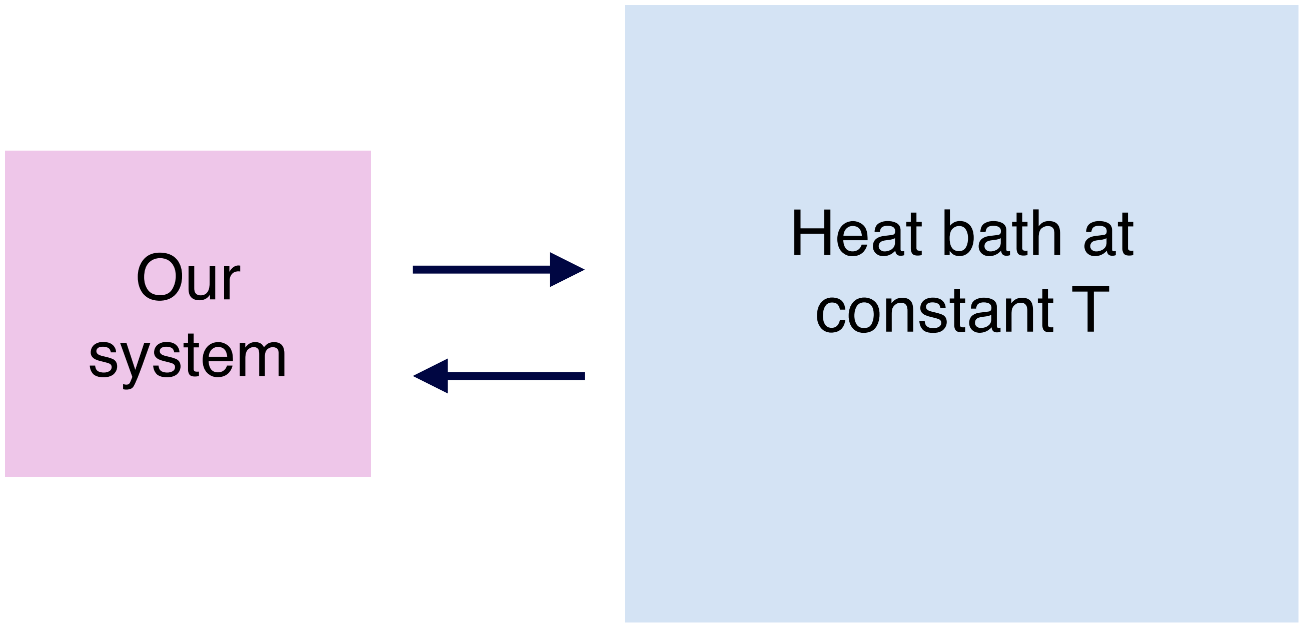
Why do we need to thermostat molecular dynamics?

- Molecular dynamics is a solution to the initial value partial differential equations defined by Newton's equation of motion
- Given x,v (or dx/dt) for all particles
  - F=ma **means** $d^2x/dt^2 = -1/m\ dU/dx$
  - We want to know: what are x and dx/dt of each particle at all future times?
- Newton's laws of motion conserve energy
- Therefore this gives constant number of particles, constant volume, and **constant energy**
- But we want answers at, say 300 K.

We want to simulate properties at constant T



Exchange thermal energy until at equilibrium

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
  - If they are cooler than the system average, then energy goes out of the system

How do 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?

- This will define a **rate** at which excess kinetic energy is removed

$$dK/dt = k(K-K_0)$$

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

Bussi-Parrinello

- A simulation should not have a single 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 NVT and exp(-βU)

$$dK/dt = k(K-K_{\text{Maxwell-Boltzmann}})$$

In GROMACS, `tcoupl = v-rescale`

Nose-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

$$dK/dt = k^2(K-K_0)^2$$

- Relaxes to right K faster than Berendsen
- BUT oscillates back and forth
- *Pretty much* gives the right NVT

Langevin dynamics

$$m dv/dt = F + R - \gamma v$$

- 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 awa with a friction term
  - Like moving through a fluid, which d necessarily make sense if you are simulating your fluid already . . .

- 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.**

Do I need a thermostat?

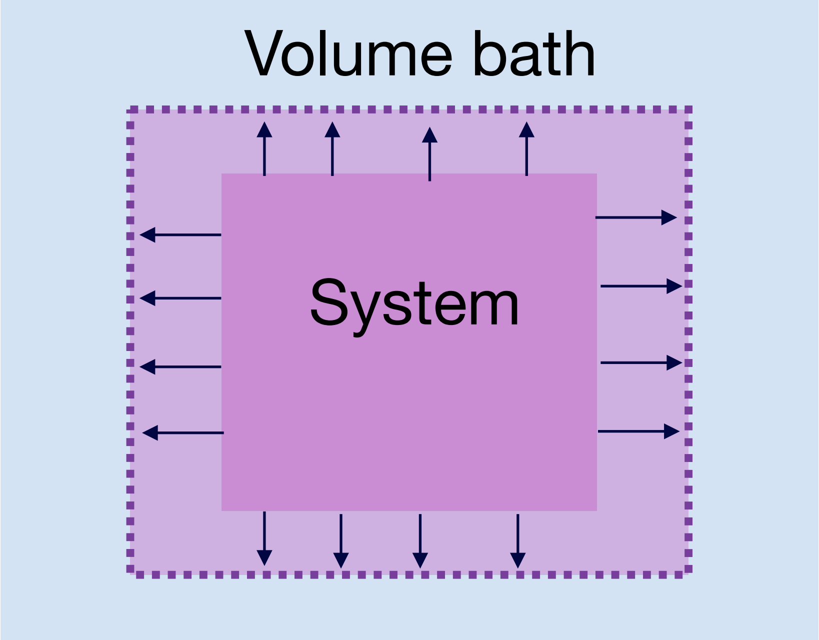
- Unless you are measuring transport properties, almost certainly yes.
- It is VERY EASY to start a simulation at too high a total E. Need to maintain T!
- If your simulation is SLIGHTLY not obeying F=ma, it will keep the simulation from accumulating or losing much energy

Thermostat recommendations

- Langevin is most stable, but slows system down a bit, especially at large
  - The ONLY one you should use for gases
- Bussi-Parrinello is simple and usually very good
- Parrinello-Rahman is OK for liquids, not so good for gases/solids
- Berendsen is only good to equilibrate a system.

Barostats

- So: We know how to handle NVT
- What about N**P**T
  - 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?

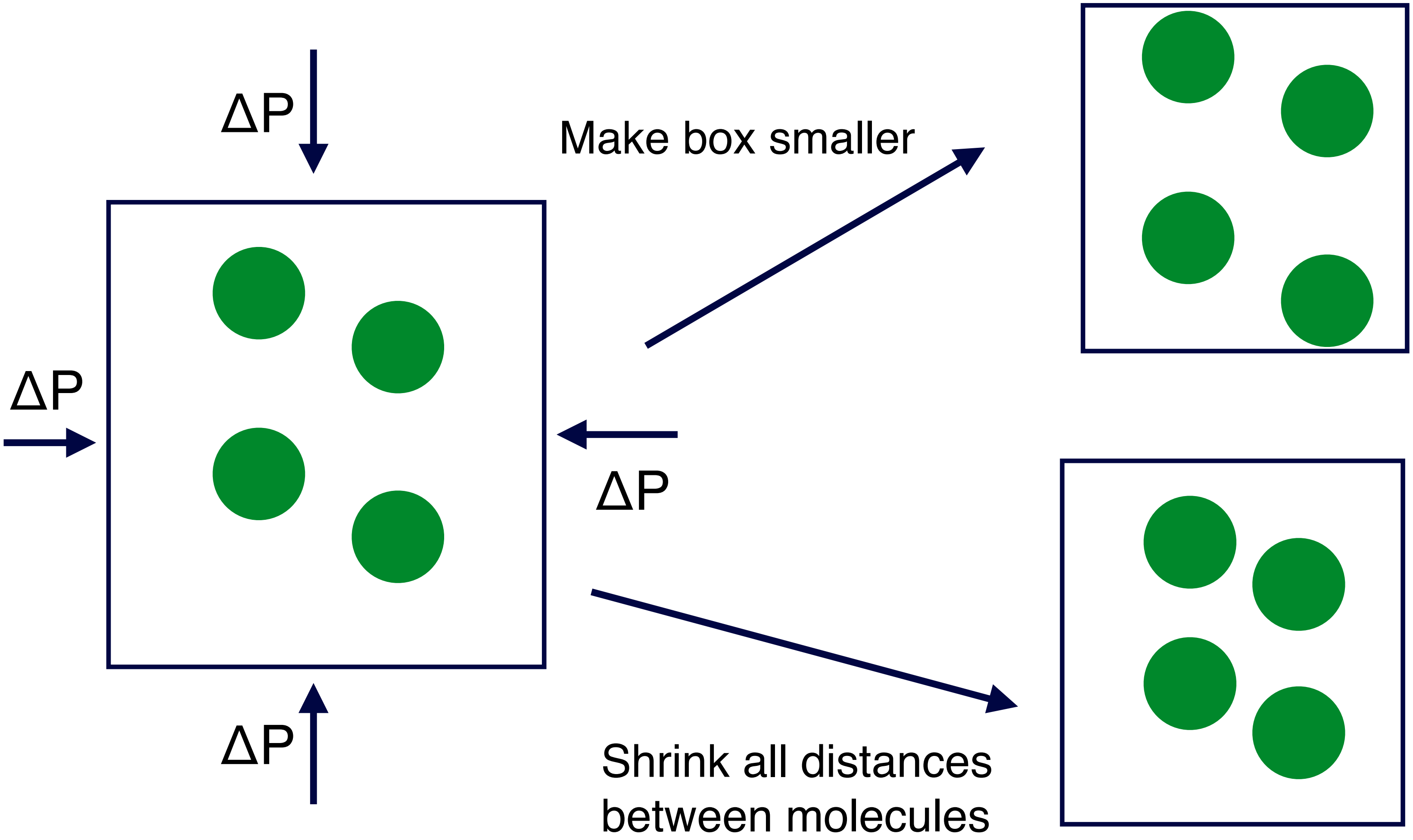
- Particles collide with your system.
  - If there are lots of hard collisions with your system, it pushes your system together
  - If there are few hard collisions with your system, your system pushes out.

How does an MD barostat work?

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

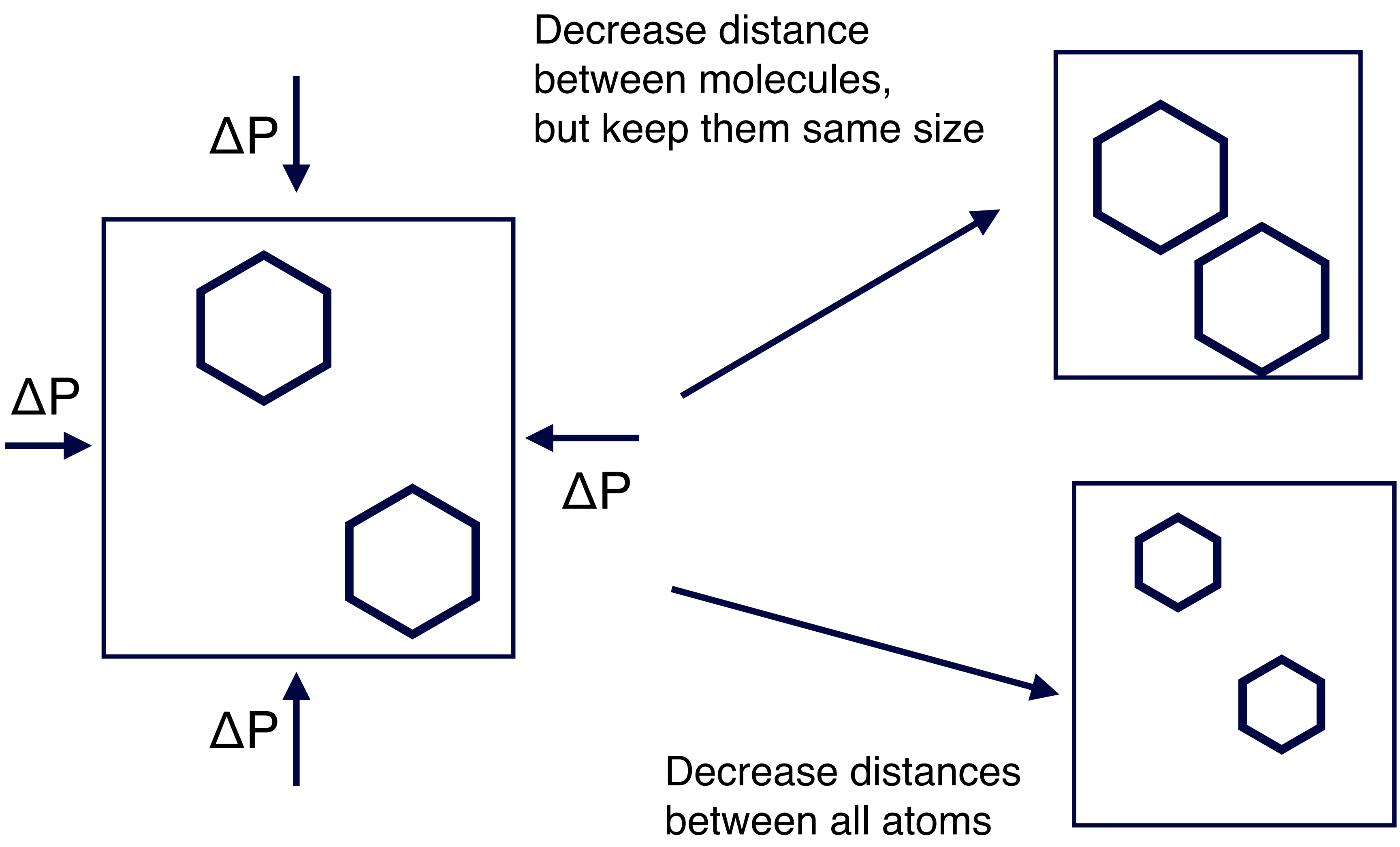
- We compare the quantity in the brackets to P
- It can oscillate wildly (like, several 1000 atm)
- Again, there is no such thing as an instantaneous pressure.
  - Just frames where more of the system wants to breath out, and frames where more of the system wants to breath in.

How does an MD barostat work?



i-Co SE

Do we shrink molecules??



Decrease distance
between molecules,
but keep them same size

Decrease distances
between all atoms

How a barostat works

- "P"$_{estimate}$ > P$_{desired}$
  - Shrink box by some factor s
- "P"$_{estimate}$ < P$_{desired}$
  - Expand box by some factor s

- HOW BIG should s be?
- Assume s is proportional to P$_{estimate}$ -P$_{desired}$
- What is the constant of proportionality, i.e. how QUICKLY does one respond to differences in pressure?
- Governed by `tau_p`

- If we say: $V_{new}/V_{old} = k(P_{estimate}-P_{desired})$
- Berendsen's weak coupling barostat
- `pcouple=Berendsen`, and set `ref_p` as well!

Parinello-Rahman barostat

- Parrinello-Rahman barostat
  - Similar theory to Nose-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`

Parrinello-Rahman barostat

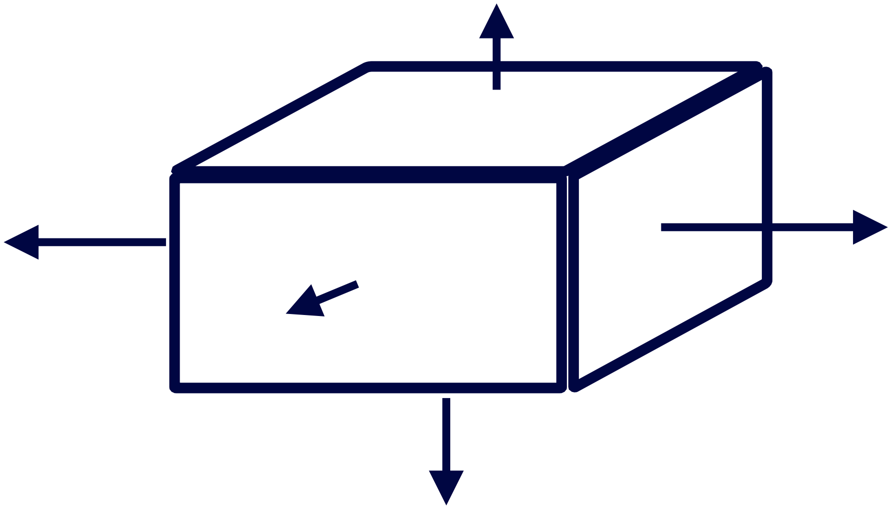
- Cell-resizing thermostat
  - Very new!
  - Similar theory to Bussi-Parrinello barostat
  - Like Berendsen, but with added fluctuations to get NPT right.
  - Rigorously gives NPT
  - Not that well understood where it might fail, but in theory it might be the best wone.
  - `pcoupl = c-rescale`

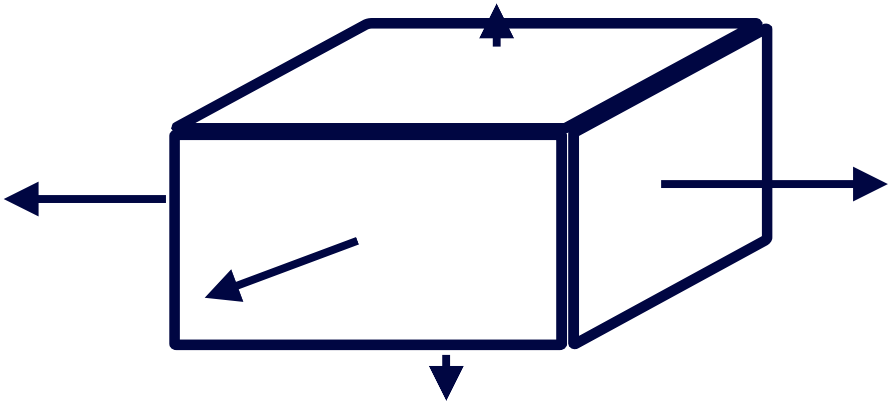
Monte Carlo Barostat

- Change the system volume by a small amount
- If U+PV is lower, then accept the change
- If U+PV is higher, then roll a random number to accept
  - If accepted, then shrink/expand the box
- Gives correct NPT volume distributions!
- Breaks the dynamics
- Barostats **already** break the dynamics, since particles don't magically move together and apart in nature

Other possibilities

- 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
- In GROMACS, `pcoupltype = anisotropic`





What can go wrong with a barostat

- The systems of equations are significantly less stable than MD alone or MD + barostat.
- 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 you 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.
- 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:
  - Pressure will frequently oscillate between +/- 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 (2x-5x). Results will be valid as long as `tau_p` is < 0.1-1% of the simulation run.