ADVANCED TECHNIQUES (MC/MD)

A (seemingly) random selection.

Daan Frenkel
U. Cambridge
13/01/2017

But first: beyond Newtonian MD

1. Langevin dynamics
2. Brownian dynamics
3. Stokesian dynamics
4. Dissipative particle dynamics
5. Etc. etc.

WHY?

1. Can be used to simulate molecular motion in a viscous medium, *without solving the equations of motion for the solvent particles.*
2. Can be used as a *thermostat*.

Friction force

Conservative force

First, consider n ion with frictio alone:

$$\xcancel{m\dot{\vec{v}} = -\gamma\vec{v}(t') - \nabla U \ ,}$$

After a short while, all particles will stop moving, due to the fricti

"random" force

Better:

$$m\dot{\vec{v}} = -\gamma\vec{v}(t') - \nabla U + \vec{\zeta}(t) \ ,$$

There is a relation between the correlation function of the random force and the friction coefficient:

$$< \zeta_x(0)\zeta_x(t) >= 2kT\gamma\delta(t)$$

The derivation is straightforward, but beyond the scope of this lecture.

The KEY point is that the friction force and the random force ARE RELATED.

Limiting case of Langevin dynamics:

No inertial effects (m=0)

$$m\dot{\vec{v}} = -\gamma\vec{v}(t') - \nabla U + \vec{\zeta}(t) \ ,$$

Becomes:

$$0 = -\gamma\vec{v}(t') - \nabla U + \vec{\zeta}(t) \ ,$$

"Brownian Dynamics"

(But still the friction force and the random force are related)

What is missing in Langevin dynamics and Brownian dynamics?

1. Momentum conservation
2. Hydrodynamics

(1 and 2 are not independent).

Is this serious?

Not always: it depends on the time scales.

Momentum "diffuses" away in a time L^2/ν. After that time, a "Brownian" picture is OK.

However: hydrodynamics makes that the friction constant depends on the positions of all particles (and so do the random forces…).

Momentum conserving, coarse-grained schemes:

- Dissipative particle dynamics
- Stochastic Rotation Dynamics

These schemes represent the solvent explicitly (i.e. as particles), but in a highly simplified way.

ADVANCED MC SAMPLING

Is the rejection of Monte Carlo trial moves wasteful?

Conventional MC performs a random walk in configuration space, such that the number of times that each point is visited, is proportional to its Boltzmann weight.

$$n(\mathbf{r}^N) = c\,\mathrm{exp}[-\beta\mathcal{U}(\mathbf{r}^N)]$$

$$\frac{\mathrm{acc}(o \to n)}{\mathrm{acc}(n \to o)} = \frac{\mathcal{N}(n)}{\mathcal{N}(o)} = \mathrm{exp}\{-\beta[\mathcal{U}(n) - \mathcal{U}(o)]\}$$

Metropolis,
Rosenbluth,Rosenbluth,

Teller and Teller choice:

$$\mathrm{acc}(o \to n) = \mathrm{min}\left(1, \mathrm{exp}\{-\beta[\mathcal{U}(\mathbf{r}'^N) - \mathcal{U}(\mathbf{r}^N)]\}\right)$$

Satisfactory?

Solution of conflict: play with the a-priori probabilities of trial moves:

$$\alpha(o \to n) \neq \alpha(n \to o)$$

$$\frac{\mathrm{acc}(o \to n)}{\mathrm{acc}(n \to o)} = \frac{\alpha(n \to o)}{\alpha(o \to n)} \exp\{-\beta[\mathcal{U}(n) - \mathcal{U}(o)]\}.$$

In particular, if:

$$\frac{\alpha(n \to o)}{\alpha(o \to n)} = \exp\{-\beta[\mathcal{U}(o) - \mathcal{U}(n)]\}.$$

Then

$$\frac{\mathrm{acc}(o \to n)}{\mathrm{acc}(n \to o)} = 1$$ (100% acceptance)

100% acceptance can be achieved in special cases: e.g. Swendsen-Wang, Wolff, Luyten, Whitelam-Geissler, Bortz-Kalos-Lebowitz, Krauth…

General idea: construct "cluster moves"

Simplest example: Swendsen-Wang

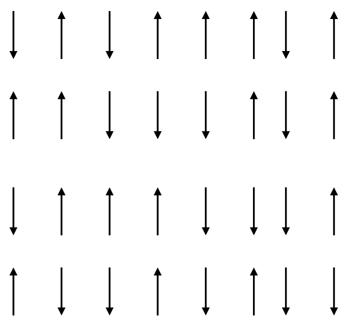
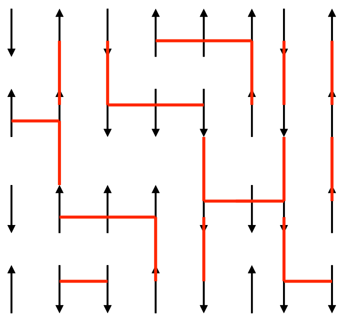


Illustration: 2D Ising model.

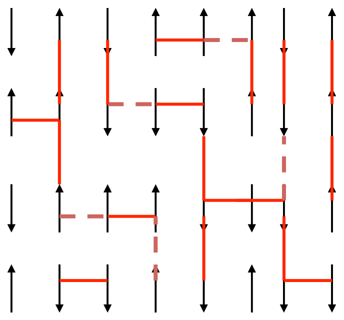
Snapshot: some neighbors are parallel, others anti-parallel



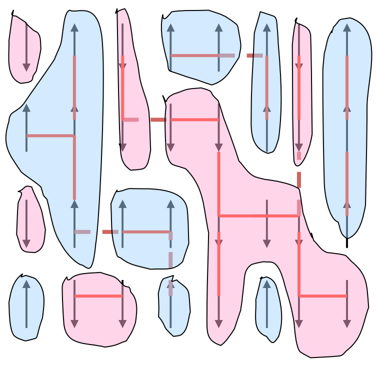
Number of parallel nearest-neighbor pairs: $\mathbf{N_p}$

Number of anti-parallel nearest neighbor pairs is: $\mathbf{N_a}$

Total energy: $\mathbf{U = (N_a - N_p)\ J}$

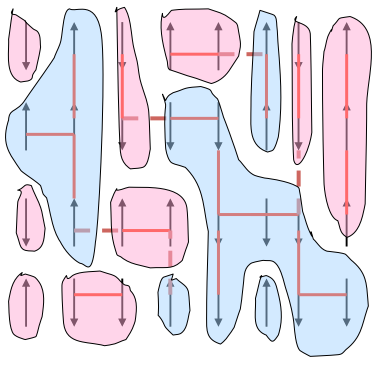


Make "bonds" between parallel neighbors. The probability to have a bond (red line) between parallel neighbors is **p** (as yet undetermined). With a probability **1-p**, parallel neighbors are not connected (blue dashed line).



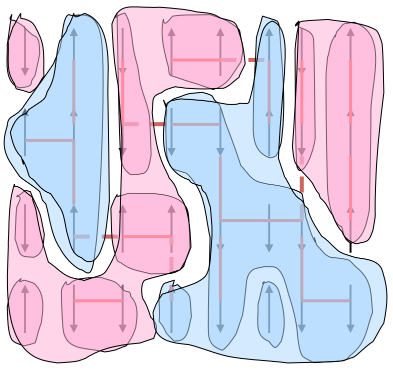
Form clusters of all spins that are connected by bonds. Some clusters are all “spin up” others are all “spin down”.

Denote the number of clusters by **M**.

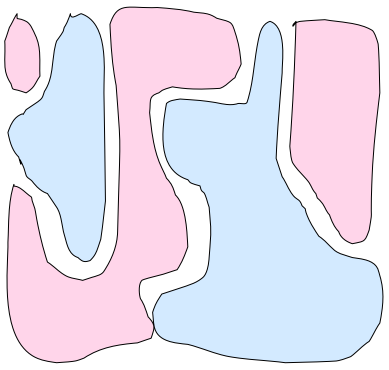


Now randomly flip clusters. This yields a new cluster configuration with probability $\mathbf{P_{(flip)}} \mathbf{=(1/2)^{M}}$.

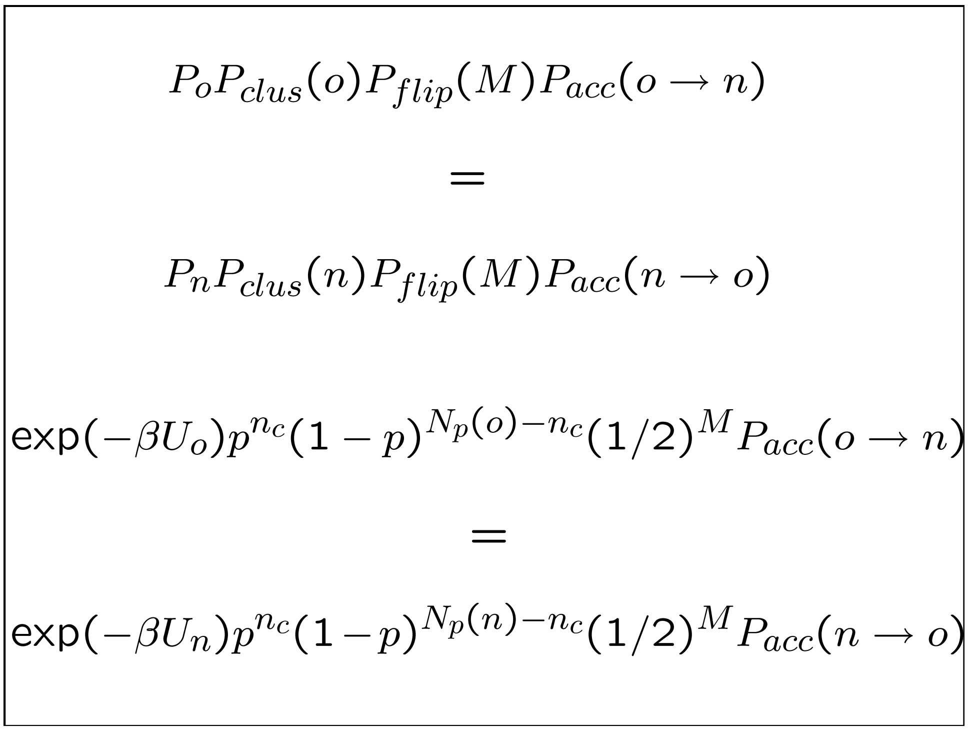
Then reconnect parallel spins



Next: forget about the "bonds"…



New spin configuration!



$$P_o P_{clus}(o) P_{flip}(M) P_{acc}(o \to n)$$

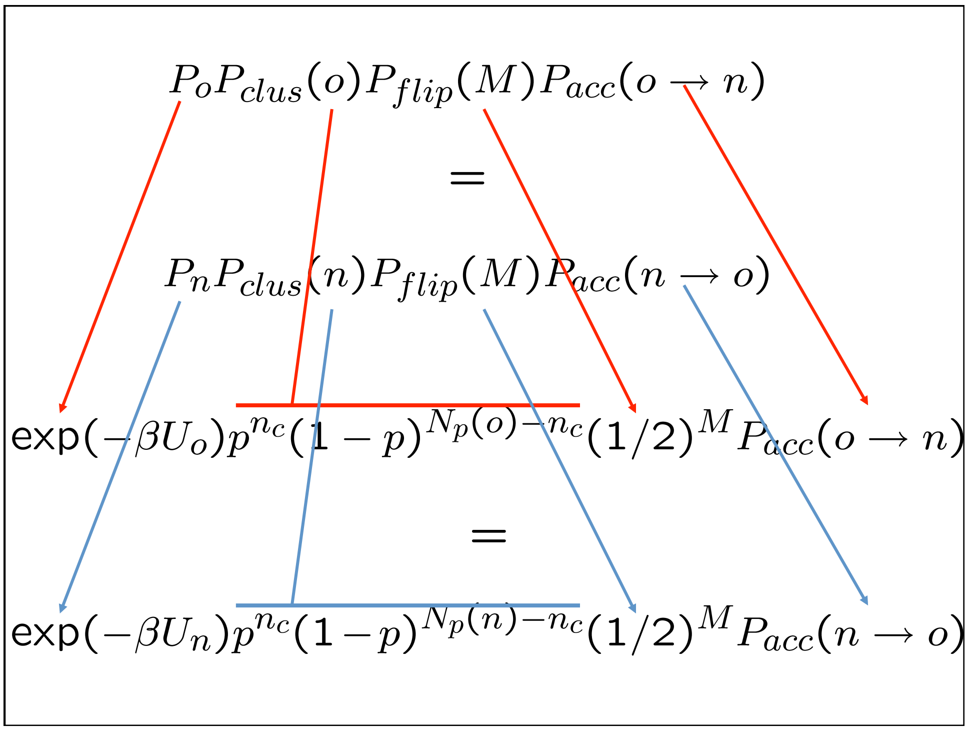
$$=$$

$$P_n P_{clus}(n) P_{flip}(M) P_{acc}(n \to o)$$

$$\exp(-\beta U_o) p^{n_c} (1-p)^{N_p(o)-n_c} (1/2)^M P_{acc}(o \to n)$$

$$=$$

$$\exp(-\beta U_n) p^{n_c} (1-p)^{N_p(n)-n_c} (1/2)^M P_{acc}(n \to o)$$



$$P_o P_{clus}(o) P_{flip}(M) P_{acc}(o \to n)$$

$$=$$

$$P_n P_{clus}(n) P_{flip}(M) P_{acc}(n \to o)$$

$$\exp(-\beta U_o) \underline{p^{n_c} (1-p)^{N_p(o)-n_c}} (1/2)^M P_{acc}(o \to n)$$

$$=$$

$$\exp(-\beta U_n) \underline{p^{n_c} (1-p)^{N_p(n)-n_c}} (1/2)^M P_{acc}(n \to o)$$

Moreover, we want 100% acceptance, i.e.:

$$P_{acc}(o \rightarrow n) = P_{acc}(n \rightarrow o) = 1$$

$$\exp(-\beta U_o)\cancel{p^{n_c}}(1-p)^{N_p(o)-\cancel{n_c}}\cancel{(1/2)^M}P_{acc}(o \to n)$$

$$=$$

$$\exp(-\beta U_n)\cancel{p^{n_c}}(1-p)^{N_p(n)-\cancel{n_c}}\cancel{(1/2)^M}P_{acc}(n \to o)$$

Hence:

$$\exp(-\beta U_o)(1-p)^{N_p(o)} = \exp(-\beta U_n)(1-p)^{N_p(n)}$$

$$\exp(\beta(U_n - U_o)) = (1-p)^{N_p(n)-N_p(o)}$$

But remember:

$$U_n - U_o = J(N_a(n) - N_p(n)) - J(N_a(o) - N_p(o))$$

or

$$\Delta U = J(\Delta N_a - \Delta N_p)$$

But: $\Delta N_a = -\Delta N_p$
and therefore
$\Delta U = -2J\Delta N_p$

$$\exp(\beta(U_n - U_o)) = \exp(-2\beta J(N_p(n) - N_p(o)))$$

Combining this with:

$$\exp(\beta(U_n - U_o)) = (1 - p)^{N_p(n) - N_p(o)}$$

we obtain:

$$p = 1 - \exp(-2\beta J)$$

100% acceptance!!!

WASTE RECYCLING MC

Include "rejected" moves in the sampling

This is the key:

$$\sum_m \rho(m)\pi_{mn} = \rho(n)$$

The transition matrix π leaves the equilibrium distribution ρ unchanged.

$$\langle A \rangle_\rho = \sum_n A_n \rho_n$$

This, we can rewrite as:

$$\sum_n A_n \rho_n = \sum_n \sum_m A_n \rho_m \pi_{mn} = \sum_m \rho_m \sum_n A_n \pi_{mn}$$

$$= \sum_m \rho_m \sum_n A_n \pi_{mn} \Leftrightarrow \langle A \rangle_\rho = \left\langle \sum_n \pi_{mn} A_n \right\rangle_{\rho_m}$$

$$\langle A \rangle_\rho = \left\langle \sum_n \pi_{mn} A_n \right\rangle_{\rho_m}$$

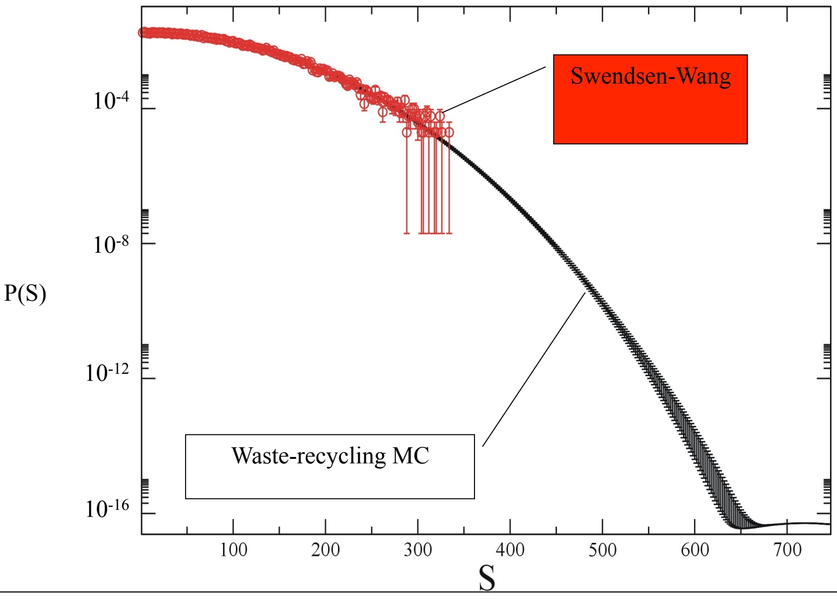
Note that <A> is no longer an average over "visited" states – we also include "rejected" moves in the sampling.

Slightly dishonest and slightly trivial example:

Sampling the magnetization of a 2D Ising system

Compare:

1. Normal (Swendsen-Wang) MC (sample **one** out of 2^n states)
2. Idem + "waste recycling" (sample **all** 2^n states)



Monte Carlo sampling with noisy weight functions.

Two possible cases:

1. **The calculation of the energy function is subject to statistical error** (Ceperley, Dewing, J. Chem. Phys. **110**, 9812 (1999).)

$$u_{\text{computed}} = u_{\text{real}} + \delta u$$

with:

$$< \delta u >= 0$$
$$< (\delta u)^2 >= \sigma_s^2$$

We will assume that the fluctuations in ***u*** are Gaussian. Then:

Now consider that we do Monte Carlo with this noisy energy function:

$$\frac{P_n(\mathbf{x}_n)}{P_o(\mathbf{x}_o)} = \exp[-\beta\Delta u]$$

with

$$\Delta u = u_n + \delta u_n - u_o - \delta u_o$$

Then:

$$\left\langle \frac{P_n}{P_o} \right\rangle = \exp[-\beta\langle\Delta u\rangle + (\beta\sigma)^2/2]$$

With: $\sigma^2 = 2\sigma_s^2$

As a consequence, we sample the states with the wrong weight.

However, we can use another acceptance rule:

$$P_{\text{acc}} = \text{Min}\{1, \exp[-\beta\Delta u - (\beta\sigma)^2/2]\}$$

In that case:

$$\left\langle \frac{P_n}{P_o} \right\rangle = \exp[-\beta\langle\Delta u\rangle + (\beta\sigma)^2/2] \times \exp[-(\beta\sigma)^2/2$$

$$= \exp[-\beta\langle\Delta u\rangle]$$

In other words:

If the statistical noise in the energy is Gaussian,

and its variance is constant,

then we can perform rigorous sampling, even when the energy function is noisy

2. **The weight function is noisy, but its average is correct** (not so common in molecular simulation, but quite common in other sampling problems)

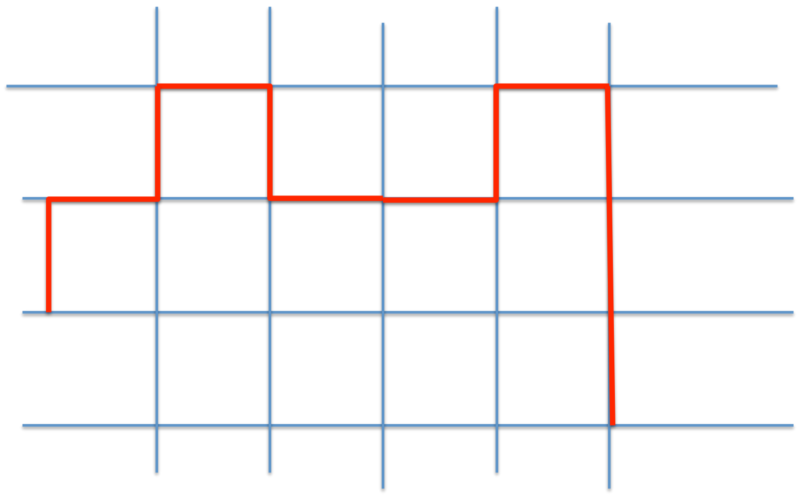
(can also be sampled rigorously – but outside the scope of this lecture)

Recursive sampling

Outline:

1. Recursive enumeration
   a) Polymer statistics (simulation)
   b) ..
2. Molecular Motors (experiments!)
   (well, actually, simulated experiments)

Lattice polymers:



Consider a lattice (e.g. 2D-square).

At a given point x_i, the potential energy is $U(x_i)$.

The Boltzmann factor for a particle at point x_i is

$\exp(-\beta U(x_i)) \equiv z_i^1$

Consider a lattice (e.g. 2D-square).

At a given point x_i, the potential energy is $U(x_i)$.

The Boltzmann factor for a particle at point x_i is

$\exp(-\beta U(x_i)) \equiv z_i^1$

The partition function for a single point particle is

$Z_1 \equiv \sum_i z_i^1$

Dimers

The Boltzmann factor for a dimer on points x_i and x_{i+1} is

$\exp(-\beta(U(x_i) + U(x_{i+1})) = z_i^1 \times z_{i+1}^1$

The Boltzmann factor for all dimers terminating

on point x_i is

$z_i^{(2)} \equiv z_i^1 \times \sum_{jnni} z_j^1$

The partition function for a single dimer is

$Z_2 \equiv \sum_i z_i^{(2)}$

n-mers

The Boltzmann weight for an n-mer terminating on point x_i is

$z_i^{(n)} = z_i^1 \times \sum_{jnni} z_j^{(n-1)}$

and the corresponding partition function is

$Z^{(n)} = \sum_i z_i^{(n)}$

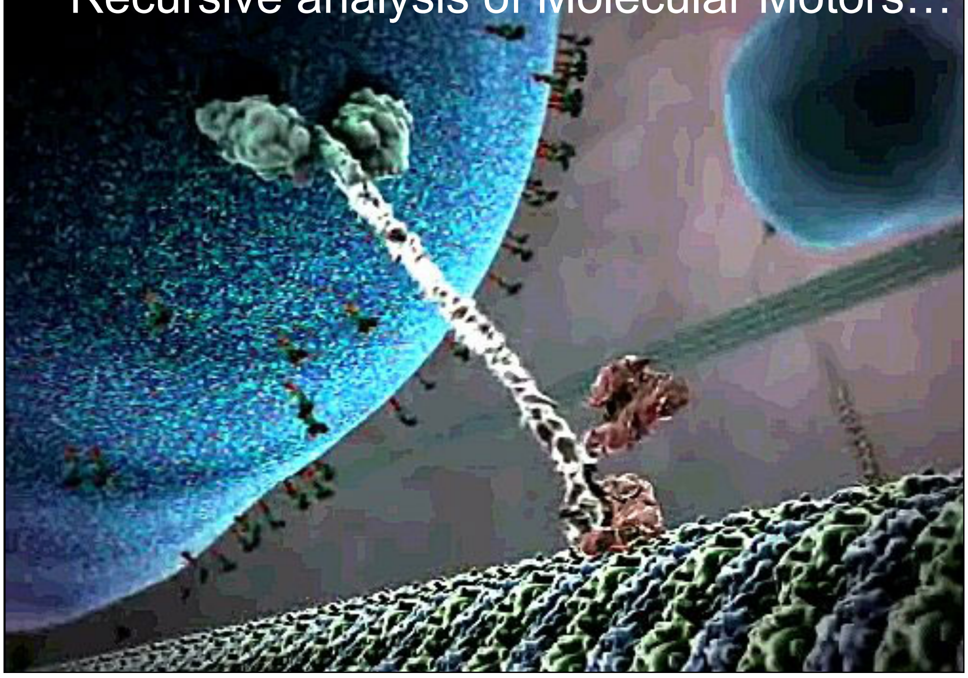
This method is exact for non-self-avoiding, non-interacting lattice polymers.

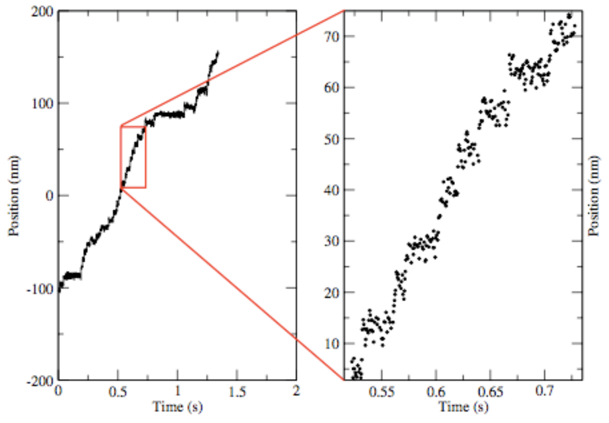
It can be used to speed up MC sampling of (self)interacting polymers

B. Bozorgui and DF, Phys. Rev. E 75, 036708 (2007))

NOTE: `MFOLD' also uses recursive sampling to predict RNA secondary structures.

Recursive analysis of Molecular Motors…

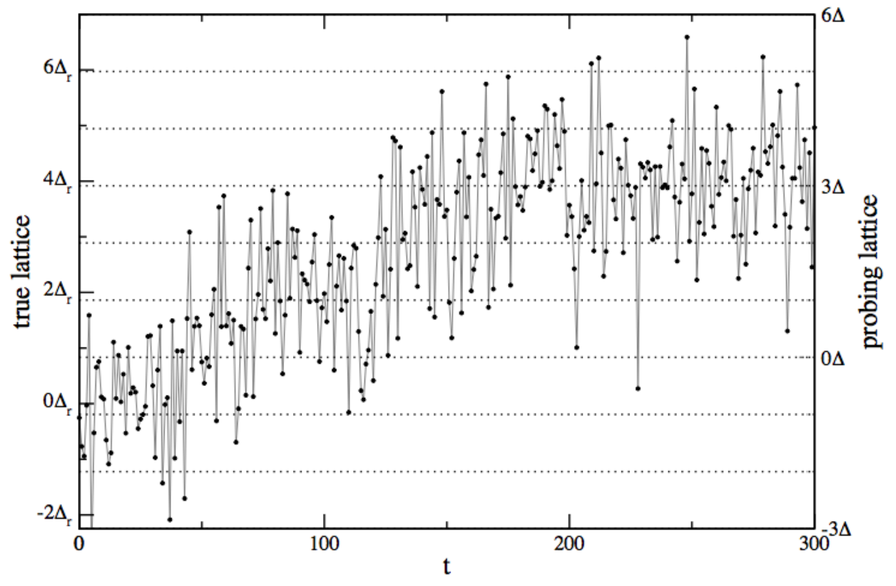




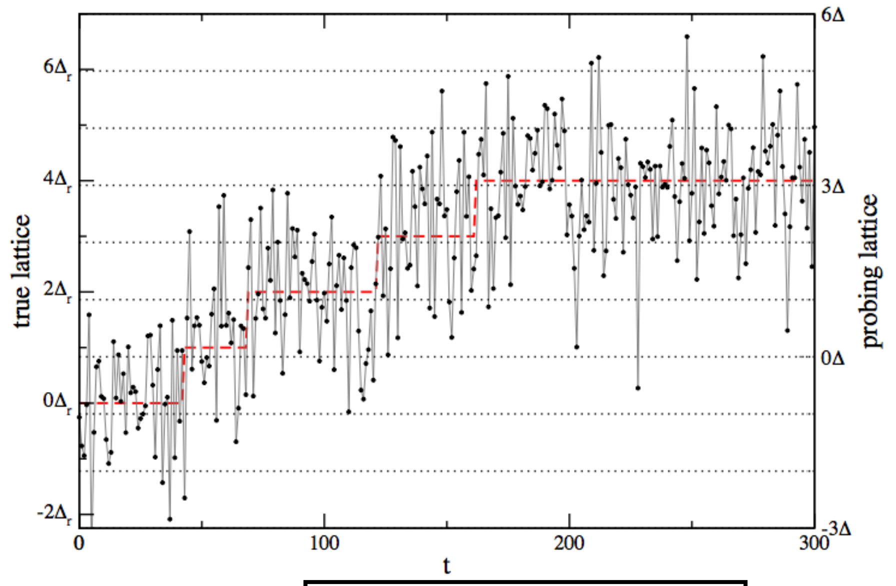
Kinesin motor steps along micro-tubules with a step size of 8nm

Experimentally, the step size is measured by fitting the (noisy) data.

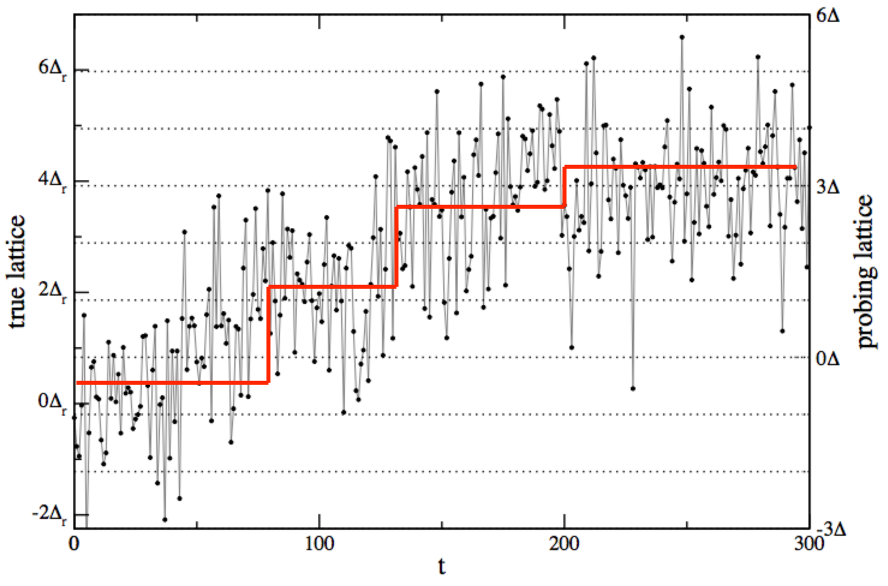
Example: noisy "synthetic data"



Example: noisy "synthetic data"



Best practice: “fit steps to data”



J.W.J. Kerssemakers et al. , Nature 442,709 (2006)

How well does it perform?

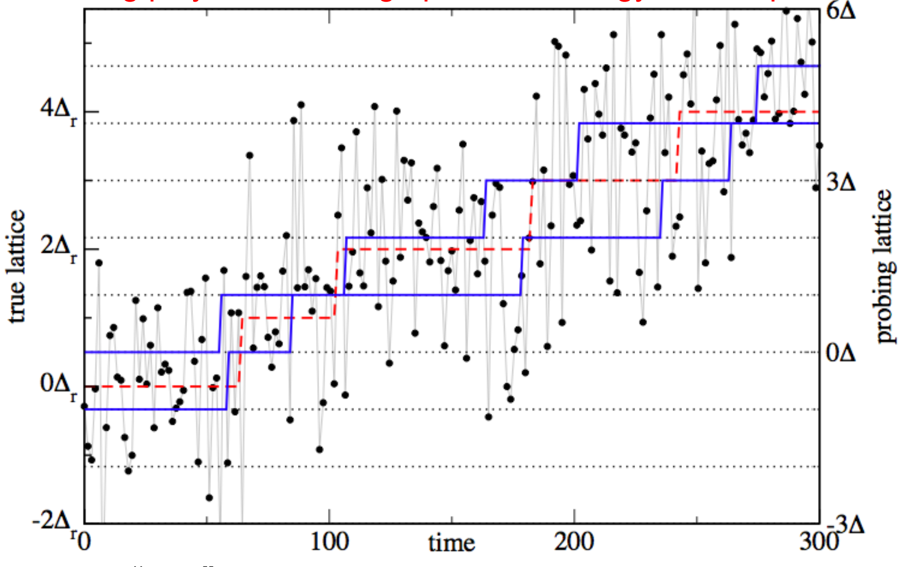
1. It can be used if the noise is less than 60% of the step size.
2. It yields a distribution of step sizes (even if the underlying process has only one step size)

Observation:

We want to know the step size and the step frequency but…

We do not care which trace is the "correct" trace.

Bayesian approach: compute the partition function **Q** of non-reversing polymer in a rough potential energy landscape



"true" trace

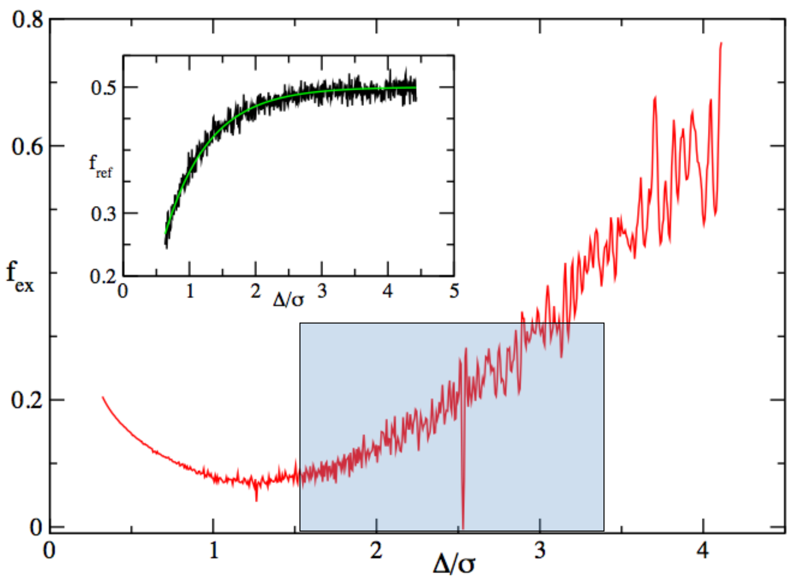
Other directed walks

As shown before: we can enumerate **Q** exactly (and cheaply).

From **Q** we can compute a "free energy"

$$(F = -\ln Q)$$

Compute the "excess free energy" with respect to reference data

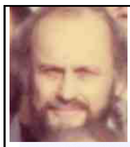


MBAR

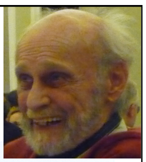
(Multistate Bennett Acceptance Ratio)

Method to obtain the best estimate of free-energy differences from umbrella sampling

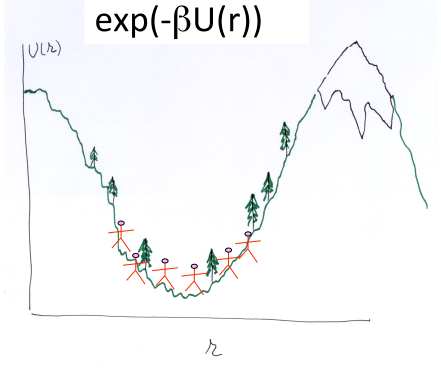
Shirts, M. R., and Chodera, J. D. (2008) Statistically optimal analysis of samples from multiple equilibrium states. *J. Chem. Phys.* 129, 129105.



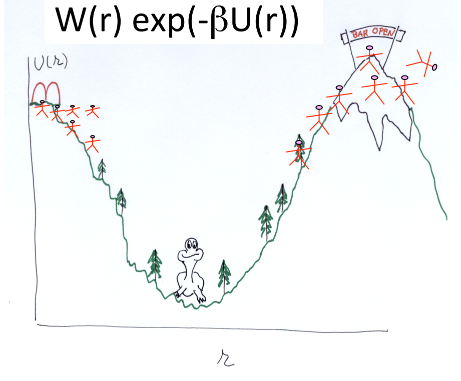
Umbrella sampling



exp(-βU(r))



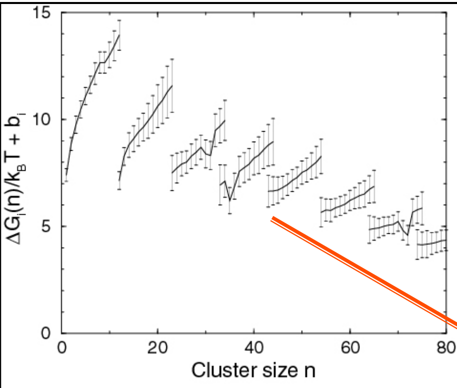
W(r) exp(-βU(r))

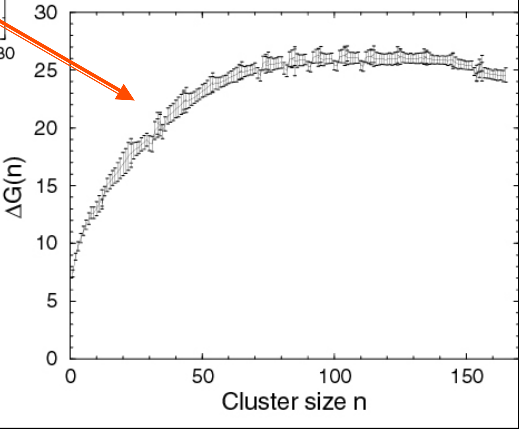


METROPOLIS SAMPLING

UMBRELLA SAMPLING

COMBINING HISTOGRAMS: HOW?





Problems:

1. What is the `best' bin width
2. How do we stitch histograms together?

MBAR: No binning and `optimal' stitching.

We start from:

$$Z = \int d\mathbf{R}^N \ \exp[-\beta U(\mathbf{R}^N)]$$

and

$$F = -k_B T \ln Z$$

Suppose we have k different samples (e.g. in umbrella sampling), biased with potentials $V_k(\mathbf{R}^N)$. Assume that we have N_k points for sample k We can then define ‘partition functions Z_k for the biased systems as

$$Z_k \equiv \int d\mathbf{R}^N \exp(-\beta[U(\mathbf{R}^N) + V_k(\mathbf{R}^N)])$$

and

$$F_k \equiv -k_B T \ln Z_k$$

In what follows, we will use:

$$\Delta F_k \equiv F_k - F = k_B T \ln(Z/Z_k)$$

The key assumption of MBAR is that the true (as opposed to the sampled) distribution function is a weighted set of delta-functions *at the points that have been sampled*.

In words: we do not assume anything about points that we have not sampled.

The distribution function is then of the form:

$$P(\mathbf{R}^N) = \mathcal{Z}^{-1} \sum_{j=1}^{K} \sum_{n=1}^{N_k} p_{j,n} \delta\left(\mathbf{R}^N - \mathbf{R}^N_{j,n}\right)$$

Where the $p_{j,n}$ are (as yet) unknown.

The normalization factor is defined as:

$$\mathcal{Z} \equiv \sum_{j=1}^{K} \sum_{n=1}^{N_k} p_{j,n}$$

Once the full distribution is known, the biased distributions follow:

$$P_k(\mathbf{R}^N) = \mathcal{Z}_k^{-1} \sum_{j=1}^{K} \sum_{n=1}^{N_k} p_{j,n} \exp(-\beta V_k(\mathbf{R}^N)) \delta\left(\mathbf{R}^N - \mathbf{R}^N_{j,n}\right)$$

The normalization factor Z_k is defined as:

$$\mathcal{Z}_k \equiv \sum_{j=1}^{K} \sum_{n=1}^{N_k} p_{j,n} \exp(-\beta V_k(\mathbf{R}^N))$$

Now we must compute the unknown weights $p_{j,n}$

We do this, using `maximum likelihood’.

That is: we impose that the values of the $p_{j,n}$ should be such that the probability of obtaining the observed histograms is maximised

We define the *likelihood* **L:**

$$L \equiv \prod_{j=1}^{K} \left[\prod_{n=1}^{N_k} P_k(\mathbf{R}_{j,n}^{N})) \right]$$

L depends on all $p_{j,n}$

We determine $p_{j,n}$ by imposing that **L**, or equivalently ln **L** is maximal.

If we look at ln **L**

$$\ln L \equiv \sum_{j=1}^{K}\sum_{n=1}^{N_j} \ln \frac{p_{j,n}}{\mathcal{Z}_j} \exp(-\beta V_j(\mathbf{R}_{j,n}^N))$$

We see that ln $p_{j,n}$ and Z_k depend on $p_{j,n}$
But the Boltzmann factor does not.

Therefore:

$$\begin{aligned}\ln L &= \text{constant} + \sum_{j=1}^{K}\sum_{n=1}^{N_j} [\ln p_{j,n} - \ln \mathcal{Z}_j] \\ &= \text{constant} + \sum_{j=1}^{K}\sum_{n=1}^{N_j} \ln p_{j,n} - \sum_{j=1}^{K} N_j \ln \mathcal{Z}_j\end{aligned}$$

Now, we can differentiate with respect to $p_{j,n}$
The constant yields zero.
The second term: $1/p_{j,n}$
The third term follows if we use:

$$\mathcal{Z}_k \equiv \sum_{j=1}^{K}\sum_{n=1}^{N_j} p_{j,n} \exp(-\beta V_k(\mathbf{R}_{j,n}^N))$$

Our condition for maximum likelihood is then

$$0 = \frac{1}{p_{j,n}} - \sum_{k=1}^{K} N_k \frac{\exp[-\beta V_k(\mathbf{R}_{j,n}^N))]}{\mathcal{Z}_k}$$

Or:

$$p_{j,n}/\mathcal{Z} = \frac{1}{\sum_{k=1}^{K} N_k \frac{\exp[-\beta V_k(\mathbf{R}_{j,n}^N))]}{(\mathcal{Z}_k/\mathcal{Z})}}$$

The probability to observe a given point (j,n) given the optimal $p_{j,n}$ is then

$$p_{j,n}/\mathcal{Z} = \frac{1}{\sum_{k=1}^{K} N_k \exp[-\beta(V_k(\mathbf{R}_{j,n}^N) - \Delta F_k)]}$$

Where we have used

$$\Delta F_k \equiv F_k - F = k_B T \ln(Z/Z_k)$$

We can rewrite our result as an implicit equation for the ΔF_i :

$$\Delta F_i = -k_B T \ln \sum_{j=1}^{K} \sum_{n=1}^{N_j} \frac{\exp[-\beta(V_i(\mathbf{R}^N_{j,n})]}{\sum_{k=1}^{K} N_k \exp[-\beta(V_k(\mathbf{R}^N_{j,n}) - \Delta F_k)]}$$

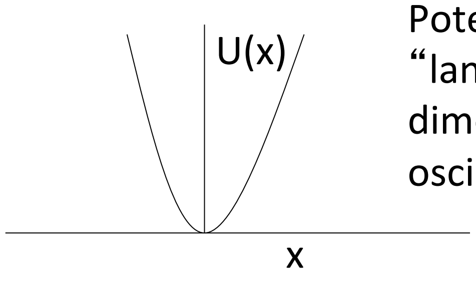
These are the MBAR equations that must be solved self-consistently

Advantages of MBAR over all earlier schemes (except Bennett)

- It does not use bins.
- it makes no assumption about the form of the distribution function where it has not been sampled.
- different biased runs may sample different points in parameter space
- the method yields the *best* (in the sense of `the most likely') estimate for the histograms and the free energy differences.

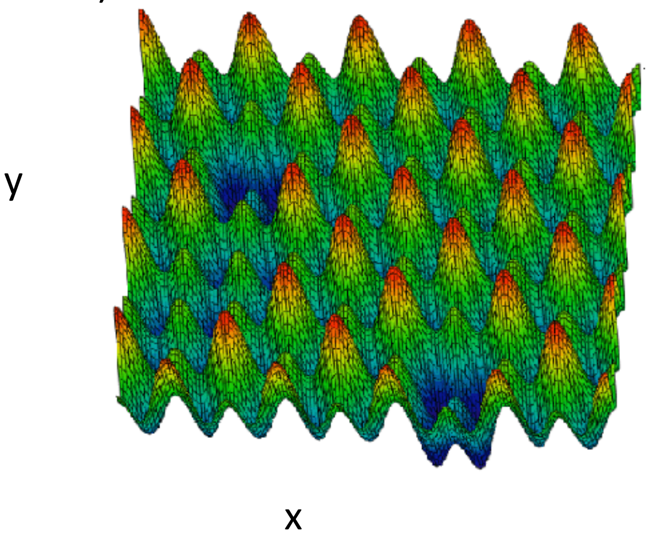
First question:
What **IS** a free-energy landscape?

Simpler question:
What is an energy landscape?



Potential-energy "landscape" of a 1-dimensional harmonic oscillator

2-dimensional potential energy landscape (e.g. surface potential experienced by an adsorbate atom)



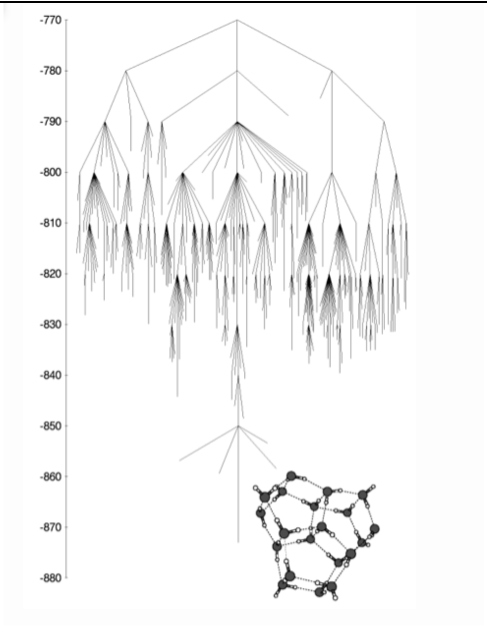
General potential energy landscape:

$U(\mathbf{r}^N)$ with $\mathbf{r}^N \equiv (x_1, y_1, z_1, \cdots, x_i, y_i, z_i \cdots x_N, y_N, z_N)$

High dimensional - not easy to visualise.

Visual aid: "disconnectivity graphs"

U

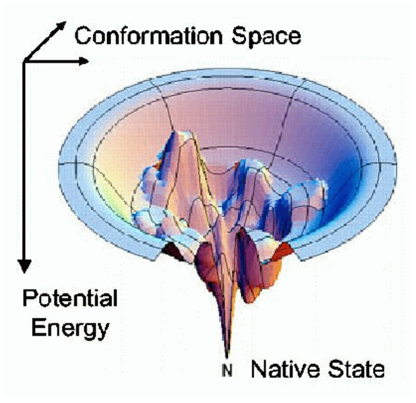


X? $\longrightarrow$

“tree” of energy minima.

Topology - not distance

Free energy landscape?



But what are the landscape coordinates?

In order to define a free energy it is necessary to specify the coordinates of the landscape.

Other coordinates => other free-energy landscape"

Statistical mechanics: Boltzmann weight.

$$P(r^N) = \frac{e^{-U(\mathbf{r}^N)/kT}}{Z}$$

What is the probability that the center of mass of the system is at a coordinate X?

$$P(X) = \frac{\int dx^N e^{-U(x^N)/kT} \delta(X - N^{-1} \sum_i x_i)}{Z}$$

We now *define* the free energy associated with center-of-mass coordinate X as:

$$e^{-F(X)/kT} \equiv \int dx^N e^{-U(x^N)/kT} \delta(X - N^{-1} \sum_i x_i)$$

The free energy is to the “collective” coordinate X, what the potential energy is to the individual coordinates.

In general, there may be several coordinates, X, Y, Z etc.

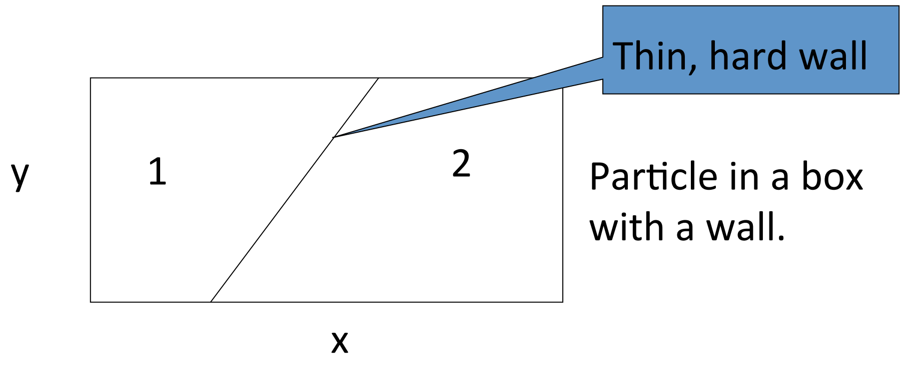
They may be complicated functions of $\mathbf{r}^N$, and they may be discrete.
e.g.
X = radius of gyration of a protein
Y = number of native contacts

One message:

There is no such thing as **the** free energy landscape of a system.

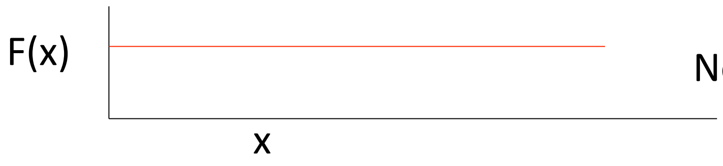
We can only define F(X,Y,…) *after* choosing the relevant coordinates X,Y,…

Note: there may be a landscape but not a road…



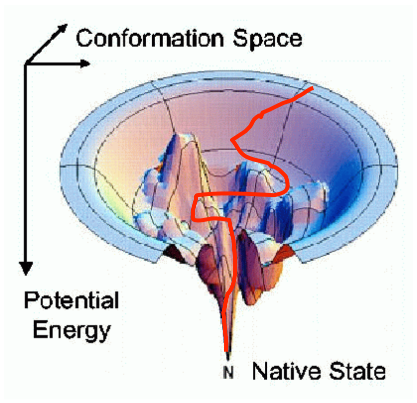
Particle in a box with a wall.

Integrate over y => F(x): no barrier but…



No road!

A "funnel" may, or may not contain a "road".



Usually, it does.

Disconnectivity graphs **always** contain a road (but no distance).

There is NEVER enough computer time

Solutions:

1. **Wait until there is enough computer power**

Moore's law: computing power Γ increases a factor 10 every 5 years

$$\Gamma(\Delta t) = \Gamma_0 e^{\Delta t/\tau}$$

with $\tau \approx 2.17$ years.

When to start a major calculation?

Either start now, or wait a time Δt and use the computers that are available then:

For a computation involving X operations, the expected time to completion, using the current computing power Γ_0, is: $t_0 = \frac{X}{\Gamma_0}$.

But if we first wait a time Δt, it is:

$$t = \Delta t + \frac{t_0}{e^{\Delta t/\tau}}$$

Optimum? Compute extremum:

$$0 = 1 - \frac{t_0}{\tau} e^{-\Delta t/\tau}$$

Hence

$$1 = \frac{t_0}{\tau} e^{-\Delta t/\tau}$$

No solutions if $t_0 < \tau$, i.e. if the estimated computing time is less than 2.17 years.

In that case: start right away!

Otherwise:

wait for a time interval

$$\Delta t = \tau \ln(t_0/\tau)$$

then buy a computer, and start!!!

Solutions:

1. Wait until there is enough computer power
2. **Use cheaper models/ more efficient algorithms.**