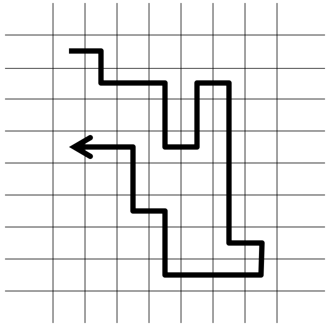
Monte Carlo and MD simulations

Andrew Torda, April 2015 strukt und sim

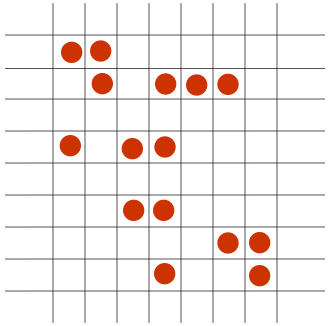
What we observe in any system ?

- averages of observables (pressure, energy, density)

Given enough time system will visit all states



time



random hopping

My observable $\mathcal{A}$

$$\mathcal{A}_{obs} = \frac{1}{b-a}\int_a^b \mathcal{A}_t dt \qquad \mathcal{A}_{obs} = \frac{1}{N_{obs}}\sum_{i=1}^{N_{obs}} \mathcal{A}_i$$

Time and space averages

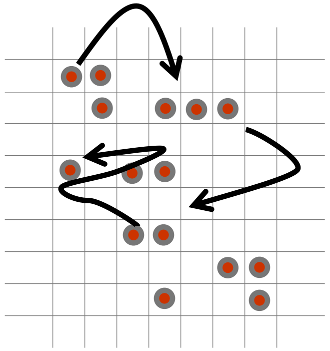
If we believe $\mathcal{A}_{obs} = \frac{1}{N_{obs}} \sum_{i=1}^{N_{obs}} \mathcal{A}_i$

then

$$\mathcal{A}_{obs} = \sum_{j}^{states} p_j \mathcal{A}_j$$

$$\equiv \langle \mathcal{A} \rangle$$

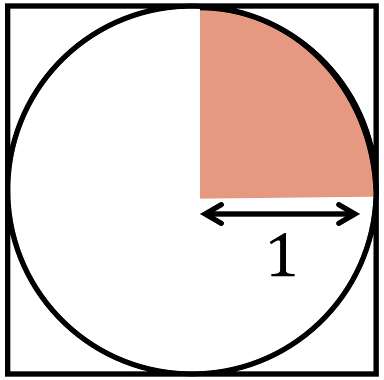
and p_j is the probability of state j



- $\langle \mathcal{A} \rangle$ is ensemble average and usually $\bar{\mathcal{A}}$ is time average
- if sample with correct probability, we can find $\mathcal{A}_{obs}$
- order of visiting states does not matter

Monte Carlo

How to calculate π with random numbers



$$\frac{points_{red}}{points_{square}} = \frac{{}^{1}\!/_{4}\,\pi r^2}{\text{area in square}}$$

$$\pi = 4\,\frac{points_{red}}{points_{square}}$$

```
while ( not converged)
    pick random $x, y$
    $n_{square}$++
    if (($x^2$+$y^2$) < 1)
        $n_{red}$++
print 4 $\frac{n_{red}}{n_{square}}$
```

diagram loosely from Allen, MP and Tildesley, DJ

Generating distributions / Monte Carlo

Generating points in a circle ? (generating function)

$$p_{in\_circle} = \begin{cases} 1 & x^2 + y^2 \leq 1 \\ 0 & x^2 + y^2 > 1 \end{cases}$$

- we could work out the area of a circle (integrate) by picking random numbers

What does Monte Carlo simulation mean ?

- generating points according to some distribution to find an average or integral
- what is our distribution in physical systems ?
  - Boltzmann distribution

Monte Carlo and Boltzmann distributions

Boltzmann probability distribution

$p_i = \frac{e^{\frac{-E_i}{kT}}}{\sum_j e^{\frac{-E_j}{kT}}}$ often written as $p_i = \frac{e^{\frac{-E_i}{kT}}}{Z}$ since we define $Z = \sum_j e^{\frac{-E_j}{kT}}$

- if we could generate this distribution,
  we could reproduce most properties of a system
- leads to a scheme (not possible)

correct, but not practical scheme

```
while (not happy)
    generate configuration $\mathbf{r}_i$ (conformation of protein, …)
    calculate $p_i$                    (number between 0 and 1)
    generate random number x
    if ($x < p_i$)
        accept $\mathbf{r}_i$
    else
        reject $\mathbf{r}_i$
```

$$p_i = \frac{e^{\frac{-E_i}{kT}}}{\sum_j e^{\frac{-E_j}{kT}}}$$

- result ? a set of $\mathbf{r}_i$ with Boltzmann distribution
- problem ? we do not know $\sum_j e^{\frac{-E_j}{kT}}$

a better scheme

We cannot generate points from $p_i = \frac{e^{\frac{-E_i}{kT}}}{\sum_j e^{\frac{-E_j}{kT}}}$

What if we have two configurations ?

$$\frac{p_i}{p_j} = \frac{e^{\frac{-E_i}{kT}}}{Z} \; \frac{Z}{e^{\frac{-E_j}{kT}}}$$

$$= e^{\frac{E_j - E_i}{kT}}$$

$$= e^{\frac{-\Delta E}{kT}}$$

a better scheme

$$\frac{p_i}{p_j} = e^{\frac{-\Delta E}{kT}}$$

If we have one configuration to start

- we can work out the relative probability of a second

Convenient convention

- going from old→new $\Delta E < 0$
  - $E_{new} - E_{old} < 0$ energy is better / more negative

Does it matter where you start ? What is i ?

Metropolis Monte Carlo

- generating a distribution $\frac{p_i}{p_j} = e^{\frac{-\Delta E}{kT}}$
- if $\Delta E < 0$, new is likely (more than 1)
- if $\Delta E > 0$, old is p_{new} is possible

```
generate starting configuration $\mathbf{r}_o$
while (not happy)
      generate $\mathbf{r}_{new}$
      calculate $E_{new}$ and $\Delta E$
      if $\Delta E$ < 0
            set $\mathbf{r}_o$ to $\mathbf{r}_{new}$
      else
            x = rand [0:1]
                  if$\left( x \le e^{-\Delta E / kT} \right)$
                  set $\mathbf{r}_o$ to $\mathbf{r}_{new}$
```

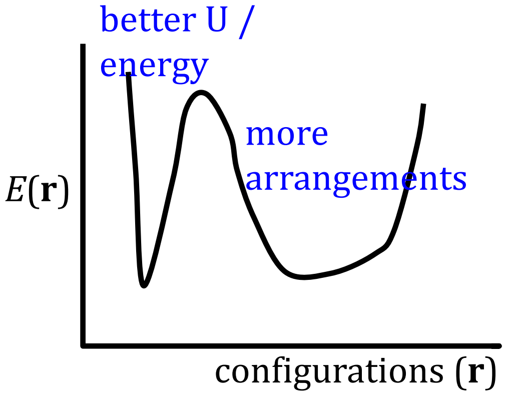
- what if ΔE slightly > 0 ?
  - 0.0000000001
- what if $\Delta E = 10^6$?
- small uphill moves are OK
- bigger moves are less likely

Properties of Monte Carlo

The set of $\mathbf{r}_o$ is a valid distribution (ensemble)

- for some property $\mathcal{A}$

$$\mathcal{A}_{obs} = \langle\mathcal{A}\rangle = \frac{1}{N_{visited}} \sum_{i}^{N_{visited}} \mathcal{A}_i$$



- $\mathcal{A}$ could be density, structural property, E, …
- only works for one temperature T

Look at picture.. could I calculate entropy / free energy ?

- for simple systems

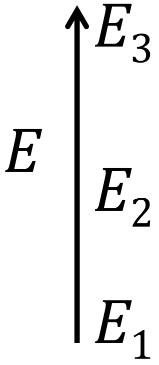
Equilibrium

MC results (observables / averages)

- only for system at equilibrium
- simulations generate system at equilibrium

What happens for a system out of equilibrium ?

- Toy system with 3 states
- for some *T*, at equilibrium
- $p_1 = {}^5/_8$ $p_2 = {}^1/_4$ $p_3 = {}^1/_8$
- if I have 80 copies of the system, most are in state$_1$



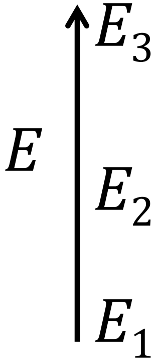
Reaching equilibrium

System wants $p_1 = {}^5/_8$ $p_2 = {}^1/_4$ $p_3 = {}^1/_8$

50 : 20 : 10

- start it with 5 : 70 : 5

- all moves 2→1 are accepted (large flux)
- the flux from 1 → 2
    - 1→ 2 moves are not always accepted
    - there are less particles in state_1



Moving to equilibrium depends on

- population
- probability

Detailed balance

For any two states (state$_i$ and state$_j$)

Flow $i \rightarrow j$ must equal $j \rightarrow i$

- otherwise ?

Flow $i \rightarrow j$ depends on

- population N_i
- probability $\pi(i \rightarrow \ j)$

Detailed balance

$$N_i \pi(i \rightarrow j) = N_j \pi(j \rightarrow i)$$

- detailed balance must apply for any pair i, j

all textbooks use π for probability here

Ergodic

Assumptions

- I can do integrals because
    - I will visit every state
    - I can calculate p_i for all states
- I will visit every state

alternatively

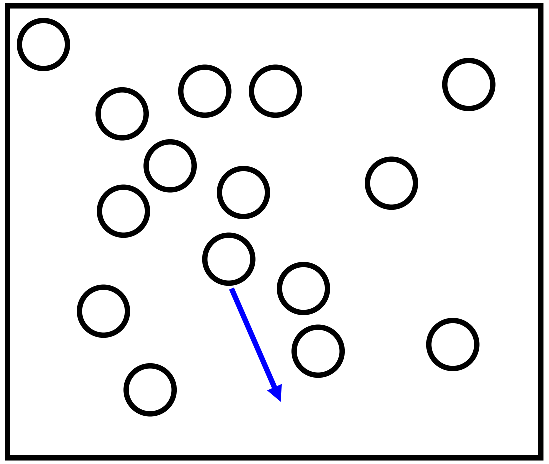
For any i, j

- $\pi(i \to j) > 0$
- may require a finite number of steps: $i \to k \to m \to j$
- must be satisfied

Moves

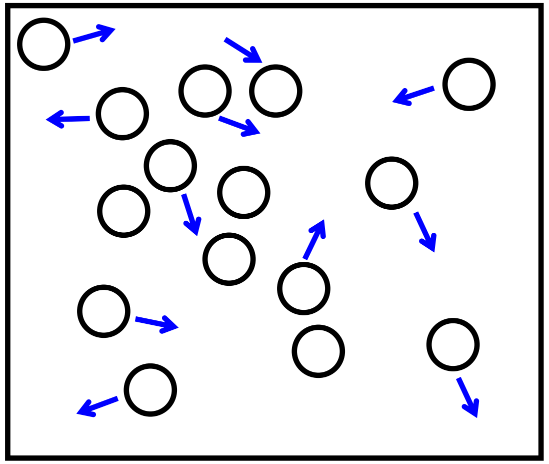
version 1

- decide on r_{max}
- pick a particle at random
- pick random $\Delta x, \Delta y, \Delta z$
  $0 < \Delta\, a < \; r_{max}$
- apply move
- accept / reject move



version 2

- decide on smaller r_{max}
- foreach particle
  - pick random $\Delta x, \Delta y, \Delta z$
    $0 < \Delta\, a < r_{max}$
- apply move
- accept / reject



Moves

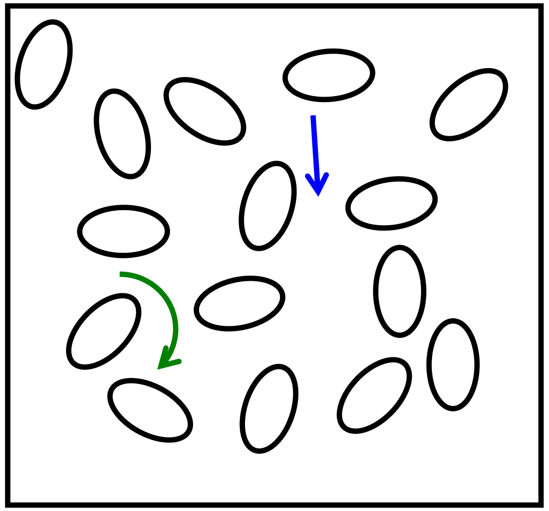
- both kinds of move OK
- note
    - "accept / reject"

More generally,

- how big is r_{max} ?
- big
    - system moves faster
    - more moves rejected

What if my particles are not spheres ?

- rotations also necessary



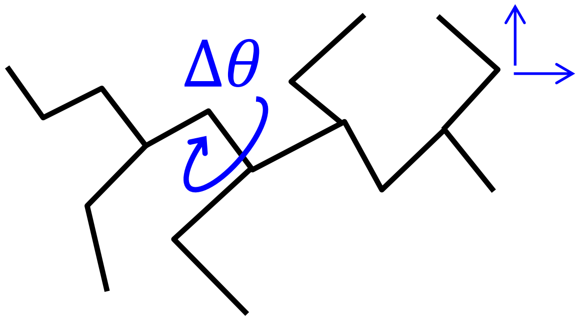
- time has no meaning

Bonded systems

Protein (lipid, polymer, ..)

Random Δ x ?

- nearly all will stretch a bond
  - high energy : rejected move
- only feasible method
  - random rotations $\Delta\theta$



In general

- most kinds of simple moves OK
- must maintain detailed balance, ergodicity
- question of efficiency
  - high rejection rate means lots of wasted calculations

More moves – *N* particles

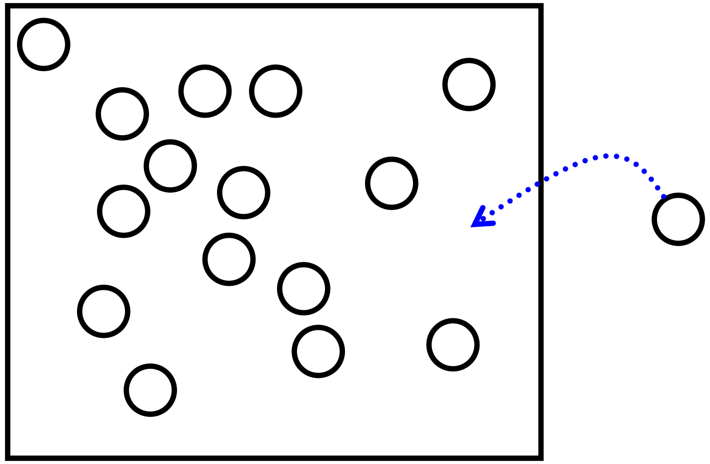
$$\frac{p_{new}}{p_{old}} = e^{-\Delta E/_{kT}}$$

I have defined temperature

- and $N_{particles}$ and V
- called NVT simulation

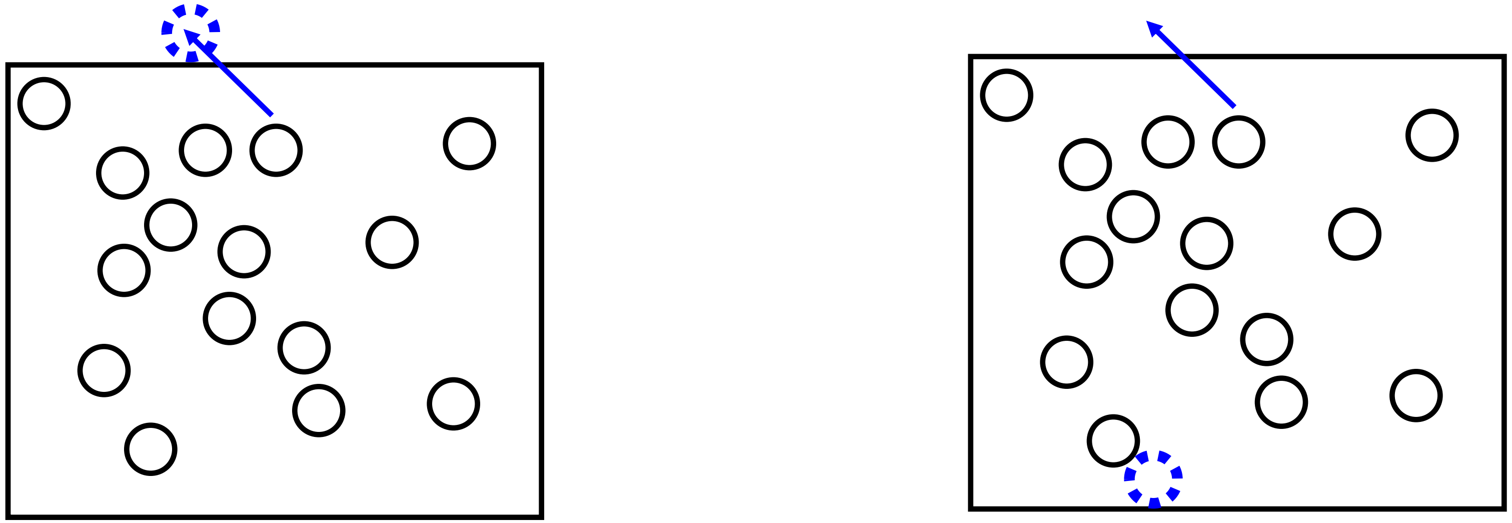
Could I have varied something else ?

- what if I tried to put particles in / take out ?
  - sometimes energy ↑sometimes↓
- system will fluctuate around $\langle N \rangle$
- this would not be NVT

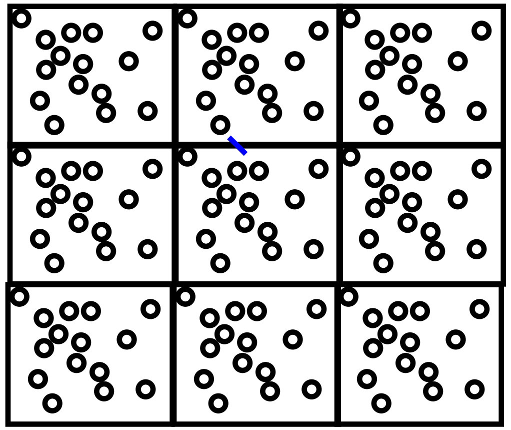


Periodic Boundary Conditions

Technical point relevant to gases, proteins in water…



Behaves like an infinite system

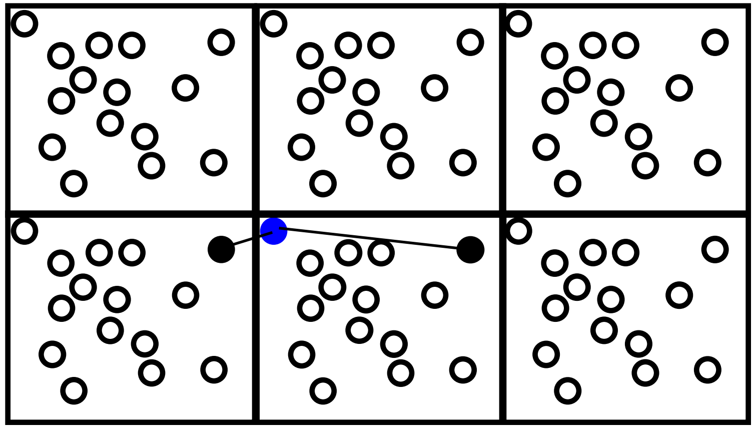


Infinite interactions ?

Neighbours of blue particle

- only use the nearer
- not really an infinite system

- volume defined by box



Problems with Monte Carlo

```
while (not happy)
        propose move
        accept / reject move
```

Small steps ?

- system moves slowly: long time to visit all states

Big steps ?

- calculate energy
- reject move
    - no progress, wastes time

Dense Systems and Monte Carlo

Random moves ?

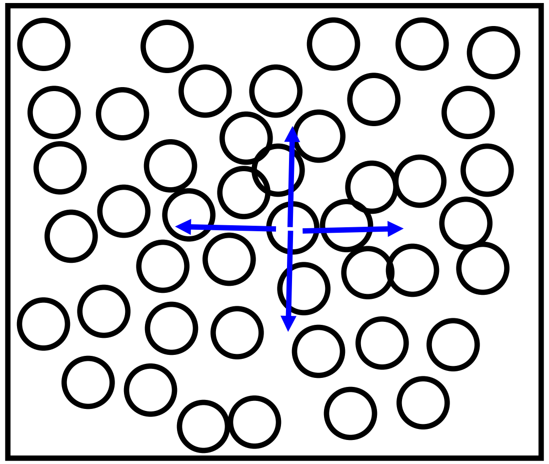
- most moves rejected

Dense systems ?

- liquids
- proteins, polymers, …

Solutions

- cleverer MC moves (later)
- MD



Why do molecular dynamics simulations ?

Real world

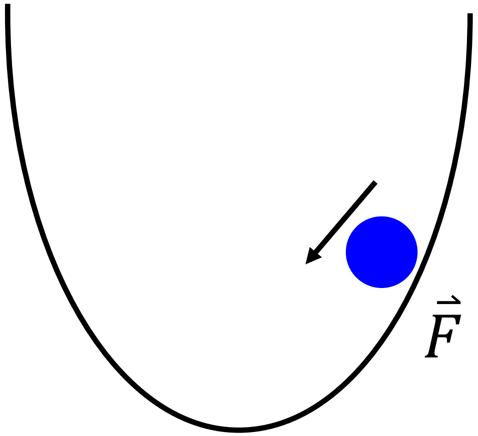
- box of gas, molecule in space, protein molecule in water
- atoms hit each other,
  - share energy, box expands/contracts, ..
  - soon reaches equilibrium
  - visits low energies (often), high energies (less often)
  - visits entropically favoured regions
- we stick in a thermometer
- measure density, …

What have the atoms done ?

- feel forces and move
- an MD simulation just copies this

What do we expect ? Molecular Dynamics

one particle in a well



$$\vec{F} = \frac{-\partial U}{\partial \vec{r}}$$

Unlike MC, particles have kinetic energy E_{kin}

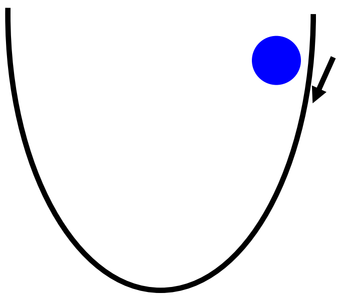
Kinetic and potential energy

Our system is isolated (no work done)

E_{tot} never changes

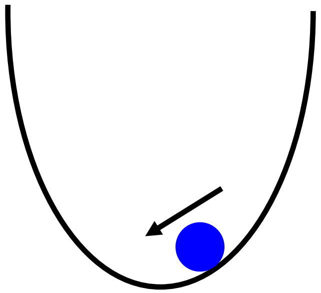
- conserves energy (no work done on system)

$$E_{tot} = E_{pot} + E_{kin}$$



E_{kin} small

E_{pot} big

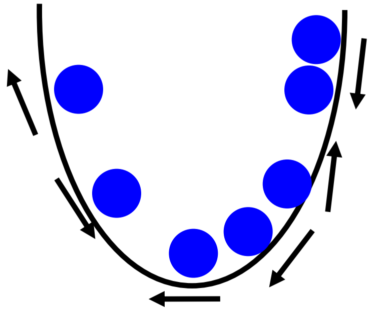


E_{kin} big

E_{pot} small

For one particle $E_{tot} = E_{pot} + E_{kin}$ = constant

Lots of particles



Particles hitting each other

- exchanging energy

Total system

- conserves energy

One particle ?

- maybe at bottom but moving slow (E_{kin} + E_{pot} small)
- per particle energy no longer conserved (may gain or lose)

Many particles

- distribution of velocities
- distribution of potential energies

Boltzmann distribution in real world

One version of real world (N, V, T)

- constant number of particles, volume, temperature
- today $E = E_{kin} + E_{pot}$
- Z is partition function
- earlier $Z = \sum_i e^{\frac{-\Delta E_i}{kT}}$

But now we have kinetic energy $E_{kin}(\mathbf{p})$

- where $\mathbf{p} = m\dot{\mathbf{x}}$
  - potential energy $E_{pot}(\mathbf{r})$
- if we write in continuous form …

Partition function for MD

Usually write $\mathcal{H}(\mathbf{p},\mathbf{r}) = E_{kin}(\mathbf{p}) + E_{pot}(\mathbf{r})$

- "Hamiltonian"

All the states are defined by all possible momenta and coordinates

- sum over these: $Z(N,V,T) \propto \int d\mathbf{p} \int d\mathbf{r}\, e^{\frac{-\mathcal{H}(p,r)}{kT}}$

often see $\mathrm{H}(\mathbf{p},\mathbf{r})$ or $\mathcal{H}(\boldsymbol{\Gamma})$

MD Method

For any particle we can calculate forces

Newtons law

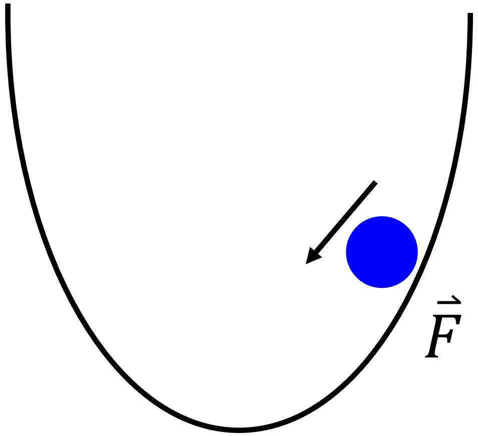
$F = ma$ often better written $\ddot{\vec{x}} = \vec{F}m^{-1}$

If we know acceleration

- we can get velocity

from velocity

- can get coordinates



$$\vec{F} = \frac{-\partial U}{\partial \vec{r}}$$

```
while (nstep < max_step)
     calculate forces
     integrate to get new coordinates
     nstep ++
```

averaging,
sampling,
…

Starting system

Initial coordinates

- protein model
- protein from protein data bank (PDB)
- protein + proposed ligand
- box of liquid

Do initial coordinates matter ?

- in principle: no
  infinitely long simulation visits all configurations, reaches equilibrium
- in practice: yes
  - bad examples
    - no simulation is long enough to predict protein conformation
  - take water configuration and run at ice temperature

Initial velocities

First consider temperature – reflects kinetic energy

$$\left\langle \frac{1}{2} m v_\alpha^2 \right\rangle = \frac{1}{2} kT$$

where v_α^2could be v_x, v_y, v_z

leads to definition

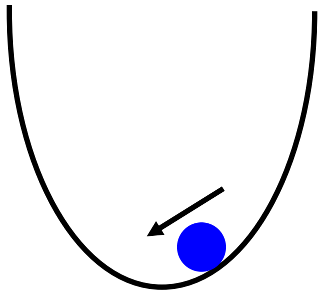
$$T(t) = \sum_{i=1}^{N} \frac{m_i v_i^2(t)}{k N_f}$$

- where N_f is number degrees of freedom $\approx 3N$
- we could use this to get initial velocities $\langle v_\alpha^2 \rangle = \frac{kT}{m}$

Initial velocities

Would one <v^2> be OK ?

- not very good
  - E_{kin} correlated with E_{pot}



Either

- use more sophisticated distribution
- do not worry
  - system will go to equilibrium
    - velocities will reach sensible values

Getting new velocities / coordinates

constant acceleration

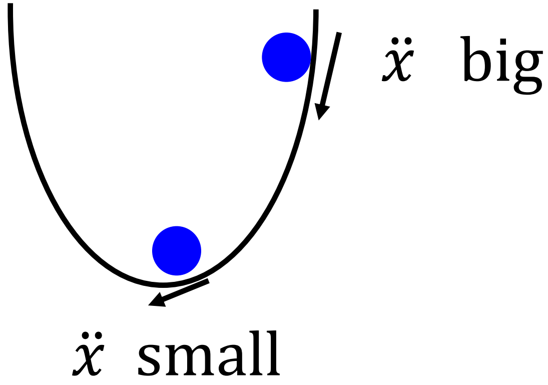
$$x_t = x_0 + vt + \frac{1}{2}at^2$$

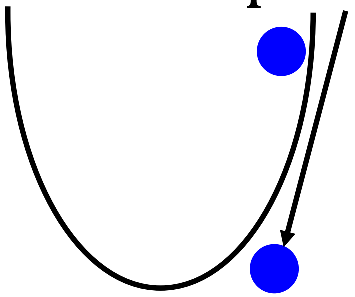
or

$$x_t = x_0 + \dot{x}t + \frac{1}{2}\ddot{x}t^2$$

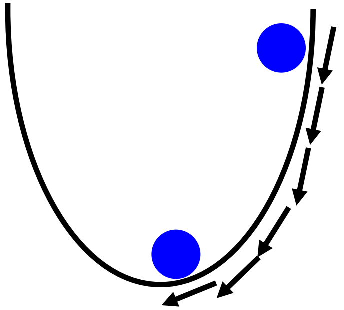
OK for constant acceleration

- try to use formula to predict future time





big Δt / step
big error



small Δt / step
small error
slow

Fundamental problem with integration

- We want to use big Δt (speed)
- We must use small Δt (accuracy)

All Δt will give us some error

- numerical integration is never perfect

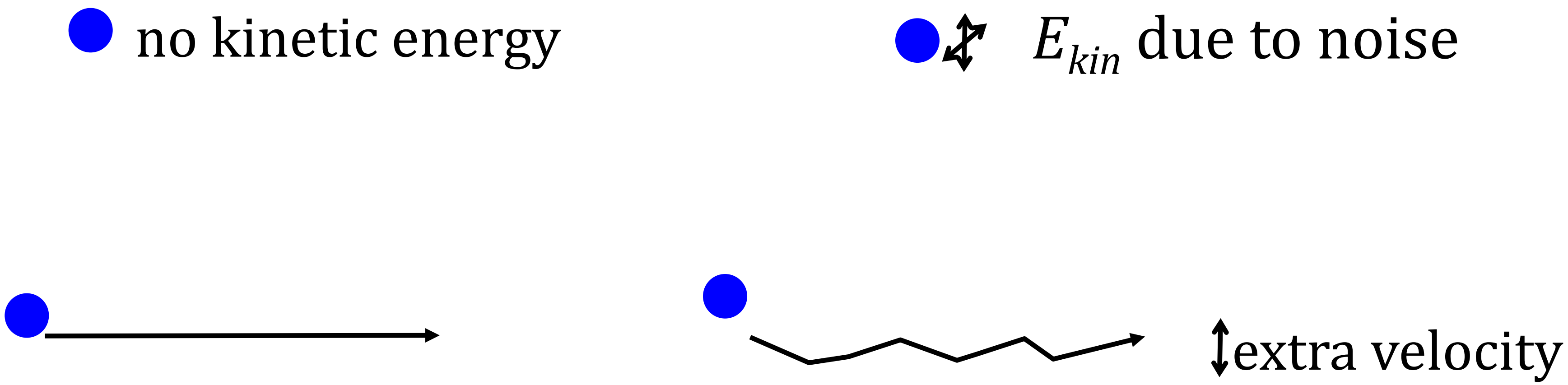
How small is Δt ?

- depends on fastest frequency / steepest walls in energy
  - usually bonds
- for proteins at room temperature
  - $\Delta t \approx 1$ fs (femtosecond 10^{-15} s)
- high temperature Δt should be smaller

Noise and heating

General rule

- noise heats the system
- formally difficult to prove
- $E_{kin} = ½\, mv^2$



no kinetic energy

E_{kin} due to noise

extra velocity

Noise-free Simulation

Energy conservation : Absolute rule $E_{pot} = f(\mathbf{r})$

- no time component
- invariant under translation, rotation

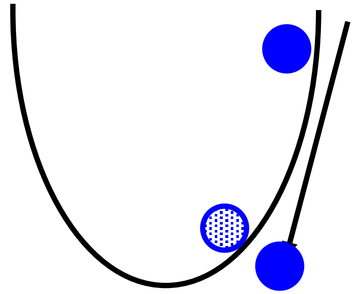
When violated ?

- $(\mathbf{r})$ does not change, but E_{pot} changes: E_{tot} changes

Noise Sources

Integrator

- coordinates do not match velocity
  E_{kin} wrong: $\left(E_{kin} + E_{pot}\right) \neq$ constant
- energy not conserved

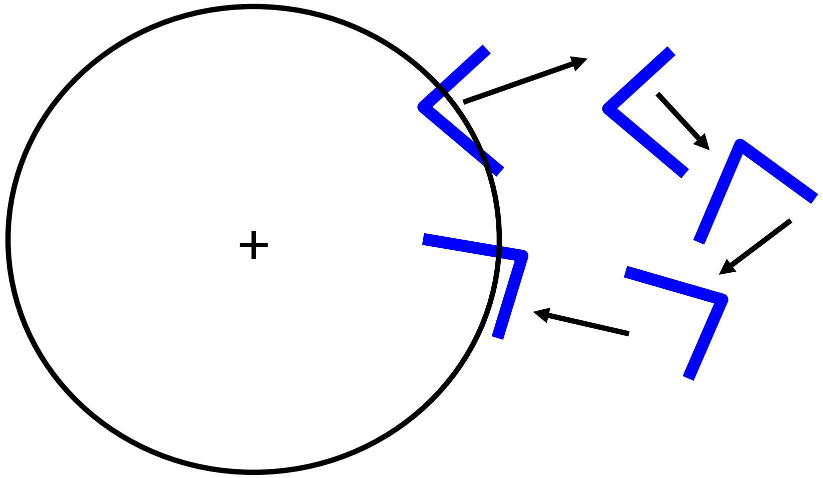


Numerical noise

- $E_{pot} = f(\mathbf{r})$
- initial coordinates $(\mathbf{r})$ quoted to 3 decimal places
- really less accurate

Cutoffs

- within cutoff rotation restricted
- outside cutoff rotation suddenly free



Result

- heating

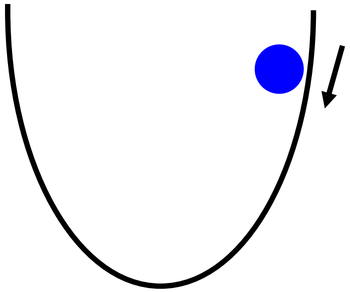
Equilibrium

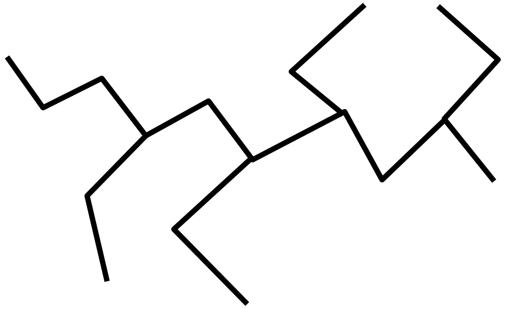
Remember MC story

- system not at equilibrium ? eventually equilibrates

MD

- start in high energy E_{pot}
- E_{pot} converted to E_{kin}





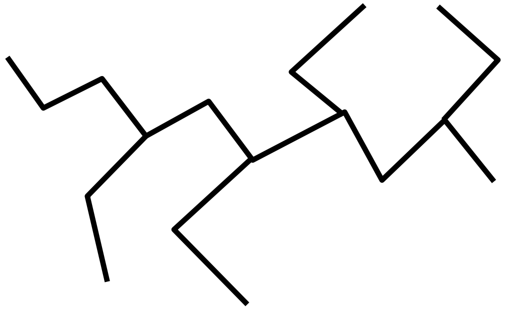
Some high energy conformation

- relaxes
- E_{pot} converted to E_{kin}

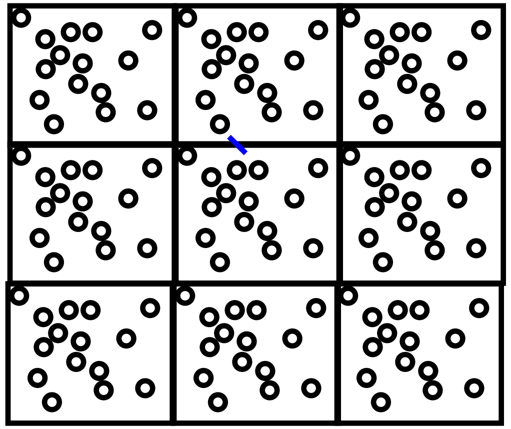
MD system will not

- really find low energy
- known temperature

MD in a closed system



- An isolated molecule should not lose energy
- A repeated box will not lose energy
- Formally system is
  - NVE (constant $N_{particles}$, volume, energy)



Problems

- we want to set the temperature of the system
- we may have noise / heat creating energy

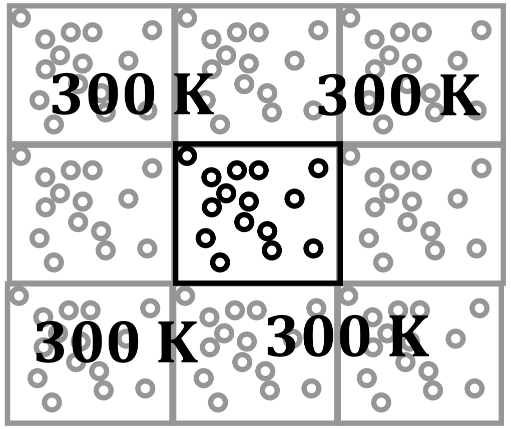
Cure

- thermostat

Bath

imagine infinite bath at desired temperature

- heat will flow in or out
- at equilibrium no flow of heat
    - maybe removal of noise/heat



How to implement ? Many ways

Occasionally:

1. introduce a fake particle desired temperature / collide
2. pick a particle at random / give average *v* for temperature
3. Easy method –weak coupling…

Weak Coupling

Remember temperature* $E_{kin} = \sum_i^N \frac{1}{2} m_i v_i^2 = \frac{3}{2} NkT$

Goal: heat leaves system depending on how wrong temperature is

$$\frac{dT(t)}{dt} = \frac{T_0 - T(t)}{\tau_T}$$

- T_0 is reference temperature
- τ_t is a coupling / relaxation constant
  - τ_t tiny, heat moves fast. τ_t big, …
- to implement this idea ? Multiply velocities

*Slight simplification of formula

Classic reference: Berendsen, HJC, Postma, JPM, van Gunsteren, WF, DiNola, A, Haak, JR, "Molecular dynamics with coupling to an external bath", J. Chem Phys, 81, 3684, (1984)

Implementation of weak coupling

Scale velocities, $v_{new} = \lambda\, v_{old}$ and $\lambda = \left(1 + \frac{\Delta t}{\tau_T}\left(\frac{T_0}{T} - 1\right)\right)^{1/2}$

Intuitively

- Δt (time step) big ? temperature will change more
- what if $T_0 = T$?
- square root ?
  - wrong T reflects a difference in v^2

Importance of heat baths

Does not conserve energy

In principle

- bring a system to equilibrium for temperature

In practice

- avoid damage due to numerical errors / approximations

For a system at equilibrium

- heat bath should do nothing

Does allow artificial tricks

- gently heat a system and watch behaviour
- gently cool a system and "anneal" it (more later)

Extension to other properties

  - analogous reasoning for pressure bath

dynamics versus Monte Carlo

MC	MD
any cost/energy OK	requires continuous $E_{pot}(\boldsymbol{r})$
time usually invalid	gives time scales
most moves OK	physical trajectories
temperature from acceptance/rejection	has explicit E_{kin} and temperature bath
easy to program	can be difficult
both yield a Boltzmann distribution	
both include entropy	