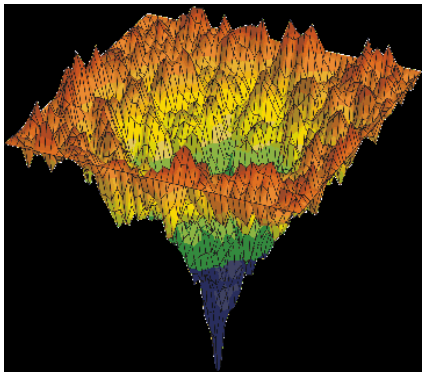
CHEM-E4155
Computational Chemistry I (5op)
2nd part: molecular modelling

Book Chapters 8.1-8.13

Monte Carlo simulations

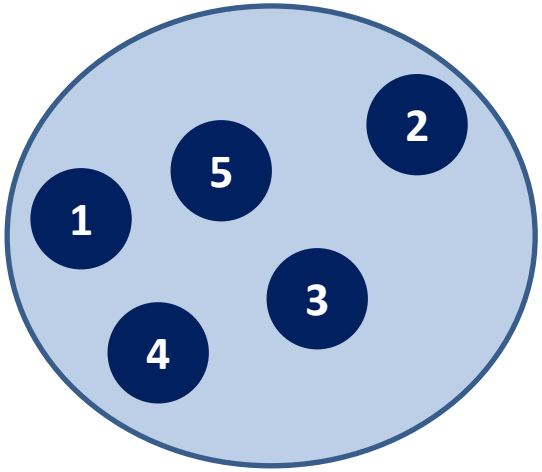
From potential energy surface to molecular modelling and simulations



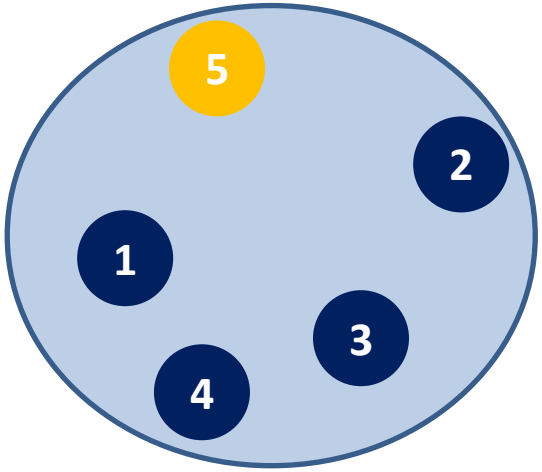
- Global minimum energy configuration
- T>0 also higher internal energies
- Ergodic sampling of the potential energy landscape
- Typical methodology:
  - Minimum energy configuration determination (no dynamics, just potential energy surface)
  - Molecular dynamics (deterministic dynamic sampling of the potential energy surface)
  - Monte Carlo (stochastic sampling of the potential energy surface)

Monte Carlo basics

- Potential energy functional E (function of nuclei positions) -> probability of configuration
- Configurations sampled with some ”random” algorithm (typically random number generator based trial moves) and new configuration accepted based on its energy (Metropolis Monte Carlo)
  - high energy configurations accepted with low probability, low energy configurations with high
  - Boltzmann distribution
  - Average over a large set provides physically measurable property
  - No deterministic dynamics



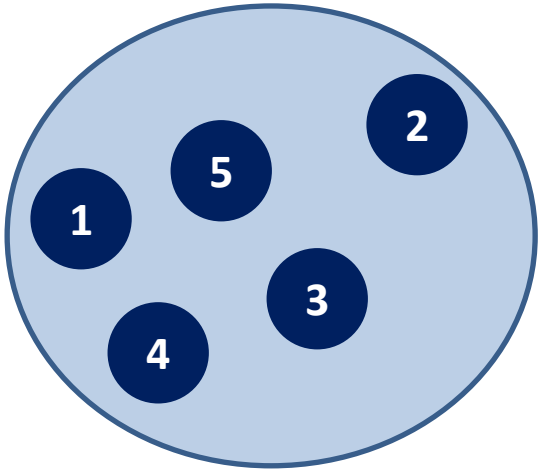
Energy E1



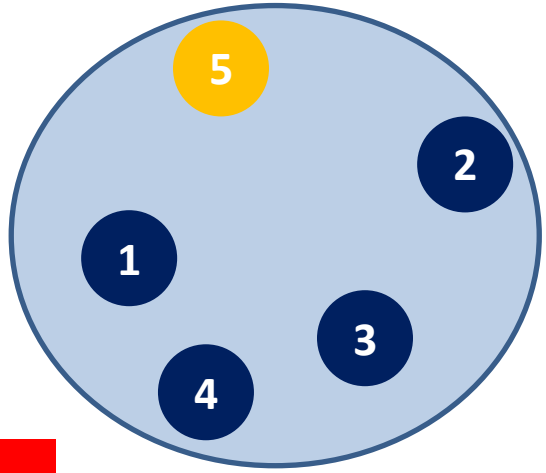
Energy E2

Monte Carlo basics

- Potential energy functional E (function of nuclei positions) -> probability of configuration
- Configurations sampled with some ”random” algorithm (typically random number generator based trial moves) and new configuration accepted based on its energy (Metropolis Monte Carlo)
  - high energy configurations accepted with low probability, low energy configurations with high
  - Boltzmann distribution
  - Average over a large set provides physically measurable property
  - No deterministic dynamics



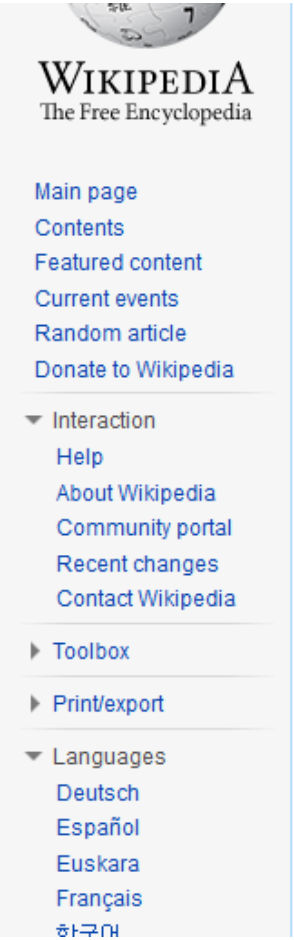
Energy E1



Energy E2

Let’s take a step back to more general “Monte Carlo” and “Monte Carlo simulations”

Monte Carlo



Monte Carlo (disambiguation)

From Wikipedia, the free encyclopedia

Monte Carlo is an administrative area of Monaco.

Monte Carlo or **Montecarlo** may also refer to:

Geography

- Montecarlo, Tuscany, a town in Italy
- Montecarlo (Misiones), a town in Argentina
- Monte Carlo (Macau), a football club in Macau
- Monte Carlo Resort and Casino, a luxury hotel on the Las Vegas Stri
- Monte Carlo (San Marino), a mountain in San Marino

Special events

- Monte Carlo Rally, a rallying event organized by the *Automobile Club*
- Monte-Carlo Masters, a tennis tournament
- Circuit de Monaco, a street circuit laid out on the city streets of Mont

Transportation

- Chevrolet Monte Carlo, an American automobile built by Chevrolet
- Lancia Montecarlo, an Italian automobile
- Monte Carlo (racing car), an open-wheel racing car
- Monte Carlo (yacht), a motor yacht

- Monte Carlo stock, a style of rifle buttstock

Media

- *Monte Carlo* (musical), an 1896 West End musical by Howard Talbot
- *Monte Carlo* (1926 film)
- *Monte Carlo* (1930 film), a 1930 American film
- *The Monte Carlo Story*, a 1957 American film
- *Monte Carlo* (2011 film), distributed by 20th Century Fox
- *Monte Carlo* (song), a 2004 song by The Verve
- Monte Carlo (solitaire), a solitaire card game
- *Monte Carlo* (video game), a 1987 computer game

Science

- Monte Carlo method, a class of computational algorithms
- Monte Carlo integration, a method of numerical integration
- Monte Carlo option model, an option valuation model using Monte Carlo m
- Monte Carlo algorithm, a randomized algorithm

People

- Sophia Montecarlo (born 1986), former contestant on the reality show *Bor*
- Monte Carlo (composer) (born 1883), Danish-born Broadway composer an

See Also

- Monte Cristo (sandwich)

Monte Carlo (in science)

- Wide sense: A simulation which uses random numbers
- Named after: Monte Carlo Casino (randomness in games)
- Name dates to ~1940s
- Pioneers: John von Neumann, Stanislaw Ulam and Nicholas Metropolis working on Manhattan Project (nuclear weapon projects) in the Los Alamos National Laboratory.
- Also called: stochastic simulations

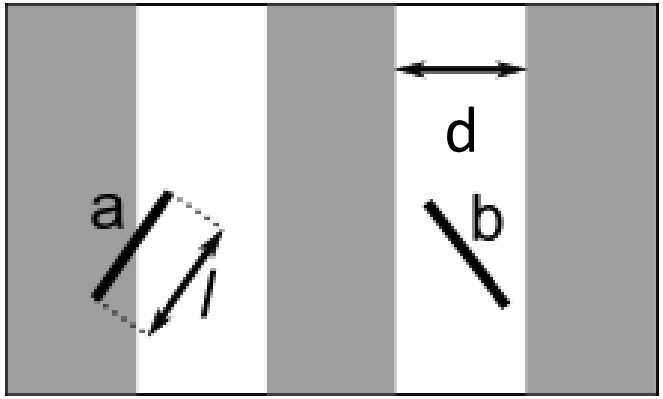
Monte Carlo simulations

- Basic approach (general)
  1. Define what the inputs can be (range of inputs)
  2. Generate inputs in a random way from the range of possible inputs (stochastic choice of input)
  3. Perform a computation (deterministically) based on the input
  4. Collect the results and aggregate them

History: Buffons needle(Georges-Louis Leclerc, Comte de Buffon:1707 - 1788)

- Suppose we have a floor made of parallel strips of wood, each the same width d, and we drop a needle onto the floor. What is the probability that the needle will lie across a line between two strips?

$$P_{hit} = \frac{2l}{\pi d}$$



http://en.wikipedia.org/wiki/Buffon%27s_needle

https://mste.illinois.edu/activity/buffon/

http://www.youtube.com/watch?v=kazgQXaeOHk (see comments!)

Example: Pitfall of stochastic simulations 1

In 1901, Italian mathematician Mario Lazzarini performed the Buffon's needle experiment. Tossing a needle 3408 times, he obtained the well-known estimate 355/113 for π, which is a very accurate value, differing from π by no more than 3×10^{-7}. This is an impressive result, but is something of a cheat, as follows.

Lazzarini chose needles whose length was 5/6 of the width of the strips of wood. In this case, the probability that the needles will cross the lines is 5/3π. Thus if one were to drop *n* needles and get *x* crossings, one would estimate π as

$$\pi \approx 5/3 \cdot n/x.$$

http://en.wikipedia.org/wiki/Buffon%27s_needle

Example: Pitfall of stochastic simulations 2

$$\pi \approx 5/3 \cdot n/x.$$

π is very nearly 355/113; in fact, there is no better rational approximation with fewer than 5 digits in the numerator and denominator. So if one had n and x such that:

$$355/113 = 5/3 \cdot n/x$$

or equivalently,

$$x = 113n/213$$

one would derive an unexpectedly accurate approximation to π, simply because the fraction 355/113 happens to be so close to the correct value. But this is easily arranged. To do this, one

http://en.wikipedia.org/wiki/Buffon%27s_needle

Example: Pitfall of stochastic simulations 3

the correct value. But this is easily arranged. To do this, one should pick n as a multiple of 213, because then $113n/213$ is an integer; one then drops n needles, and hopes for exactly $x = 113n/213$ successes.

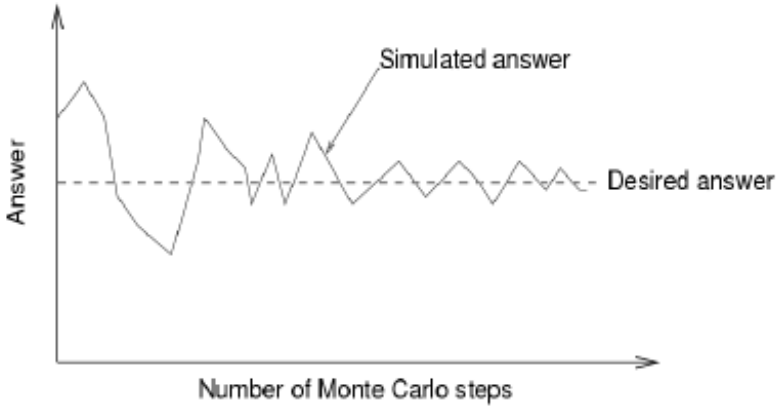
If one drops 213 needles and happens to get 113 successes, then one can triumphantly report an estimate of π accurate to six decimal places. If not, one can just do 213 more trials and hope for a total of 226 successes; if not, just repeat as necessary. Lazzarini performed $3408 = 213 \cdot 16$ trials, making it seem likely that this is the strategy he used to obtain his "estimate".

http://en.wikipedia.org/wiki/Buffon%27s_needle

Example: Pitfall of stochastic simulations: Summary

- This illustrates a potential pitfall in Monte Carlo simulations
- More simply: If the answer you want to get is known, and then do Monte Carlo simulation and look at the intermediate answer after every step, your result will behave something like the picture below
- If you plan the simulation or stop it when you happen to cross the 'right answer' line, you will get the right answer! But doing this is of course utterly wrong.

Issue with aiming for an answer in stochastic simulations!



Pitfalls of stochastic simulations

- In practice, the answer is seldom known in advance (why would one simulate if one knows the answer). But a
- More dangerous (and common) pitfall is that one wishes for a low, or high, value or a value matching with experiments or other data, and stops the simulation at a value.
- **Always decide in advance how many Monte Carlo steps to do!**

From Buffon's needle to random numbers and Monte Carlo simulations

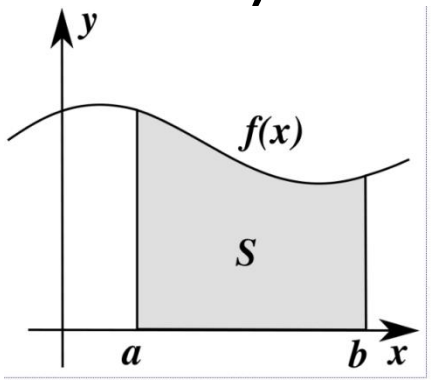
- After Buffon's needle, random numbers have been used in statistics
  - Wide use began in 1940's
- Origins of Monte Carlo simulations obscured because part of Manhattan project
- Two early publications
  - Note on census-taking in Monte Carlo calculations E. Fermi and R.D. Richtmyer 1948. A declassified report by Enrico Fermi. From the Los Alamos Archive.
  - The Monte Carlo Method N. Metropolis and S. Ulam 1949 Journal of the American Statistical Association, 44, 335 (1949)

Monte Carlo simulations

- Next
  - Monte Carlo integration
  - Metropolis Monte Carlo
    - Molecular simulations algorithm

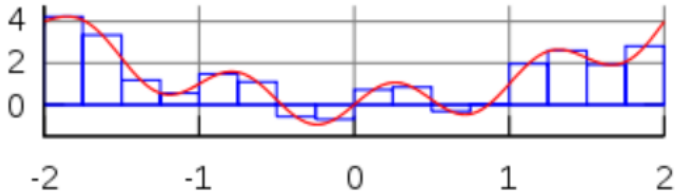
Monte Carlo Integration

- numerical integration using random numbers
- algorithms for the approximate evaluation of definite integrals (usually multidimensional ones)

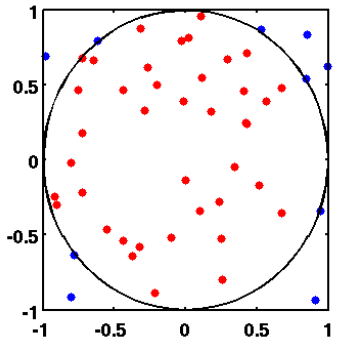


Analytical integration

$$\int_a^b f(x)\,dx \approx (b-a)\,f\left(\frac{a+b}{2}\right).$$



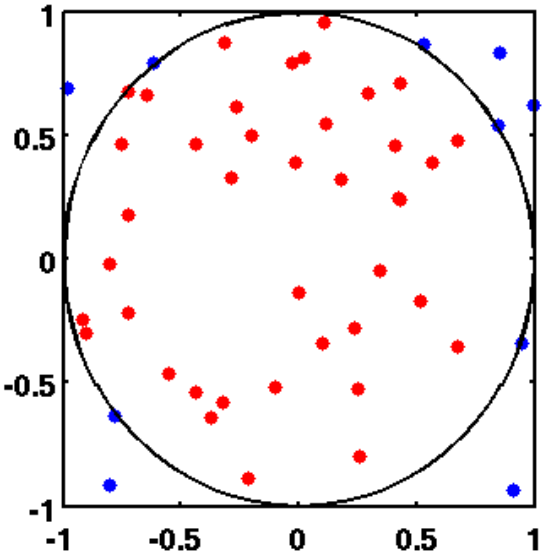
Numerical finite difference integration (here: rectangle rule)



Monte Carlo integration

Monte Carlo integration:

- Consider a circle in a unit square. Given that the circle and the square have a ratio of areas that is $\pi/4$, the value of π can be approximated using a Monte Carlo method
  - Draw a square on the ground, then inscribe a circle within it
  - Uniformly scatter some objects of uniform size (grains of rice or sand) over the square
  - Count the number of objects inside the circle and the total number of objects.
  - The ratio of the two counts is an estimate of the ratio of the two areas, which is $\pi/4$. Multiply the result by 4 to estimate π.



Example of calculating π: http://en.wikipedia.org/wiki/File:Pi_30K.gif

Monte Carlo integration: When to use

- For N dimensional integral m^N function evaluations, where m is number of points needed in each dimension
- Finite difference methods volume $$V^{(M+1)} = \frac{V^{(M)}}{N} \sum_{i_1=1}^{N_1} \sum_{i_2=1}^{N_2} \cdots \sum_{i_M=1}^{N_M} f(\mathbf{x}_i)$$
- Compare: Stochastic sampling estimate $$V^{(M+1)} \approx \frac{V^{(M)}}{N} \sum_{i=1}^{N} f(\mathbf{x}_i)$$
- with increasing numbers of dimensions M, doing the M sums becomes increasingly cumbersome, and eventually using the Monte Carlo approach with only one sum needed will be simpler

To illustrate this, I did the following test. I calculated the volume of a sphere in M dimensions with direct numerical integration (using the midpoint method) and MC integration.

- The number of intervals was 20 in the numerical integration in each dimension, and the number of attempts in the MC simulation was always 10^5.
- This happened to give results of comparable, about ~ 0.5 % accuracy. I timed the result simply with the Unix `time` command.

The results are as follows. The first column gives the number of dimensions M, the next two the numerical execution time, the next two the MC results in the same way, and the last column the correct answer (known analytically). The times are in seconds.

	numerical		MC		
M	time	result	time	result	Correct
2	0.00	3.1524	0.01	3.1435	3.1415
3	0.00	4.1737	0.07	4.1896	4.1887
4	0.00	4.9023	0.08	4.9330	4.9348
5	0.02	5.2381	0.10	5.2787	5.2637
6	0.30	5.1451	0.13	5.1748	5.1677
7	5.02	4.6704	0.15	4.7098	4.7247
8	89.9	3.9595	0.17	4.0479	4.0587
9	1320	3.3998	0.20	3.3191	3.2985

So we see that for $M < 6$ the numerical method is faster, but after that becomes terribly much slower.

The results are as follows. The first column gives the number of dimensions M, the next two the numerical execution time, the next two the MC results in the same way, and the last column the correct answer (known analytically). The times are in seconds.

	numerical		MC		
M	time	result	time	result	Correct
2	0.00	3.1524	0.01	3.1435	3.1415
3	0.00	4.1737	0.07	4.1896	4.1887
4	0.00	4.9023	0.08	4.9330	4.9348
5	0.02	5.2381	0.10	5.2787	5.2637
6	0.30	5.1451	0.13	5.1748	5.1677
7	5.02	4.6704	0.15	4.7098	4.7247
8	89.9	3.9595	0.17	4.0479	4.0587
9	1320	3.3998	0.20	3.3191	3.2985

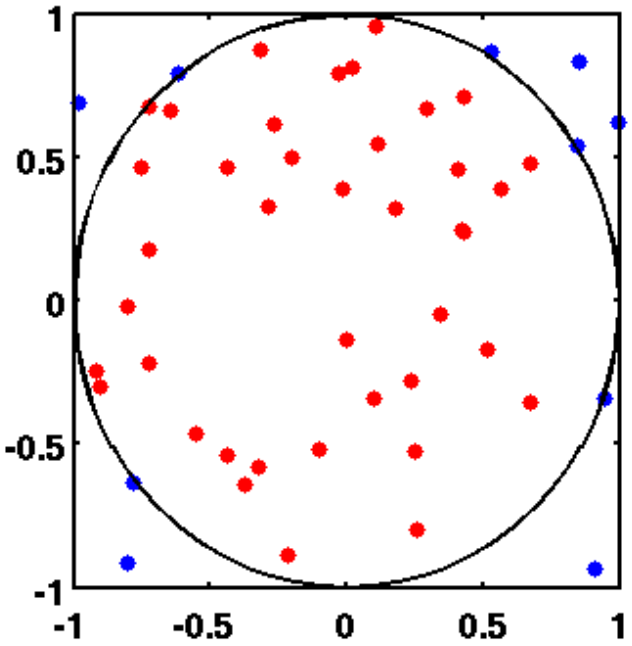
So we see that for $M < 6$ the numerical method is faster, but after that becomes terribly much slower.

- What is most interesting is that the time required by the MC method is not rising almost at all, even though the accuracy stays the same. This is what makes it so interesting for high-dimensional integration!

Basics of Monte Carlo simulations, Kai Nordlund 2006

Monte Carlo integration: continuation

- Traditionally evaluation points distributed uniformly over the integration region
- Importance or weighted sampling possible -> faster convergence



Statistical thermodynamics basis of Metropolis Monte Carlo algoritm

- All microstates are equally probable
  - Consider: dice
- Microstates can correspond to same outcome (multiplicity of a state)
- Higher multiplicity - > higher probability of getting that outcome
  - Probability of getting an outcome i, p_i, depends on how many different ways that outcome can be achieved
  - Consider:
    - 3 dice and getting 1,1,1 vs 1,3,5

Statistical thermodynamics basis of Metropolis Monte Carlo algoritm

- Expectation value of a measurable quantity A

$$\langle A \rangle = \sum_i p_i A_i$$

- Probability distribution that maximizes entropy for molecular conformations (states) i is the Boltzmann distribution

$$p_i = \frac{e^{(-E_i/k_B T)}}{\sum_j e^{(-E_j/k_B T)}}$$

Statistical thermodynamics basis of Metropolis Monte Carlo algoritm

- Boltzmann distribution: $p_i = \frac{e^{(-E_i/k_B T)}}{\sum_j e^{(-E_j/k_B T)}}$
- E_i, E_j energies of state i
- N_i Number of particles in state i
- $\frac{N_i}{N} = p_i$ Fraction of particles in state i, probability of state i

Metropolis Monte Carlo

Computational approach (i.e., algorithm) for generating a set of N configurations X_i, X_j, X_k, … of the system such that $\lim_{N\to\infty} \frac{N_i}{N} = p_i$ for all X_i, X_j, X_k, … where p_i follow a given probability distribution.

In Boltzmann distribution $p_i = \frac{e^{(-E_i/k_B T)}}{\sum_j e^{(-E_j/k_B T)}}$

N_i is the number of particles in state X_i (configurations).

Outcoming configurations match p_i in their observation frequency

Metropolis Monte Carlo Algorithm

1) Pick any configuration X_n

2) Pick a *trial configuration* X_t

3) Compute the acceptance ratio based on probabilities of the configurations $R = \frac{p(X_t)}{p(X_n)}$

4) Pick a random number *u* with value between 0 and 1. Make X_{n+1} = X_t if $u \leq R$

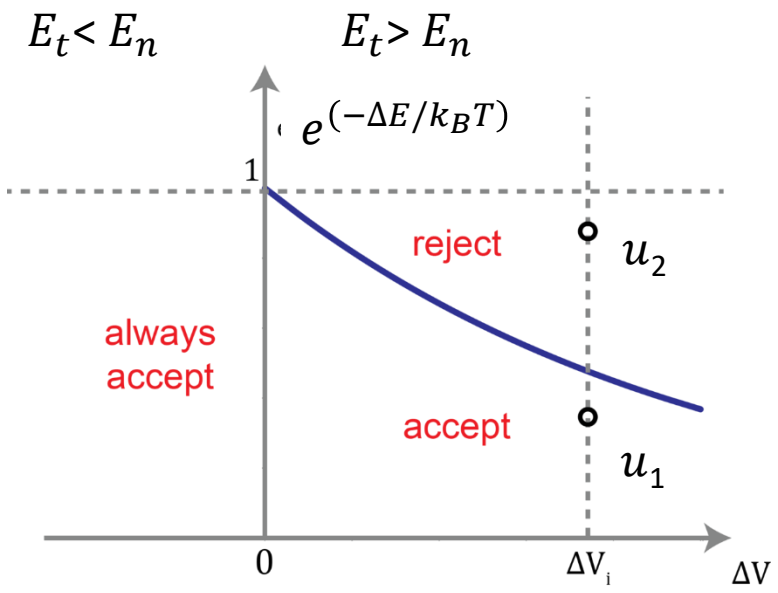
and otherwise X_{n+1} = X_n

5) Goto 2 replacing X_n by X_{n+1}

Repeat N times, where N is a sufficiently large number

Metropolis Monte Carlo Algorithm

1) Pick any configuration X_n
2) Pick a *trial configuration* X_t
3) Compute the acceptance ratio based on probabilities of the configurations $R = \frac{p(X_t)}{p(X_n)}$
4) Pick a random number *u* with value between 0 and 1. Make $X_{n+1} = X_t$ if $u \leq R$
and otherwise $X_{n+1} = X_n$
5) Goto 2 replacing X_n by X_{n+1}
Repeat N times, where N is a sufficiently large number

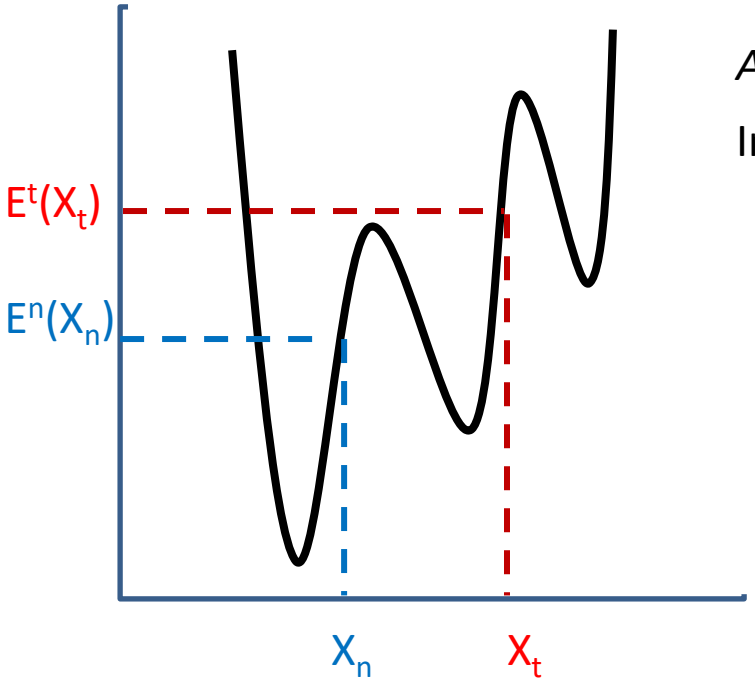


- Expectation value of a measurable quantity A

$\langle A \rangle = \sum_i p_i A_i = \sum \frac{A_{X_n}}{N}$

Because A_{X_n} follow p_i in their frequency,
p_i is the probability of state *i*

Metropolis Monte Carlo compares energies, no forces calculated



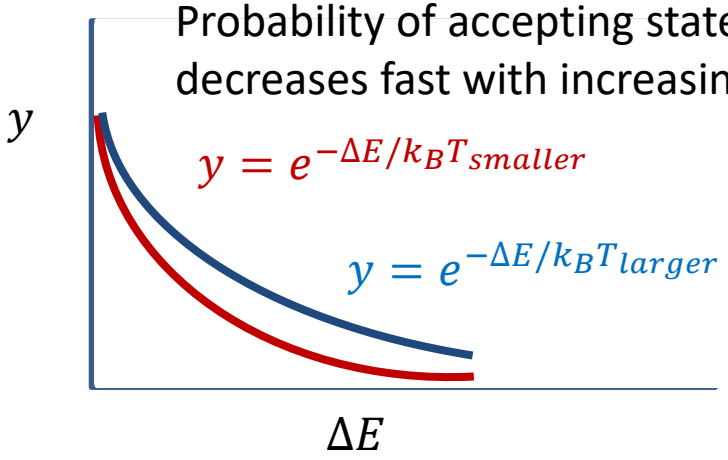
Acceptance ratio $R = \frac{p(X_t)}{p(X_n)}$

In Boltzmann distribution

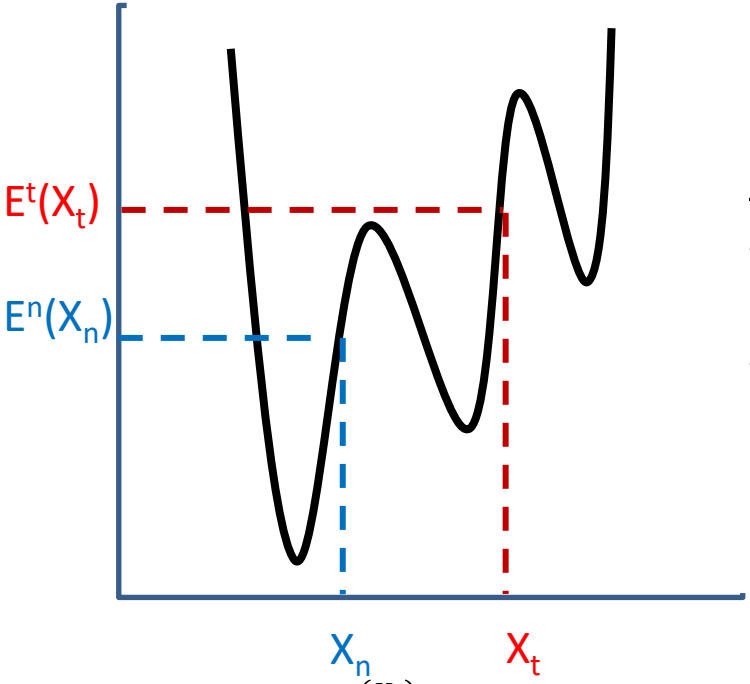
$$p(X_i) = p(E_i) = \frac{e^{(-E_i/k_BT)}}{\sum_j e^{(-E_j/k_BT)}}$$

$$\frac{p(X_t)}{p(X_n)} = \frac{p(E^t)}{p(E^n)} = e^{-(E^t - E^n)/k_BT} = e^{-\Delta E/k_BT}$$

Probability of accepting state decreases fast with increasing ΔE



Metropolis Monte Carlo on the macroion model in exercise



probability $R = \frac{p(X_t)}{p(X_n)}$

In Boltzmann distribution

$$p(X_i) = p(E_i) = \frac{e^{(-E_i/k_BT)}}{\sum_j e^{(-E_j/k_BT)}}$$

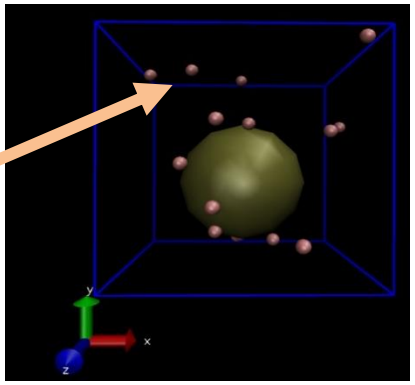
$$\frac{p(X_t)}{p(X_n)} = \frac{p(E^t)}{p(E^n)} = e^{-(E^t - E^n)/k_BT} = e^{-\Delta E/k_BT}$$

Monte Carlo trial move:

An ion moves

in random direction

A distance of L_{max}



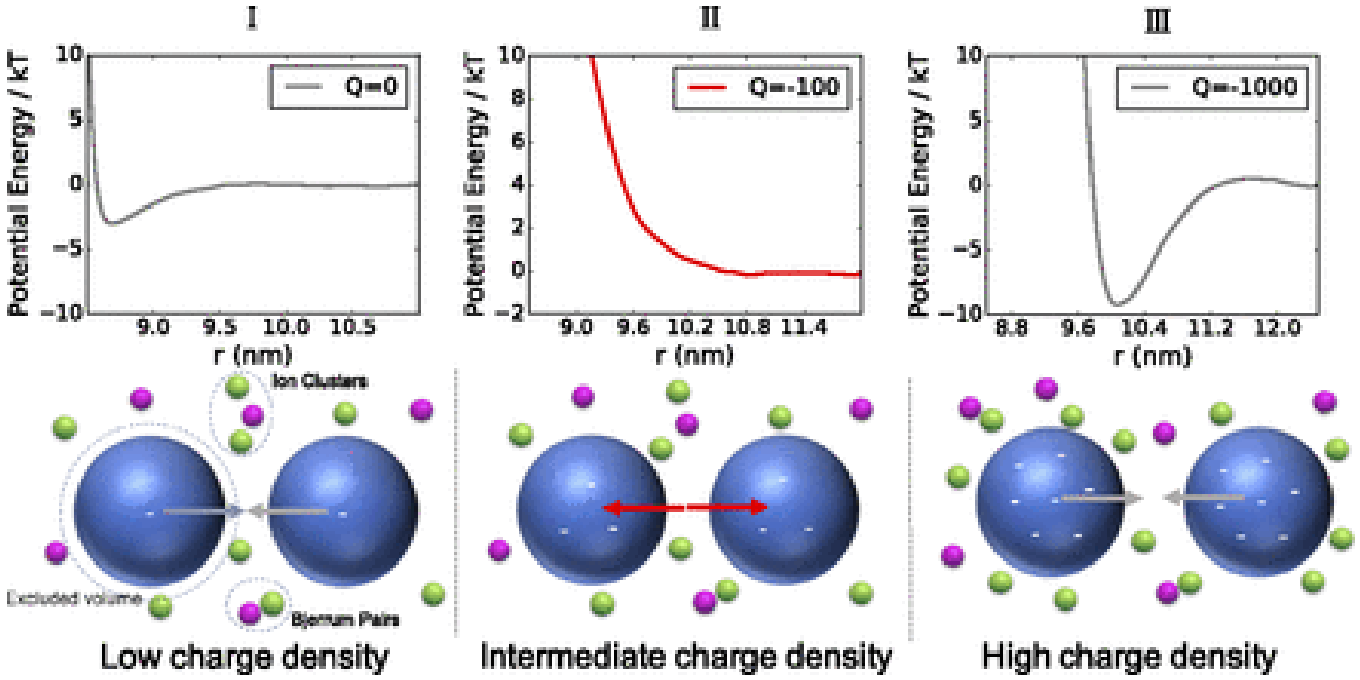
$$E^{ES} = \frac{1}{4\pi\varepsilon}\frac{q_i q_j}{r_{ij}} = \frac{\ell_B q_i q_j}{r_{ij}}$$

$$E^{HS}_{ion,ion} = \begin{cases} \infty, & r_{ij} < d \\ 0, & r_{ij} \geq d \end{cases}$$

$$E^{HS}_{p,ion} = \begin{cases} \infty, r_{ij} < \dfrac{(d+d_0)}{2} \\ 0, r_{ij} \geq \dfrac{(d+d_0)}{2} \end{cases}$$

Research example: macroions in salt

PNAS November 7, 2017 114 (45) 11838-11843



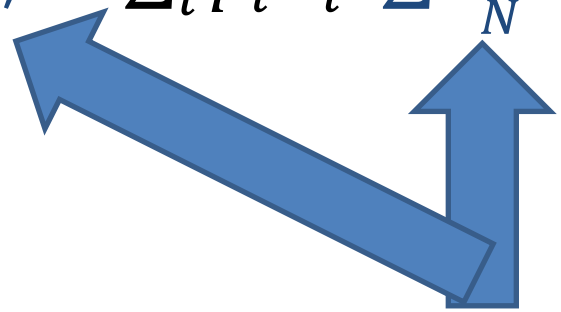
In regime I, where the macroions have zero or low charge densities, their interaction is attractive due to depletion interactions mediated by the clustering of ions. In regime II, where the charge is sufficiently large, a repulsion that is stronger and longer range than the double-layer force predicted by the DLVO theory rises. In regime III, where the macroions have very high charge densities resulting in strong counterion condensation, a strong short-range repulsion and a deep long-range attractive well rises. van der Waals interactions between macroions are not included in the model (purely electrostatics).

Metropolis Monte Carlo Algorithm

If configurations X_n come from Metropolis algorithm (previous slide)

- Expectation value of a measurable quantity

$$\langle A \rangle = \sum_i p_i A_i = \sum \frac{A_{X_n}}{N}$$



Because A_{X_n} follow p_i in their frequency N configurations

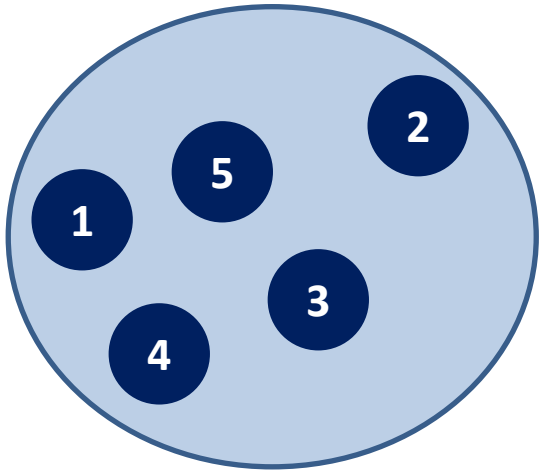
Key to calculating a physical quantity from Metropolis algorithm based simulation

Metropolis Monte Carlo Algorithm

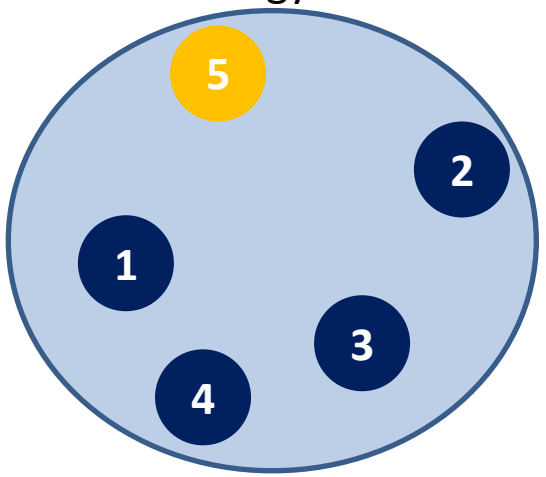
- Note that acceptance ratio (acceptance probability) of a configuration X_t is
- $$P_{X_n,X_t} = \begin{cases} R = \frac{p(X_t)}{p(X_n)}, \text{when } p(X_t) < p(X_n) \\ 1, \text{otherwise} \end{cases}$$
- Most typical variable measuring the state of the system (and probability) is energy E in molecular simulations but this is not necessary

Metropolis Monte Carlo moves

- Consequent states must be uncorrelated
- Trial move can be practically anything as long as it is reversible and probability of reverse action is equal



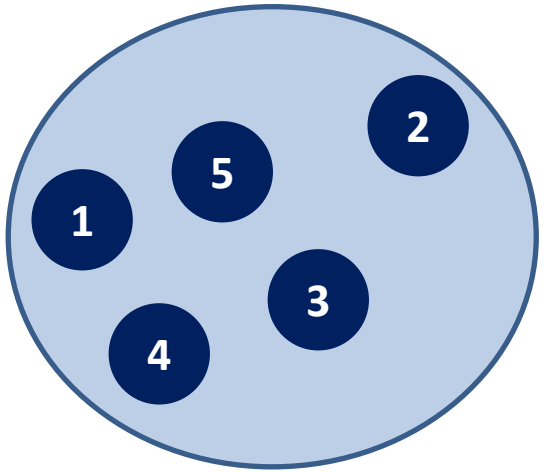
Energy E1



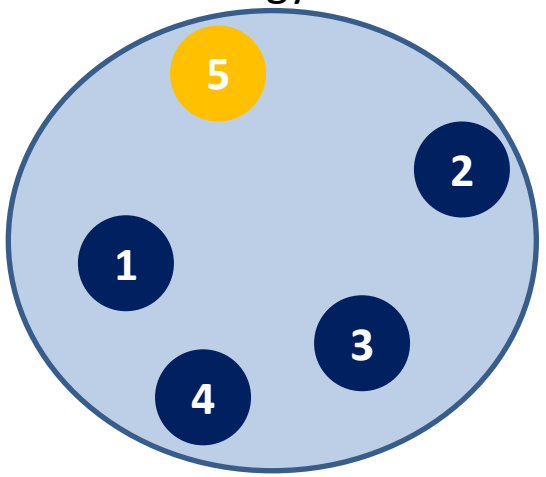
Energy E2

Metropolis Monte Carlo moves

- Common examples
  - Pick a random atom, move a random distance shorter than L_{max} in random direction
  - Switch the positions of two randomly chosen particles
  - Rotation of a randomly chosen molecule or atom pair (random angle)



Energy E1



Energy E2

Monte Carlo

- Thermodynamic quantities, conformation properties as ensemble average using numerical integration, note density can be included by N (numerical integration over ensemble)
- M number of configurations

$$< A >= \int \int d\boldsymbol{p}^N d\boldsymbol{r}^N A(\boldsymbol{p}^N, \boldsymbol{r}^N) \rho(\boldsymbol{p}^N, \boldsymbol{r}^N)$$

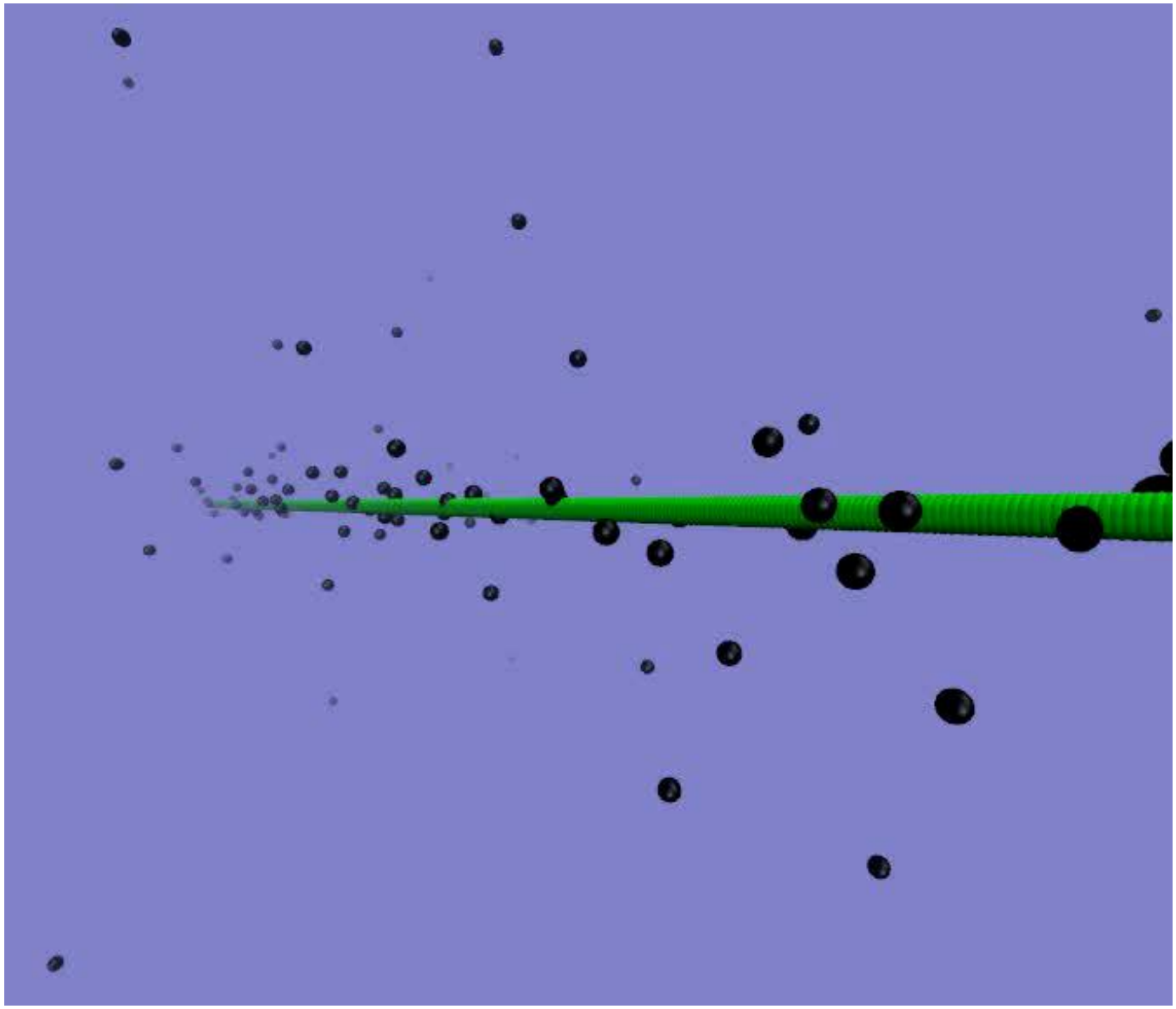
$$< A >= \frac{1}{M} \sum_{i=1}^{M} A(\boldsymbol{r}^N)$$

Differences between molecular dynamics (MD) and Monte Carlo (MC)

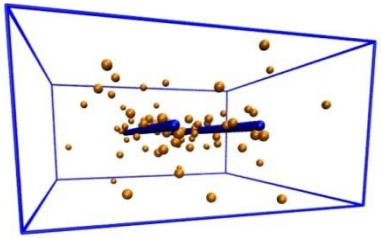
- MD provides deterministic time dependence information
- MC has no temporal relationship between consequent configurations
- With MD possible to predict any future or past state knowing current state
- MC next state acceptance depends on only one prior state (prior state energy) (Metropolis algorithm)

Differences between molecular dynamics (MD) and Monte Carlo (MC)

- MD has kinetic energy explicitly present as velocities
- MC total energy in the potential energy function
- MD inherently in microcanonical (NVE) ensemble (without pressure or thermostat algorithmic corrections to ensemble)
- MC inherently in canonical (NVT) ensemble

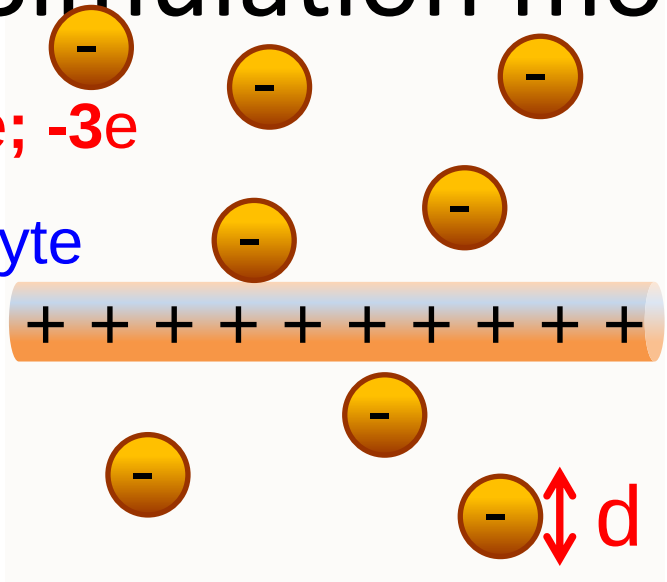


Simulation model



counterion
q= **-1**e; **-2**e; **-3**e

polyelectrolyte
$\tau=Q_p/L$



$$U^{ES} = \frac{1}{4\pi\varepsilon}\frac{q_i q_j}{r_{ij}} = \frac{\ell_B q_i q_j}{r_{ij}}$$

$$U^{HS}_{ion,ion} = \begin{cases} \infty, & r_{ij} < d \\ 0, & r_{ij} \geq d \end{cases}$$

$$U^{HS}_{p,ion} = \begin{cases} \infty, r_{ij} < \dfrac{(d+d_0)}{2} \\ 0, r_{ij} \geq \dfrac{(d+d_0)}{2} \end{cases}$$

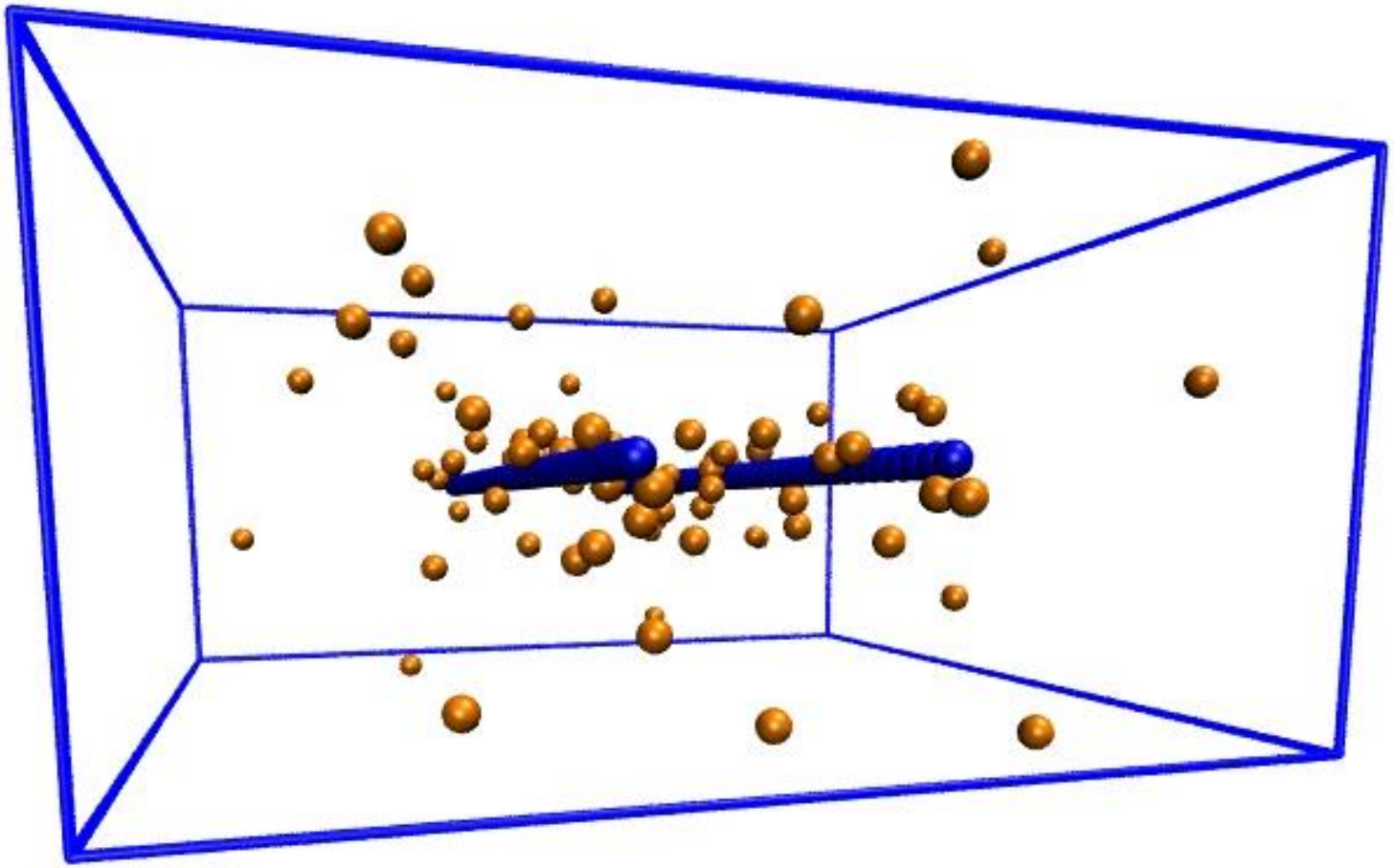
Model: Monte Carlo (Metropolis), $p=e^{-U/kT}$
Ewald summation

Polyelectrolyte: hard rod, discrete axial charge, d_0 varies, τ varies

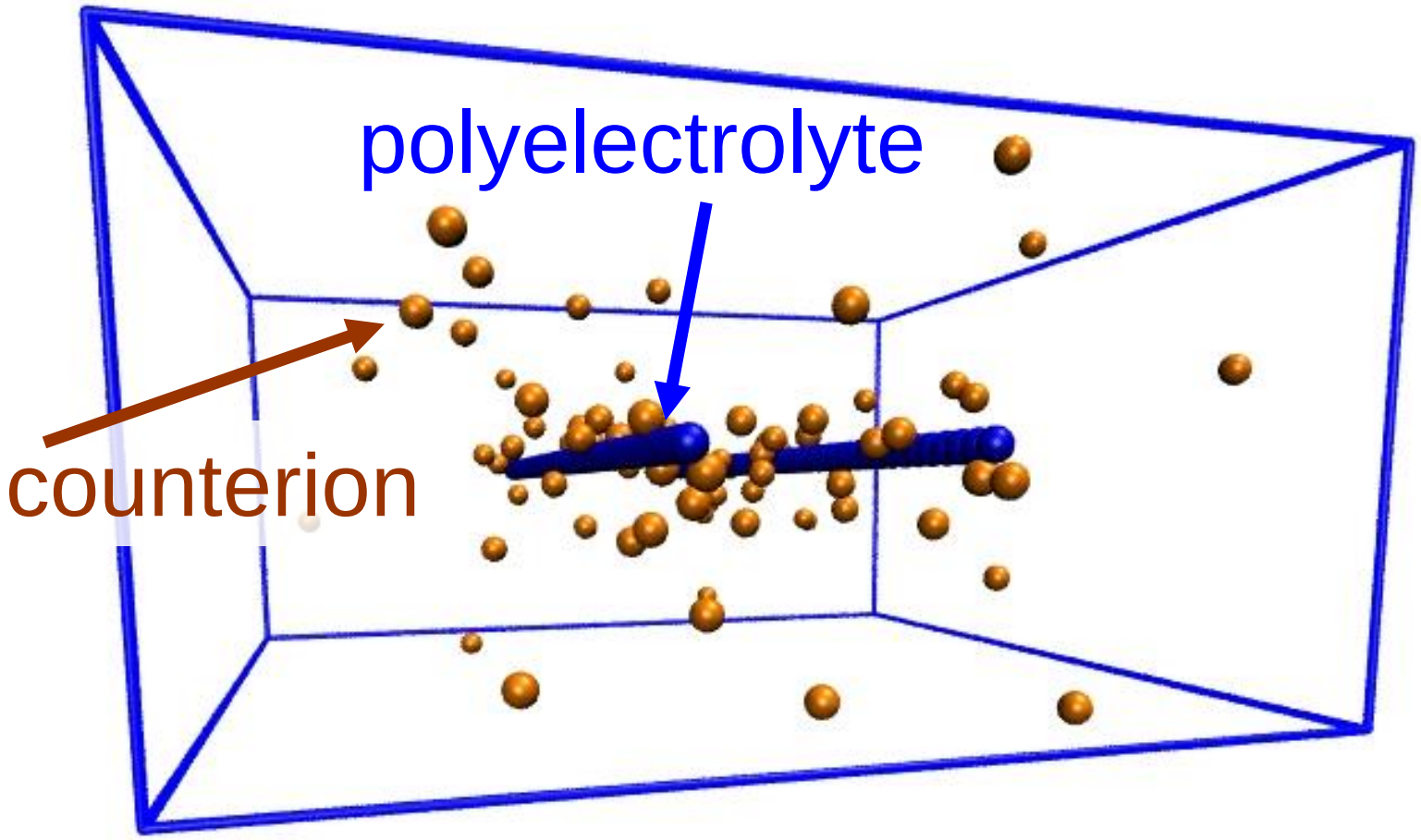
Ions: hard charged beads, d varies, q varies

Bjerrum length $\ell_B=e^2/4\pi\varepsilon kT$ (=0.71nm at 25°C,water)

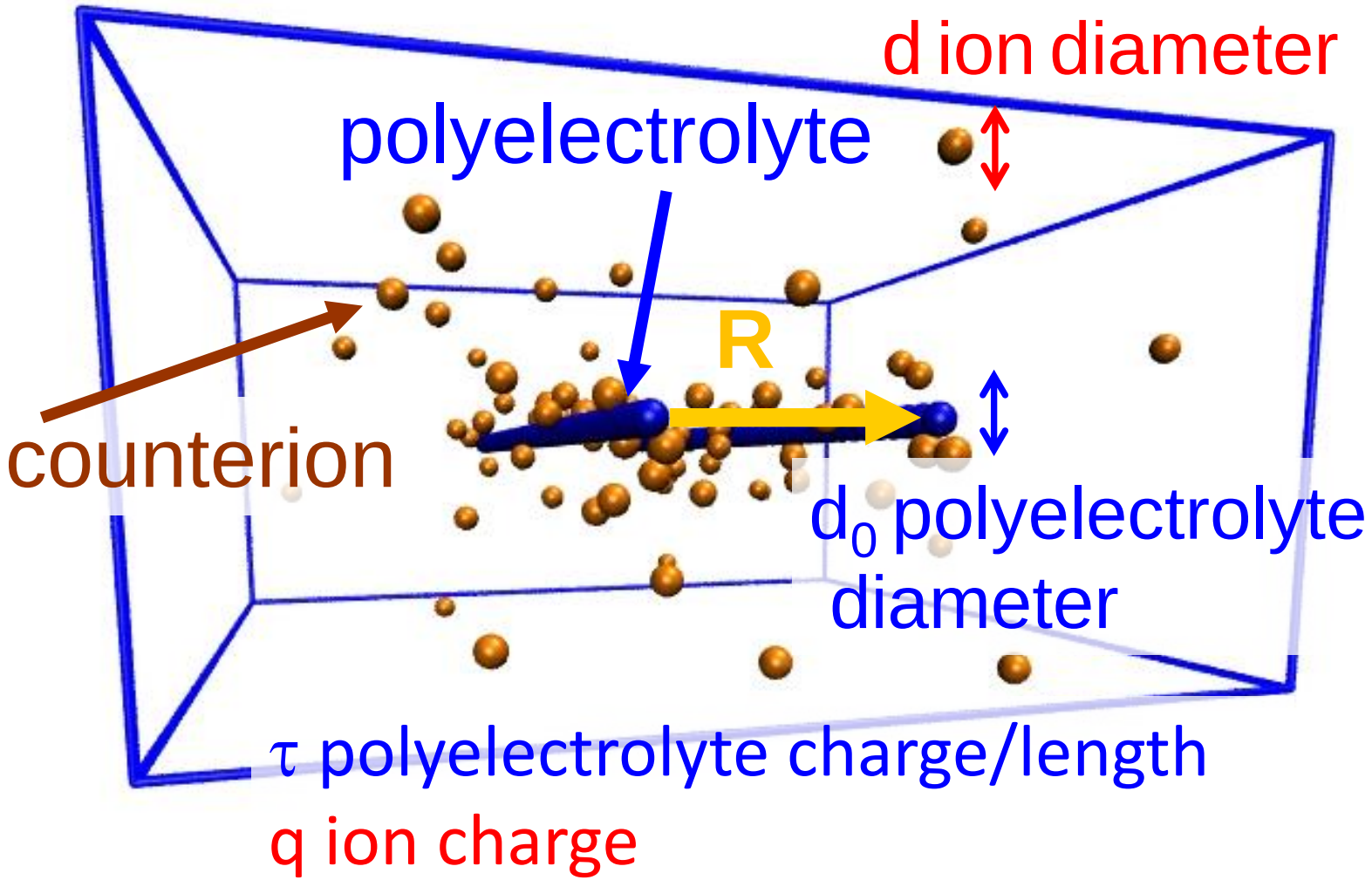
Simulation model

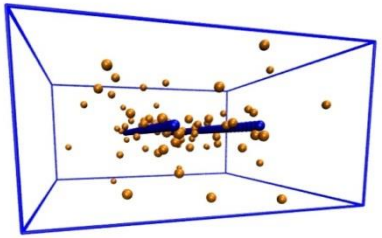


Simulation model

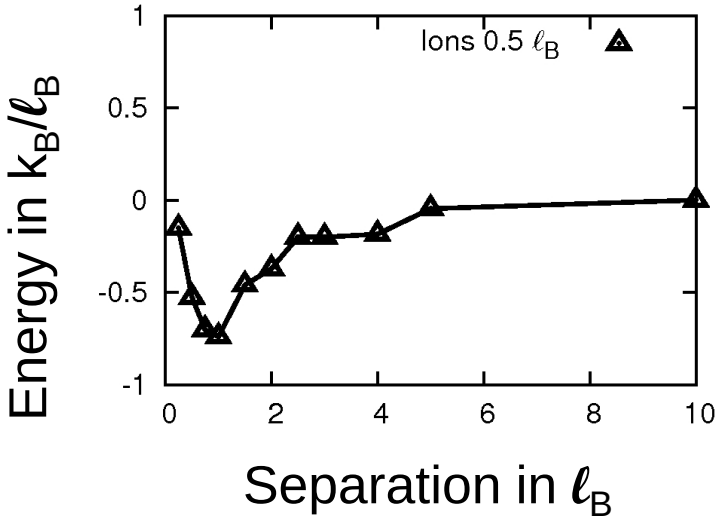


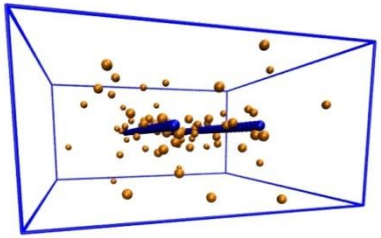
Simulation model





Like charge attraction





Ions: $q = -2e$

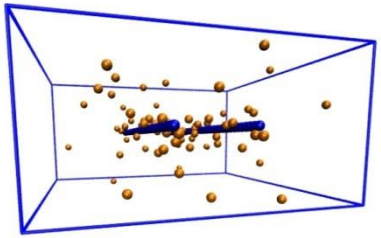
$d = 0.5\ell_b$

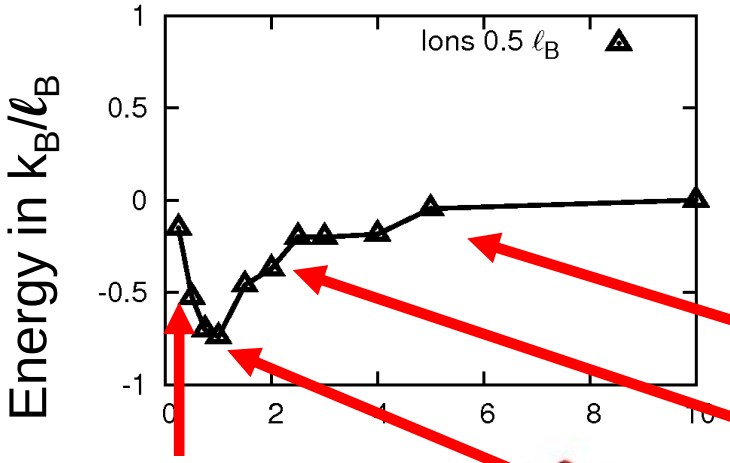
Polyelectrolyte: $\tau = e/\ell_b$

$d_0 = 0.25\ell_b$

➔ Ion coupling $\Gamma = 3.3$

Like charge attraction





Ions:

Polyelectrolyte:

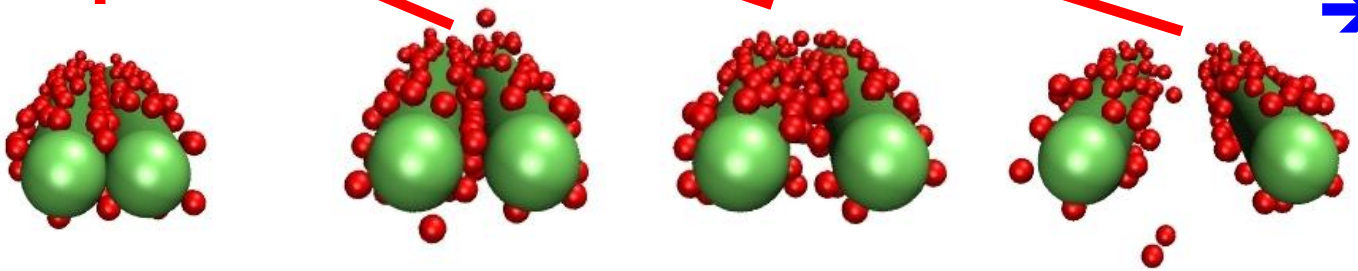
$q = -2e$

$d = 0.5\ell_b$

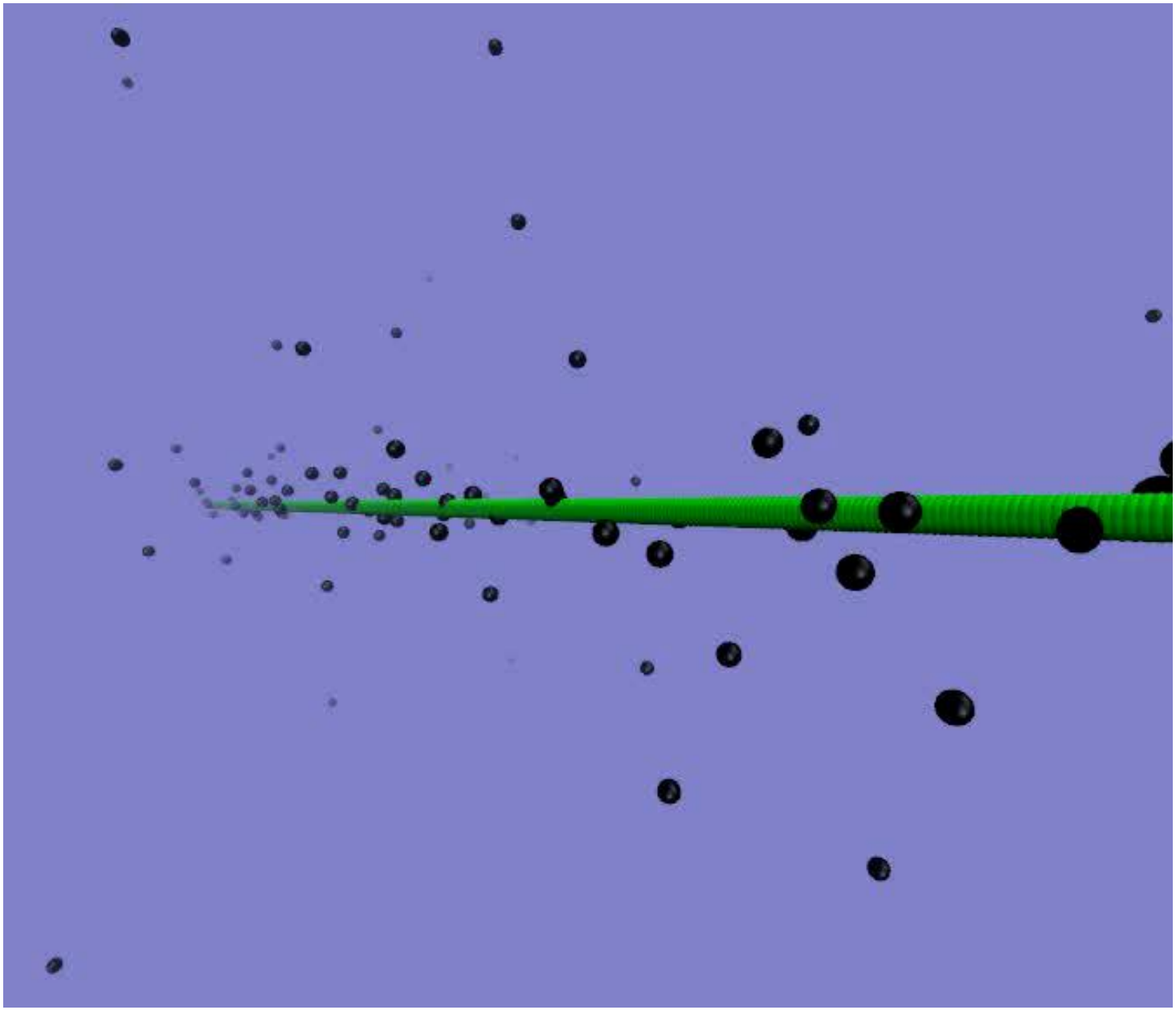
$\tau = e/\ell_b$

$d_0 = 0.25\ell_b$

➔ $\Gamma = 3.3$



<u>Like charge attraction</u>:
Correlation among counterions
(Ions locate between polymers!)



Monte Carlo in different ensembles

- Typically Monte Carlo (Metropolis) samples from NVT ensemble
- NPT ensemble: simulation box size must change
  - Combine random displacements of particles with random box volume changes
  - New volume $V_{\text{new}} = V_{\text{old}} + \delta V_{\max}(2\xi - 1)$
  - When volume changes, in principle, interaction energy must be recalculated for the whole system

Monte Carlo in NPT

Simple potentials trick-of-trade for recalculating the energy when volume changes: Scaled coordinates

s_{ij} scaled coordinate; $s_{ij} = L_{old}\, r_{ij}$ Here Lennard-Jones

$$\mathcal{V}_{\text{old}}(\mathbf{r}^N) = 4\varepsilon \sum_{i=1}^{N} \sum_{j=i+1}^{N} \left(\frac{\sigma}{L_{\text{old}} s_{ij}} \right)^{12} - 4\varepsilon \sum_{i=1}^{N} \sum_{j=i+1}^{N} \left(\frac{\sigma}{L_{\text{old}} s_{ij}} \right)^{6}$$

$$\mathcal{V}_{\text{new}}(\mathbf{r}^N) = 4\varepsilon \sum_{i=1}^{N} \sum_{j=i+1}^{N} \left(\frac{\sigma}{L_{\text{new}} s_{ij}} \right)^{12} - 4\varepsilon \sum_{i=1}^{N} \sum_{j=i+1}^{N} \left(\frac{\sigma}{L_{\text{new}} s_{ij}} \right)^{6}$$

$$\mathcal{V}_{\text{new}}(\mathbf{r}^N) = \mathcal{V}_{\text{old}}(12) \left\{ \frac{L_{\text{old}}}{L_{\text{new}}} \right\}^{12} + \mathcal{V}_{\text{old}}(6) \left\{ \frac{L_{\text{old}}}{L_{\text{new}}} \right\}^{6}$$

Change in the energy due to box size change:

$$\Delta\mathcal{V}(\mathbf{r}^N) = \mathcal{V}_{\text{old}}(12) \left\{ \frac{L_{\text{old}}}{L_{\text{new}}} - 1 \right\}^{12} + \mathcal{V}_{\text{old}}(6) \left\{ \frac{L_{\text{old}}}{L_{\text{new}}} - 1 \right\}^{6}$$

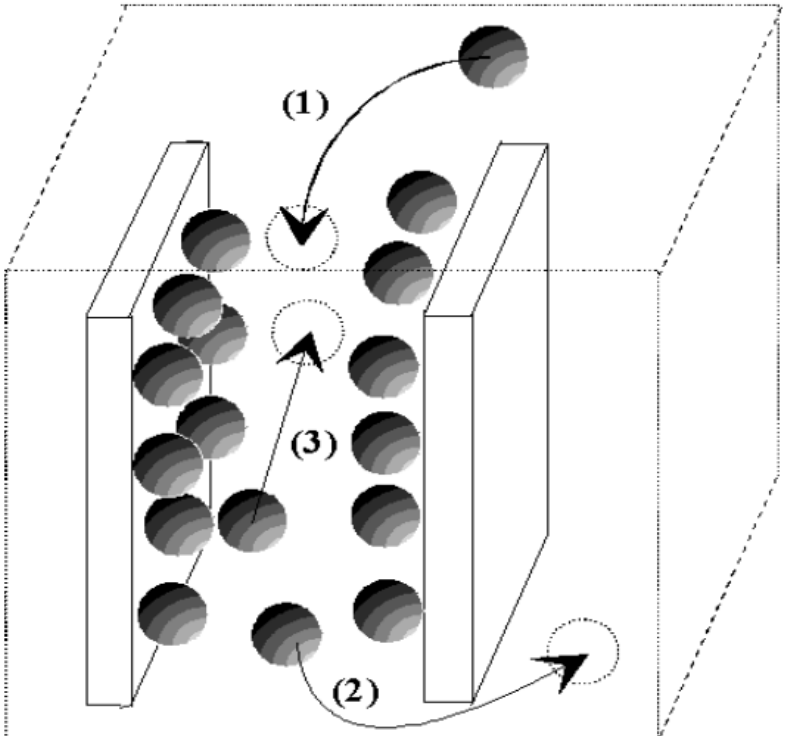
Monte Carlo in NPT

- Long-range interactions and their treatment (cut-offs, …) problematic in NPT
- Simple scaling does not work on molecules / interactions more complicated than distance-based
  - Total energy must be recalculated with simulation box volume change: Computationally expensive
- Criterion for accepting or rejecting a new configuration

$$\Delta H(\mathbf{r}^N) = \mathscr{V}_{\text{new}}(\mathbf{r}^N) - \mathscr{V}_{\text{old}}(\mathbf{r}^N) + P(V_{\text{new}} - V_{\text{old}}) - Nk_{\text{B}}T \ln\left(\frac{V_{\text{new}}}{V_{\text{old}}}\right)$$

Grand Canonical Monte Carlo Simulations

- Key feature: Number of particles may change during simulation
- Three basic moves in Grand Canonical Monte Carlo simulation
  - A particle is created (1)
  - A particle is destroyed (2)
  - A particle is displaced (regular Metropolis move) (3)
- Probability of creating a particle should be equal to particle being destroyed !



Grand Canonical Monte Carlo Simulations

To determine whether destruction move is accepted, calculate:

$$\Delta D = \frac{[\mathcal{V}_{\text{new}}(\mathbf{r}^N) - \mathcal{V}_{\text{old}}(\mathbf{r}^N)]}{k_B T} - \ln\left(\frac{N}{zV}\right)$$

μ Chemical potential
Λ de Broglie wavelength
z activity

$$\mu = k_b T ln \Lambda^3 z$$
$$\Lambda = \sqrt{h^2/2\pi m k_b T}$$

To determine whether creation move is accepted, calculate:

$$\Delta C = \frac{[\mathcal{V}_{\text{new}}(\mathbf{r}^N) - \mathcal{V}_{\text{old}}(\mathbf{r}^N)]}{k_B T} - \ln\left(\frac{zV}{N+1}\right)$$

- If ΔD / ΔC negative, move accepted, otherwise

$$\exp(-\Delta D/k_B T) \text{ or } \exp(-\Delta C/k_B T)$$

is calculated and compared with a random number (regular probability of acceptance comparison)

Grand Canonical Monte Carlo simulations

- Creation / destruction steps in dense systems often fail due to large energy penalty of adding / removing one particle from the (dense) system
- Step-wise fading in / out of a particle
- Configurational bias Monte Carlo
- Example: http://terpconnect.umd.edu/~denesyuk/polymer_images.html

Monte Carlo simulations: vocabulary 1

- Metropolis Monte Carlo
  - A simulation algorithm, central to which is the formula which determines whether a process should happen or not. Originally used for simulating atom systems in an NVT thermodynamic ensemble in physical sciences, but nowadays generalized to many other problems e.g. process design, process optimization, economics, biology, finance and business
- Markov Chain Monte Carlo, MCMC
  - A version of Metropolis Monte Carlo in which a sequence of points in some known distribution is generated.
- Simulated annealing
  - The Metropolis MC idea generalized to optimization, i.e. finding minima or maxima in a system.
  - Can be used in a very wide range of problems
- Thermodynamic Monte Carlo
  - Monte Carlo when used to determine thermodynamic properties, usually of atomic systems

Adapted from: Basics of Monte Carlo simulations, Kai Nordlund 2006

Monte Carlo simulations: vocabulary 2

- Lattice Monte Carlo, LMC
  - MC used on a lattice. Used to distinguish MC done on crystal lattices from those done in a random/amorphous medium
- Kinetic Monte Carlo, KMC
  - MC used to simulate activated processes, i.e. processes which occur with an exponential probability $e^{-Ea/kT}$, for example migration of defects in a solid. KMC can be done free of lattice (non-directional system), or on a lattice. In the latter case one could also talk about lattice kinetic MC.
- Variational MC, VMC & Diffusion MC, DMC
  - These terms are nowadays used often to signify a variety of electronic structure calculations where MC techniques are used to obtain the ground state electron configuration (the terms are wider in principle). Sometimes not mentioned that one deals with electronic structure, which may lead to confusion (“diffusion MC” obviously could mean many other things as well.

Adapted from: Basics of Monte Carlo simulations, Kai Nordlund 2006

Monte Carlo simulations: vocabulary 3

- Quantum Monte Carlo, QMC
  - Used in many different contexts
  - Electronic structure calculation methods VMC and DMC are often called QMC
  - Quantum mechanical simulations of spin systems (which does not necessarily relate to electronic structure-determining calculations in any direct way) are also called QMC...

- Summary: One should define clearly which Monte Carlo method is used

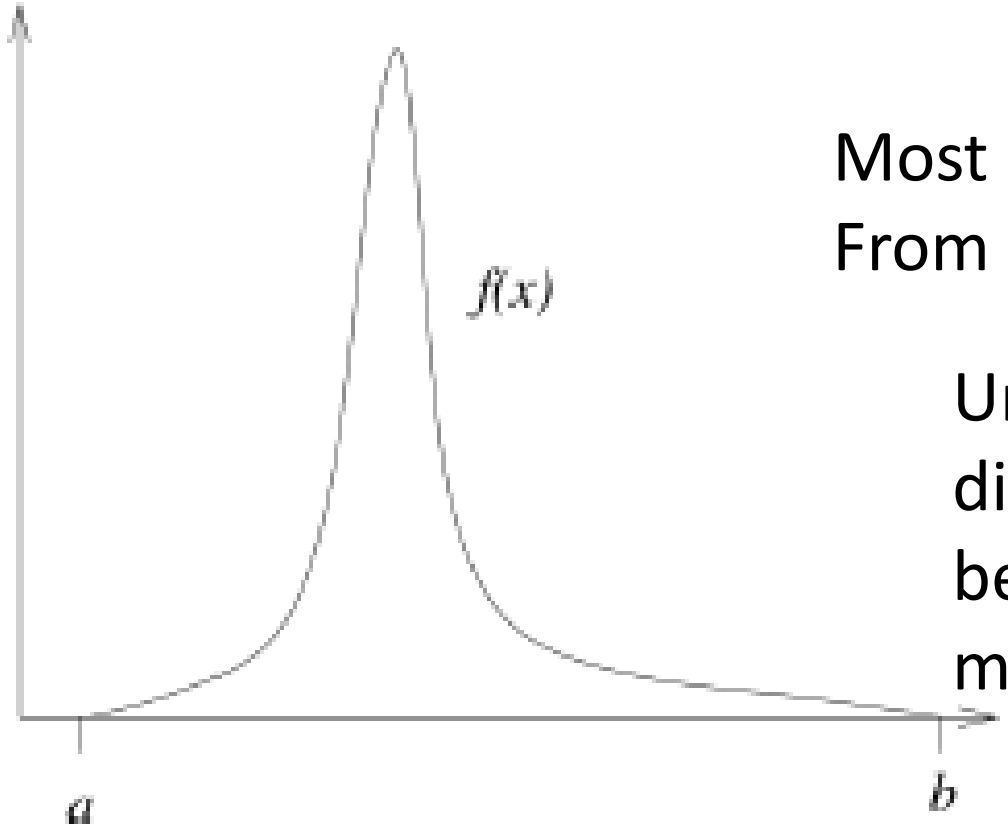
Monte Carlo simulations

- Next
  - Advanced sampling methods
  - Weighted importance sampling
  - Stratification

Importance sampling

- One approach to improving the MC accuracy is reducing the variance σ^2 in the data.
- σ^2 for any non-constant data distribution goes towards some finite, non-zero value when N approaches infinity
- error goes to 0 with increasing N.
- MC error is proportional to σ/sqrt(N) (more on this later)
  - if variance σ^2 goes down, the error will also go down for the same N

Importance sampling



Most contribution to integral
From peak

Uniformly randomly
distributed points
between a and b
mostly outside the peak

Importance sampling: transform f(x) into another,
flatter function which is then Monte Carlo integrated.
Required: back-transformation to give the original integral

Importance sampling: f(x)/g(x) flatter function than f(x)

$$I = \int_a^b f(x)dx = \int_a^b \frac{f(x)}{g(x)} g(x)dx = \int_a^b \frac{f(x)}{g(x)} dG(x)$$

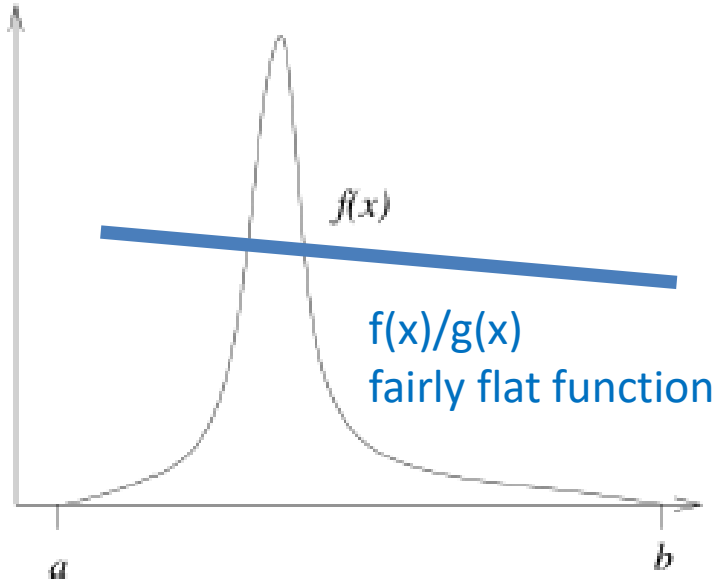
$$G(x) = \int_a^x g(x)dx$$

a variable change $r = G(x)$

$$I = \int_{G(a)}^{G(b)} \frac{f(G^{-1}(r))}{g(G^{-1}(r))} dr$$

Monte Carlo Integration

$$I = \frac{1}{N} \sum_{i=1}^{N} \frac{f(G^{-1}(r_i))}{g(G^{-1}(r_i))}$$



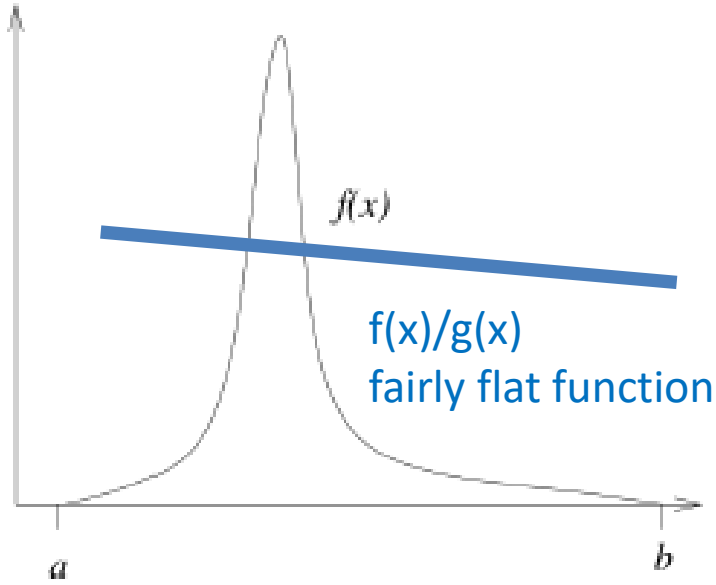
Importance sampling: f(x)/g(x) flatter function than f(x)

Monte Carlo Integration

$$I = \frac{1}{N}\sum_{i=1}^{N}\frac{f(G^{-1}(r_i))}{g(G^{-1}(r_i))}$$

- This requires G^{-1}
- Alternative: generate by any means random numbers $x_i^{(g)}$ distributed as g(x):

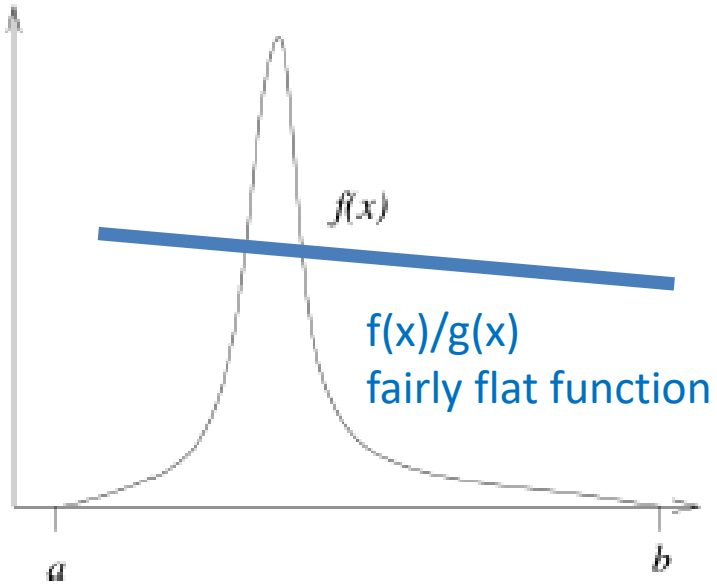
$$I = \frac{1}{N}\sum_{i=1}^{N}\frac{f(x_i^{(g)}))}{g(x_i^{(g)})}$$



Importance sampling

- Reduces misses a lot
- Decreases variance
- -> decreases error for same N

- Practical consequences
  - Computationally heavier
  - Reduces the number of steps required to get close to right answer

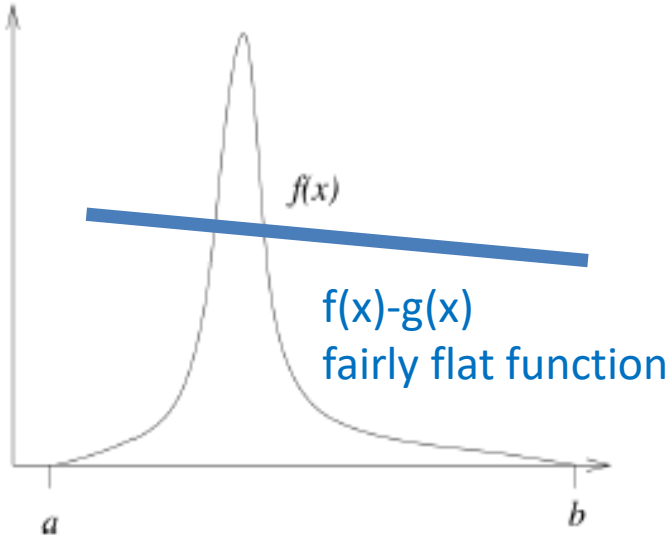


Control variates method

- Same idea as with importance sampling (make the function flatter) but now with subtraction instead of division

$$I = \int_a^b = \int_a^b (f(x) - g(x))dx + \int_a^b g(x)dx$$

- Advantages to importance sampling
  - g(x) can be zero or negative
  - No need for g(x) distributed random numbers



Stratification sampling

- Importance sampling and control variates methods are good ways to improve on MC integration, but require that the function form has to be known. However, often it is not
- Sampling of subregions (subpopulations) separately and independently