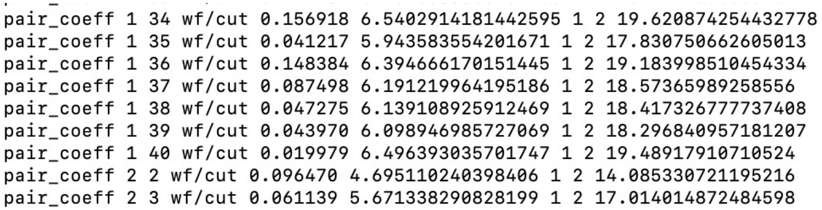
LAMMPS parameter file: Timestep, thermostat etc

K25 MAEPR QEFEV MEDHA GTYGL GDRKD QGGYT MHQDQ EGDTD AGLKA EEAGI GDTPS LEDEA AGHVT QARMV
SKSKD GTGSD DKKAK GADGK TKIAT PRGAA PPGQK GQANA TRIPA KTPPA PKTPP SSGEP PKSGD RSGYS
SPGSP GTPGS RSRTP SLPTP PTREP KKVAV VRTPP KSPSS AKSRL

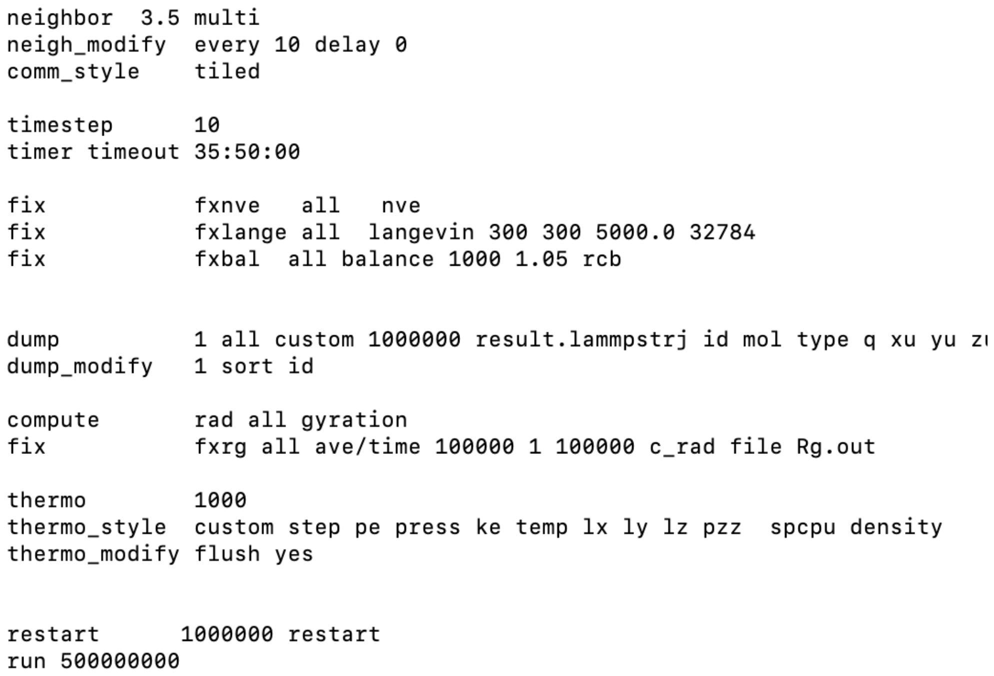


```
bond_coeff          1     9.600     3.81
pair_style  hybrid/overlay wf/cut 25.0 coul/debye 0.131 0.0

pair_coeff 1 1 wf/cut 0.039564 6.467950871367303 1 2 19.403852614101908
pair_coeff 1 2 wf/cut 0.068017 5.576178403526399 1 2 16.728535210579196
pair_coeff 1 3 wf/cut 0.032686 6.5677835361737795 1 2 19.70335060852134
pair_coeff 1 4 wf/cut 0.035161 6.178228033595178 1 2 18.534684100785533
pair_coeff 1 5 wf/cut 0.171515 6.643964892255349 1 2 19.93189467676605
pair_coeff 1 6 wf/cut 0.044522 5.868502140125925 1 2 17.605506420377775
```





```
pair_coeff 1 34 wf/cut 0.156918 6.5402914181442595 1 2 19.620874254432778
pair_coeff 1 35 wf/cut 0.041217 5.943583554201671 1 2 17.830750662605013
pair_coeff 1 36 wf/cut 0.148384 6.394666170151445 1 2 19.183998510454334
pair_coeff 1 37 wf/cut 0.087498 6.191219964195186 1 2 18.5736598925856
pair_coeff 1 38 wf/cut 0.047275 6.139108925912469 1 2 18.417326777737408
pair_coeff 1 39 wf/cut 0.043970 6.098946985727069 1 2 18.296840957181207
pair_coeff 1 40 wf/cut 0.019979 6.496393035701747 1 2 19.48917910710524
pair_coeff 2 2 wf/cut 0.096470 4.695110240398406 1 2 14.085330721195216
pair_coeff 2 3 wf/cut 0.061139 5.671338290828199 1 2 17.01401487248459
```

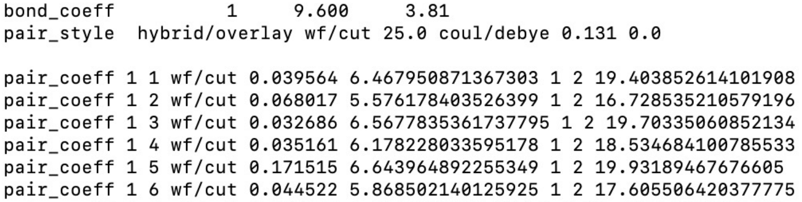


```
neighbor  3.5 multi
neigh_modify  every 10 delay 0
comm_style    tiled

timestep      10
timer timeout 35:50:00

fix           fxnve   all   nve
fix           fxlange all  langevin 300 300 5000.0 32784
fix           fxbal  all balance 1000 1.05 rcb


dump          1 all custom 1000000 result.lammpstrj id mol type q xu yu zu
dump_modify   1 sort id

compute       rad all gyration
fix           fxrg all ave/time 100000 1 100000 c_rad file Rg.out

thermo        1000
thermo_style  custom step pe press ke temp lx ly lz pzz  spcpu density
thermo_modify flush yes


restart      1000000 restart
run 500000000
```

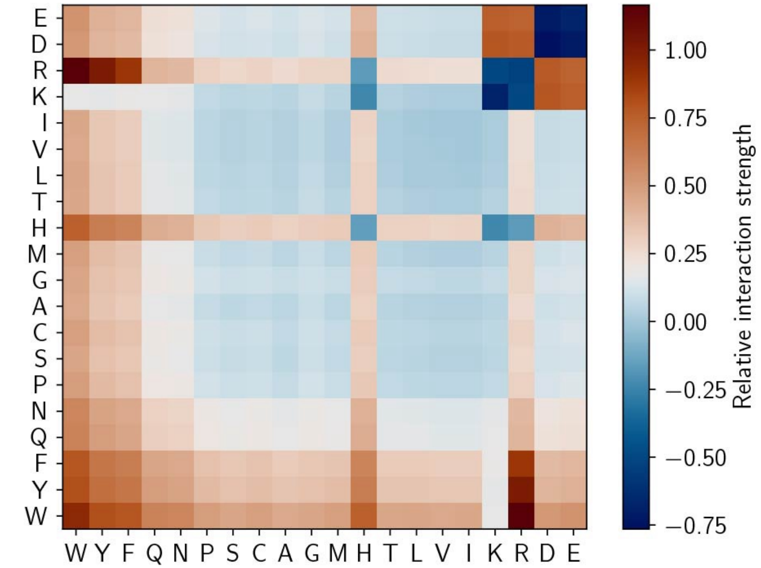
Simulation output file: Timestep vs *R*g

K25

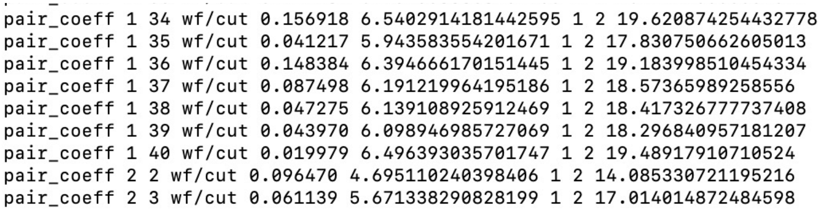
```
MAEPR QEFEV MEDHA GTYGL GDRKD QGGYT MHQDQ EGDTD AGLKA EEAGI GDTPS LEDEA AGHVT QARMV
SKSKD GTGSD DKKAK GADGK TKIAT PRGAA PPGQK GQANA TRIPA KTPPA PKTPP SSGEP PKSGD RSGYS
SPGSP GTPGS RSRTP SLPTP PTREP KKVAV VRTPP KSPSS AKSRL
```



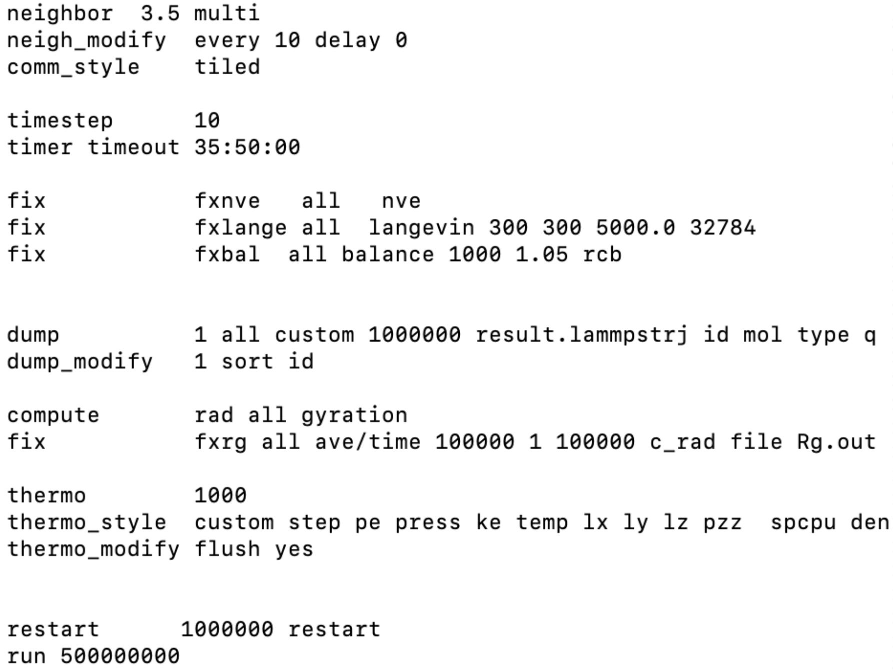
```
bond_coeff          1      9.600      3.81
pair_style  hybrid/overlay wf/cut 25.0 coul/debye 0.131 0.0

pair_coeff 1 1 wf/cut 0.039564 6.467950871367303 1 2 19.403852614101908
pair_coeff 1 2 wf/cut 0.068017 5.576178403526399 1 2 16.728535210579196
pair_coeff 1 3 wf/cut 0.032686 6.5677835361737795 1 2 19.70335060852134
pair_coeff 1 4 wf/cut 0.035161 6.178228033595178 1 2 18.534684100785533
pair_coeff 1 5 wf/cut 0.171515 6.643964892255349 1 2 19.93189467676605
pair_coeff 1 6 wf/cut 0.044522 5.868502140125925 1 2 17.605506420377775
```



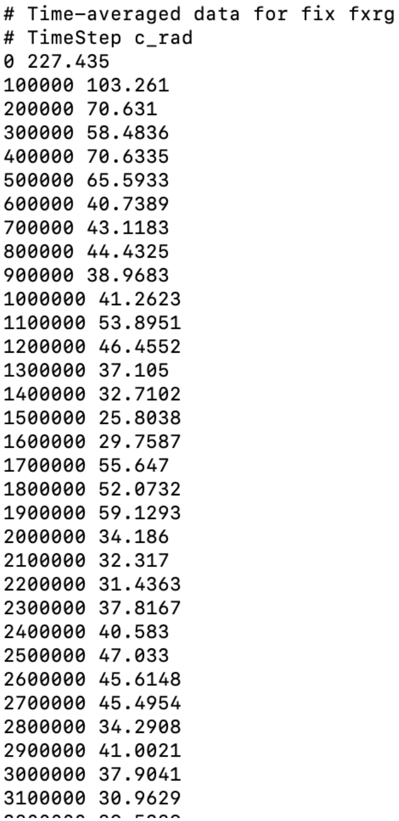
6



```
pair_coeff 1 34 wf/cut 0.156918 6.5402914181442595 1 2 19.620874254432778
pair_coeff 1 35 wf/cut 0.041217 5.943583554201671 1 2 17.830750662605013
pair_coeff 1 36 wf/cut 0.148384 6.394666170151445 1 2 19.183998510454334
pair_coeff 1 37 wf/cut 0.087498 6.191219964195186 1 2 18.5736598925855
pair_coeff 1 38 wf/cut 0.047275 6.139108925912469 1 2 18.417326777737408
pair_coeff 1 39 wf/cut 0.043970 6.098946985727069 1 2 18.296840957181207
pair_coeff 1 40 wf/cut 0.019979 6.496393035701747 1 2 19.48917910710524
pair_coeff 2 2 wf/cut 0.096470 4.695110240398406 1 2 14.085330721195216
pair_coeff 2 3 wf/cut 0.061139 5.671338290828199 1 2 17.014014872484598
```



```
neighbor   3.5 multi
neigh_modify   every 10 delay 0
comm_style     tiled

timestep       10
timer timeout 35:50:00

fix            fxnve   all   nve
fix            fxlange all  langevin 300 300 5000.0 32784
fix            fxbal   all balance 1000 1.05 rcb


dump           1 all custom 1000000 result.lammpstrj id mol type q
dump_modify    1 sort id

compute        rad all gyration
fix            fxrg all ave/time 100000 1 100000 c_rad file Rg.out

thermo         1000
thermo_style   custom step pe press ke temp lx ly lz pzz  spcpu den
thermo_modify  flush yes


restart       1000000 restart
run 500000000
```



```
# Time-averaged data for fix fxrg
# TimeStep c_rad
0 227.435
100000 103.261
200000 70.631
300000 58.4836
400000 70.6335
500000 65.5933
600000 40.7389
700000 43.1183
800000 44.4325
900000 38.9683
1000000 41.2623
1100000 53.8951
1200000 46.4552
1300000 37.105
1400000 32.7102
1500000 25.8038
1600000 29.7587
1700000 55.647
1800000 52.0732
1900000 59.1293
2000000 34.186
2100000 32.317
2200000 31.4363
2300000 37.8167
2400000 40.583
2500000 47.033
2600000 45.6148
2700000 45.4954
2800000 34.2908
2900000 41.0021
3000000 37.9041
3100000 30.9629
```

Tutorial on MD Software by Nathaniel Hess (including introduction to LAMMPS)

https://drive.google.com/drive/folders/1XaKymQV4uPrjMCQr2crKHWvSGJsr0ZCH?usp=sharing