

# An Introduction to Molecular Dynamics

Prof. Michael Shirts

University of Colorado Boulder

7th i-CoMSE Workshop: Molecular Dynamics

Boise State University, July 8, 2024

Licence: CC-BY 4.0



# Why do molecular modeling?

- Why do we bother doing molecular modeling?
- Much of engineering is about modeling a system well enough to make predictions about how to design and control the system.
- So why not model at a **molecular** level?

# What can we get out of molecular simulations?

- Thermodynamics
  - Ensemble averages of configurations
  - Free energy related quantities
  - Distributions of molecular configurations
- Kinetics/transport
  - Dependent on the paths that molecules take
  - Rates of transfer between configurations
  - Diffusion, viscosity, protein conformations, polymer aging

# Molecular dynamics simulations are classical approximations to the QM equations

- Closest to truth: relativistic, time-dependent, Schrödinger equation
  - Assumption #1: Almost always with molecules, you can ignore relativity
  - Even then, anything more complex than a few atoms cannot be handled at this level of theory
- Assumption #2: We can assume that the nuclei move like classical point masses (Born-Oppenheimer approximation)
  - Occasionally (for proton transport) this has some effects.
- We can then make many, many more numerical approximations that make computing the electron energies faster
- This allows us to do *Ab Initio* Molecular Dynamics (AIMD)
  - Use some approximation for Schrodinger's equations to solve for the electronic wavefunction
  - Then use Coulomb's law to describe how it the electron forces move nuclei around
- STILL VERY, VERY SLOW!!

# Molecular dynamics simulations are classical approximations to the QM equations

- But what if we just give up on modeling the electrons, and instead **average out the electronic degrees of freedom**?
- A type of coarse-graining, where we just model the nuclei and electrons **lumped together**
- Approximate the interactions of the electrons and nuclei **together**, calculating an effective potential between the nuclei
- And model the physics of **this** system?
- This is **classical molecular dynamics**.

# Classical MD simulation

- Cons:
  - As we average out electron behavior more and more, we lose accuracy
  - Usually, have to add in bonds making/breaking by hand.
  - Interactions like pi-cations hard to model with simple potentials.
  - No understanding of interaction of light with models.
- Pros:
  - Orders of magnitude faster  $10^3$ - $10^9$
  - Actually more accurate for many problems in condensed phase solutions
    - Converging QM to high levels of accuracy is HARD, sometimes you can do better by picking an approximate interaction term.
  - It's a model - you use **the model that works for your questions.**

# A classical system follows Newton's equations of motion

$$m\vec{a}_i(t) = \vec{F}_i(\vec{x}(t)) = -\nabla U(\vec{x}_i) = -\left[\frac{\partial U}{\partial x_i}, \frac{\partial U}{\partial y_i}, \frac{\partial U}{\partial z_i}\right]$$
$$\frac{\partial^2 x_i}{\partial t^2} = -\frac{1}{m} \nabla U(\vec{x})$$

- This is system of initial value partial differential equations
- General strategy:
  - Pick an energy function  $U(x)$ 
    - **IMPORTANT CHOICE!**
  - Specify the positions and velocities at time  $t$ .
  - How we get "good" initial positions and velocities?
    - **IMPORTANT CHOICE!**
  - Approximate the positions and velocities at time  $t+\Delta t$  using the differential equation

# Go to the notebook!

- Simple\_MD.ipynb
- We will discuss:
  - Taylor series solution
  - Verlet and higher order solutions
  - Energy drift
  - Shadow Hamiltonians
  - Stiffness



# What sort of ensemble does MD generate?

- MD simulates the NVE ensemble (it conserves energy!)
- With a bit of tweaking, MD can also allow us to capture canonical (NVT) and isobaric-isothermal (NPT) ensembles

# What comes out of an MD simulation?

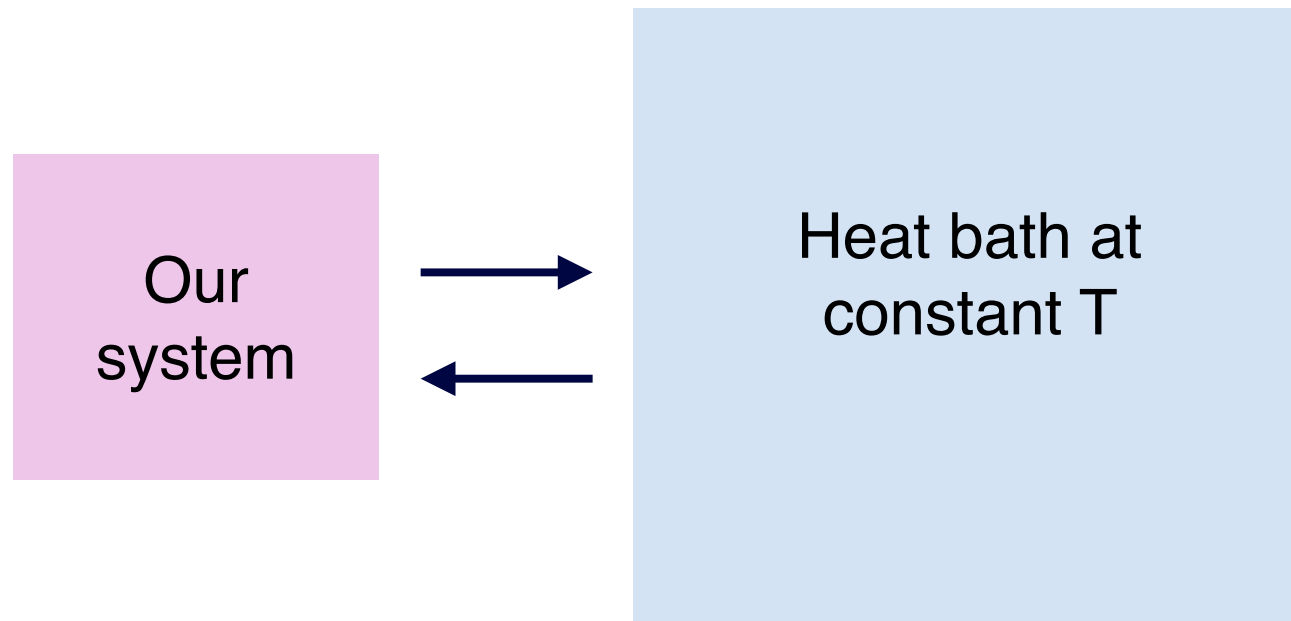
- Time series of the energies and forces (+ other derived quantities like pressure, etc).
- Time series of the coordinates/velocities
- This is sufficient to estimate **pretty much any corresponding experimental observable**.
  - What are some of these observables?
- The **key problem of MD analysis** is how to **extract data of interest** when you have  $3N \times$  number of frames data points!

# Why do we need thermostats for molecular dynamics?

- Molecular dynamics gives:
  - Solutions to the initial value partial differential equations defined by Newton's equation of motion
- Newton's laws of motion conserve **energy**
- Therefore this gives ensembles
  - constant number of particles
  - constant volume
  - **constant energy**
- But we want answers at, say 300 K?

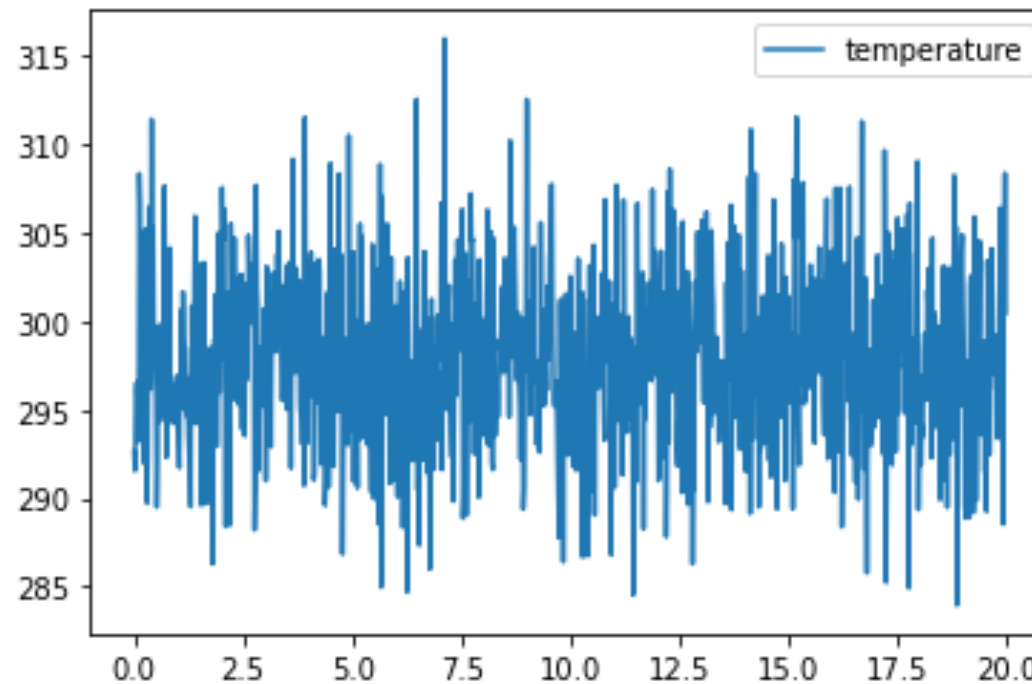
# We want to simulate properties at constant T

What does it mean to be a constant T ensemble?



Exchange thermal energy  
until at equilibrium

# What does it mean if you see a graph like this from a simulation?



- This is actually a graph of  $\frac{2KE}{N_{DOF}k_B}$
- $T = \frac{2\langle KE \rangle}{N_{DOF}k_B}$ , but T is constant in an NVT simulation.

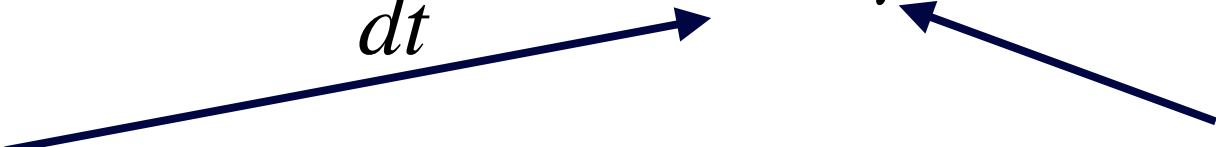
# How does REAL LIFE thermal equilibration happen?

- Our system is surrounded by other molecules
- They bang into our system, and exchange kinetic energy
  - If they are hotter than the system average, then energy goes into the system, and it heats up
  - If they are cooler than the system average, then energy goes out of the system, and it cools down
  - Until it is in thermal equilibrium with the surroundings

# A very simple thermostat (Andersen)

- "Thermo" = temperature, "stat" - keeps the same
- With some probability  $p$  (or frequency  $N$ ), our particles collide
  - (With what? Some imaginary particle from the multiverse)
  - And we randomize their velocity to the Maxwell-Boltzmann distribution
- This is the Andersen thermostat
- In GROMACS, `tcoupl = anderson-massive`
- Need to set `ref_t` as well!
- Go to the notebook `Example_NVT`!

# Langevin dynamics

$$m \frac{dv}{dt} = F + R - \gamma v$$


- Add thermal energy by adding random noise
  - Like a particle colliding!
  - But, all of the particles are colliding, even ones inside your system...
- Suck excess thermal energy away with a friction term
  - Like moving through a fluid, which doesn't not necessarily make sense; the simulation IS the fluid . . .
- Balance the friction loss and random noise to be equal when you are at the right T  $\langle R \rangle = 0 \quad \langle R^2 \rangle = 2k_B T \gamma$ 
  - Implemented as `integrator = sd` in gromacs.
  - Very good stability!
  - If you add too much random noise, and suck out too much velocity, it **slows the dynamics**.



# How else could we get thermal energy into and out of the system?

- We modify the kinetic energies of the particles
- First guess:
  - If the average thermal energy is too **high**, decrease the velocity of **all** particles
  - If the average thermal energy is too **low**, increase the velocity of **all** particles
- This is the weak coupling thermostat
- In GROMACS, `tcoupl = berendsen`
- Need to set `ref_t` as well!

# How quickly do we pump energy in or out?

- Let's say the system has 110 kJ mol. At the temperature you specified, it should have 100 kJ/mol on average
  - Do you take out 10 kJ/mol each step?
  - Or do you take out 1 kJ/mol each step?
- We need to define a **rate** at which excess kinetic energy is removed

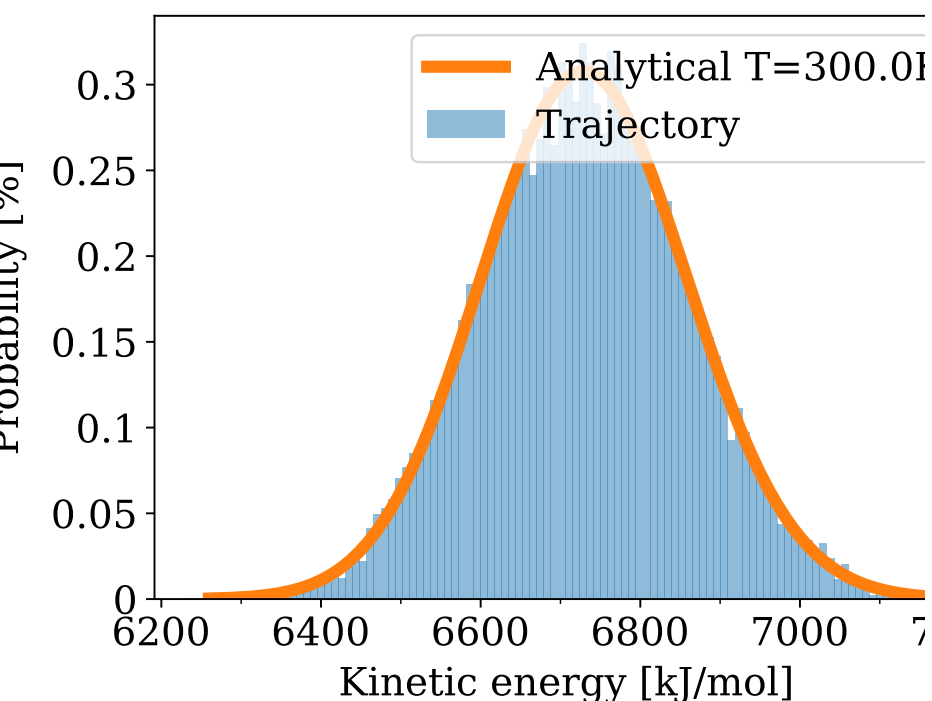
$$\frac{dKE}{dt} = k(KE_0 - KE)$$

- $k$  has units of  $1/\text{time} = 1/\tau$
- In GROMACS, this is `tau_t`.

# There's a problem with Berendsen!

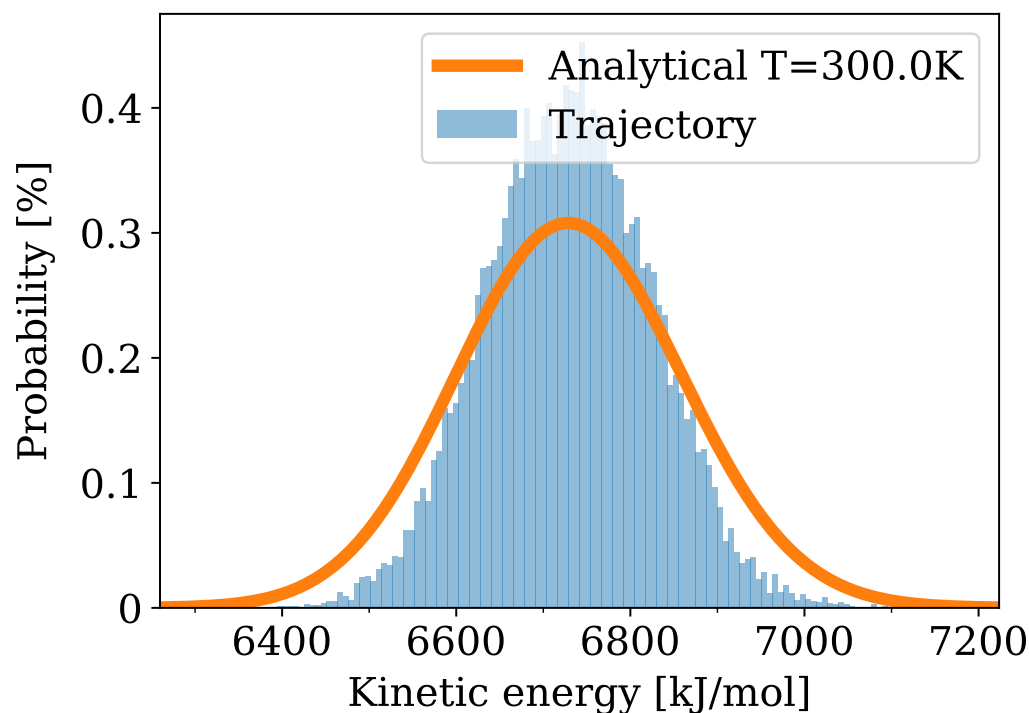
- It suppresses fluctuations!

True thermostat



$T_{\mu} = 299.94 \pm 0.04 \text{ K}$   
 $T_{\sigma} = 301.01 \pm 1.47 \text{ K}$   
 $p = 0.214045$

Berendsen thermostat



$T_{\mu} = 299.99 \pm 0.03 \text{ K}$   
 $T_{\sigma} = 222.60 \pm 1.16 \text{ K}$   
 $p = 8.42116 \times 10^{-83}$

# Bussi-Parrinello

- A simulation should not have a single target kinetic energy. It should have a **Maxwell-Boltzmann kinetic energy distribution**.
- Don't target a **fixed** kinetic energy  $K$ . Target at **random**  $K$ , chosen from the Maxwell-Boltzmann distribution
- This gives a **correct distribution of kinetic energies**
- **Rigorously consistent** with KE NVT distribution

$$\frac{dKE}{dt} = k(KE_{\text{Maxwell-Boltzmann}} - KE)$$

In GROMACS, `tcoupl = v-rescale`

# Nosé-Hoover thermostat

- Also a scaling thermostat:
  - If kinetic energy  $K$  is too big, make all particles slower
- Math is too complicated for today
- **But** it is essentially a **second-order** thermostat

$$\frac{dKE}{dt} = k(KE_0 - KE)^2$$

- Relaxes to right  $K$  **faster** than Berendsen
- BUT can oscillate back and forth
  - This can be bad for solids
- *Almost always* gives the right NVT

# Do I need a thermostat?

- Unless you are measuring transport properties, almost certainly yes.
- It is VERY EASY to start a simulation at too high a total E. Need to maintain T!
- If your simulation is SLIGHTLY not obeying  $F=ma$ , it will also keep the simulation from accumulating or losing much energy

# Thermostat recommendations

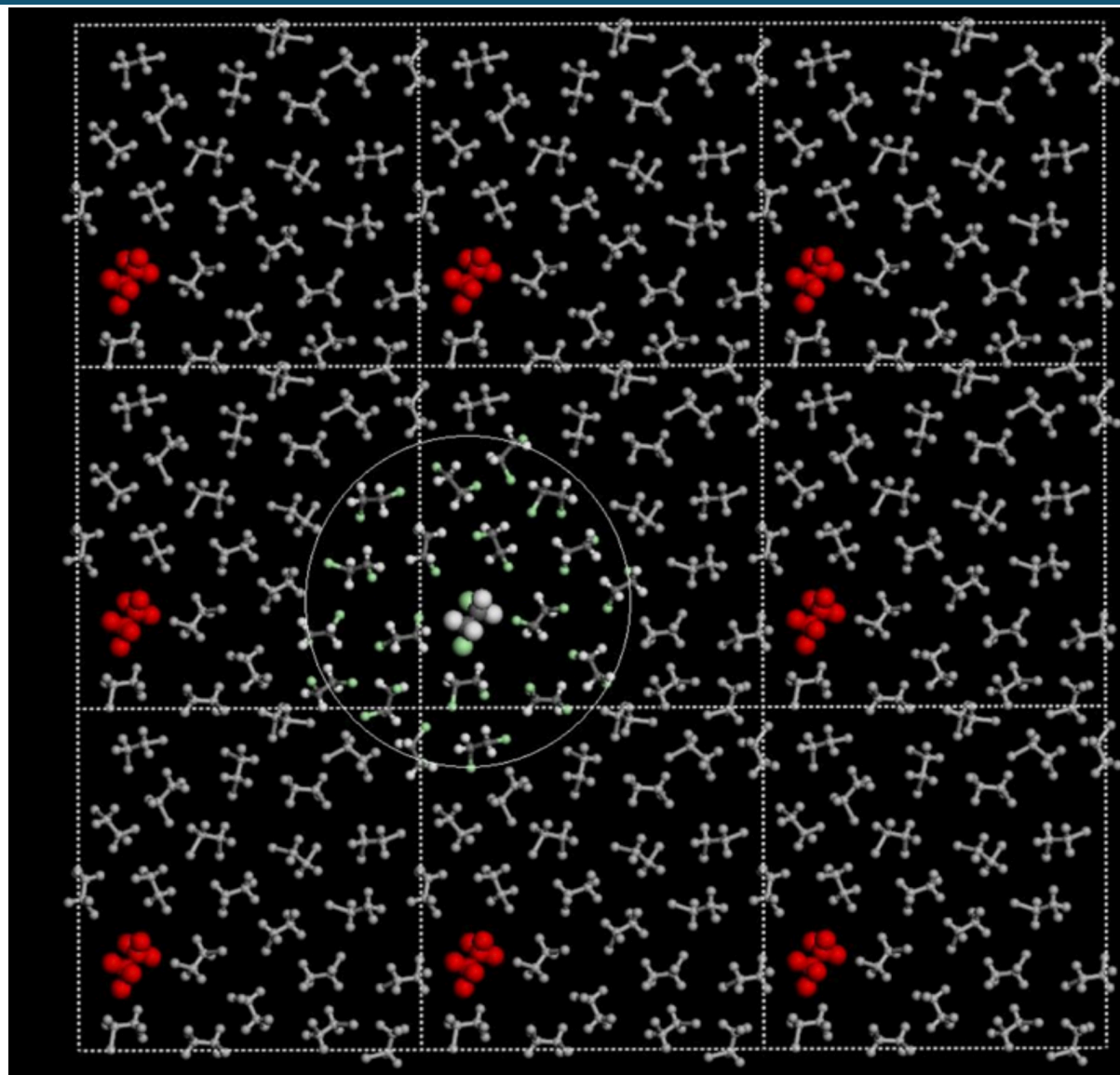
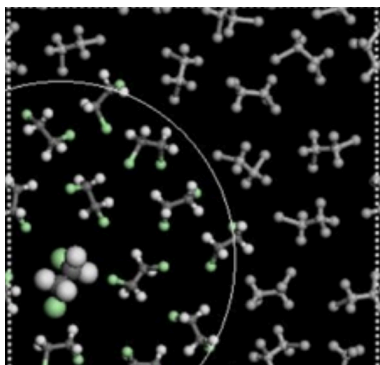
- Langevin is most stable, but slows system down a bit, especially at large friction values
  - Anderson is slows things down a bit more.
  - The ONLY ones you should use for gases
- Bussi-Parrinello is simple and usually very good
- Nosé-Hoover is OK for liquids, not so good for gases/solids
- Berendsen (weak coupling) is only good to equilibrate a system.

# How do we simulate bulk properties?

- Our simulation box is usually only a few nanometers across!
- We could simulate with infinitely hard walls, or simulate a droplet
  - But surface effects are at **least** 1-2 nanometers.
- Periodic boundary conditions:
  - We model the bulk by REPEATING the simulation cell.
- When a particle goes **out** one side, it comes **back** on the other.
- To calculate forces, we calculate the force from **the neighboring copies**.

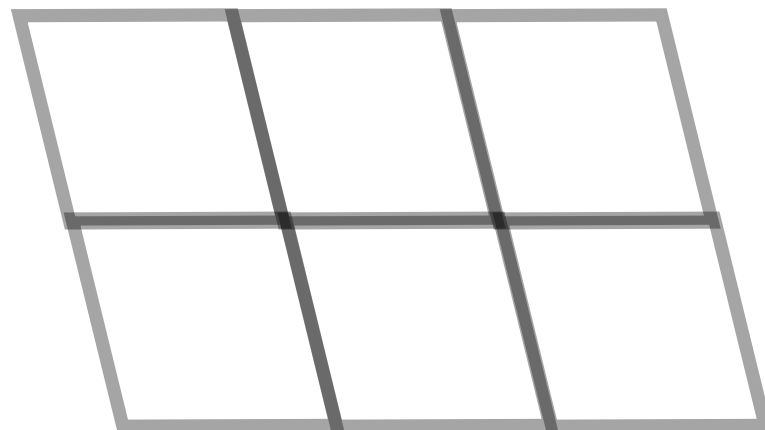
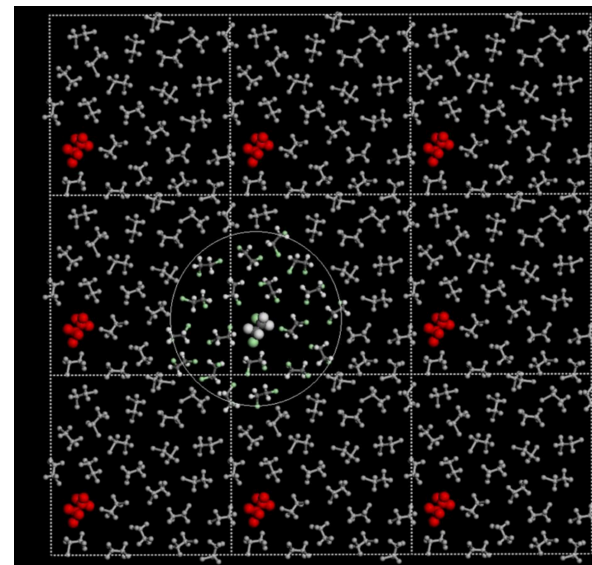


# Periodic Boundary Conditions



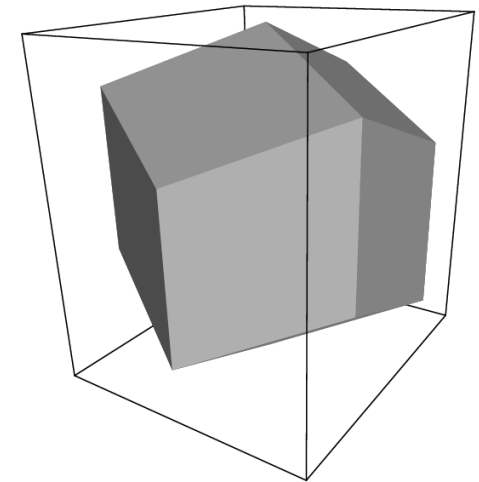
# Periodic Boundary Conditions

- Boxes can be cubic or tilted
- When not to use:
  - Single molecule
  - Droplet
- Molecules can't be longer than the box, or you get weirdness.

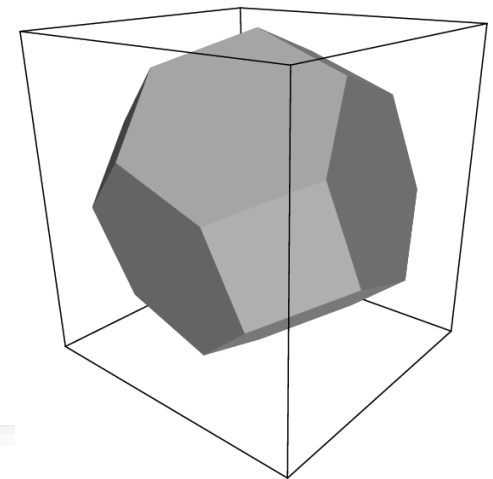


# Possible ways to tile 3d space

box type	image distance	box volume	box vectors			box vector angles		
			a	b	c	$\angle bc$	$\angle ac$	$\angle ab$
cubic	$d$	$d^3$	$d$	0	0	$90^\circ$	$90^\circ$	$90^\circ$
			0	$d$	0			
			0	0	$d$			
rhombic dodcahdron (xy-square)	$d$	$\frac{1}{2}\sqrt{2}d^3$ $0.707d^3$	$d$	0	$\frac{1}{2}d$	$60^\circ$	$60^\circ$	$60^\circ$
			0	$d$	$\frac{1}{2}d$			
			0	0	$\frac{1}{2}\sqrt{2}d$			
rhombic dodcahdron (xy- hexagon)	$d$	$\frac{1}{2}\sqrt{2}d^3$ $0.707d^3$	$d$	$\frac{1}{2}d$	$\frac{1}{2}d$	$60^\circ$	$60^\circ$	$60^\circ$
			0	$\frac{1}{2}\sqrt{3}d$	$\frac{1}{6}\sqrt{3}d$			
			0	0	$\frac{1}{3}\sqrt{6}d$			
truncated octahedron	$d$	$\frac{4}{9}\sqrt{3}d^3$ $0.770d^3$	$d$	$\frac{1}{3}d$	$-\frac{1}{3}d$	$71.53^\circ$	$109.47^\circ$	$71.53^\circ$
			0	$\frac{2}{3}\sqrt{2}d$	$\frac{1}{3}\sqrt{2}d$			
			0	0	$\frac{1}{3}\sqrt{6}d$			



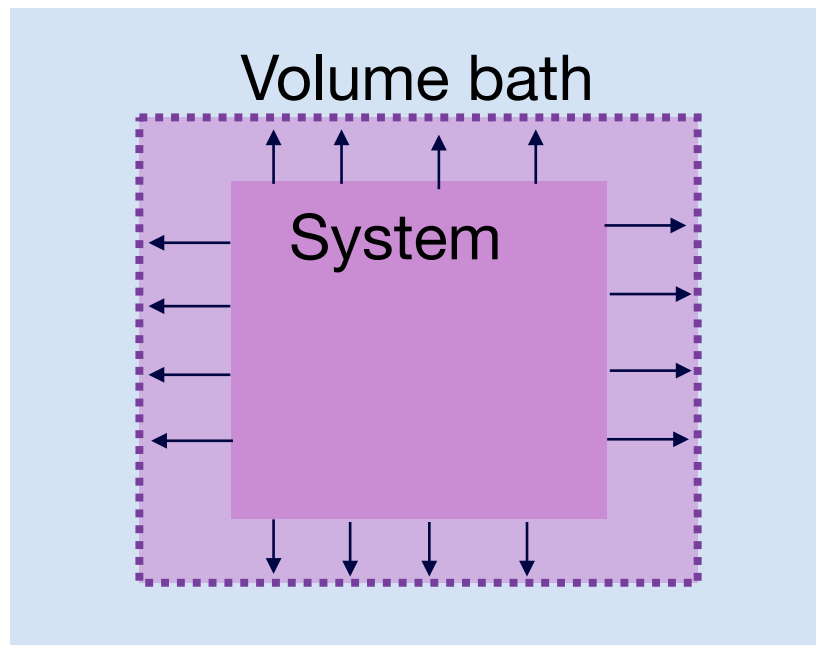
rhombic dodecahedron



truncated octahedron

# Barostats

- So: We know how to
  - Handle thermal equilibrium
  - What box volume means for a periodic system
- What about **NPT**
  - Where the volume is not constant?

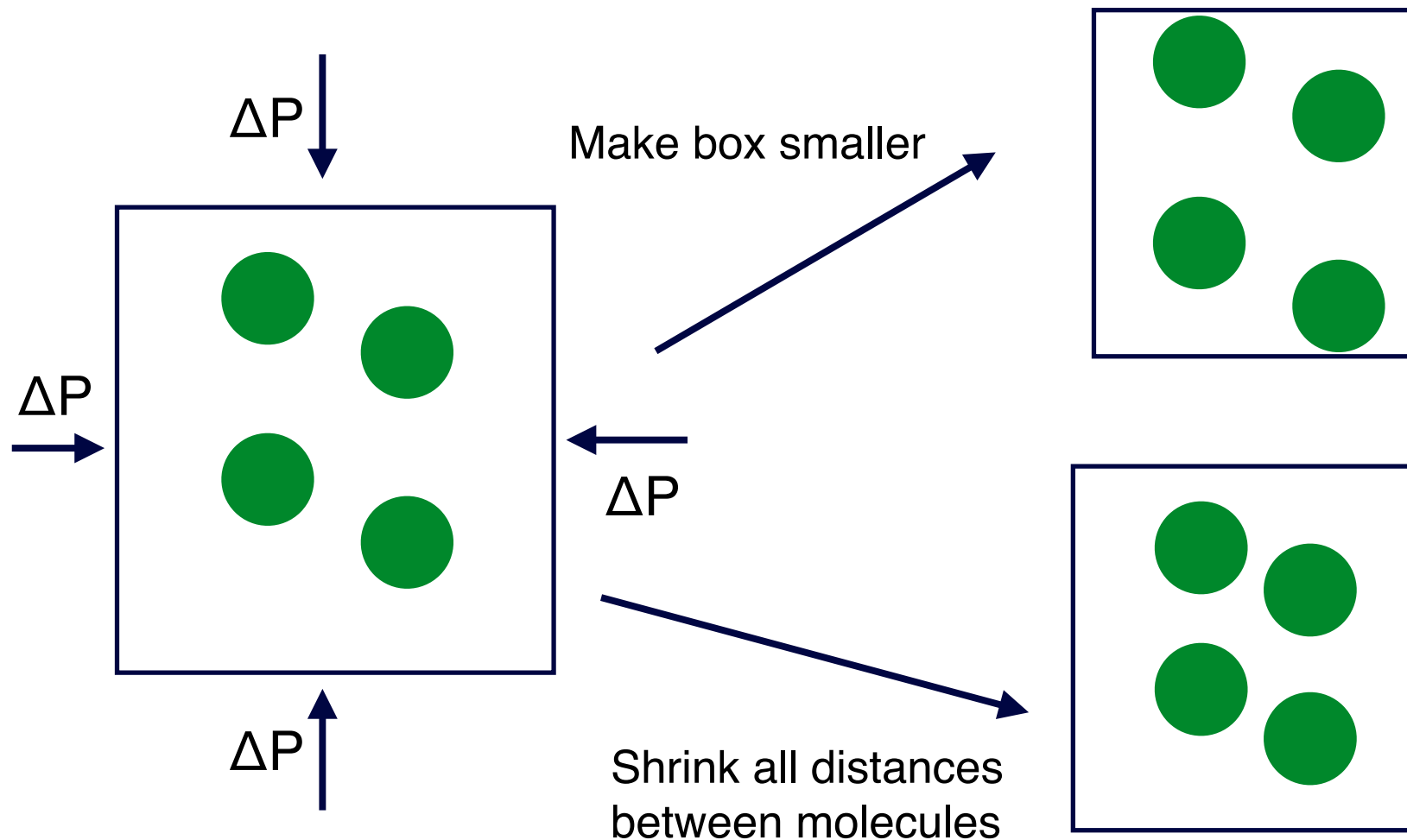


- System exchanges volume with the surroundings until in mechanical equilibrium with its environment.

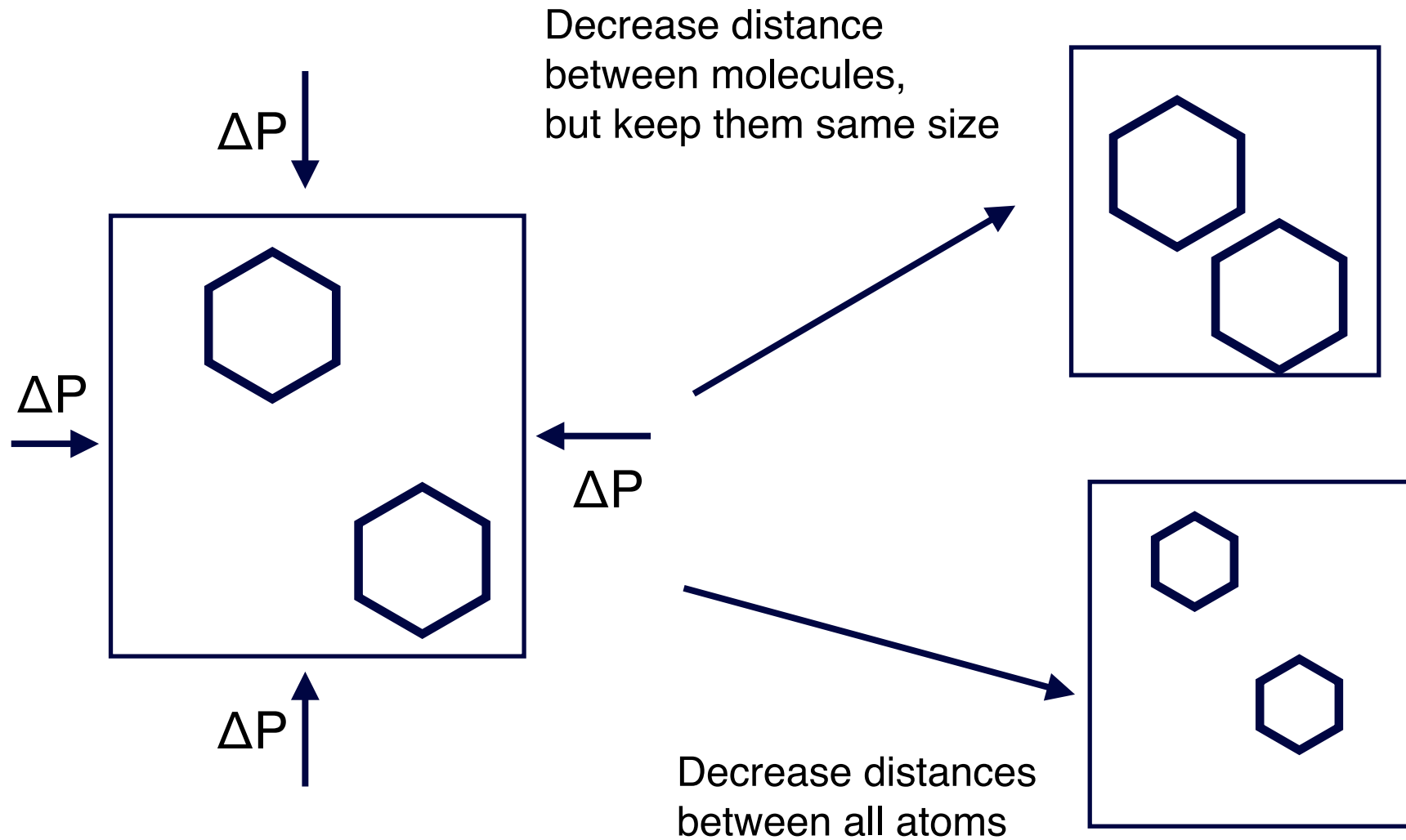
# How does a real life barostat work?

- Outside particles collide with your system.
  - If there are lots of hard collisions with your system, it pushes your system together
  - If there are few hard collisions with your system, your system pushes out.
- How can we model that in a periodic boundary system?
  - There is no **outside**. . .

# How does an MD barostat work?



# Do we shrink molecules??



# How do we know whether to shrink or grow?

$$P = \left\langle \frac{1}{V} \left( 2KE + \frac{1}{3} \sum_{i>j} r_{ij} \cdot F(r_{ij}) \right) \right\rangle$$

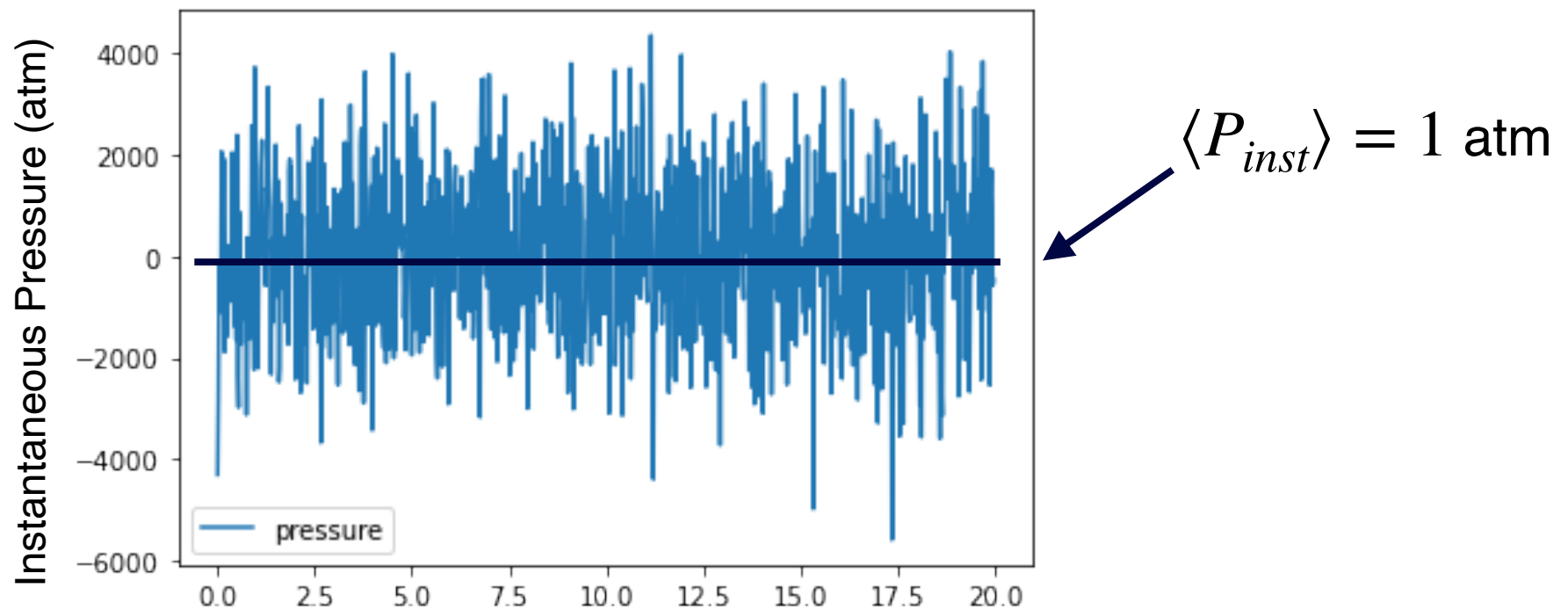
- The **virial** compares the "pushing out" of the kinetic energy with the "pulling in" of the forces
- This is an "instantaneous pressure" not the external pressure. The pressure is constant.
- We compare the quantity inside the averages to  $P$
- If it's too big, we shrink!
- If it's too small, we grow!



# Properties of "instantaneous pressure"

$$P = \left\langle \frac{1}{V} \left( 2KE + \frac{1}{3} \sum_{i>j} r_{ij} \cdot F(r_{ij}) \right) \right\rangle$$

- Can oscillate wildly (like, several 1000 atm!)



# How a simple barostat works

- " $P$ "<sub>estimate</sub> >  $P$ <sub>desired</sub>
  - Shrink box by some factor  $s$
- " $P$ "<sub>estimate</sub> <  $P$ <sub>desired</sub>
  - Expand box by some factor  $s$
- HOW BIG should  $s$  be?
- Let's make  $s$  is proportional to  $P_{\text{target}} - P_{\text{instant}}$
- What is the constant of proportionality, i.e. how QUICKLY does one respond to differences in pressure?
- If we say:  $V_{\text{new}}/V_{\text{old}} = k(P_{\text{target}} - P_{\text{instant}})$ 
  - in GROMACS, `tau_p` sets  $k$
- Berendsen's weak coupling barostat
- `pcouple=Berendsen`, and set `ref_p` as well!
- **Guess what!** Weak coupling suppresses true fluctuations in  $V$ !

# Parrinello-Rahman barostat

- Parrinello-Rahman barostat
  - Similar theory to Nosé-Hoover thermostat
  - 2nd order, so approaches true answer with some oscillations
  - Common in most MD packages
  - Rigorously\* gives NPT
    - Has some minor errors if you have *constraints*, like rigid water, hydrogen bond constraints
  - Tends to be somewhat numerically unstable, especially for small systems
- `pcoupl = Parrinello-Rahman`

# Cell-resizing barostat

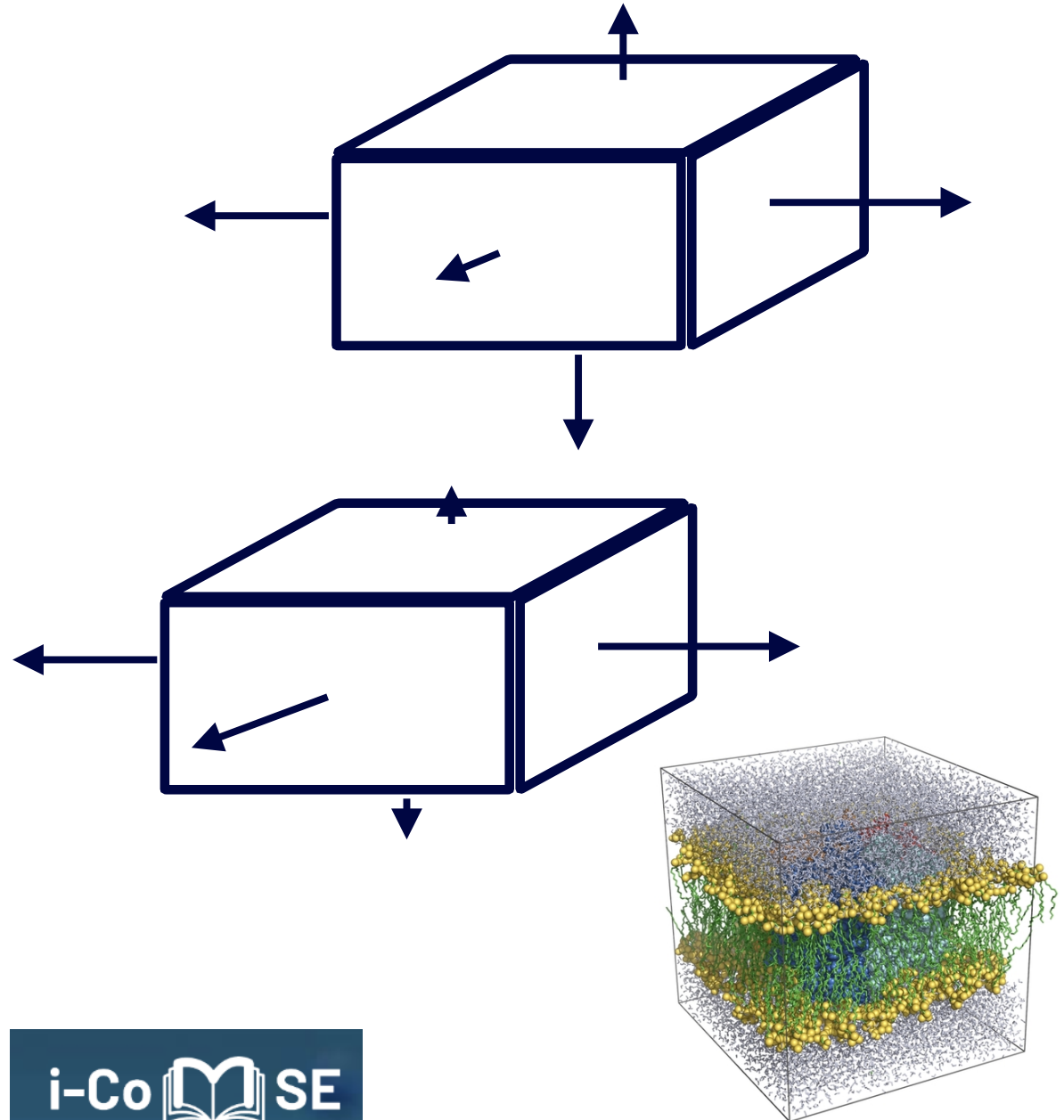
- Very new!
- Similar theory to Bussi-Parrinello thermostat
- Like Berendsen, but with added fluctuations to get NPT right.
- Rigorously gives NPT
- Not that well understood where it might fail, but in theory it should be the best barostat
- `pcoupl = c-rescale`

# Monte Carlo Barostat

- Change the system volume by a random small amount
  - If  $U+PV$  is lower, then accept the change
  - If  $U+PV$  is higher, then roll a random number to accept
  - If change is accepted, then shrink/expand the box
- Gives correct NPT volume distributions!
  - implemented in OpenMM
- Doesn't this breaks the dynamics
- Barostats **already** break the dynamics, since particles don't magically move together and apart in nature without direct action from other molecules

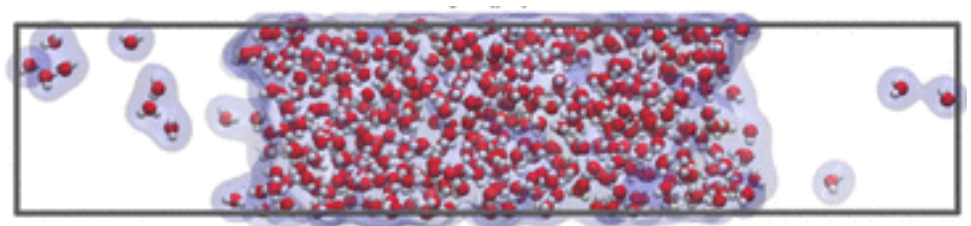
# Other possibilities for a barostat

- Isotropic
  - Box side ratios do not change
- Anisotropic:
  - All box sides can move independently.
  - The ratio of the sides can change
- Semiisotropic
  - x and y isotropically, but not z
  - Common for bilayer/lamellar simulations



# What can go wrong with a barostat?

- The systems of equations for MD are significantly less stable than MD alone or MD + thermostat
  - Because particles can get "scaled" into each other!
- More likely to crash.
  - If you start from a bad configuration.
  - If the system is poorly equilibrated
  - If the system is very small.
  - If there is some sort of phase separation or self-assembly



# Do I need a barostat?

- You almost certainly need a barostat to relax your simulation to the equilibrium volume.
  - You don't want to run at a density that is too high or too low
  - Like running at 1000 atm, or bubbles could form.
- BUT if your system is **highly incompressible**,
  - You may not need to run your simulation at NPT once it reaches the right P
  - You could average the volume over the last 100 ps, find the average volume, and pick a frame with that average volume, and run with that barostat
- OFTEN results are OK with the Berendsen barostat



# Some guidelines

- Thermostats:

- For equilibration, have a SHORT  $\tau_t$ 
  - 0.1 to 1 ps
- For production, have a LONGER  $\tau_t$ 
  - 5 or 10 ps

- Barostats:

- Instantaneous pressure will frequently oscillate between  $\pm 1000$  atm for a low compressibility fluid. THIS IS NORMAL!
- Generally, keep  $\tau_p$  about 5x  $\tau_t$
- If the simulation crashes because of volume fluctuations, try a somewhat longer  $\tau_p$  (2x-5x). Results will be valid as long as  $\tau_p$  is  $< 0.1-1\%$  of the simulation run.