Free energy calculations

Berk Hess

Why do free energy calculations?

The free energy G gives the population of states:

$$\frac{P_1}{P_2} = \exp\left(\frac{\Delta G}{k_B T}\right) \quad , \qquad \Delta G = G_2 - G_1$$

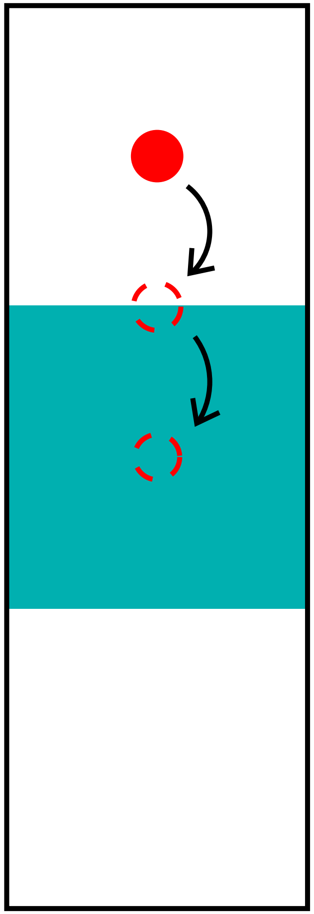
Since we mostly simulate in the NPT ensemble we will use the Gibbs free energy G (not the NVT Helmholtz free energy A)

A free energy difference can be split into two terms:

$$\Delta G = \Delta H - T\Delta S \quad , \qquad \Delta H = \Delta U + P\Delta V$$

ΔG is less costly to calculate than ΔU

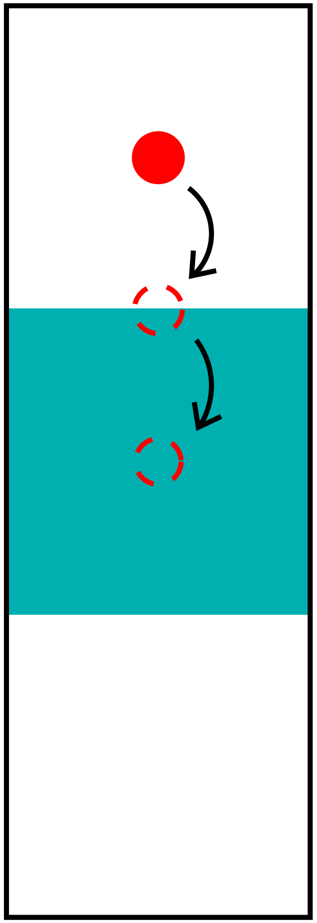
Example: solvation free energy

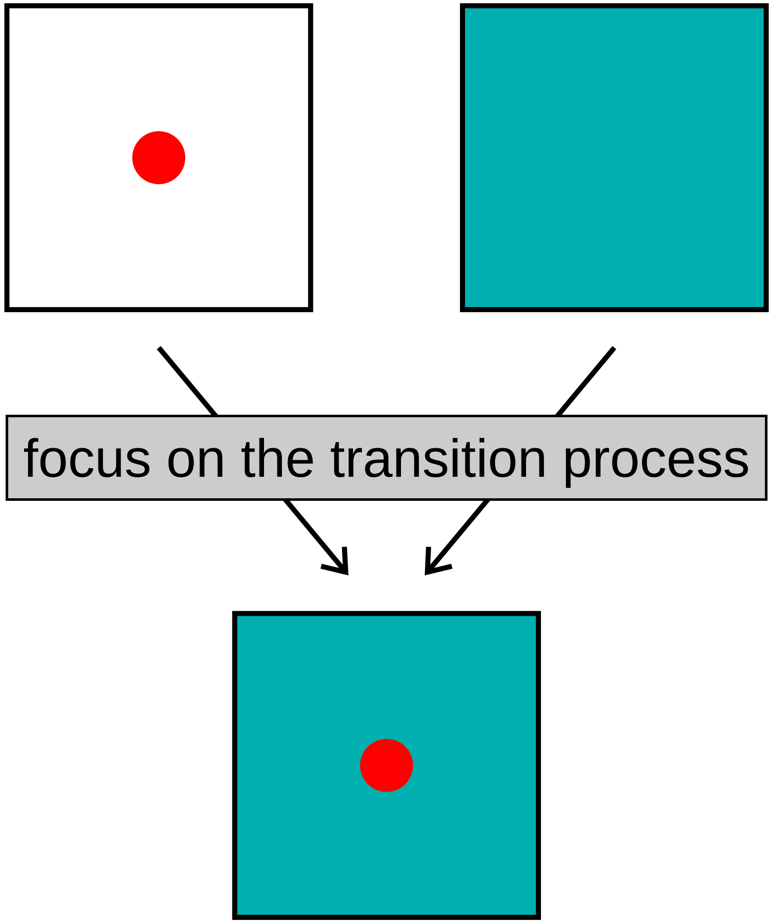


Measuring the probabilities by calculating distributions directly can be very inefficient:

- path between two states infrequently sampled
  - barrier high
  - narrow connection
- probability of the two states can differ very much (large free energy difference)

Example: solvation free energy





focus on the transition process

The coupling parameter approach

We add a coupling parameter λ to the Hamiltonian or potential:

$$V = V(\mathbf{r}, \lambda)$$

$$V(\mathbf{r}, 0) = V^{\mathrm{A}}(\mathbf{r}) \quad , \qquad V(\mathbf{r}, 1) = V^{\mathrm{B}}(\mathbf{r})$$

The free energy difference between state A and B is then given by:

$$G^{\mathrm{B}}(p,T) - G^{\mathrm{A}}(p,T) = \int_0^1 \left\langle \frac{\partial H}{\partial \lambda} \right\rangle_{NPT;\lambda} d\lambda$$

Free energy perturbation

At a given value of λ:

$$A(\lambda) = -k_B T \log \left[c \int e^{-\beta V(\mathbf{r},\lambda)} d\mathbf{r} \right]$$

Usually impossible to calculate from simulations.
But possible to calculate as a *perturbation* from an *ensemble average*:

$$\begin{aligned} A(\lambda + \Delta\lambda) - A(\lambda) &= -k_B T \log \frac{\int \exp[-\beta V(\mathbf{r}, \lambda + \Delta\lambda)] d\mathbf{r}}{\int \exp[-\beta V(\mathbf{r}, \lambda)] d\mathbf{r}} \\ &= -k_B T \log \left\langle e^{-\beta[V(\mathbf{r},\lambda+\Delta\lambda) - V(\mathbf{r},\lambda)]} \right\rangle_\lambda \end{aligned}$$

Free energy integration

At a given value of λ:

$$A(\lambda) = -k_B T \log \left[c \int e^{-\beta V(\mathbf{r},\lambda)} d\mathbf{r} \right]$$

Usually impossible to calculate from simulations.
Also possible to compute $dA/d\lambda$ from an *ensemble average*:

$$\frac{dA}{d\lambda} = \frac{\int \frac{\partial V}{\partial \lambda} \exp[-\beta V(\mathbf{r},\lambda)] d\mathbf{r}}{\int \exp[-\beta V(\mathbf{r},\lambda)] d\mathbf{r}} = \left\langle \frac{\partial V}{\partial \lambda} \right\rangle_\lambda$$

Averages are taken over an equilibrium path using $V(\lambda)$

Usually intermediate points are used.
About 5-20 points required to integrate $\partial V/\partial\lambda$ properly

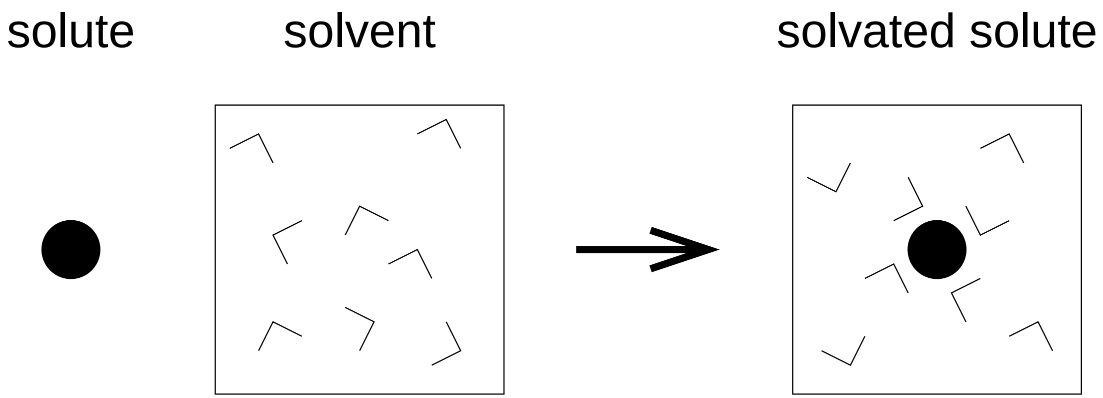
Slow growth

Integrate $\left\langle \frac{\partial V}{\partial \lambda} \right\rangle_\lambda$ while changing λ at every MD step

Issue: no ensemble average is taken: hysteresis is likely

Only use this for a simple, rough initial estimate,
use free energy integration for quantitative results

Free energy of solvation

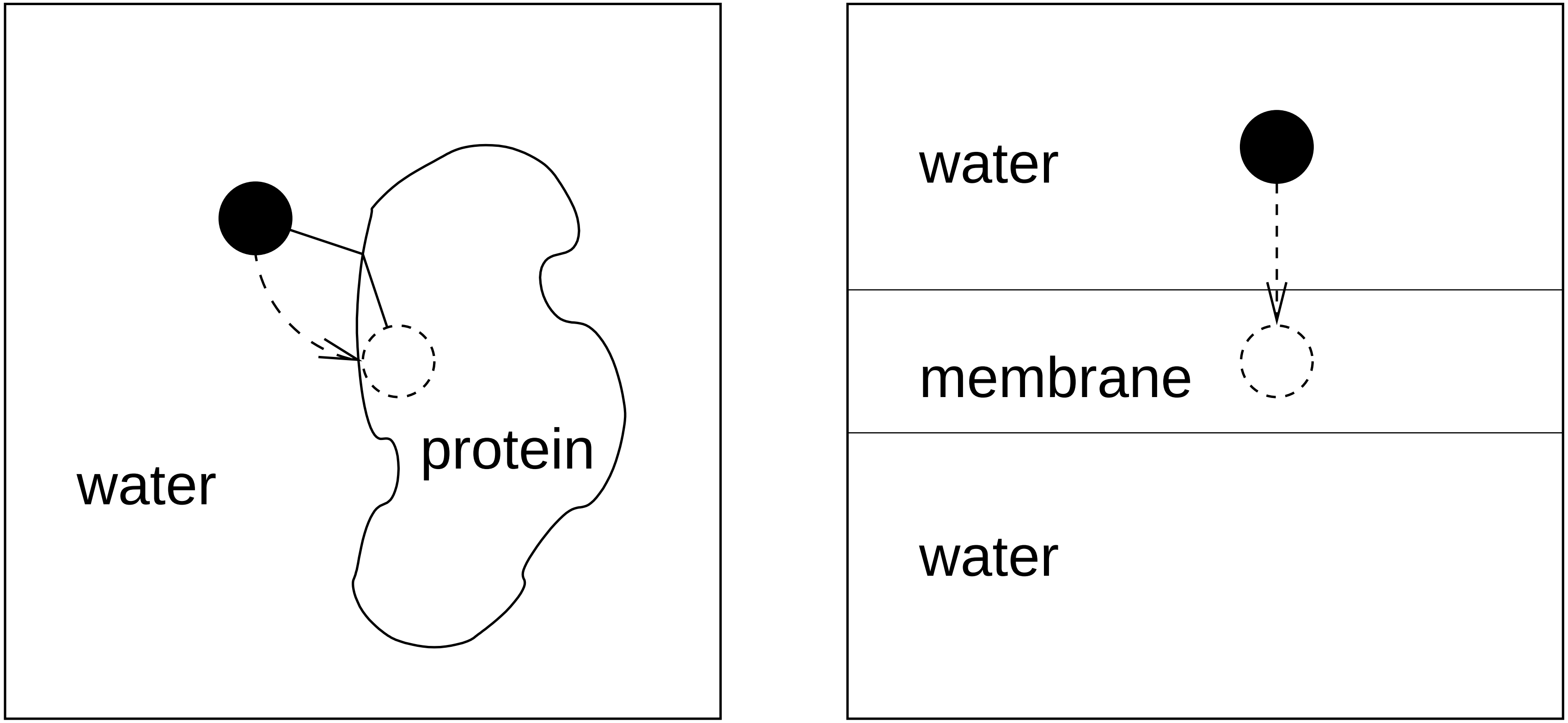


ΔG of solvation is often used to parametrize force fields

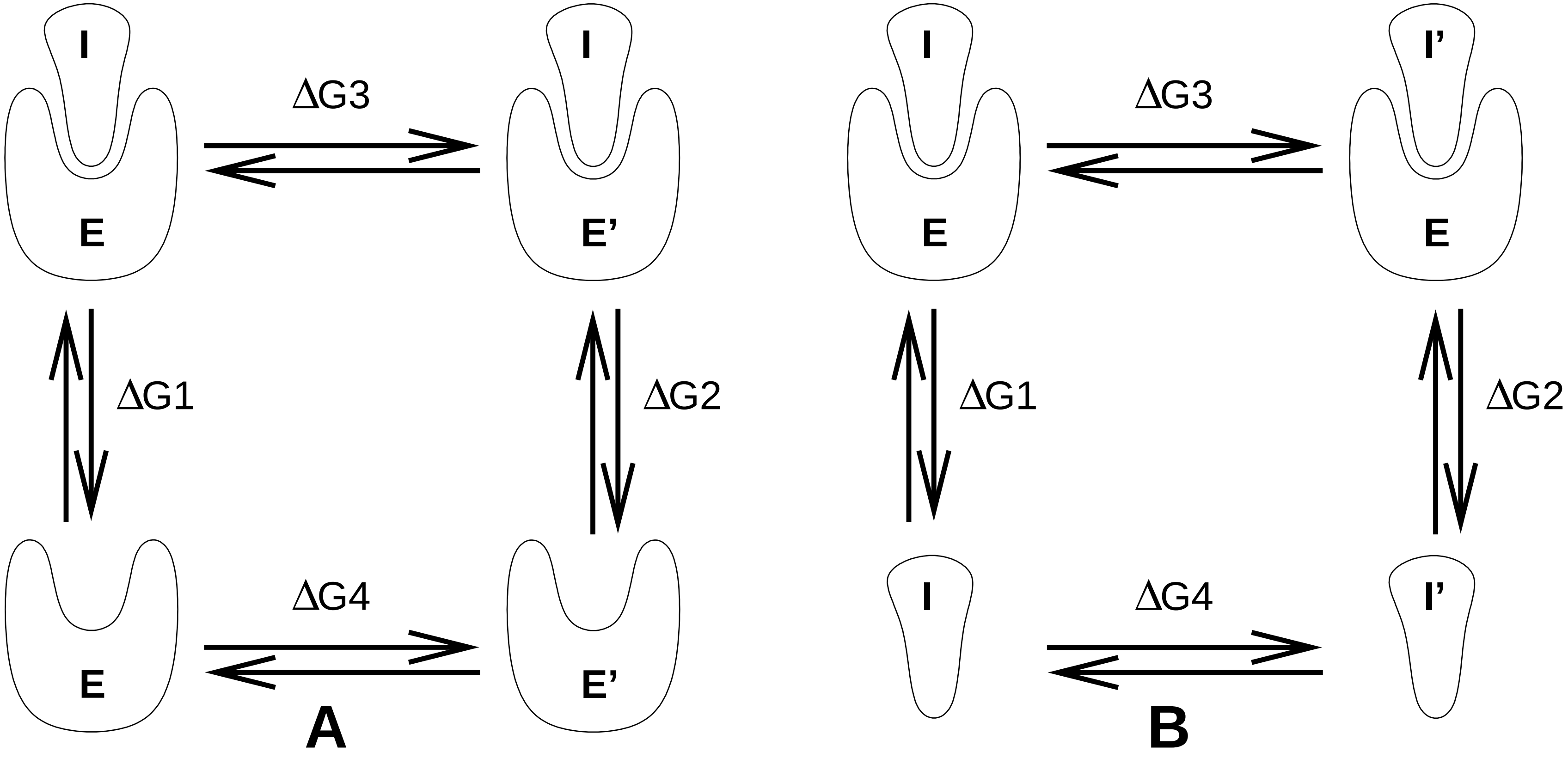
Partitioning properties:
similarly one can determine the free energy of transfer from a polar (water) to an apolar solvent (octane, cyclohexane)

This is important for protein folding and peptide-membrane interactions

Partitioning free-energies

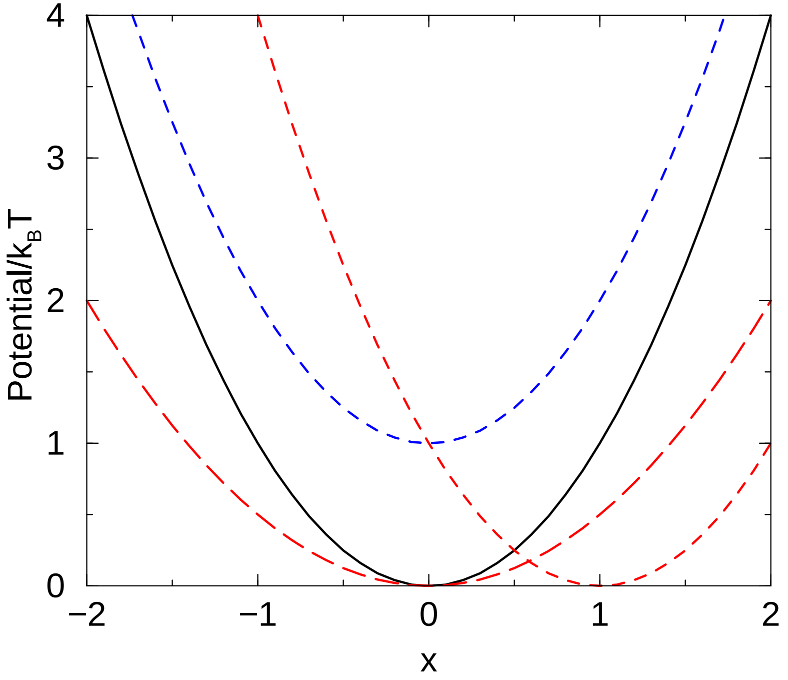


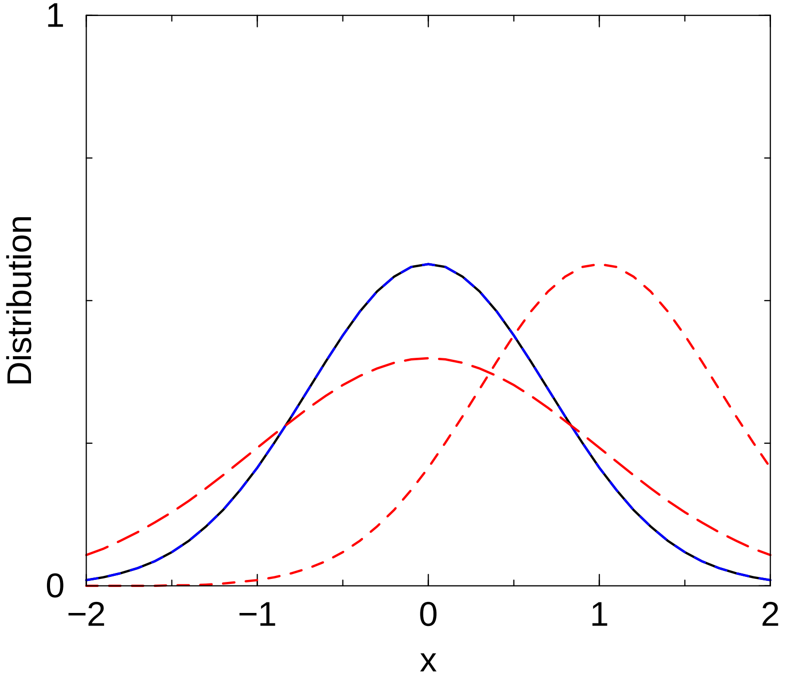
Binding free energies



$$\Delta\Delta G = \Delta G_1 - \Delta G_2 = \Delta G_3 - \Delta G_4$$

Distributions





ΔG *between similar states*

When the states of interest are very similar, e.g. small changes in charge or LJ parameters, only the potential energy difference matters and one can do single step perturbation

In Gromacs this can be done in two ways:

- Run a simulation for state A
- Do `mdrun -rerun traj.trr` with B-state parameters
- Determine the potential energy difference

or

- Run a simulation for state A with also a B-state topology and get the potential energy difference from the free energy code

ΔG *between different states*

With frequent transitions between states and small ΔG:

- simulate the system
- count the populations

With infrequent transitions between states or large ΔG:

We can modify the Hamiltonian to gradually move the system from state A to state B

The free energy difference is then given by the total work associated

with changing the Hamiltonian

The coupling parameter approach

The λ dependence of $V(\mathbf{r}m\lambda)$ can be chosen freely, as long as the end points match the states

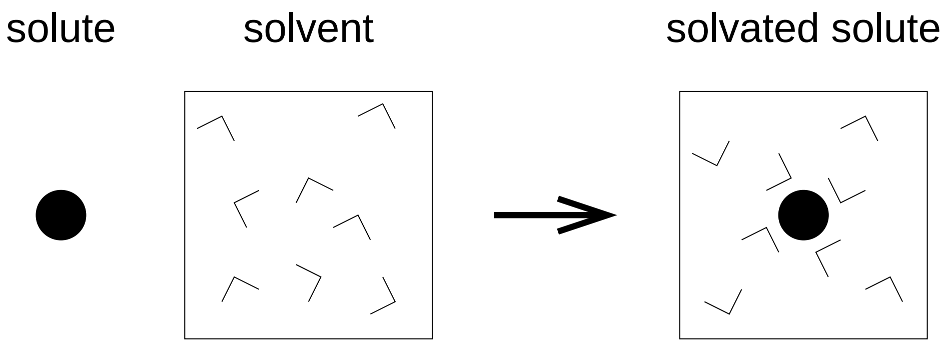
The simplest approach is linear interpolation:

$$V(\mathbf{r},\lambda) = (1-\lambda)V^{\mathrm{A}}(\mathbf{r}) + \lambda V^{\mathrm{B}}(\mathbf{r})$$

This works fine, except when potentials with singularities are affected

(LJ, Coulomb)

The coupling parameter approach



State A: solute fully coupled to the solvent
State B: solute fully decoupled from the solvent

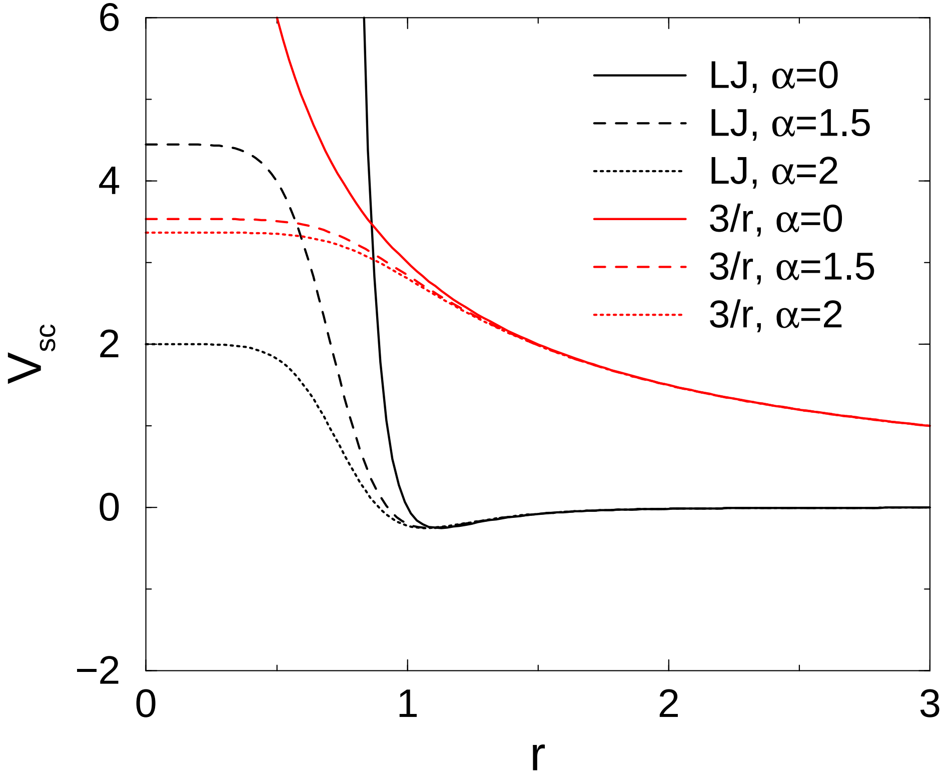
An example approach:
Run simulations at: $\lambda = 0, 0.1, \ldots, 0.9, 1.0$

Soft-core interactions

Instead of linear interpolation we use:

$$V_{sc}(r) = (1-\lambda)V^A(r_A) + \lambda V^B(r_B)$$

$$r_A = \left(\alpha\sigma_A^6\lambda^p + r^6\right)^{\frac{1}{6}} \quad , \qquad r_B = \left(\alpha\sigma_B^6(1-\lambda)^p + r^6\right)^{\frac{1}{6}}$$



Solvation free energy in practice

Make a topology where:

- the B-state atom types of the solute have LJ parameters zero
- the B-state charges of the solute are zero

1) Simulate the system with solvent at several λ-values 0,...,1

2) Simulate the system in vacuum at several λ-values 0,...,1

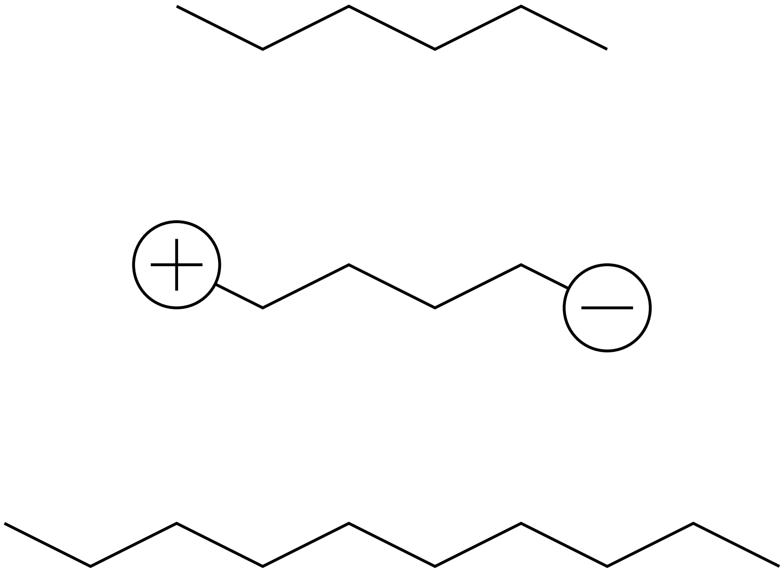
The solvation free energy is given by:

$$\Delta G_{solv} = \int_0^1 \left\langle \frac{\partial H}{\partial \lambda} \right\rangle_{2)} d\lambda - \int_0^1 \left\langle \frac{\partial H}{\partial \lambda} \right\rangle_{1)} d\lambda$$

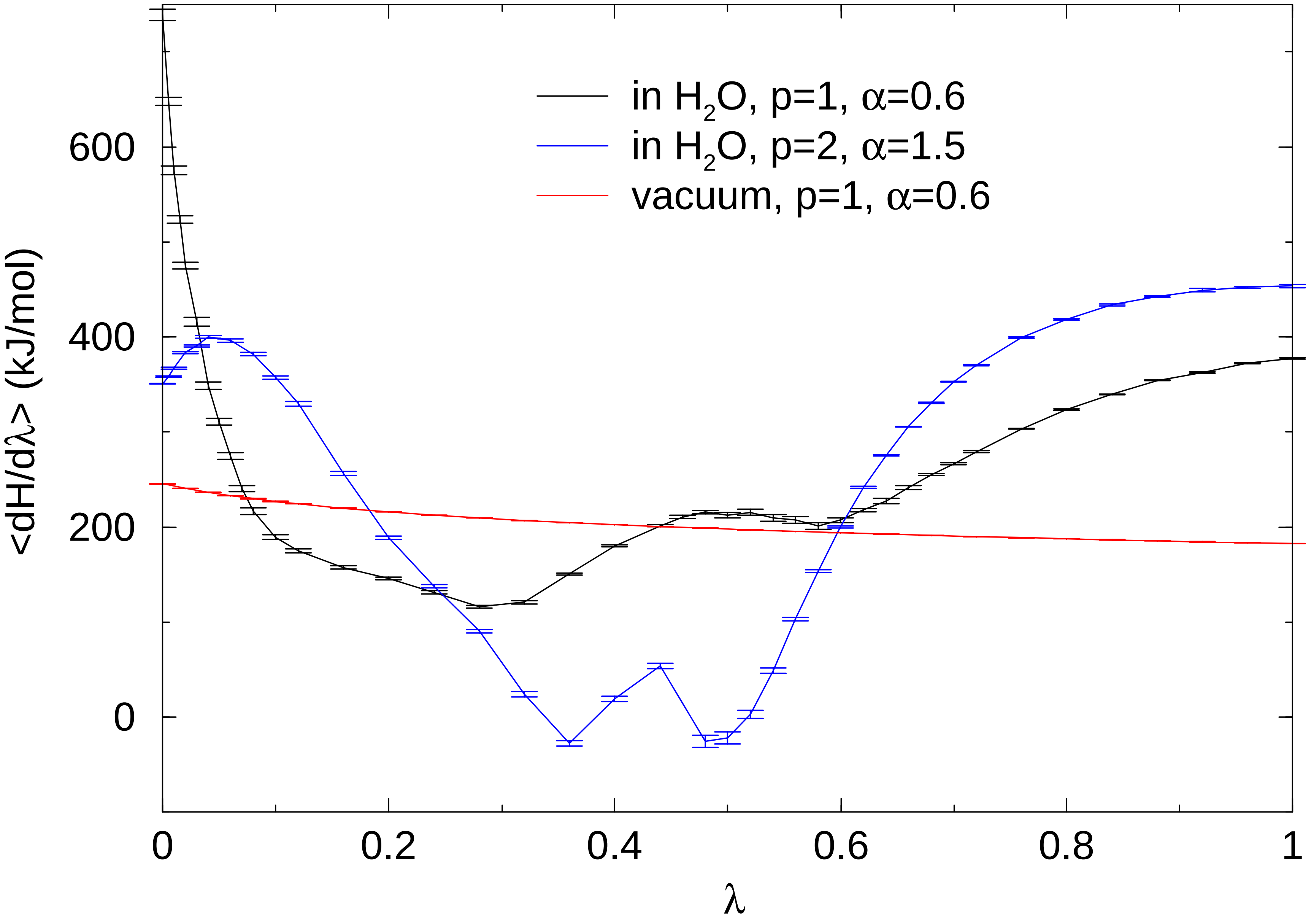
Turning off non-bonded interactions

In many cases it is more efficient to also turn off all non-bonded intramolecular interactions

(separate calculation of the intramolecular contribution required)



Solvation of ethanol in water



Free energy topology

```
; Include forcefield parameters
#include "ffoplsaa.itp"

[ moleculetype ]
; Name            nrexcl
methanol               3

[ atoms ]
;nr  type  resnr  res atom cgnr charge   mass    typeB chargeB massB
 1  opls_080   1  MET  CH3   1   0.266  15.035   opls_068  0  15.035
 2  opls_078   1  MET  O     1  -0.674  15.9994  opls_071  0  14.027
 3  opls_079   1  MET  H     1   0.408   1.008   opls_068  0  15.035

[ bonds ]
;  ai    aj funct   dA   kA   dB   kB
    1     2     1
    2     3     1

[ angles ]
;  ai    aj    ak funct  thetaA   kA    thetaB   kB
    1     2     3     1
```

Potential of mean force

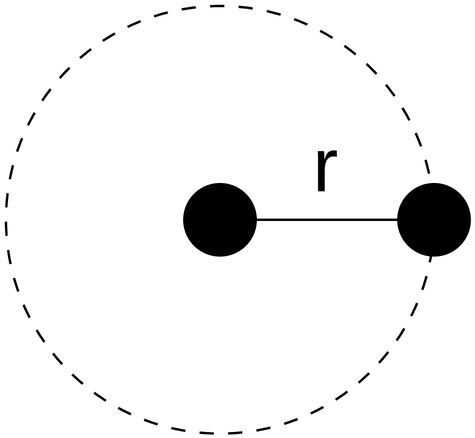
A potential of mean force (PMF) is the free energy along one or more degrees of freedom

The name comes from a common way to derive a PMF:
by integrating the mean force working on a certain degree of freedom

One can also obtain a potential of mean force by Boltzmann inverting a pair correlation function $g(r)$:

$$PMF(r) = -k_B T \log(g(r)) + C$$

Entropic effects



The phase-space volume available to the system is: $4\pi r^2$

Thus the entropic distribution is:

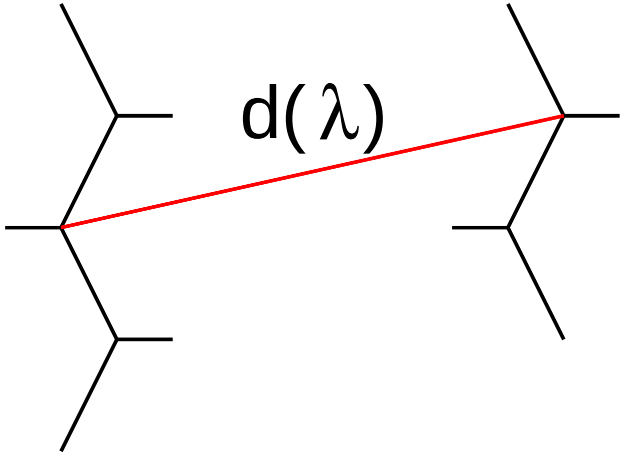
$$T\Delta S = k_B T \log(4\pi\, r^2) = 2k_B T \log(r) + C$$

This result can also be obtained by integrating the centrifugal force

$$F_c(r) = \frac{2k_B T}{r}$$

PMF with a coupling parameter

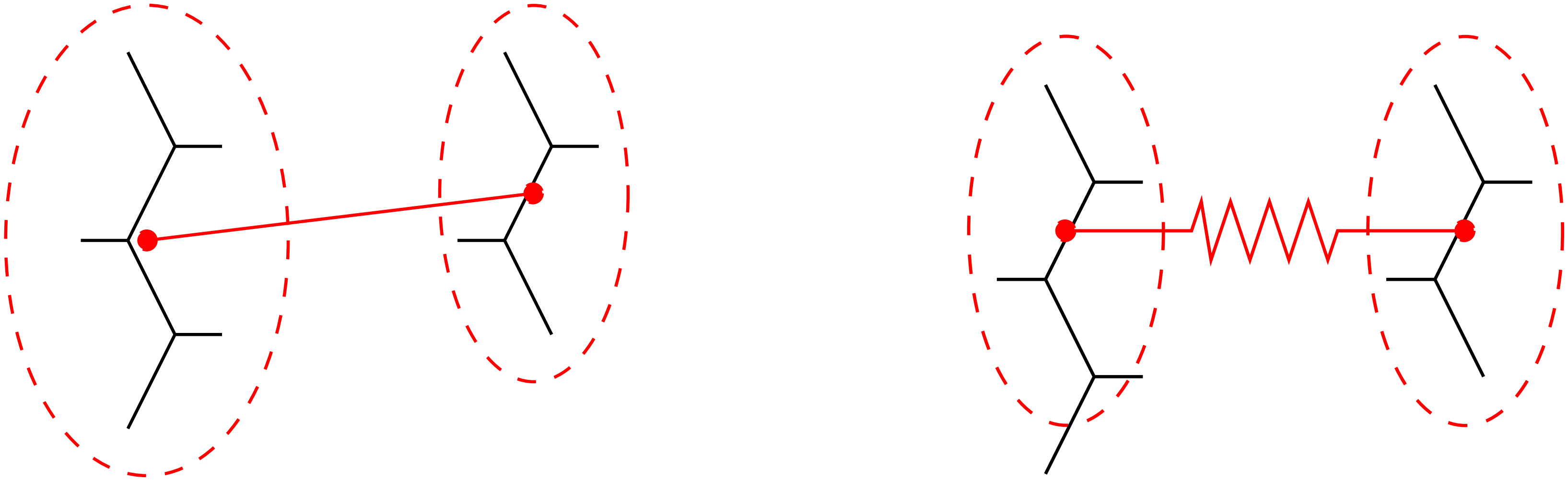
Since constraints are part of the Hamiltonian, they can also be coupled with lambda



The mean force is given by: $-\left\langle \frac{\partial H}{\partial \lambda} \right\rangle$

With this approach one can one make PMF for distances between atoms, not between centers of mass

PMF with the pull code

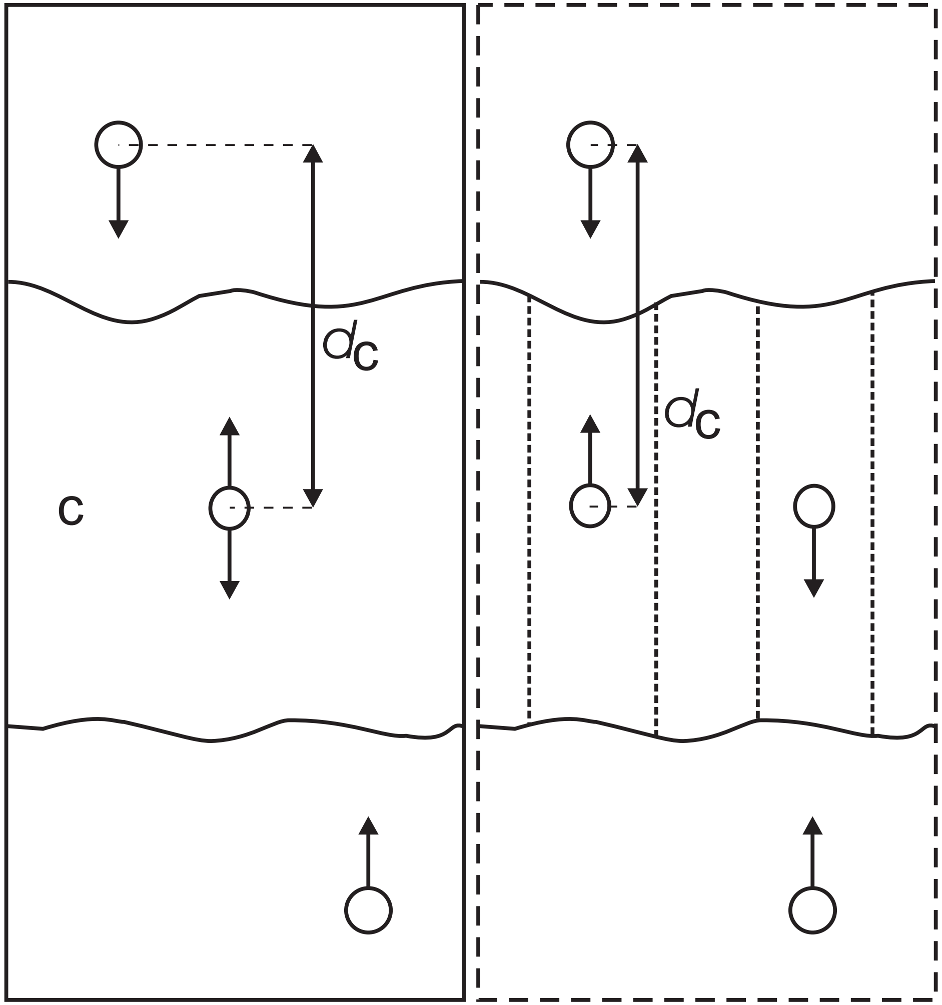


With a constraint: PMF between whole molecules

With a spring (umbrella sampling): PMF between anything
The mean force is the mean force working on the spring

Confusingly there are similar option AFM and $umbrella$

The pull code for membranes



The pull code in practice

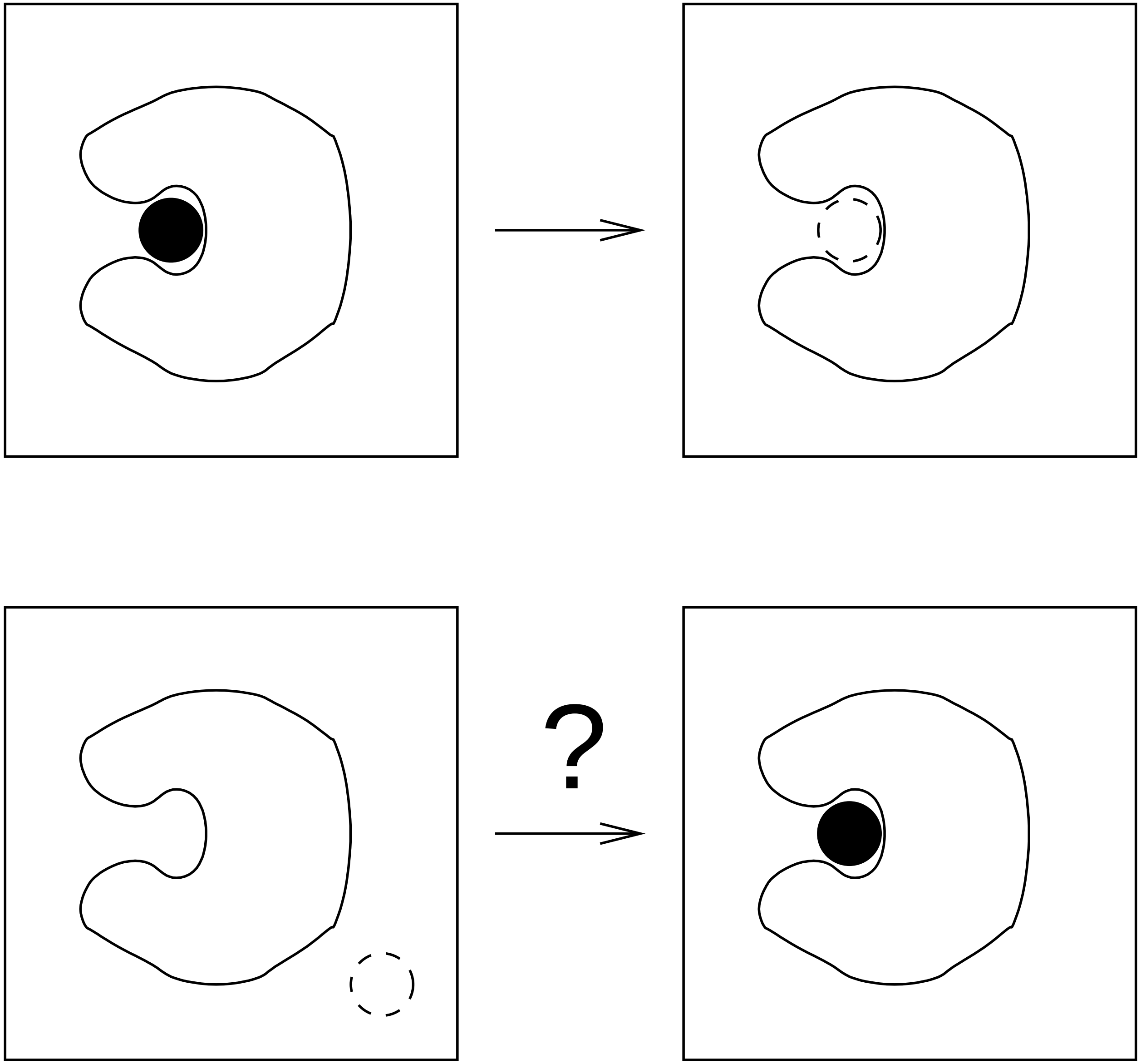
The pull code is controlled by extra input files for `mdrun`

An index file to define the groups

An `.ppa` file for the pull parameters:
Constraint pulling: constraint length, pull rate
“Spring” pulling: distance, pull rate, force constant

Output: `pull.pdo` with the pull forces

Reversibility



Reversibility check

The forward and backward path should give the same answer

So one can check:

ΔG making molecule dissappear

=

-ΔG make molecule appear

or

ΔG pulling molecule out of a protein

=

-ΔG pushing molecule into a protein

Free energy from non-equilibrium processes

If you change the Hamiltonian too fast, the ensemble will “lag behind” the change in the Hamiltonian: work partly irreversible
Total work W done on the system exceeds the reversible part ΔA:

$$W \geq \Delta A$$

In 1997 Jarzynski published the relation between the work and free energy difference for an *irreversible process* from λ=0 to 1:

$$A_1 - A_0 = -k_B T \log \left\langle e^{-\beta W} \right\rangle_{\lambda=0}$$

averaged over ensemble of initial points for $\lambda = 0$

Remarkable as W is very much process dependent

All methods up till now

Jarzynski's relation:

$$A_1 - A_0 = -k_B T \log \left\langle e^{-\beta W} \right\rangle_{\lambda=0}$$

Free energy perturbation is Jarzynski in one step

Free energy integration is the slow limit of Jarzynski's equation

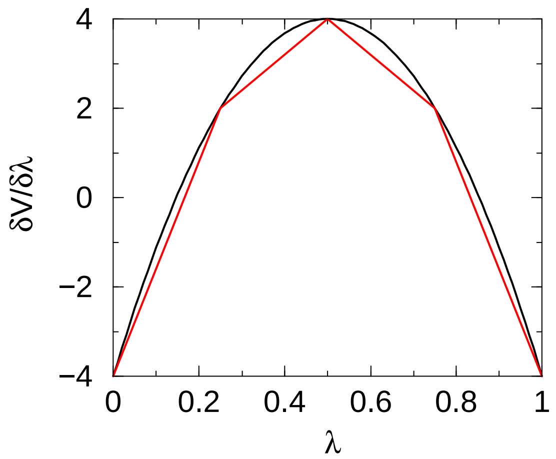
Slow growth is incorrect, as it should use Jarzynski's relation

Jarzynski's relation often not of practical use, as exponential averaging is inconvenient; only works well for $\sigma(W) < 2\, k_B T$

Free energy integration error

For highly curved $\partial V/\partial\lambda$, as is often the case, the discretized integral will give a systematic error

Solution:
use Bennett acceptance ratio method



Instead of $\partial V/\partial\lambda$, calculate:
$\langle V(\lambda_{i+1}) - V(\lambda_i)\rangle_{\lambda_i}$ and
$\langle V(\lambda_{i+1}) - V(\lambda_i)\rangle_{\lambda_{i+1}}$
and from that determine $A(\lambda_{i+1}) - A(\lambda_i)$
using a maximum likelyhood estimate

This is the method of choice

Other non-equilibrium methods

Crooks method using many transitions between λ=0 and λ=1 from two (or more) equilibrium simulations at λ=0 and λ=1

Reading

Daan & Frenkel Part III.7

exercise will come

Estimating errors

Block averaging

Estimating error is important for any quantitative analysis

A simple method that can always be applied
is to split your data into m blocks and estimate the error by
assuming the blocks are independent

Say we have n data points: $x(i\Delta t)$, $i = 0, \ldots, n-1$

We can estimate the error of the average from m block averages B_j:

$$E = \sqrt{\frac{1}{m(m-1)} \sum_{j=0}^{m-1} (B_j - \langle B \rangle)^2}$$

$$B_j = \frac{1}{\ell} \sum_{i=j\,\ell}^{(j+1)\ell-1} x(i\Delta t) \quad , \qquad \ell = \frac{n}{m}$$

Block averaging

But the blocks will always be correlated

We can assume a double exponential correlation,
fast decay correlation time τ_1, slow decay τ_2:

$$\langle x(0)\, x(t)\rangle = ae^{-t/\tau_1} + (1-a)e^{-t/\tau_2}$$

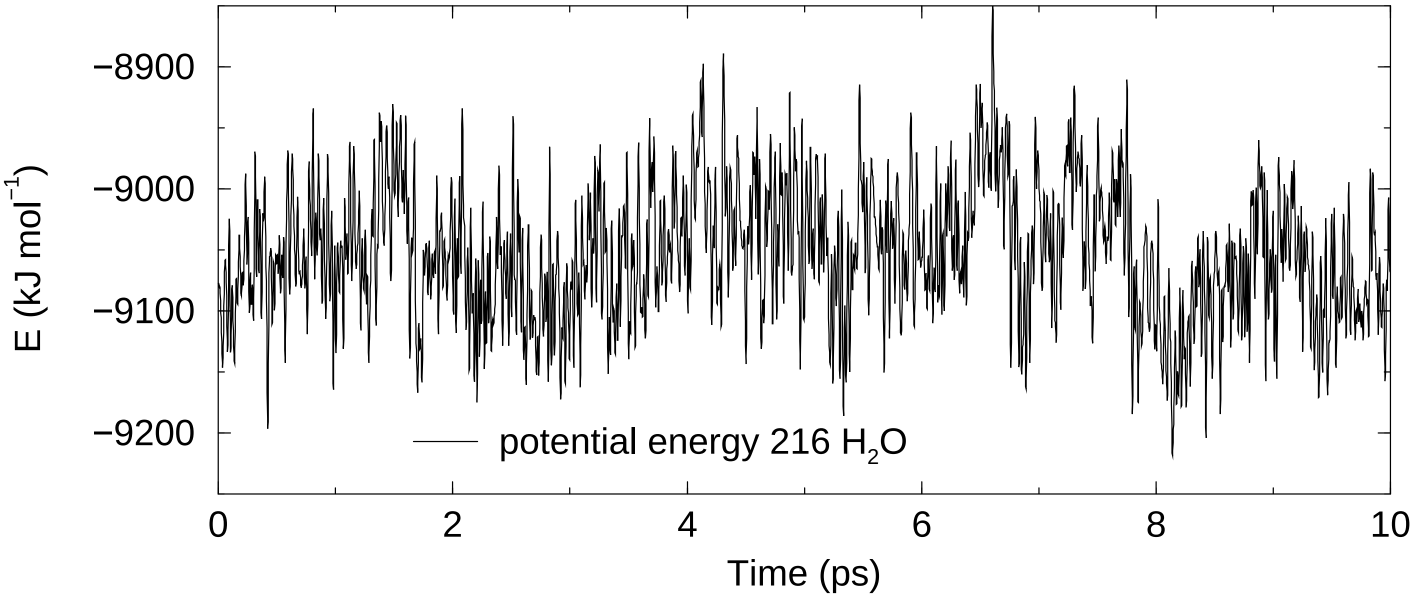
$$E^2(t) = \sigma^2 \frac{2}{n\Delta t}\left\{a\,\tau_1\left[(e^{-t/\tau_1}-1)\frac{\tau_1}{t}\right] + (1-a)\tau_2\left[(e^{-t/\tau_2}-1)\frac{\tau_2}{t}\right]\right\}$$

The error estimate for the average is then given by:

$$\lim_{t->\infty} E(t) = \sigma\sqrt{\frac{2}{n\Delta t}\{a\,\tau_1 + (1-a)\tau_2\}}$$

This analysis can be done with `g_analyze -ee`

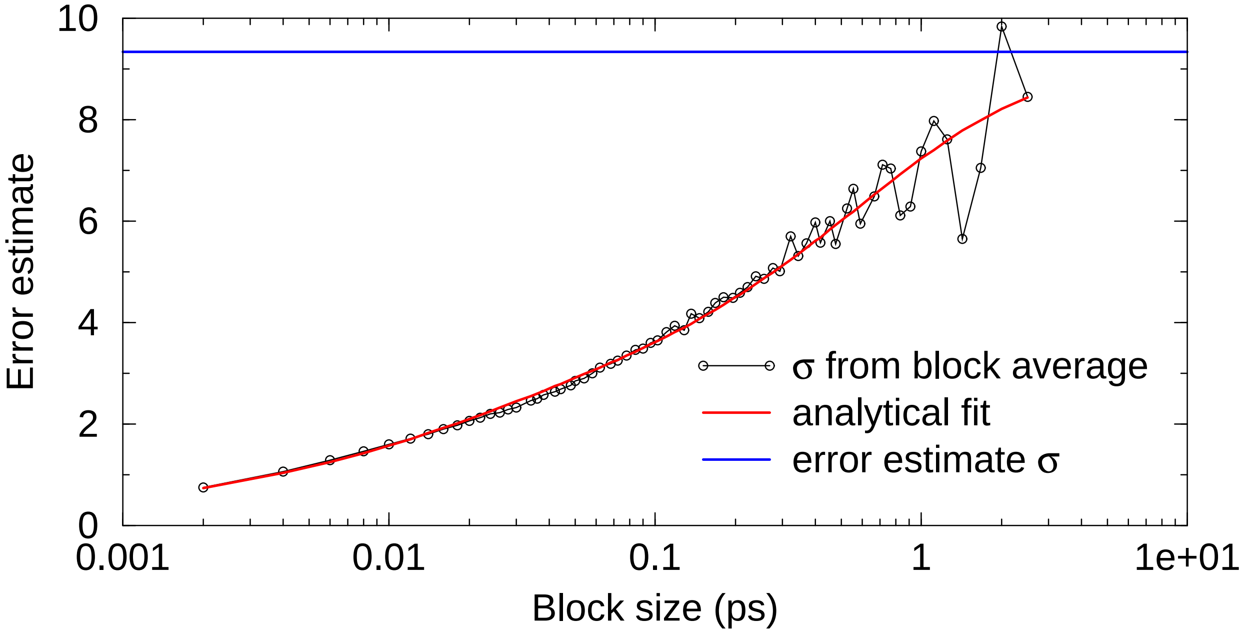
Error estimate example



a = 0.72

τ_1 = 0.016 ps

τ_2 = 0.50 ps


Error estimate
1e+01
Block size (ps)
σ from block average
analytical fit
error estimate σ