

Campinas Advanced School of Thermodynamics

Workshop on Molecular Simulations

Session 1 – Basic MD

Edward Maginn and Eliseo Marin

Department of Chemical and Biomolecular Engineering

University of Notre Dame, USA

ed@nd.edu

All materials available on
MaginnGroup GitHub site

28 October, 2022



Set up conda environments

- Follow the instructions provided earlier.
- Activate LAMMPS environment

> conda activate lammmps-tutorial

- Make sure Ovito is installed
- Download workshop materials

<https://github.com/MaginnGroup/UNICAMP-workshop>

History of “models”

- *It seems to me that the test of “Do we or not understand a particular subject in physics?” is “Can we make a mechanical model of it?”*

-W. Thomson (Lord Kelvin)
“Baltimore Lectures on Molecular Dynamics and the Wave Theory of Light”, 1904.

Essence of molecular simulation

- Predict the macroscopic properties of many-body systems
- Described by wave function $\psi(q, t)$
 - \mathbf{q} is position of nuclei and electrons at time t .
 - Solving wave equation gives expectation value of any mechanical property
 - Rigorous, but difficult to do for large systems

Subject of electronic structure theory.
Simplification: treat system *classically*.

- State of system given by generalized coordinates \mathbf{q}^N and generalized momenta \mathbf{p}^N .
- Born-Oppenheimer approximation: neglect position of electrons and focus only on nuclei

Uncertainty principle: we cannot know both \mathbf{q}^N and \mathbf{p}^N exactly. **Let's ignore this!**

Phase space

State vector or system point defined as

$$\mathbf{x}^N \equiv (\mathbf{q}^N, \mathbf{p}^N) = (\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_N; \mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_N)$$

6-N dimensional set from which \mathbf{x}^N takes a value is called the *phase space* of the system Γ

A point on Γ represents a *microstate* of the entire system.

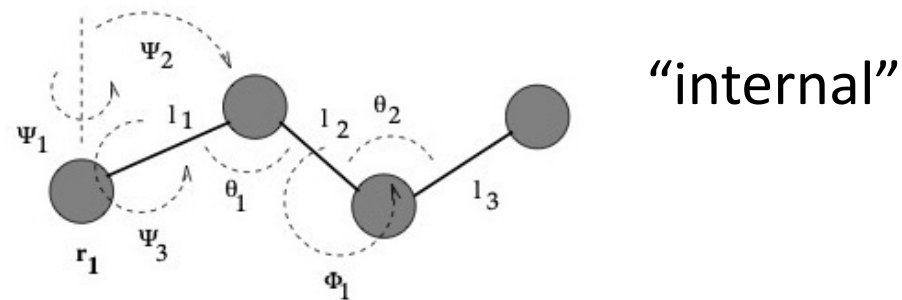
If \mathbf{x}^N is known at one time, it is *completely determined* for any other time (past and present) through the classical equations of motion.

Phase space and state points

- “phase space” is the $6N$ generalized coordinates describing the state of a system: $\mathbf{x}^N(\mathbf{q}^N, \mathbf{p}^N)$
- Positions: \mathbf{q}^N
- Momenta \mathbf{p}^N
- Several coordinate systems possible

N atoms
Bold type = vector

Cartesian:
 $\mathbf{r}(x, y, z)$



Classical energies, forces; equations of motion

$$H(\mathbf{q}, \mathbf{p}) = K + \mathcal{V} \quad \text{“Hamiltonian”}$$

$$K = \sum_{i=1}^N \frac{m_i v_i^2}{2} \quad \text{KE depends on atomic velocities}$$

$$\mathcal{V} = \mathcal{V}(\mathbf{q}) \quad \text{PE depends on atomic positions}$$

$$\mathbf{F} = -\nabla_{\mathbf{q}} \mathcal{V} \quad \text{Force between pairs of atoms is conservative}$$

Energy is conserved

$$dH/dt = 0 - \text{a } \textit{microcanonical ensemble}$$

Newton's equations of motion

$$\frac{d\mathbf{p}}{dt} = m \frac{d\mathbf{v}}{dt} = m\ddot{\mathbf{r}} = m\mathbf{a}$$

$$m_i \ddot{\mathbf{r}}_i = \mathbf{F}_i \quad \leftarrow \text{3N 2nd order differential eqns.}$$

$$\dot{\mathbf{r}}_i = \frac{\dot{\mathbf{p}}_i}{m_i}$$

$$\dot{\mathbf{p}}_i = \mathbf{F}_i \quad \leftarrow \text{6N 1st order differential equations}$$

$$\mathbf{F}_i \equiv -\nabla_{\mathbf{r}_i} U$$

MD uses finite difference

- MD is a means of solving the following initial value problem: $\ddot{r}_i = \frac{F_i(r_1, \dots, r_N)}{m_i}$
- Initial conditions: $r_i(0), \dot{r}_i(0)$
- Trajectories generated numerically using finite difference methods
- We will use LAMMPS (Large-scale Atomic/Molecular Massively Parallel Simulator) lammps.org
- Classical particle simulator
 - Tremendous flexibility, very good documentation, and active user base
- We will start with some simple demonstrations and calculations to understand basics

Microstates, macrostates, and ensembles

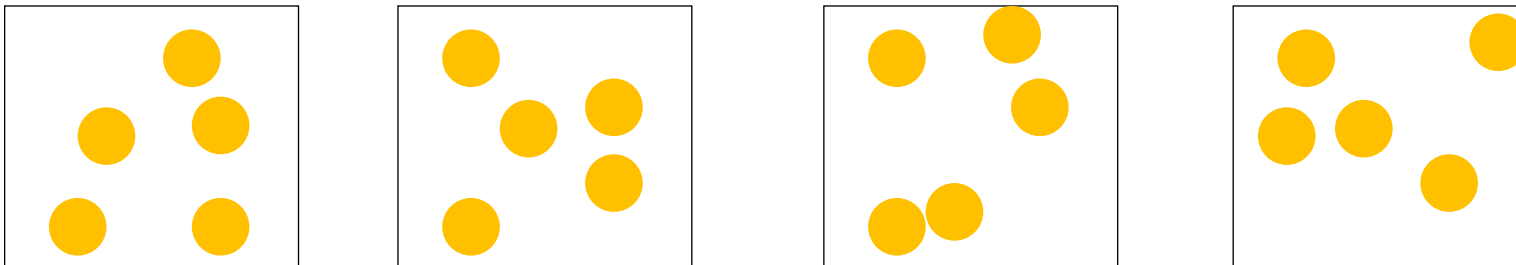
Systems with different *microstates* may exist at the same *thermodynamic state*.

Example: Many different sets of $(\mathbf{q}^N, \mathbf{p}^N)$ can be at the same temperature and pressure.

A thermodynamic state is a *collection* of many different microstates.

- Identical in composition and at the same “macrostate”
- Gibbs called this collection an “ensemble”

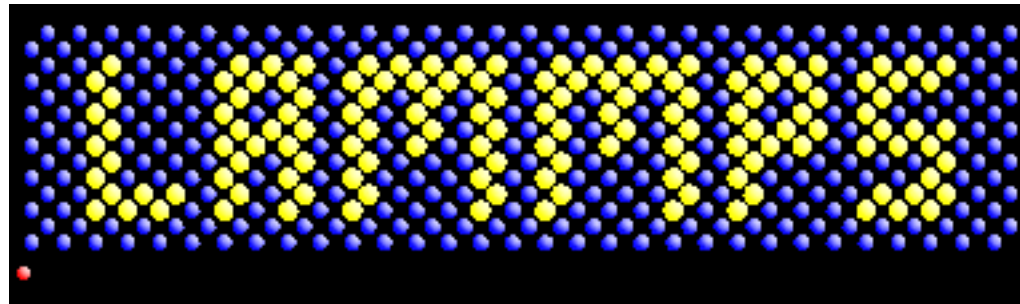
An ensemble is a collection of a large number of systems, each a replica of the thermodynamic state of the system, whose properties we are investigating



Example:
4 states having
different (\mathbf{q}, \mathbf{p}) but
same NVE

Time reversibility

Newtonian mechanics is time-reversible



$$H(\mathbf{x}^N) = H(\mathbf{p}^N, \mathbf{q}^N)$$

given $\mathbf{x}^N(t_1) \longleftrightarrow \mathbf{x}^N(t_2)$ **absolutely known**

Determinism

- Ever since Newton, deterministic, mechanical interpretation of Nature has dominated science.
- *Given for one instant an intelligence which could comprehend all the forces by which Nature is animated and the respective situation of the beings who compose it - an intelligence sufficiently vast to submit these data to analysis - it would embrace in the same formula the movements of the greatest bodies of the universe and those of the lightest atoms; for it, nothing would be uncertain and the future, as the past, would be present to its eyes.*

-- P. S. Laplace, "Philosophical Essay on Probabilities", 1814

Averages in NVE ensemble

Consider an integrable function $f(\mathbf{x}^N)$ of state point \mathbf{x}^N on a constant E-hypersurface. The phase average of $f(\mathbf{x}^N)$ is

$$\langle f \rangle_{NVE} = \frac{1}{\Sigma(E)} \int_{\Gamma} \delta(H(\mathbf{x}^N) - E) f(\mathbf{x}^N) d\mathbf{x}^N$$

$$\langle f \rangle_{NVE} = \frac{\int_{\Gamma} \delta(H(\mathbf{x}^N) - E) f(\mathbf{x}^N) d\mathbf{x}^N}{\int_{\Gamma} \delta(H(\mathbf{x}^N) - E) d\mathbf{x}^N}$$

What we do with MC

Key: phase average taken by averaging over all state points on constant E hypersurface

Could also take a “time average” over a series of snapshots as system evolves *at equilibrium*

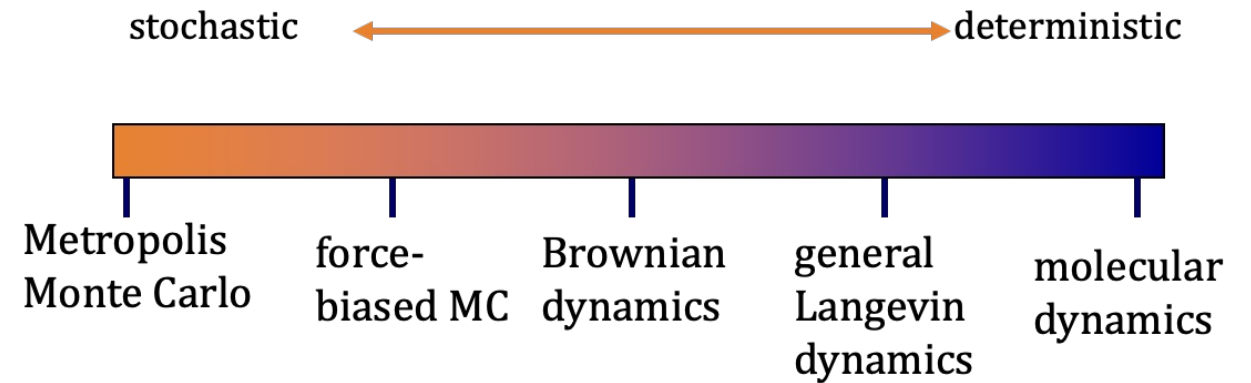
$$\langle f \rangle_t = \lim_{t \rightarrow \infty} \frac{1}{t} \int_{t_0}^{t_0+t} f(\mathbf{x}^N(t)) dt$$

What we do with MD

Steps of a molecular simulation

- A molecular simulation consists of three steps:
 - 1) construction of the molecular model
 - > How are atoms connected and how do they interact?
 - 2) calculation of molecular trajectories
 - > MC is stochastic
 - > MD is deterministic
 - > Many variants “in between”
 - 3) analysis of trajectories => properties

- A spectrum of methods

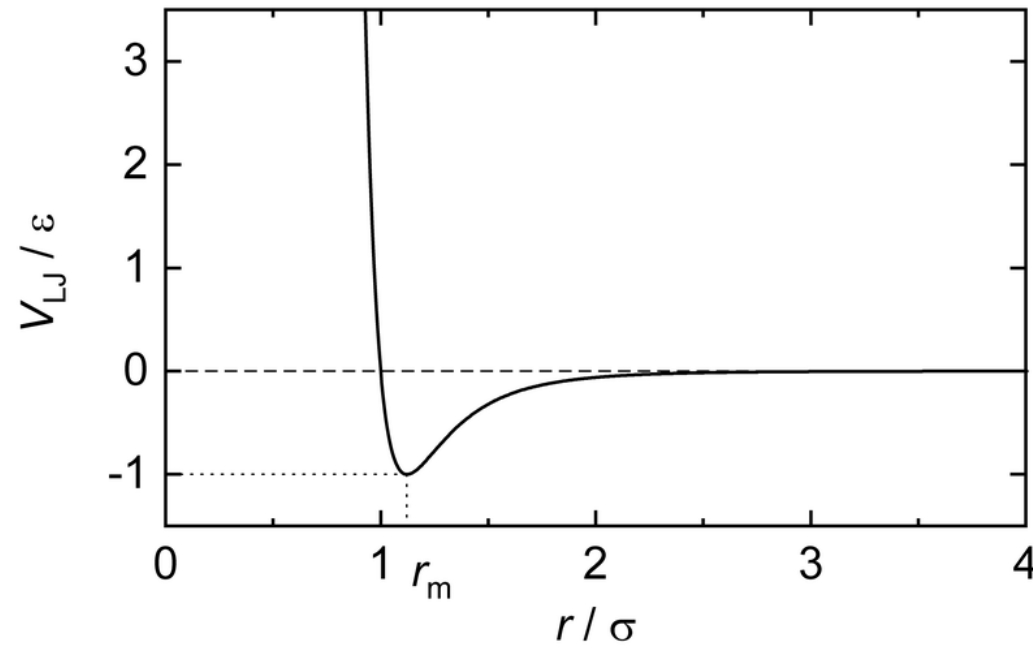


(after Ceperley and Tully)

Running a LAMMPS calculation

- Pass LAMMPS and input file - provides essential information
 - What kind of system
 - How long the run is
 - File names
 - Units
 - Many other things...
- Can provide the force field and molecule information
- A “data file” can also be used to provide force field and molecule information

Lennard-Jones 12-6 potential



$$\mathcal{V}(r_{ij}) = 4\epsilon_{ij} \left[\left(\frac{\sigma_{ij}}{r_{ij}} \right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}} \right)^6 \right]$$

LJ Units used by LAMMPS

- LAMMPS sets the fundamental quantities m , σ , ϵ , and the Boltzmann constant $k_B=1$.
- The masses, distances, energies are multiples of these fundamental values.
- 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.
- Note that using these three properties as base, the unit of time must conform to the relation

- mass = $M^*=M/m$
- distance = $x^*=x/\sigma$
- time = $\tau^*=\tau \sqrt{\epsilon/m\sigma^2}$
- energy = $E^*=E/\epsilon$
- velocity = $v^*=v \tau/\sigma$
- force = $f^*=f \sigma/\epsilon$
- temperature = reduced LJ temperature, $T^*=T k_B / \epsilon$
- pressure = reduced LJ pressure, $P^* = P \sigma^3 / \epsilon$

Example 1: Two LJ particles in an LJ sphere

Open example1.inp in your editor

```
# set name for this simulation
variable NAME index example1 # name style value
```

You can call this anything you want. Suggestion is to develop a system so you can understand and organize results

```
# set log file name. ${NAME} is the name you gave above.
# This will append "log" to it so have a logfile of the run.
log ${NAME}.log
```

comments can start a line or come after an entry

```
# configure and initialize system
units lj      # Lennard-Jones reduced units (not real)
atom_style atomic      # Simple atoms
boundary f f f # x y z (f = fixed boundary conditions)
```

LJ units are dimensionless units for the LJ system

“real” units:
Length = Å
Energy = kcal/mol
Time = fs

“fixed” boundary will be a rigid LJ sphere

```
# Define the region of the simulation andx then create
# the simulation box
region MYBOX sphere 0 0 0 5 units box
create_box 1 MYBOX # number_of_atom_types region_ID
```

**Creates the simulation
“box” of diameter 5 LJ
units and centered at 0,0,0**

```
# Create the atoms and place them at a given position
create_atoms 1 single 0 2.0 0 units box # atom_type style
args
create_atoms 1 single 0 -2.0 0 units box # second
symmetric atom, non-ergodic
#create_atoms 1 single 0 -1.0 0 units box # second
asymmetric atom, ergodic
```

**Place a single atom of type
1 at position 0,2,0**

**Place a second atom of the
same type at position 0,-
1,0.**

```
mass 1 1.0 # atom_type mass
```

**Commented out: you could
create more atoms at
other positions for fun**

**The mass of atoms of type 1 is
1.0 (LJ units)**

```
# explicitly set atom velocities, use box units, do *not* zero total momentum
velocity all set 1 0 0 units box mom no # group_ID style args keyword value
```

**Atoms need initial velocities.
Give them all +1,0,0**

```
# set forcefield parameters
pair_style lj/cut 2.5 # style args(=cutoff)
pair_coeff 1 1 1.0 1.0 2.5 # atom_i atom_j args(=epsilon,sigma,cutoff)
#pair_coeff 1 1 1.0 1.0 1.0 # repulsive only
```

**Define the LJ
potential**

```
# create a spherical wall to contain the atom(s)
fix SPHERE all wall/region MYBOX lj126 1.0 1.0 2.5 # fix_ID group_ID style
args
#fix SPHERE all wall/region MYBOX lj126 1.0 1.0 1.0 # repulsive only
```

```
# include potential energy of wall fix in thermo 'pe' output
fix_modify SPHERE energy yes # fix_ID keyword value
```

**Make the sphere have
an LJ potential (cutoff
of 1.0 is a repulsive
“Sutherland” potential**

**Include energy between atoms and
wall in PE LAMMPS calculates**

```
# configure integrator
fix NVE all nve # fix_ID group_ID style
timestep 0.003 # The default timestep size is 0.005 for LJ units.
```

**Tell LAMMPS how to integrate
the EOM**

```
# configure output of atom-specific trajectory data (e.g. coordinates)
dump DUMP all atom 50 ${NAME}.lammprj # dump_ID group_ID style interval file args
#dump_modify DUMP image yes # dump_ID keyword value
```

**“dump” coordinates to
trajectory file for analysis
every 50 steps**

```
# configure output of system property data
thermo_style custom step temp press pe ke etotal
thermo 1000 # output_interval
```

**What properties do you want
written out every 1000 steps**

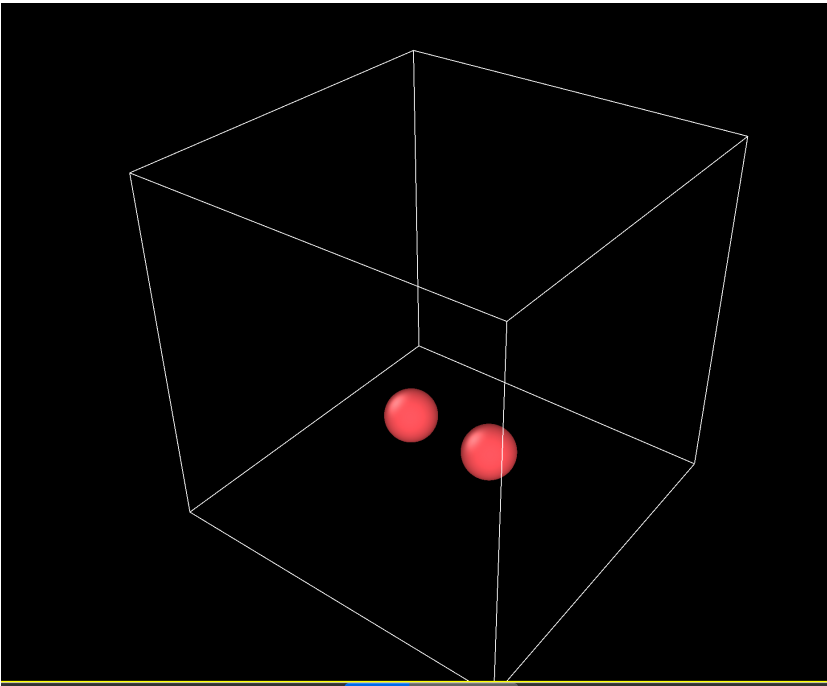
```
# run simulation
run 100000 # number_of_steps
write_restart ${NAME}.restart.*
```

Ignore other stuff for now...

**Tell LAMMPS to run MD for 100000 steps and then write
a restart file that can be used to run another simulation
from the end of the first one**

Example 1 output

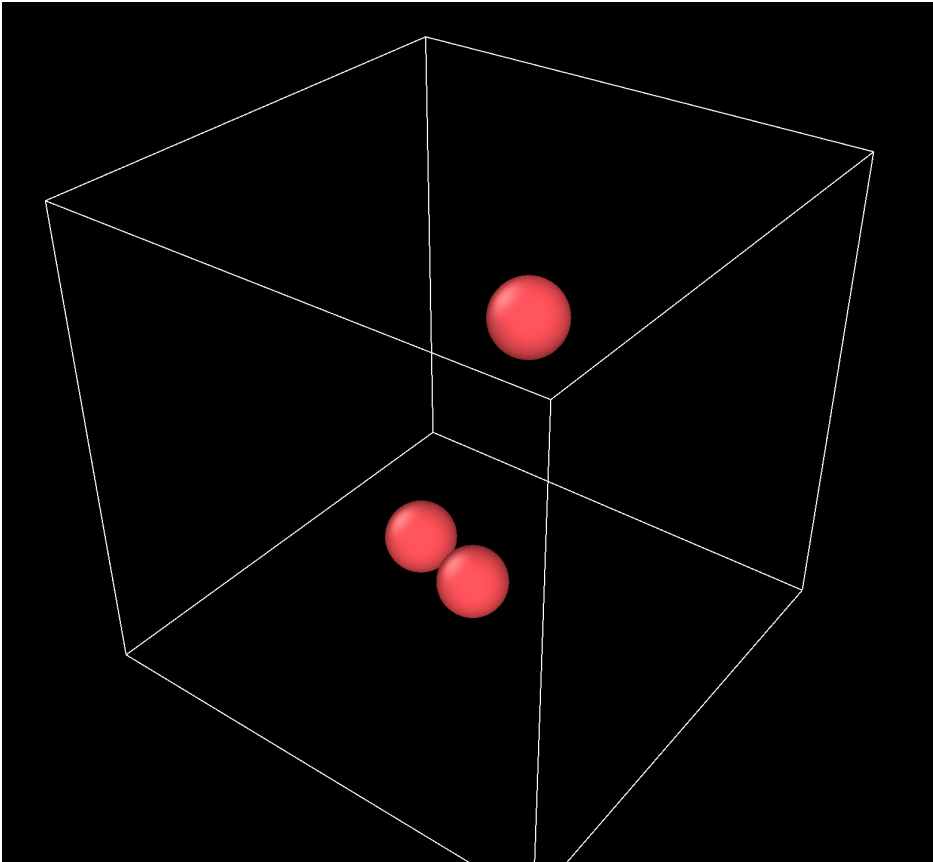
Log file has properties



Step	Temp	Press	PotEng	KinEng	TotEng
0	0.66666667	0.00066666667	0	0.5	0.5
1000	1.9276018	0.0019276018	0	1.4457013	1.4457013
2000	0.66997905	0.00066997905	0	0.50248429	0.50248429
3000	0.72222642	0.00072222642	0	0.54166981	0.54166981
4000	0.69498405	0.00069498405	0	0.52123804	0.52123804
5000	0.75054381	0.00075054381	0	0.56290786	0.56290786
6000	0.74614754	0.00074614754	0	0.55961065	0.55961065
7000	0.7478995	0.0007478995	0	0.56092462	0.56092462
8000	0.86172315	0.00077235532	-0.022603356	0.64629237	0.62368901
9000	1.1769	0.0011769	0	0.88267501	0.88267501
10000	1.8976579	0.0018976579	0	1.4232434	1.4232434
11000	1.4491928	0.003827198	-0.41223311	1.0868946	0.67466147
12000	1.1983254	0.0011983254	0	0.89874403	0.89874403
13000	0.91551879	0.00091551879	0	0.6866391	0.6866391
14000	1.7583265	0.00078678268	-0.4136154	1.3187448	0.90512944
15000	1.2961229	0.0012961229	0	0.97209217	0.97209217

- Why is energy not conserved?
- System is not “ergodic” as molecules do not explore all available space.
- Add the third atom to the box and re-run. What happens?

3 atoms - ergodic



Step	Temp	Press	PotEng	KinEng	TotEng
0	0.5	0.009	0	0.5	0.5
1000	0.83477215	0.0015143069	-0.02617744	0.83477215	0.80859471
2000	1.1149492	0.0022298985	0	1.1149492	1.1149492
3000	1.1561885	0.002312377	0	1.1561885	1.1561885
4000	0.75993387	0.0012081026	-0.054380556	0.75993387	0.70555331
5000	1.3042038	0.0016449011	-0.21512011	1.3042038	1.0890837
6000	0.82754341	0.0016550868	0	0.82754341	0.82754341
7000	1.0369676	0.0011097658	-0.21546579	1.0369676	0.82150178
8000	0.77669148	0.0013916085	-0.027556777	0.77669148	0.7491347
9000	0.52073028	0.00095543258	-0.014417638	0.52073028	0.50631264
10000	0.60680212	0.0011767117	-0.0061776457	0.60680212	0.60062447
11000	0.83508363	0.001319143	-0.06086272	0.83508363	0.77422091
12000	0.71825183	0.0013513439	-0.014351174	0.71825183	0.70390065
13000	0.70277162	0.0014055432	0	0.70277162	0.70277162

System is ergodic, but energy is still not conserved.

Integrating EOM – Verlet Algorithm

- Many “flavors” of this (good) algorithm

$$r(t + \Delta t) = 2r(t) - r(t - \Delta t) + \Delta t^2 a(t)$$

- Where did this come from, and where did the velocities go???
- Taylor series about $r(t)$

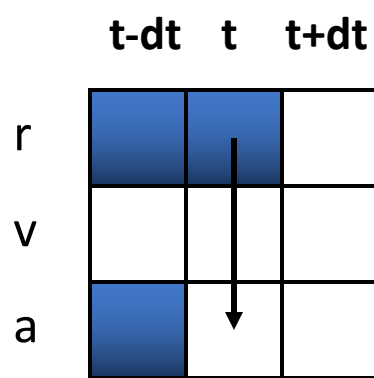
$$r(t + \Delta t) = r(t) + v(t)\Delta t + \frac{1}{2} \Delta t^2 a(t) + \frac{1}{6} \Delta t^3 b(t)$$

$$r(t - \Delta t) = r(t) - v(t)\Delta t + \frac{1}{2} \Delta t^2 a(t) - \frac{1}{6} \Delta t^3 b(t)$$

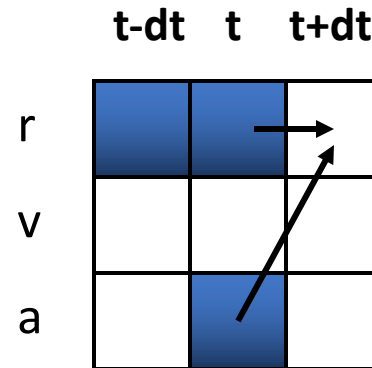
add two eqns
to get Verlet
algorithm

Verlet Algorithm

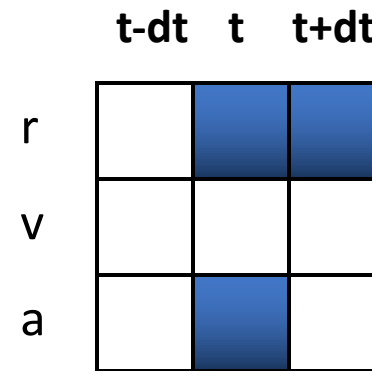
- A graphical representation



get a using $r(t)$



use a to advance
to $r(t+dt)$



arrive at $r(t+dt)$

Verlet Algorithm

- No explicit velocities...problem?
- Yes! Need them to compute temperature
- Estimate from

$$v(t) = \frac{r(t + \Delta t) - r(t - \Delta t)}{2\Delta t}$$

- Positive aspects of Verlet algorithm
- accurate to Δt^4
- time reversible
- good energy conservation
- one-step procedure (no predict-correct)
- very easy to code

Verlet Algorithm

- Drawbacks of Verlet algorithm
 - handling velocities is a bit awkward
 - more important: can lead to *numerical instabilities*

$$r(t + \Delta t) = [2r(t) - r(t - \Delta t)] + \Delta t^2 a(t)$$

↑
relatively large number

↖
very small
number
squared

rule of thumb: all numbers in an operation should be of the same order of magnitude.

Verlet “Leap Frog”

- Verlet “Leap Frog” an improvement over standard Verlet

1.
$$r(t + \Delta t) = r(t) + \Delta t v\left(t + \frac{1}{2} \Delta t\right)$$

2.
$$v\left(t + \frac{1}{2} \Delta t\right) = v\left(t - \frac{1}{2} \Delta t\right) + \Delta t a(t)$$

Default in Gromacs

Verlet “Leap Frog”

- Implementation

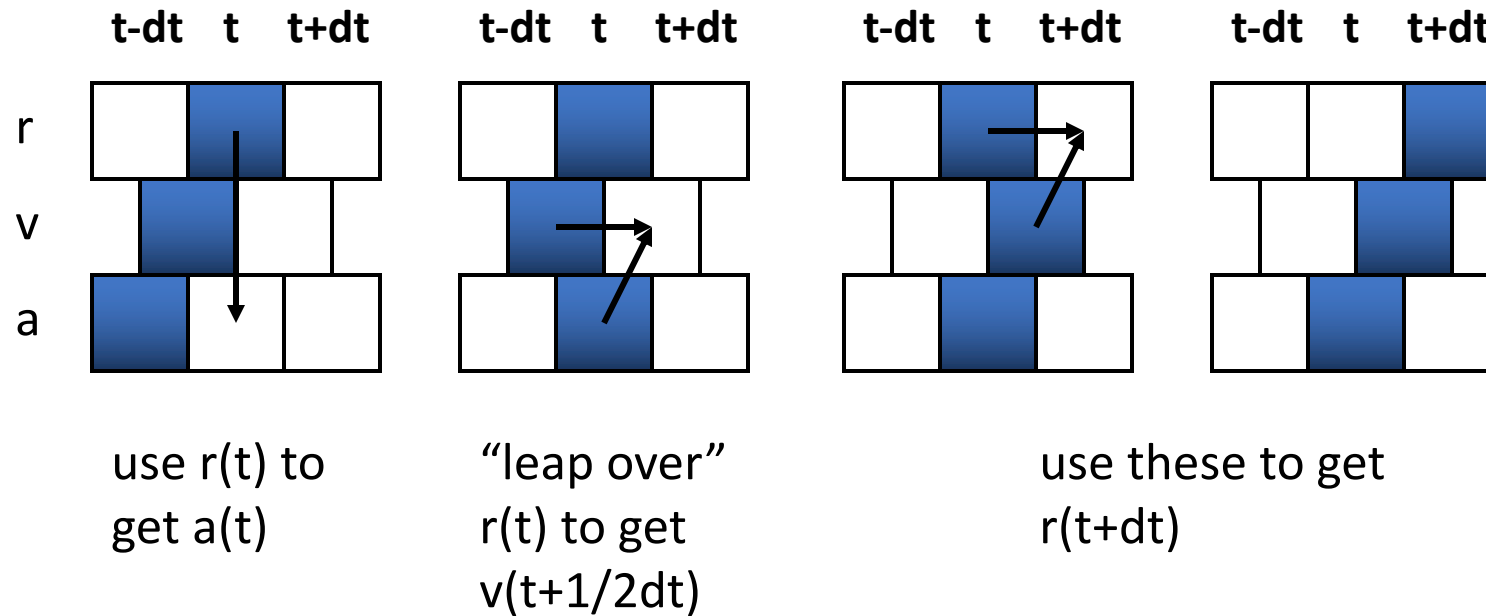
- Apply 2nd equation to allow v to “leap over” r by $1/2$ time step.
- Compute current velocities

$$v(t) = \frac{1}{2} \left[v \left(t + \frac{1}{2} \Delta t \right) + v \left(t - \frac{1}{2} \Delta t \right) \right]$$

- Use $v \left(t + \frac{1}{2} \Delta t \right)$ to allow r to “catch up”

Verlet “Leap Frog”

- Graphical representation



Velocity Verlet algorithm

- Leap frog formally equivalent to Verlet
- Does not have instability problem
- Velocities still handled in clumsy manner
- Perhaps the “best” algorithm: “Velocity Verlet” (*J. Chem. Phys.*, **76**, 637 (1982))

$$r(t + \Delta t) = r(t) + \Delta t v(t) + \frac{1}{2} \Delta t^2 a(t)$$

$$v(t + \Delta t) = v(t) + \frac{1}{2} \Delta t [a(t) + a(t + \Delta t)]$$

Velocity Verlet

- Implementation

$$1 \quad r(t + \Delta t) = r(t) + \Delta t v(t) + \frac{1}{2} \Delta t^2 a(t)$$

$$2 \quad v(t + 1/2 \Delta t) = v(t) + \frac{1}{2} \Delta t [a(t)]$$

$$3 \quad a(t + \Delta t) = -\frac{\nabla U[r(t + \Delta t)]}{m}$$

$$4 \quad v(t + \Delta t) = v\left(t + \frac{1}{2} \Delta t\right) + \frac{1}{2} \Delta t [a(t + \Delta t)]$$

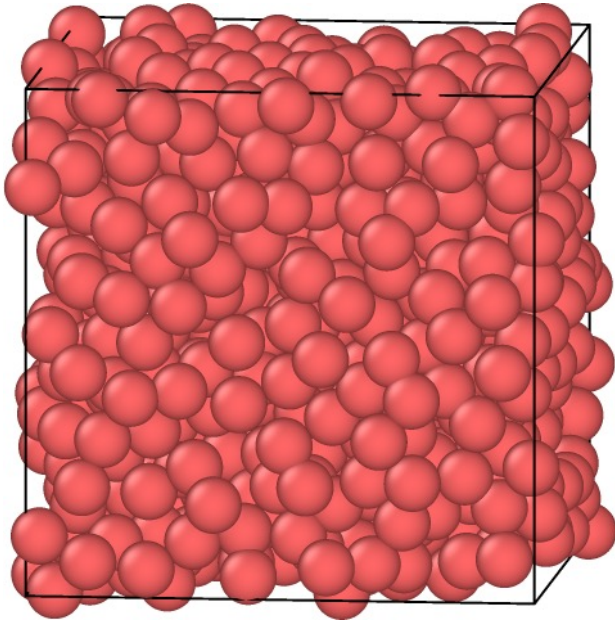
LAMMPS

run_style style args

• style = *verlet* or *verlet/split* or *respa* or *respa/omp*

“verlet” = velocity Verlet

Creating LJ molecules with LAMMPS



Reduced LJ density $N/V^* = 0.7$
N=864 molecules
 $L^* = 10.73$

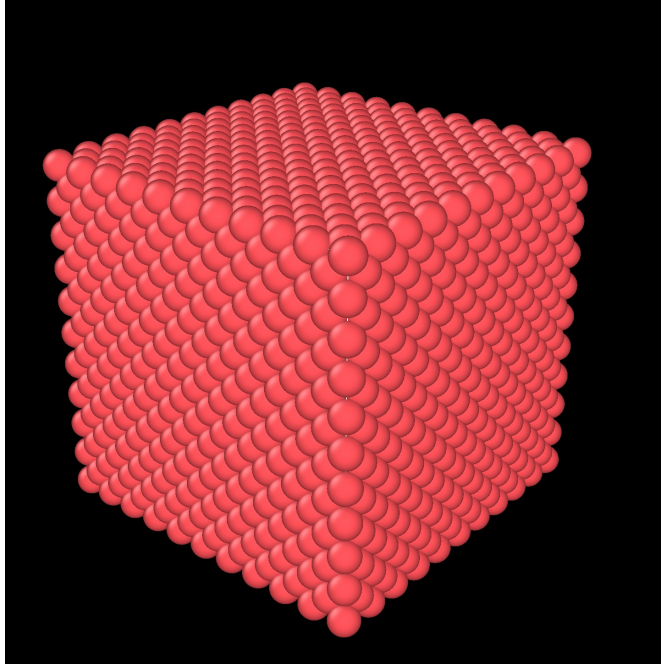
```
lattice fcc 0.70 # style scale(=reduced_density)
#lattice fcc 0.8442 # style scale(=reduced_density)
```

```
# Region in LJ units: 0 6.0 0 6.0 0 6.0 864 molecules
region MYBOX block 0 6.0 0 6.0 0 6.0
create_box 1 MYBOX # number_of_atom_types region_ID
```

```
create_atoms 1 box # atom_type style args
```

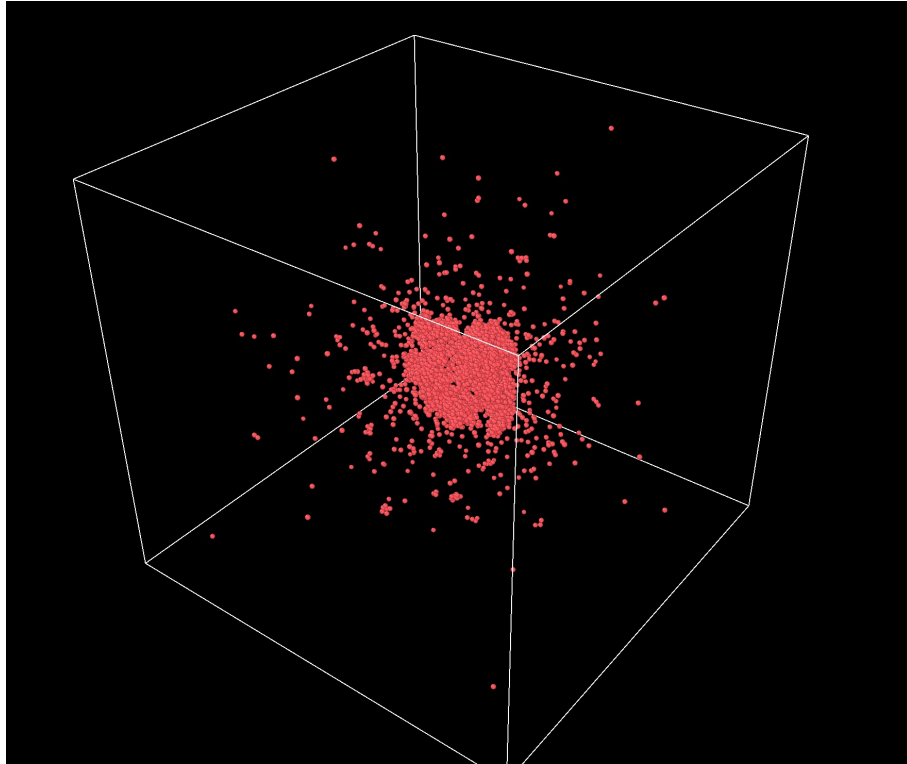
Property	Conversion
density	$\rho^* = \rho\sigma^3$
temperature	$T^* = k_B T / \epsilon$
energy	$E^* = E / \epsilon$
pressure	$P^* = P\sigma^3 / \epsilon$
time	$t^* = (\epsilon / m\sigma^3)^{1/2} t$

Run Example 2



- fcc lattice: Common initial starting configuration (get high density)
- System needs to “melt” to get liquid
- Must assign random velocities to start

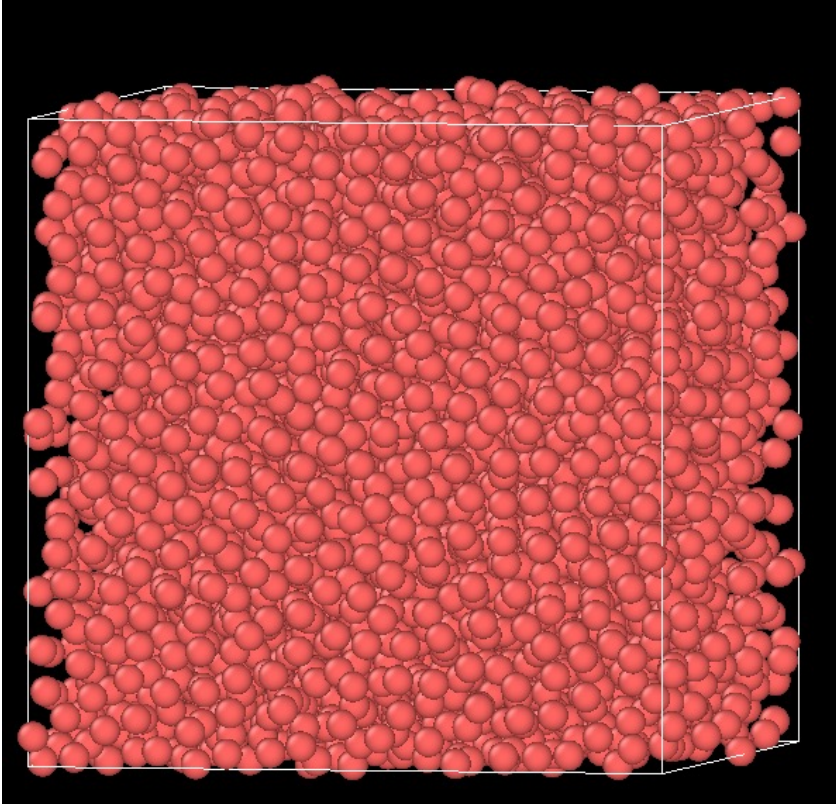
What happened?



- Atoms start at a state point that wants to be a liquid
- As atoms melt, they are exposed to a vacuum
- “Droplet” vaporizes
- V and E not constant – not microcanonical ensemble

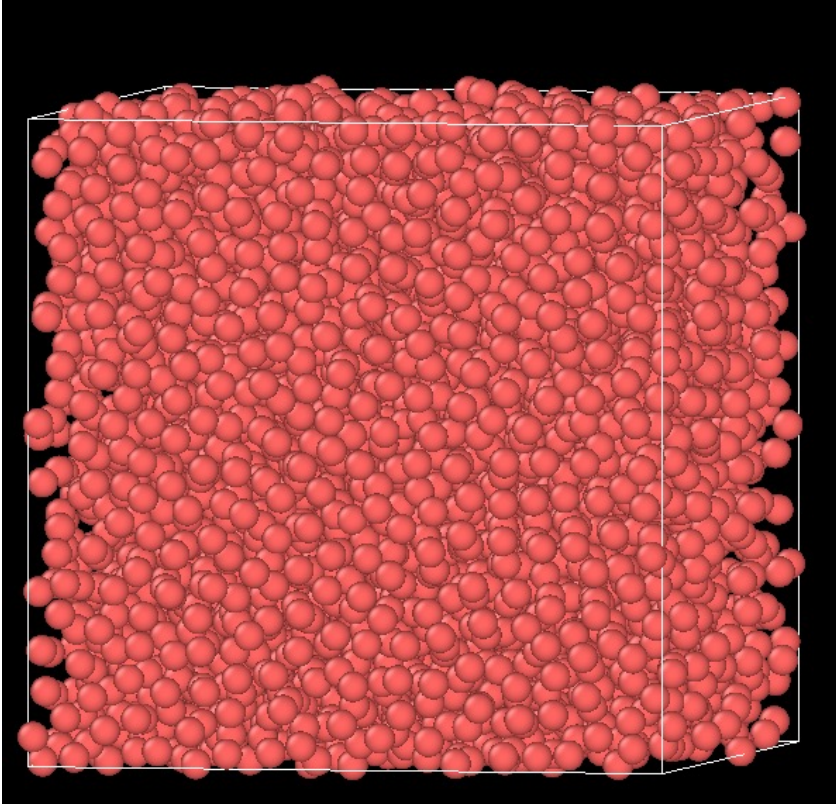
Step	Temp	Press	PotEng	KinEng	TotEng
0	3	-3.4845053	-6.0761475	4.4990283	-1.5771192
50	1.7748417	5.3570512	-4.243752	2.6616876	-1.5820644
100	1.731976	3.4268938	-4.1649181	2.5974029	-1.5675152
150	1.6790648	2.1594135	-4.0673572	2.5180534	-1.5493038
200	1.6041826	1.2516878	-3.9366626	2.4057542	-1.5309084
250	1.5091256	0.76269251	-3.7770188	2.2631995	-1.5138192
300	1.4432827	0.34099098	-3.6624767	2.1644565	-1.4980202
350	1.3603465	0.1378712	-3.5230795	2.0400792	-1.4830004
400	1.270896	0.030005588	-3.3755461	1.9059324	-1.4696137
450	1.1910348	-0.039600614	-3.2434006	1.7861664	-1.4572342
500	1.1328023	-0.090115967	-3.145119	1.6988366	-1.4462824
550	1.0687906	-0.098623743	-3.0400953	1.6028397	-1.4372555
600	1.0239435	-0.11814921	-2.9647217	1.5355836	-1.429138
650	0.97483568	-0.10659398	-2.8838015	1.4619378	-1.4218638
700	0.94299048	-0.094648988	-2.8297054	1.4141803	-1.4155252
750	0.91212726	-0.072597343	-2.7784189	1.3678955	-1.4105234
800	0.89796418	-0.063266876	-2.7533238	1.3466554	-1.4066684
850	0.88730808	-0.062104992	-2.7337885	1.3306747	-1.4031138
900	0.86641968	-0.044796873	-2.6992878	1.2993489	-1.3999389
950	0.86587028	-0.047253967	-2.6958261	1.298525	-1.3973011
1000	0.84943225	-0.0278741	-2.6692931	1.2738732	-1.3954199

Add a boundary to box



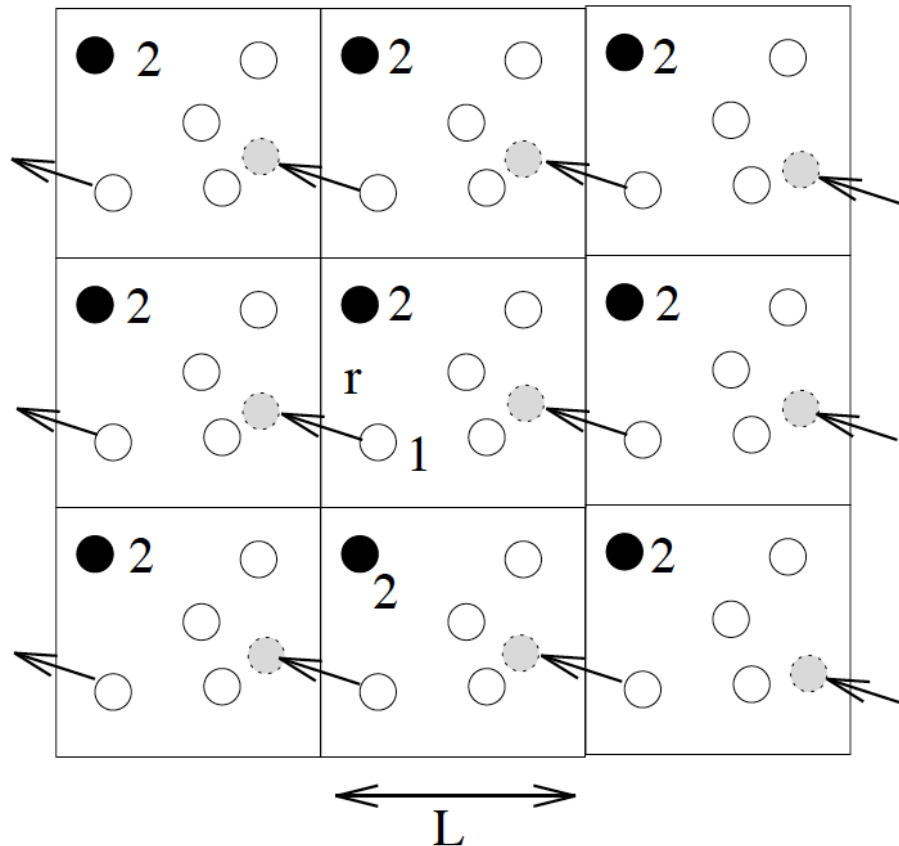
- To keep molecules inside the box, we could add a reflecting boundary or “walls”
- Keep simulation at NVE
- Do you see any problems with this?

Add a boundary to box



- To keep molecules inside the box, we could add a reflecting boundary or “walls”
- Keep simulation at NVE
- Problem?
- Simulations are ~ 1000 molecules
- Huge fraction are on surface – not representative of “bulk” behavior

Periodic boundary conditions



- Surround central box with 26 identical images
- Only track molecules in central box, but energies calculated with image particles
- $R_{\text{cut}} < \frac{1}{2} \text{ box}$

boundary p p p # x y z (p = periodic boundary conditions)

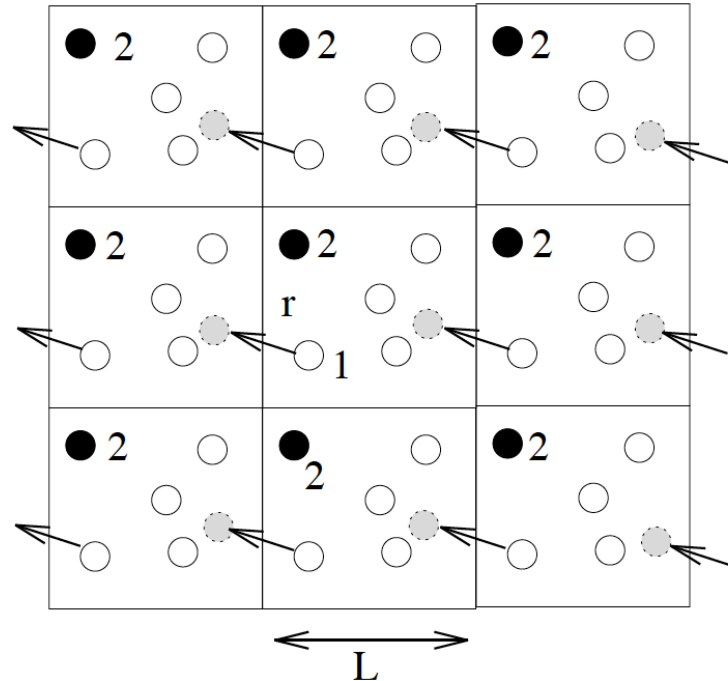
#boundary s s s # x y z (s = shrink wrapped boundary conditions)

Conserved quantities in NVE MD?

1. Hamiltonian (energy, $H = U + K$)

Conserved quantities in NVE MD?

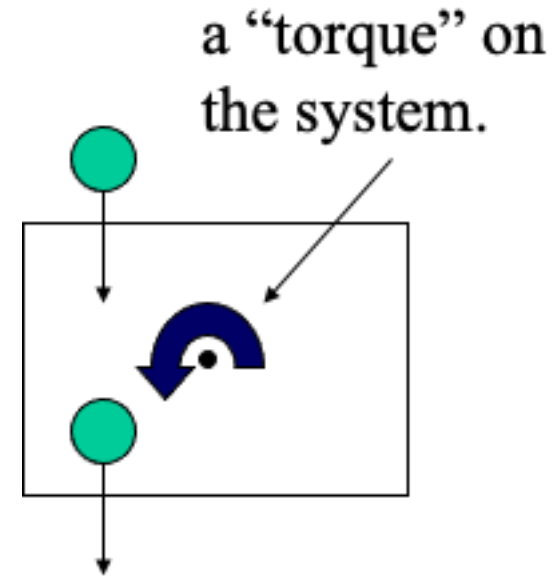
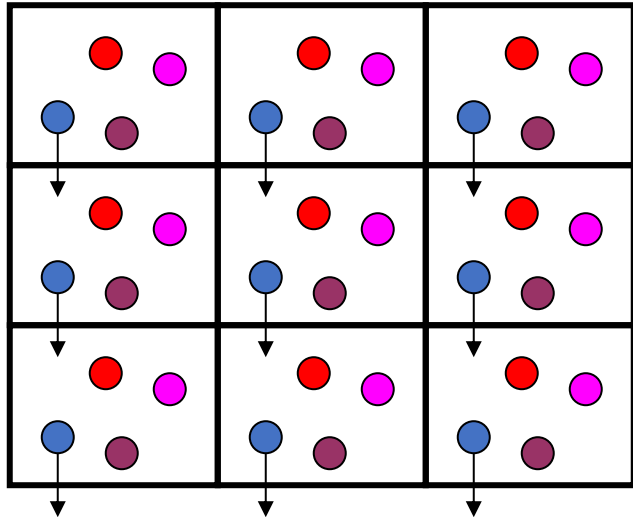
1. Hamiltonian (energy, $H = U + K$)
2. Linear momentum



Conserved quantities in NVE MD?

1. Hamiltonian (energy, $H = U + K$)
2. Linear momentum
3. NOT angular momentum

4 constraints – need to account for when using the number of degrees of freedom (more on this later)



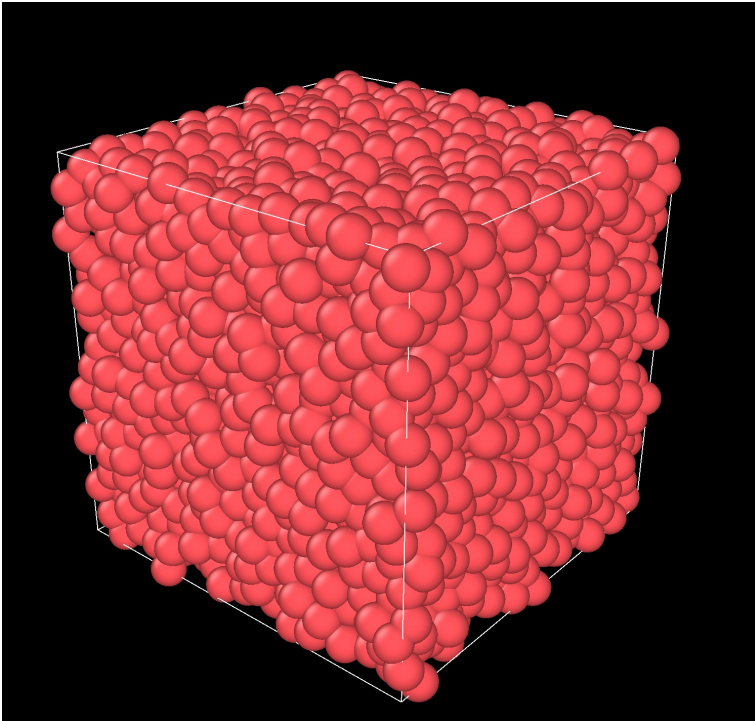
Run example 3

boundary p p p # x y z (p = periodic boundary conditions)

Implements PBC in all 3 dimensions

- What happens to NVE?
- Visualize the trajectory
 - What happens to molecules near the edges of the box?

Example 3 with PBC



Step	Temp	Press	PotEng	KinEng	TotEng
0	3	-3.7033504	-6.7733681	4.498875	-2.2744931
50	1.6762688	5.6211062	-4.7966884	2.5137746	-2.2829138
100	1.6751365	5.6745042	-4.7930126	2.5120766	-2.280936
150	1.6503181	5.8247966	-4.754385	2.4748583	-2.2795267
200	1.6439636	5.8864021	-4.7455259	2.4653289	-2.280197
250	1.6332109	5.9623581	-4.7284008	2.4492038	-2.2791969
300	1.6524852	5.8259333	-4.7570534	2.4781081	-2.2789453
350	1.6547879	5.8178479	-4.7606833	2.4815613	-2.279122
400	1.6290272	5.958913	-4.7225629	2.4429299	-2.279633
450	1.657992	5.762083	-4.7659903	2.4863663	-2.2796241
500	1.6440357	5.8575351	-4.7459147	2.4654371	-2.2804776
550	1.6348257	5.9405456	-4.7307789	2.4516255	-2.2791534
600	1.6406392	5.9643588	-4.7393391	2.4603436	-2.2789955
650	1.6450391	5.8591064	-4.7459475	2.4669417	-2.2790058
700	1.6544808	5.8686647	-4.7598262	2.4811007	-2.2787254
750	1.6355088	5.9352799	-4.7324059	2.4526499	-2.279756
800	1.6426811	5.9005846	-4.7431883	2.4634056	-2.2797827

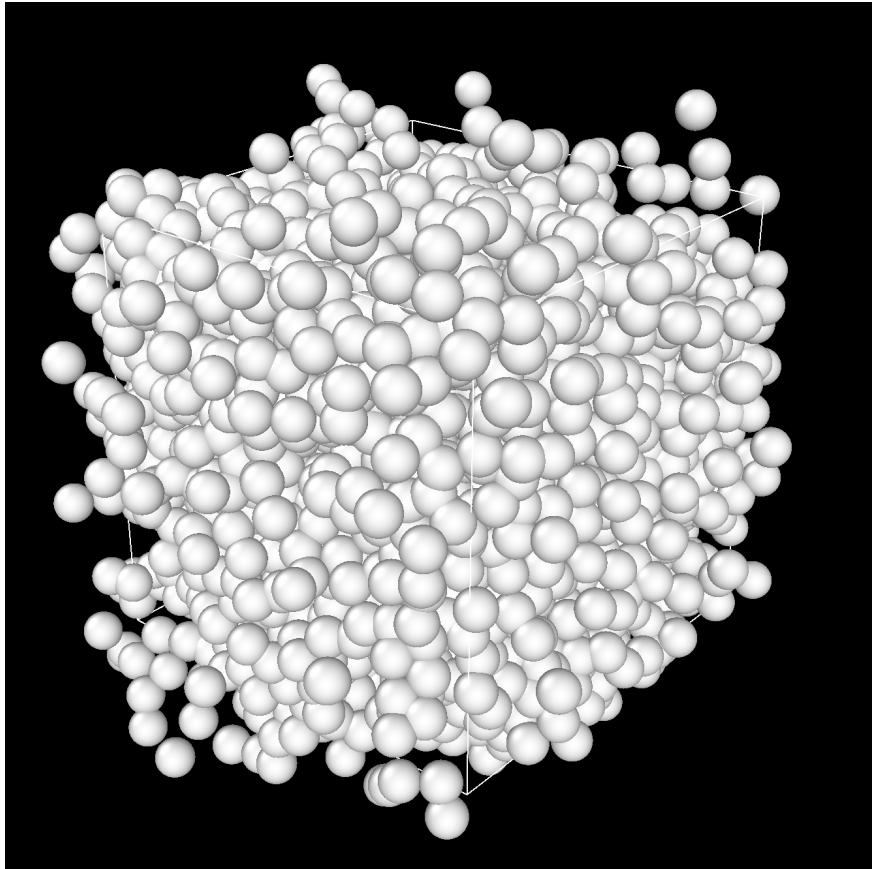
- Total energy constant (as is volume): NVE!
- Why does the potential energy go up?
- Why does the kinetic energy go down?

Run Example 4

```
# dump_ID group_ID style interval file args  
dump DUMP all custom 50 ${NAME}.lammprj xu yu zu
```

- PBC is applied to all atoms but when we write the trajectory, we don't apply PBC to them
 - Useful when computing diffusivity
- Visualize trajectory
- Compare this and thermo properties with Example 3

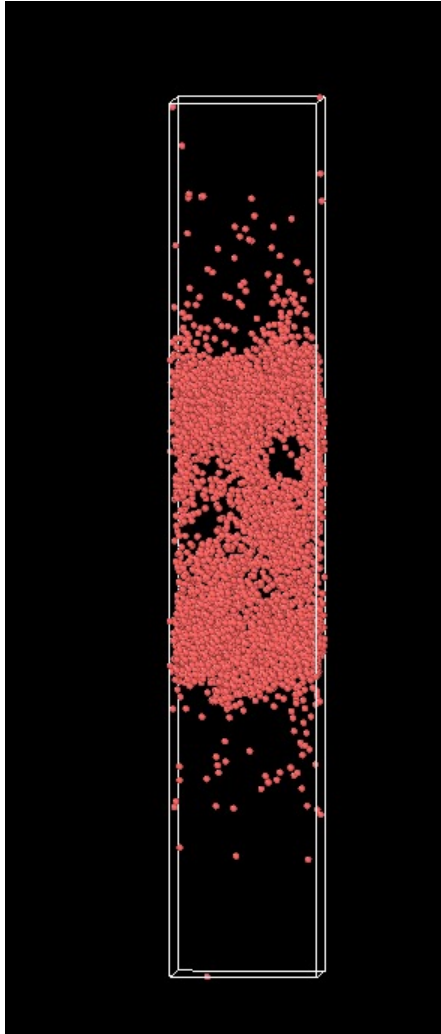
Example 4 – PBC and unwrapped coords



Step	Temp	Press	PotEng	KinEng	TotEng
0	3	-3.7033504	-6.7733681	4.498875	-2.2744931
50	1.6762688	5.6211062	-4.7966884	2.5137746	-2.2829138
100	1.6751365	5.6745042	-4.7930126	2.5120766	-2.280936
150	1.6503181	5.8247966	-4.754385	2.4748583	-2.2795267
200	1.6439636	5.8864021	-4.7455259	2.4653289	-2.280197
250	1.6332109	5.9623581	-4.7284008	2.4492038	-2.2791969
300	1.6524852	5.8259333	-4.7570534	2.4781081	-2.2789453
350	1.6547879	5.8178479	-4.7606833	2.4815613	-2.279122
400	1.6290272	5.958913	-4.7225629	2.4429299	-2.279633
450	1.657992	5.762083	-4.7659903	2.4863663	-2.2796241
500	1.6440357	5.8575351	-4.7459147	2.4654371	-2.2804776
550	1.6348257	5.9405456	-4.7307789	2.4516255	-2.2791534
600	1.6406392	5.9643588	-4.7393391	2.4603436	-2.2789955
650	1.6450391	5.8591064	-4.7459475	2.4669417	-2.2790058
700	1.6544808	5.8686647	-4.7598262	2.4811007	-2.2787254
750	1.6355088	5.9352799	-4.7324059	2.4526499	-2.279756
800	1.6426811	5.9005846	-4.7431883	2.4634056	-2.2797827

- Exactly the same as Example 3!
- Atom coordinates are “unwrapped” and allow for diffusion determination

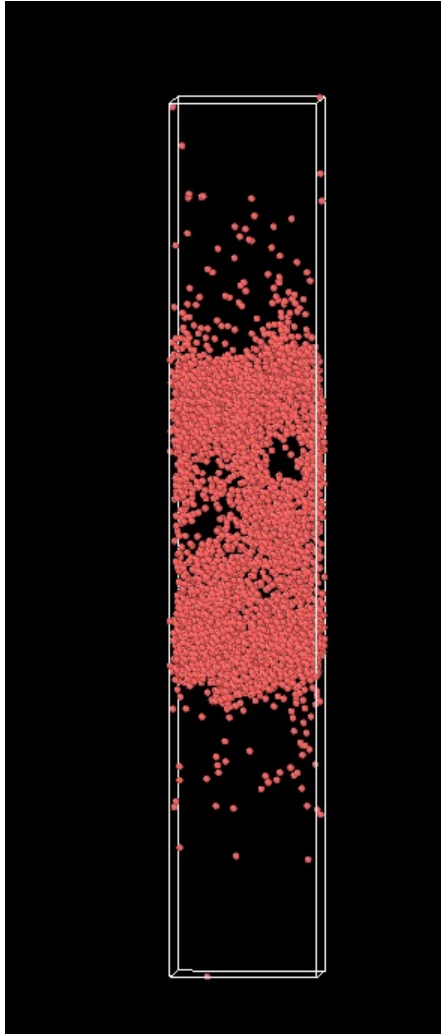
Apply PBC in 2 directions



Simulate a vacuum interface in Z; periodic liquid in X, Y

Edit example3.inp to run this simulation.

Apply PBC in 2 directions

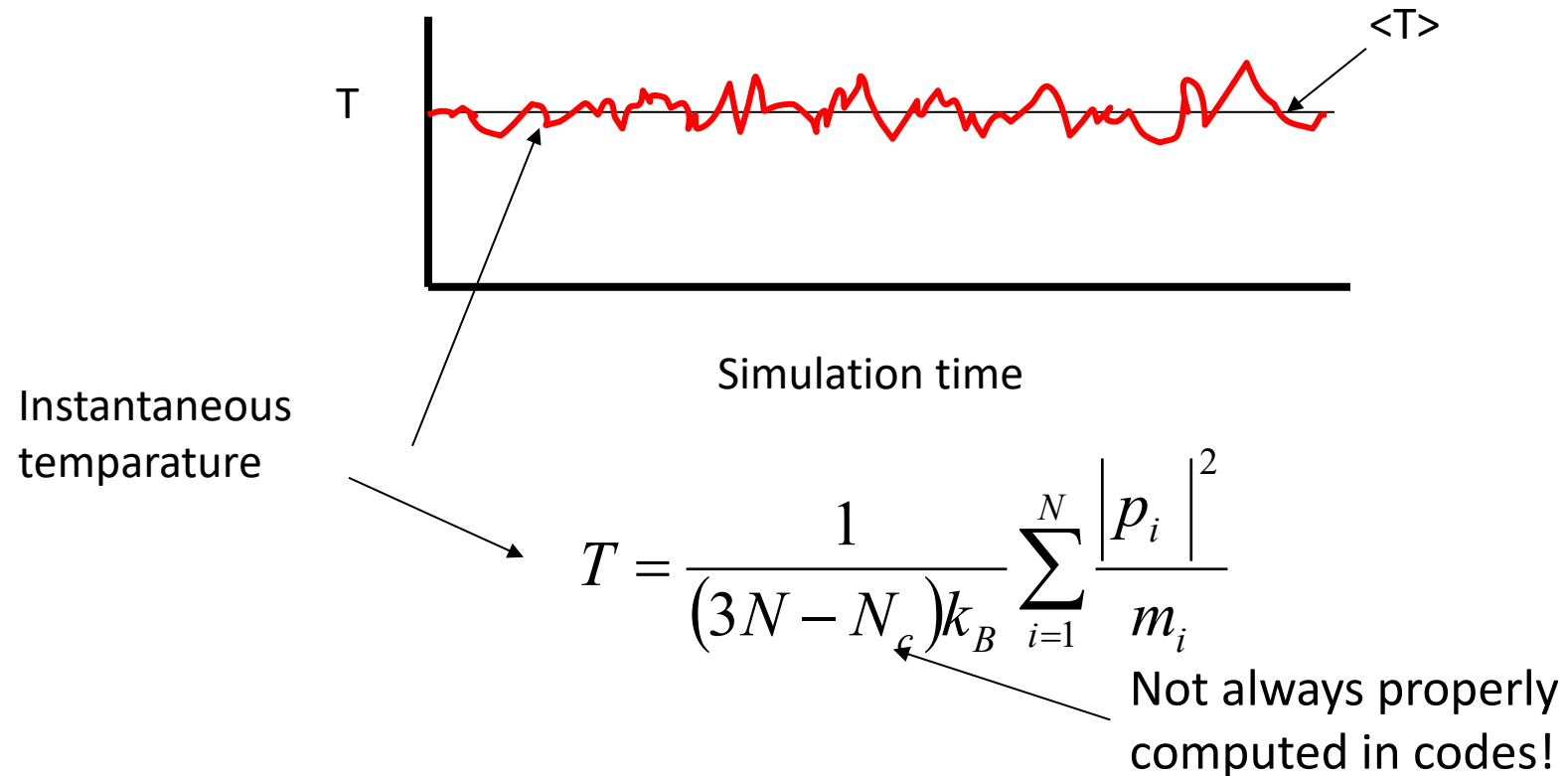


Simulate a vacuum interface in Z; periodic liquid in X, Y

```
boundary p p s # x y z (p in x, y s in z)
```

Problems With NVE MD

- Typical MD is in NVE ensemble.
- How do you compare with experiments at a given T?



Different ensembles

The NVE ensemble is the “natural” ensemble of molecular dynamics since the Newtonian equations of motion keep energy constant.

Simulate at NVE => compute what T, P are...

BUT, most experiments are at constant T and/or P. What to do?

Canonical (NVT) ensemble

$$Q(NVT) = \frac{1}{\Lambda^{3N} N!} \int d\mathbf{q}^N \exp[-\beta U(q^N)]$$

Isothermal-isobaric (NPT) ensemble

$$Q(NPT) = \frac{1}{V_0} \int dV \exp[-\beta PV] Q(NVT)$$

Simulating in these ensembles requires new equations of motion;
“thermostat” for NVT; “thermostat” and “barostat” for NPT

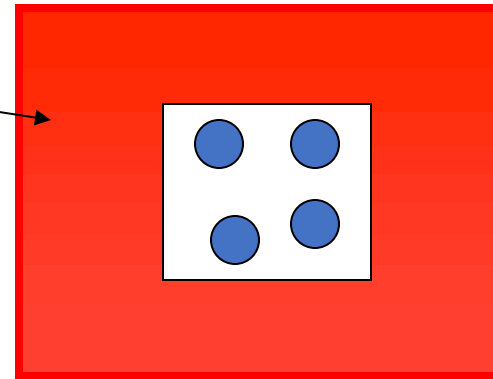
MD in Other Ensembles

- It is desirable to *set* T at start
- Perform MD in canonical (NVT) ensemble
- Can we use classical equations of motion to do this?
- No!
- A (very) crude approach: rescale v_i at each time step so

$$T = \frac{1}{(3N - N_c)k_B} \sum_{i=1}^N \frac{|p_i|^2}{m_i} = T_{set}$$

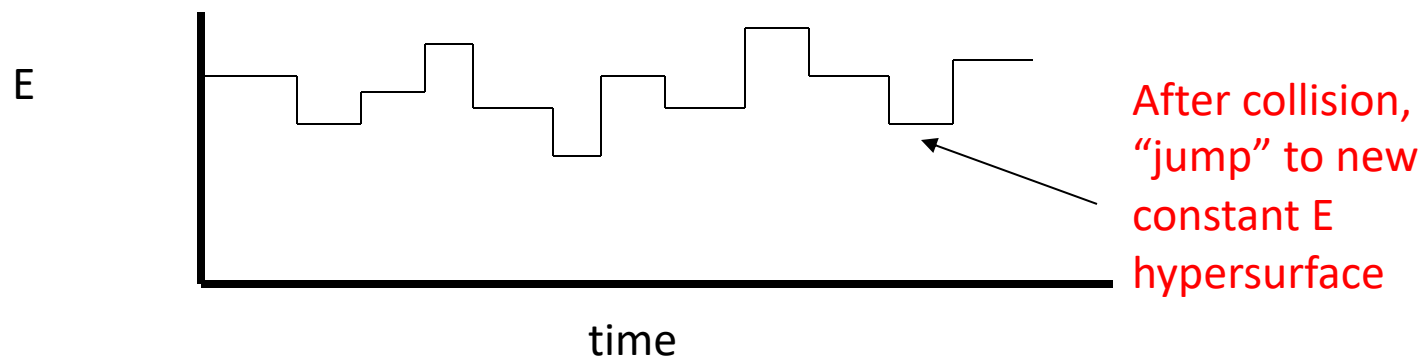
NVT MD

- Thermal reservoir at setpoint T
- At some frequency, allow a particle to “collide” with a heat bath particle.
- Reassign particle velocity from Maxwell-Boltzmann distribution at T .



NVT MD

- Phase space trajectory looks like this



- Over time, all constant E hypersurfaces sampled about T_{set}
- $\langle T \rangle \sim T_{\text{set}}$

NVT MD

- “Bath particle collision” method does yield averages consistent with canonical ensemble
- Dynamic properties may be altered however!
- This is a “zeroth order” method.
- See: H. C. Anderson, “Molecular Dynamics at Constant Pressure and/or Temperature”, *J. Chem. Phys.*, **72**, 2384 (1980).

Extended Systems

- A more sophisticated approach: “extended system” methods.
- Idea: simulate an extra degree of freedom representative of a thermal reservoir.
- Coordinate has “velocity” (rate of change)
- Coordinate has “mass” (inertia)
- Common method: Nosé-Hoover
 - S. Nosé, “A Molecular Dynamics Method for Simulations in the Canonical Ensemble”, *Mol. Phys.*, **53**, 255 (1984).
 - W. G. Hoover, “Canonical Dynamics: Equilibrium Phase-Space Distributions”, *Phys. Rev. A*, **31**, 1695 (1985).

Extended Systems

- The Nosé-Hoover “thermostat” appears naturally in the equations of motion
- A “thermal inertia” term governs rate of heat transfer and thus KE fluctuations.
- A “time constant” of $\sim 1\text{ps}$ typically used ($\sim 1000\text{ dt}$)
- Extended system methods generalized to maintain P in NPH MD
 - see *J. Chem. Phys.*, **72**, 2384 (1980).

LAMMPS *fix nvt* command

LAMMPS implements “fixes” that are any operation that is applied to the system during time stepping or minimization.

```
fix ID group-ID style_name keyword value ...  
style_name = nvt or npt or nph
```

```
fix 1 all nve                                # microcanonical for all atoms  
fix 3 all nvt temp 300.0 300.0 10.0         # canonical for all atoms  
                                              Start 300; stop 300; damping  
                                              constant 10.0  
Tdamp = temperature damping parameter  
(time units)
```

NVT simulations

- In LAMMPS, fix NVT command performs time integration on Nosé-Hoover style non-Hamiltonian equations of motion.
- The desired temperature at each timestep is a ramped value during the run from T_{start} to T_{stop} .
- T_{damp} is in time units and determines how rapidly the temperature is relaxed.
 - A value of 10.0 means to relax the temperature in a timespan of (roughly) 10 time units

Tdamp

From user documentation

- A Nosé-Hoover thermostat will not work well for arbitrary values of *Tdamp*.
- If *Tdamp* is too small, the temperature can fluctuate wildly; if it is too large, the temperature will take a very long time to equilibrate.
- A good choice for many models is a *Tdamp* of around 100 timesteps. Note that this is NOT the same as 100 time units.
- A simple way to ensure this, is via using an [immediate variable](#) expression accessing the thermo property 'dt', which is the length of the time step. Example:

```
fix 1 all nvt temp 300.0 300.0 $(100.0*dt)
```

Run Example 5: NVT

```
fix TEMP all nvt temp 1.7 1.7 0.3 # T1 T2 Tdamp
```

Step	Density	Temp	Press	PotEng	KinEng	TotEng
0	0.8442	3	-4.4654851	-7.2253807	4.498875	-2.7265057
5	0.8442	2.9479538	-4.0538529	-7.156045	4.4208252	-2.7352198
10	0.8442	2.7447312	-2.6017674	-6.8726962	4.1160675	-2.7566287
15	0.8442	2.2781444	0.33586859	-6.2053195	3.4163624	-2.7889572
20	0.8442	1.6831137	3.8926871	-5.3358811	2.5240394	-2.8118417
25	0.8442	1.4624299	5.387245	-5.0195891	2.1930964	-2.8264927
30	0.8442	1.5510769	5.0876634	-5.161707	2.3260336	-2.8356733
35	0.8442	1.6448668	4.6122982	-5.3070914	2.4666834	-2.840408
40	0.8442	1.6626398	4.4833398	-5.3367398	2.4933363	-2.8434036
45	0.8442	1.6446739	4.5495101	-5.3120582	2.466394	-2.8456641
50	0.8442	1.6294816	4.6214826	-5.2910339	2.4436113	-2.8474226
...						
950	0.8442	1.7036442	5.4694596	-5.1337292	2.5548274	-2.5789017
955	0.8442	1.7143303	5.4107694	-5.1507754	2.5708526	-2.5799228
960	0.8442	1.7223772	5.3660442	-5.1640457	2.5829199	-2.5811259
965	0.8442	1.7114629	5.4179554	-5.1482196	2.5665526	-2.581667
970	0.8442	1.7022124	5.4606582	-5.1359445	2.5526803	-2.5832642
975	0.8442	1.7034176	5.4516946	-5.1390778	2.5544876	-2.5845902
980	0.8442	1.7009249	5.4585891	-5.1366219	2.5507495	-2.5858723
985	0.8442	1.6969487	5.4686266	-5.131996	2.5447867	-2.5872092
990	0.8442	1.6991331	5.4419508	-5.1367317	2.5480624	-2.5886693
995	0.8442	1.7078689	5.3786159	-5.1514502	2.5611628	-2.5902874
1000	0.8442	1.724012	5.2839742	-5.1771125	2.5853715	-2.591741

**T slowly approaches
setpoint**

KE goes down

**Total energy is NOT
conserved**

What is the actual temperature?

$$T_{inst} = \frac{1}{(3N - N_c)k_B} \sum_{i=1}^N N \frac{|p_i|^2}{m_i} \quad \text{Instantaneous temperature}$$

$$T = T_{set}$$

We should obtain $\langle T_{inst} \rangle = T_{set}$

- We should not take averages over parts of the trajectory that have not yet reached the desired statepoint
- “Equilibration” phase is thrown away
- “Production” phase is recorded.
- Use “restart” feature to start from the end of an equilibrated trajectory

Run Example 6 – restart from end of Example 5

If you are unhappy with how well equilibrated Example 5 is, re-run it for a longer time.

Examine the KE, PE, E, T, and P

Example 7 – Play with different options

Other ways to run NVT

Play with potential cutoff method, etc.

NPT simulations

LAMMPS uses an extended system Nosé-Hoover style

```
fix TEMP all npt temp 1.7 1.7 0.3 iso 5.0 5.0 1000.0 #
```

iso values = Pstart Pstop Pdamp; scalar external pressure at start/end of run (pressure units); Pdamp = pressure damping parameter (time units)

- A Nosé-Hoover barostat will not work well for arbitrary values of *Pdamp*.
- If *Pdamp* is too small, the pressure and volume can fluctuate wildly; if it is too large, the pressure will take a very long time to equilibrate.
- A good choice for many models is a *Pdamp* of around 1000 timesteps.

Run Example 8

Run basic case and look at how T, E, P vary

Adjust the barostat (or thermostat) time constants and see what happens

```
# configure integrator and thermostats/barostats  
fix TEMP all npt temp 1.7 1.7 0.3 iso 5.0 5.0 1000.0  
timestep 0.003 # The default timestep size is 0.005 for LJ units.
```

Look at the T, P, and density.