# Advanced Monte Carlo different ensembles

# Recall

- NVT simulations
  - Detailed balance
  - Generate a Boltzmann distribution
- Non-Boltzmann sampling
  - Recover correct sampling from a non-Boltzmann distribution
- Biased Monte Carlo
  - Bias the sampling
  - Remove the bias by changing the acceptance rule

# Non-Boltzmann sampling

$$\langle A \rangle_{NVT_1} = \frac{1}{Q_{NVT_1}} \frac{1}{\Lambda^{3N} N!} \int \mathrm{d}\mathrm{r}^N A\left(\mathrm{r}^N\right) \exp\left[-\beta_1 U\left(\mathrm{r}^N\right)\right]$$

$$= \frac{\int \mathrm{d}\mathrm{r}^N A\left(\mathrm{r}^N\right) \exp\left[-\beta_1 U\left(\mathrm{r}^N\right)\right]}{\int \mathrm{d}\mathrm{r}^N \exp\left[-\beta_1 U\left(\mathrm{r}^N\right)\right]}$$

T$_1$ is arbitrary!

Why are we not using this?

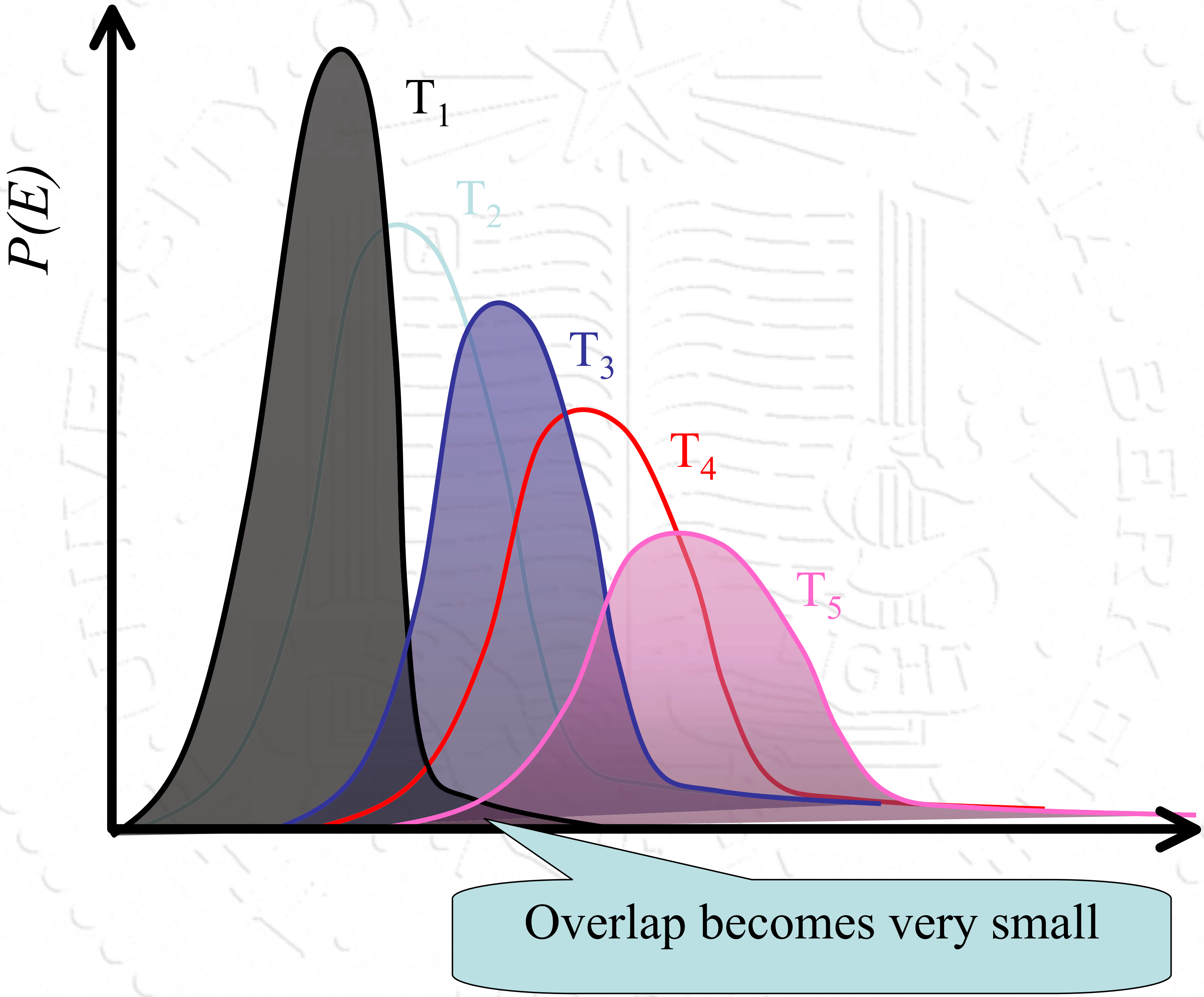
$$= \frac{\int \mathrm{d}\mathrm{r}^N A\left(\mathrm{r}^N\right) \exp\left[-\beta_1 U\left(\mathrm{r}^N\right)\right] \exp\left[\beta_2 U\left(\mathrm{r}^N\right) - \beta_2 U\left(\mathrm{r}^N\right)\right]}{\int \mathrm{d}\mathrm{r}^N \exp\left[-\beta_1 U\left(\mathrm{r}^N\right)\right] \exp\left[\beta_2 U\left(\mathrm{r}^N\right) - \beta_2 U\left(\mathrm{r}^N\right)\right]}$$

$$= \frac{\int \mathrm{d}\mathrm{r}^N A\left(\mathrm{r}^N\right) \exp\left[\beta_2 U\left(\mathrm{r}^N\right) - \beta_1 U\left(\mathrm{r}^N\right)\right] \exp\left[-\beta_2 U\left(\mathrm{r}^N\right)\right]}{\int \mathrm{d}\mathrm{r}^N \exp\left[\beta_2 U\left(\mathrm{r}^N\right) - \beta_1 U\left(\mathrm{r}^N\right)\right] \exp\left[-\beta_2 U\left(\mathrm{r}^N\right)\right]}$$

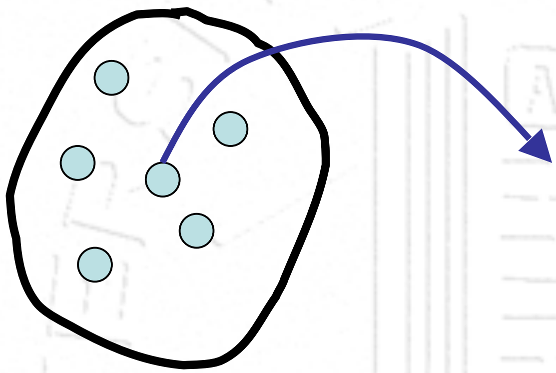
We only need a *single* simulation!

$$= \frac{\left\langle A \exp\left[\left(\beta_2 - \beta_1\right) U\right]\right\rangle_{NVT_2}}{\left\langle \exp\left[\left(\beta_2 - \beta_1\right) U\right]\right\rangle_{NVT_2}}$$

We perform a simulation at $T=T_2$ and we determine $A$ at $T=T_1$

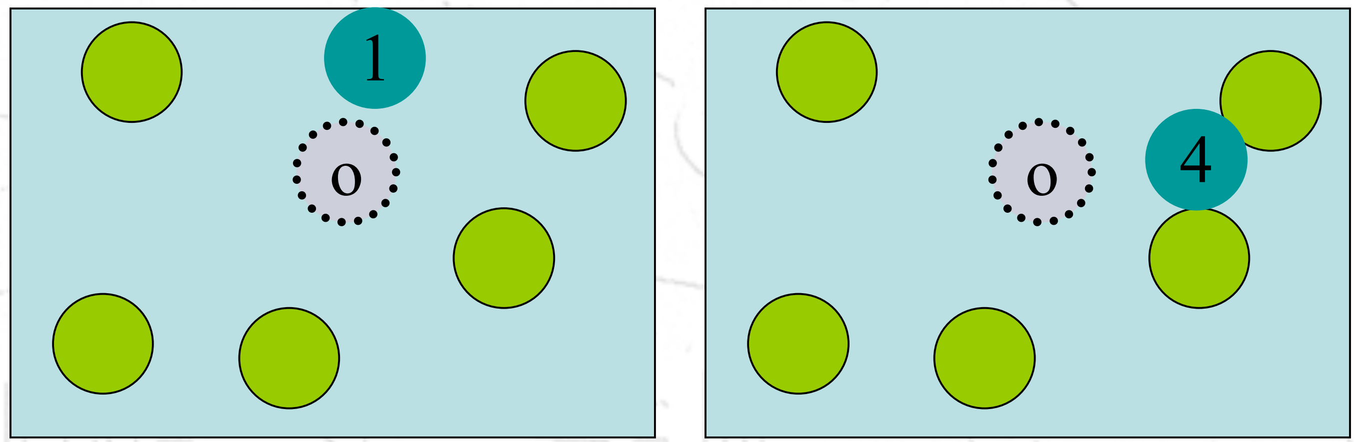

P(E)
T1
T2
T3
T4
T5
Overlap becomes very small

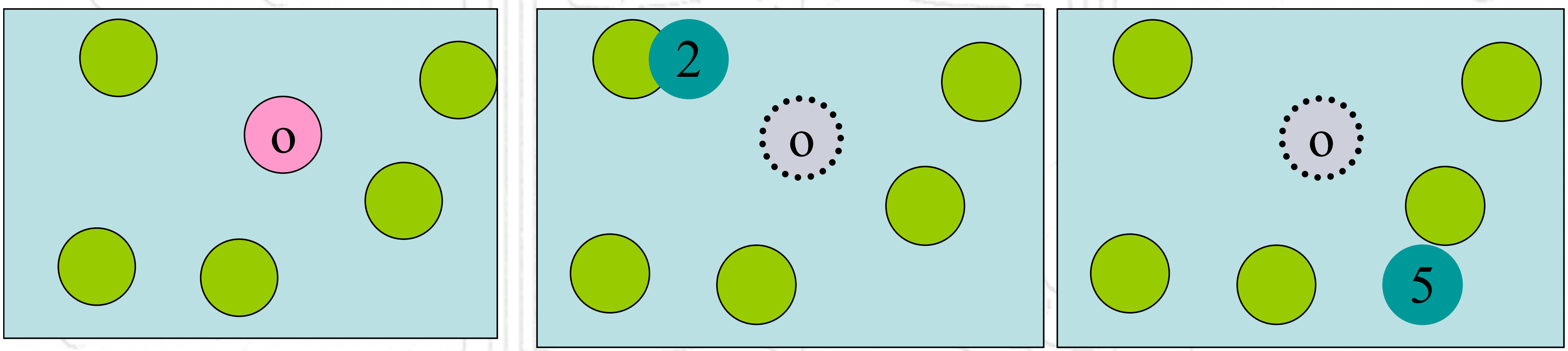
# How to do *parallel* Monte Carlo



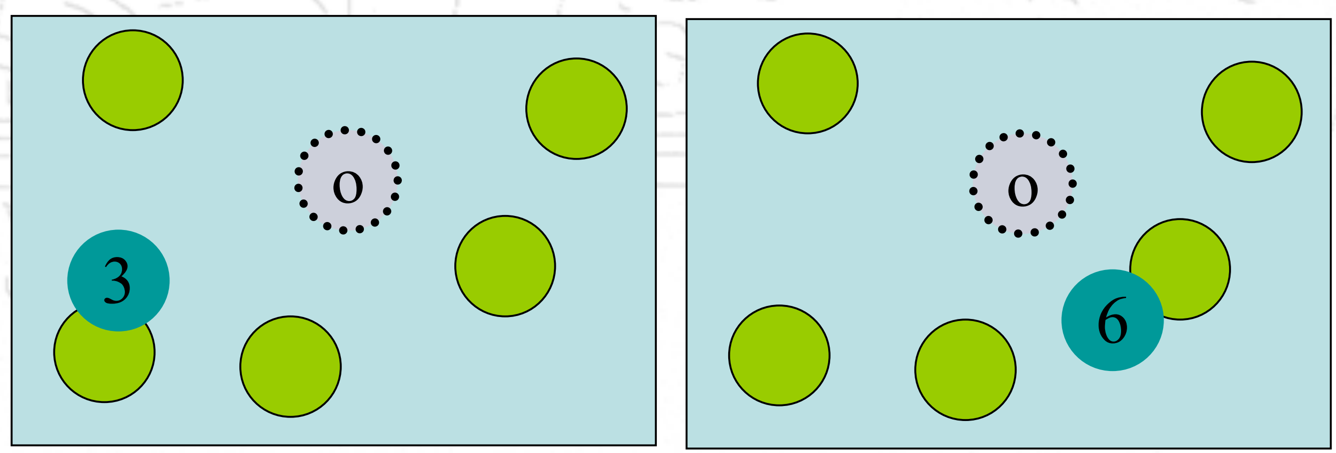
- Is it possible to do Monte Carlo in parallel
  - Monte Carlo is sequential!
  - We first have to know the fait of the current move before we can continue!

# Parallel Monte Carlo



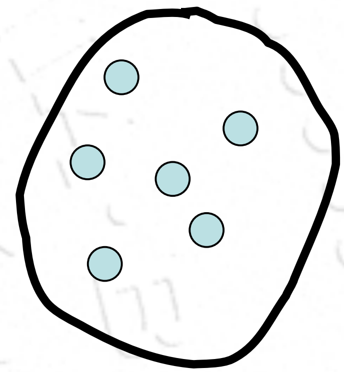




$$P(n) = \frac{\exp\left[-\beta\left(U_n\right)\right]}{\sum_{j=1}^{g} \exp\left[-\beta\left(U_j\right)\right]}$$



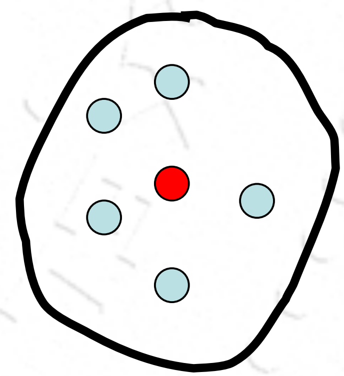
$$K(o \rightarrow n) = N(o) \times \alpha(o \rightarrow n) \times \mathrm{acc}(o \rightarrow n)$$

$$\alpha(o \rightarrow n) = \frac{\exp\left[-\beta\left(U_n\right)\right]}{\sum_{j=1}^{g} \exp\left[-\beta\left(U_j\right)\right]}$$



$$\alpha(o \rightarrow n) = \frac{\exp\left[-\beta\left(U_n\right)\right]}{W\left(n\right)}$$

$$\alpha(n \rightarrow o) = \frac{\exp\left[-\beta\left(U_o\right)\right]}{\sum_{j=1}^{g} \exp\left[-\beta\left(U_j\right)\right]}$$



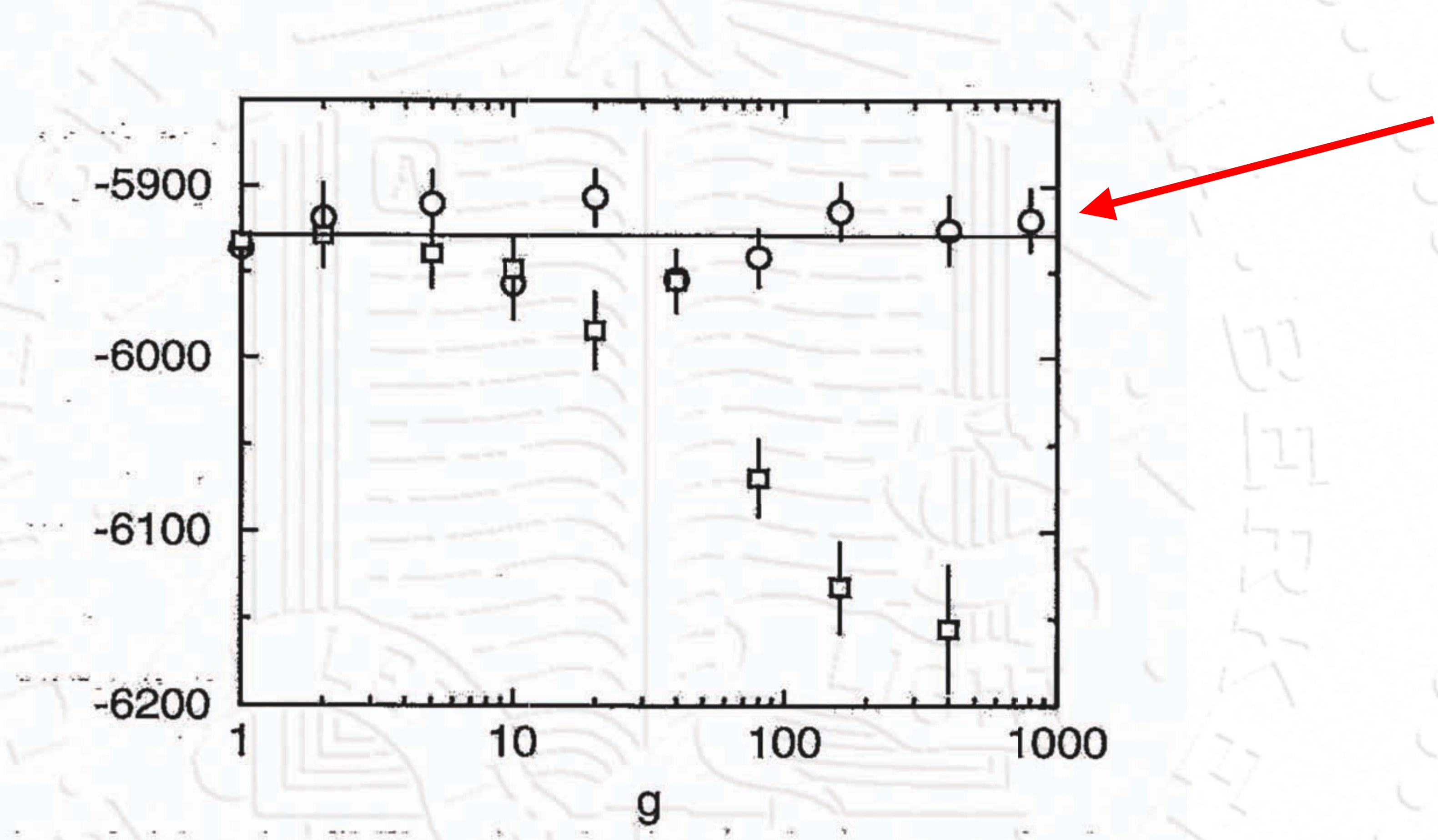
$$\alpha(n \rightarrow o) = \frac{\exp\left[-\beta\left(U_o\right)\right]}{W\left(o\right)}$$

$$\frac{\mathrm{acc}(o \rightarrow n)}{\mathrm{acc}(n \rightarrow o)} = \frac{N(n) \times \alpha(n \rightarrow o)}{N(o) \times \alpha(o \rightarrow n)}$$

$$\alpha(o \rightarrow n) = \frac{\exp\left[-\beta\left(U_n\right)\right]}{W\left(n\right)} \qquad \alpha(n \rightarrow o) = \frac{\exp\left[-\beta\left(U_o\right)\right]}{W\left(o\right)}$$

$$\frac{\mathrm{acc}(o \rightarrow n)}{\mathrm{acc}(n \rightarrow o)} = \frac{N(n) \times \dfrac{\exp\left[-\beta\left(U_o\right)\right]}{W\left(o\right)}}{N(o) \times \dfrac{\exp\left[-\beta\left(U_n\right)\right]}{W\left(n\right)}} = \frac{W(n)}{W(o)}$$

# Modified acceptance rule




Modified acceptance rule remove the ***bias*** exactly!

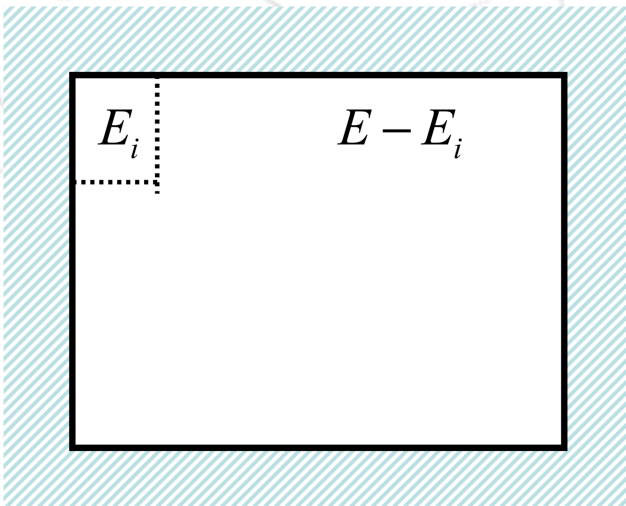
# Ensembles

- Micro-canonical ensemble: *E,V,N*
- Canonical ensemble: *T,V,N*
- Constant pressure ensemble: *T,P,N*
- Grand-canonical ensemble: *T,V,μ*

# Canonical ensemble

$1/k_B T$

Consider a small system that can exchange heat with a big reservoir




$$\ln \Omega\left(E - E_i\right) = \ln \Omega\left(E\right) - \frac{\partial \ln \Omega}{\partial E} E_i + \cdots$$

$$\ln \frac{\Omega\left(E - E_i\right)}{\Omega\left(E\right)} = -\frac{E_i}{k_B T}$$

Hence, the probability to find $E_i$:

$$P\left(E_i\right) = \frac{\Omega\left(E - E_i\right)}{\sum_j \Omega\left(E - E_j\right)} = \frac{\exp\left(-E_i / k_B T\right)}{\sum_j \exp\left(-E_j / k_B T\right)}$$

$$P\left(E_i\right) \propto \exp\left(-E_i / k_B T\right)$$

Boltzmann distribution

# Statistical Thermodynamics

Partition function

$$Q_{NVT} = \frac{1}{\Lambda^{3N} N!} \int \mathrm{dr}^N \exp\left[-\beta U\left(\mathrm{r}^N\right)\right]$$

Ensemble average

$$\langle A \rangle_{NVT} = \frac{1}{Q_{NVT}} \frac{1}{\Lambda^{3N} N!} \int \mathrm{dr}^N A\left(\mathrm{r}^N\right) \exp\left[-\beta U\left(\mathrm{r}^N\right)\right]$$

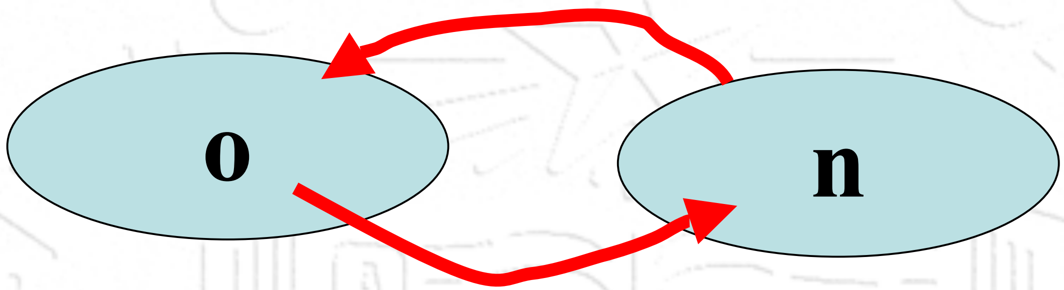
Probability to find a particular configuration

$$\mathrm{N}\left(\mathrm{r}^N\right) = \frac{1}{Q_{NVT}} \frac{1}{\Lambda^{3N} N!} \int \mathrm{dr'}^N \delta\left(\mathrm{r'}^N - \mathrm{r}^N\right) \exp\left[-\beta U\left(\mathrm{r'}^N\right)\right] \propto \exp\left[-\beta U\left(\mathrm{r}^N\right)\right]$$

Free energy

$$\beta F = -\ln\left(Q_{NVT}\right)$$

# Detailed balance




$$K(o \to n) = K(n \to o)$$

$$K(o \to n) = N(o) \times \alpha(o \to n) \times \mathrm{acc}(o \to n)$$

$$K(n \to o) = N(n) \times \alpha(n \to o) \times \mathrm{acc}(n \to o)$$

$$\frac{\mathrm{acc}(o \to n)}{\mathrm{acc}(n \to o)} = \frac{N(n) \times \alpha(n \to o)}{N(o) \times \alpha(o \to n)} = \frac{N(n)}{N(o)}$$

# *NVT*-ensemble

$$N(n) \propto \exp\left[-\beta U\left(n\right)\right]$$

$$\frac{\mathrm{acc}(o \to n)}{\mathrm{acc}(n \to o)} = \frac{N(n)}{N(o)}$$

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

## Algorithm 2 (Attempt to Displace a Particle)

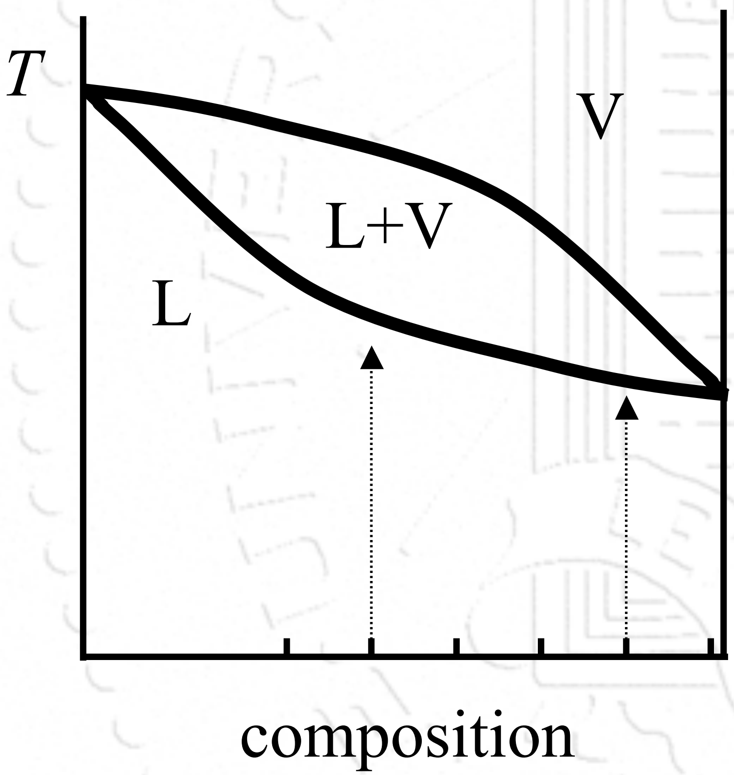
```
      SUBROUTINE mcmove               attempts to displace a particle

      o=int(ranf()*npart)+1           select a particle at random
      call ener(x(o),eno)             energy old configuration
      xn=x(o)+(ranf()-0.5)*delx       give particle random displacement
      call ener(xn,enn)               energy new configuration
      if (ranf().lt.exp(-beta         acceptance rule (3.2.1)
     +    *(enn-eno)) x(o)=xn         accepted: replace x(o) by xn
      return
      end
```

*Comments to this algorithm:*

1. *Subroutine* `ener` *calculates the energy of a particle at the given position.*
2. *Note that, if a configuration is rejected, the old configuration is retained.*
3. *The* `ranf()` *is a random number uniform in* $[0, 1]$.

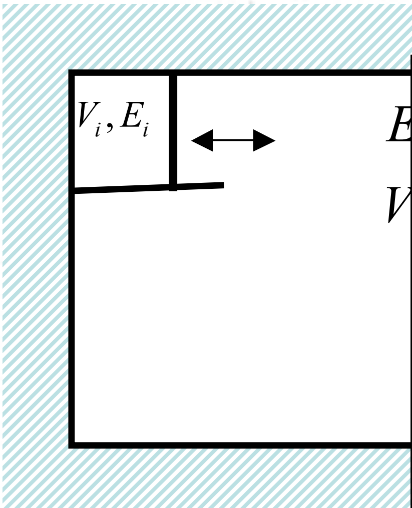
# Example (1): vapour-liquid equilibrium mixture




Measure the composition of the coexisting vapour and liquid phases if we start with a homogeneous liquid of two different compositions:

- Mimic this with the *N,V,T* ensemble?
- Or simulate at constant pressure?

# Constant pressure simulations: $N\,P\,T$ ensemble




**Thermo recall (4)**

First law of thermodynamics

$$\mathrm{d}E = T\mathrm{d}S - p\mathrm{d}V + \sum_i \mu_i \mathrm{d}N_i$$

Hence

$$\frac{1}{T} = \left(\frac{\partial S}{\partial E}\right)_{V,N}$$

and

$$\left(\frac{\partial S}{\partial V}\right)_{T,N} = \frac{p}{T}$$

$p/k_B T$

n that can exchange with a big reservoir

$$\ln\Omega\left(V - V_i, \ldots\right) \ldots - \left(\frac{\partial \ln\Omega}{\partial V}\right)_E V_i + \cdots$$

$$\ln\frac{\Omega\left(E - E_i, \ldots\right)}{\Omega\left(E, \ldots\right)}$$

Hence, the prob

$$P\left(E_i, V_i\right) = \frac{\ldots}{\sum_{j,k}\Omega\left(E - E_j, V - V_k\right)} = \frac{\ldots\beta\left(E_i + pV_i\right)\right]}{\sum_{j,k}\exp\left[-\beta\left(E_j + pV_k\right)\right]}$$

$$\propto \exp\left[-\beta\left(E_i + pV_i\right)\right]$$

# NPT Ensemble

Partition function:

$$Q_{NPT} = \frac{\beta P}{N!\Lambda^{3N}} \int \mathrm{d}V \exp\left[-\beta PV\right] \int \mathrm{dr}^N \exp\left[-\beta U\left(\mathrm{r}^N\right)\right]$$

Scaled coordinates $s_x = r_x / L_x$

$$Q_{NPT} = \frac{\beta P}{N!\Lambda^{3N}} \int \mathrm{d}V \exp\left[-\beta PV\right] V^N \int \mathrm{ds}^N \exp\left[-\beta U\left(\mathrm{s}^N; L\right)\right]$$

Probability to find a particular configuration:

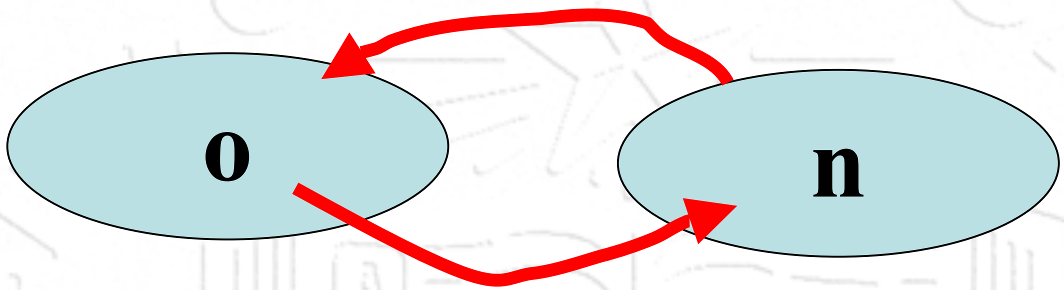
$$N_{NPT}\left(V, \mathbf{s}^N\right) \propto V^N \exp\left[-\beta PV\right] \exp\left[-\beta U\left(\mathrm{s}^N; L\right)\right]$$

Sample a particular configuration:

- Change of volume
- Change of reduced coordinates

Acceptance rules ??

# Detailed balance




$$K(o \to n) = K(n \to o)$$

$$K(o \to n) = N(o) \times \alpha(o \to n) \times \mathrm{acc}(o \to n)$$

$$K(n \to o) = N(n) \times \alpha(n \to o) \times \mathrm{acc}(n \to o)$$

$$\frac{\mathrm{acc}(o \to n)}{\mathrm{acc}(n \to o)} = \frac{N(n) \times \alpha(n \to o)}{N(o) \times \alpha(o \to n)} = \frac{N(n)}{N(o)}$$

# *NPT*-ensemble

$$N_{NPT}\left(V,\mathbf{s}^N\right) \propto V^N \exp\left[-\beta PV\right]\exp\left[-\beta U\left(\mathrm{s}^N;L\right)\right]$$

$$\frac{\mathrm{acc}(o\rightarrow n)}{\mathrm{acc}(n\rightarrow o)} = \frac{N(n)}{N(o)}$$

Suppose we change the position of a randomly selected particle

$$\frac{\mathrm{acc}(o\rightarrow n)}{\mathrm{acc}(n\rightarrow o)} = \frac{\cancel{V^N \exp\left[-\beta PV\right]}\exp\left[-\beta U\left(\mathrm{s}_\mathrm{n}^\mathrm{N};L\right)\right]}{\cancel{V^N \exp\left[-\beta PV\right]}\exp\left[-\beta U\left(\mathrm{s}_\mathrm{o}^\mathrm{N};L\right)\right]}$$

$$= \frac{\exp\left[-\beta U\left(\mathrm{s}_\mathrm{n}^\mathrm{N};L\right)\right]}{\exp\left[-\beta U\left(\mathrm{s}_\mathrm{o}^\mathrm{N};L\right)\right]} = \exp\left\{-\beta\left[U\left(n\right)-U\left(o\right)\right]\right\}$$

# *NPT*-ensemble

$$N_{NPT}\left(V,\mathbf{s}^N\right)\propto V^N\exp\left[-\beta PV\right]\exp\left[-\beta U\left(\mathrm{s}^N;L\right)\right]$$

$$\frac{\mathrm{acc}(o\rightarrow n)}{\mathrm{acc}(n\rightarrow o)}=\frac{N(n)}{N(o)}$$

Suppose we change the ***volume*** of the system

$$\frac{\mathrm{acc}(o\rightarrow n)}{\mathrm{acc}(n\rightarrow o)}=\frac{V_n^{\ N}\exp\left[-\beta PV_n\right]\exp\left[-\beta U\left(\mathrm{s}^\mathrm{N};L_n\right)\right]}{V_o^{\ N}\exp\left[-\beta PV_o\right]\exp\left[-\beta U\left(\mathrm{s}^\mathrm{N};L_o\right)\right]}$$

$$=\left(\frac{V_n}{V_o}\right)^N\exp\left[-\beta P\left(V_n-V_o\right)\right]\exp\left\{-\beta\left[U\left(n\right)-U\left(0\right)\right]\right\}$$

# Algorithm: NPT

- Randomly change the position of a particle
- Randomly change the volume

## Algorithm 10 (Basic NPT-Ensemble Simulation)

```
PROGRAM mc_npt                          basic NPT ensemble simulation

do icycl=1,ncycl                        perform ncycl MC cycles
  ran=ranf()*(npart+1)+1
  if (ran.le.npart) then
    call mcmove                         perform particle displacement
  else
    call mcvol                          perform volume change
  endif
  if (mod(icycl,nsamp).eq.0)
+   call sample                         sample averages
enddo
end
```

## Algorithm 2 (Attempt to Displace a Particle)

```
      SUBROUTINE mcmove                   attempts to displace a particle

      o=int(ranf()*npart)+1               select a particle at random
      call ener(x(o),eno)                 energy old configuration
      xn=x(o)+(ranf()-0.5)*delx           give particle random displacement
      call ener(xn,enn)                   energy new configuration
      if (ranf().lt.exp(-beta             acceptance rule (3.2.1)
     +    *(enn-eno)) x(o)=xn             accepted: replace x(o) by xn
      return
      end
```

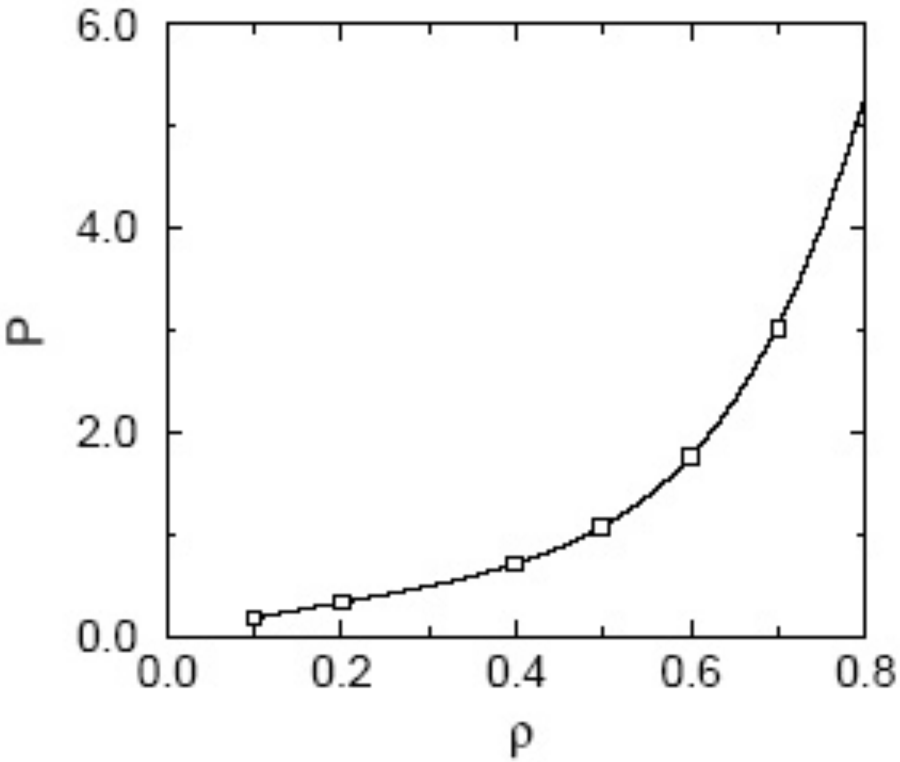
*Comments to this algorithm:*

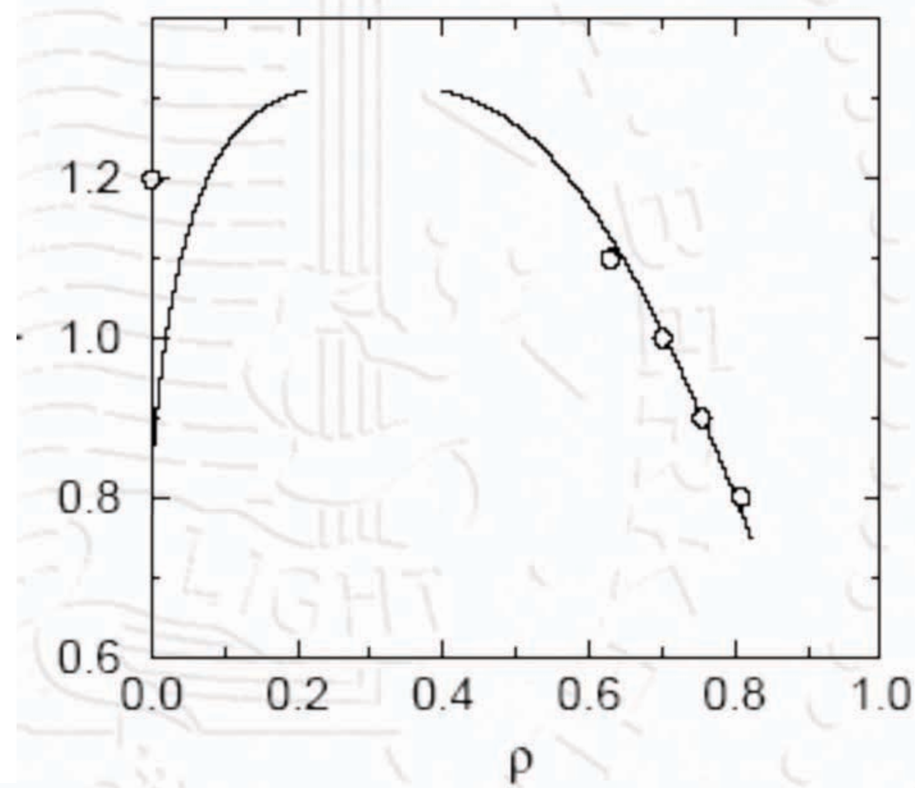
1. *Subroutine* `ener` *calculates the energy of a particle at the given position.*
2. *Note that, if a configuration is rejected, the old configuration is retained.*
3. *The* `ranf()` *is a random number uniform in* $[0, 1]$.

## Algorithm 11 (Attempt to Change the Volume)

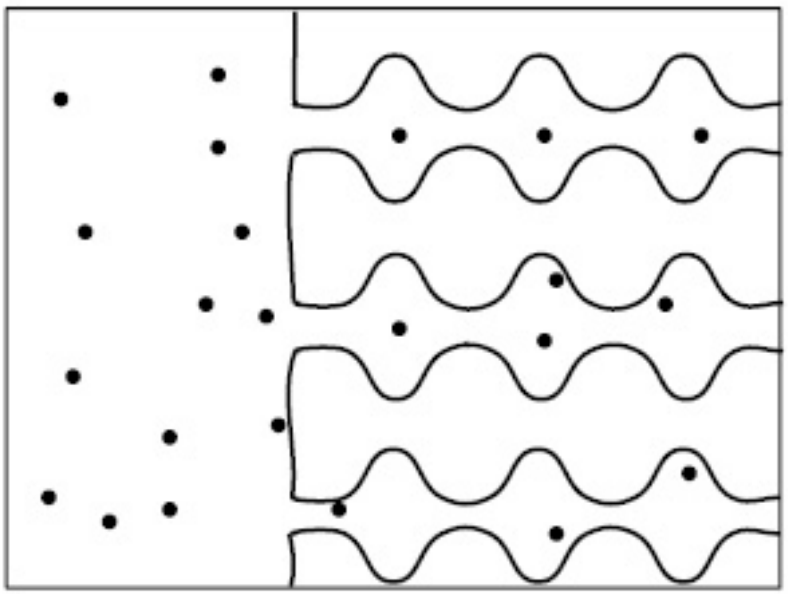
```
SUBROUTINE mcvol                          attempts to change
                                          the volume
call toterg(box,eno)                      total energy old conf.
vo=box**3                                 determine old volume
lnvn=log(vo)+(ranf()-0.5)*vmax            perform random walk in ln V
vn=exp(lnvn)
boxn=vn**(1/3)                            new box length
do i=1,npart
  x(i)=x(i)*boxn/box                      rescale center of mass
enddo
call toterg(boxn,enn)                     total energy new conf.
arg=-beta*((enn-eno)+p*(vn-vo)
+ -(npart+1)*log(vn/vo)/beta)             appropriate weight function!
if (ranf().gt.exp(arg)) then              acceptance rule (5.2.3)
  do i=1,npart                            REJECTED
    x(i)=x(i)*box/boxn                    restore the old positions
  enddo
endif
return
end
```

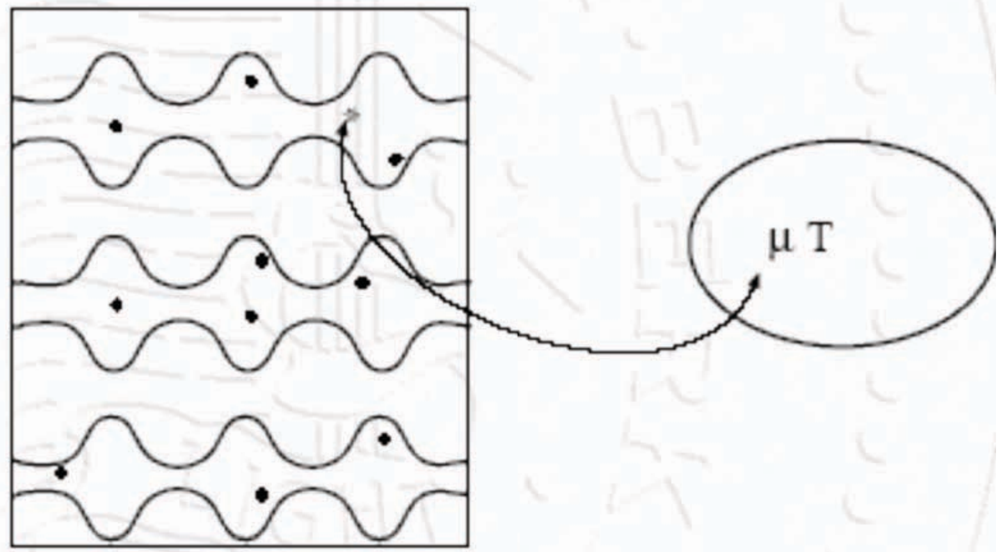
# NPT simulations



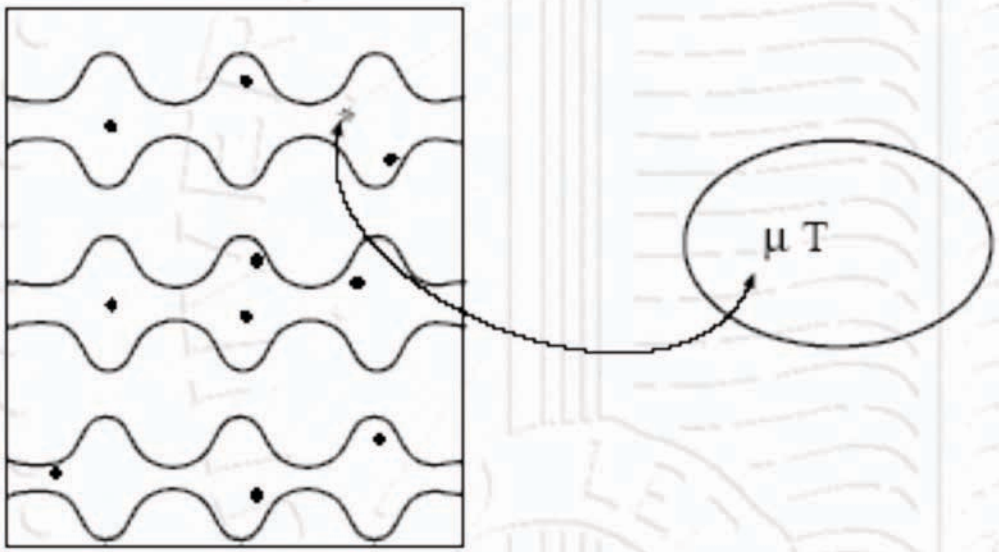
# Grand-canonical ensemble






What are the equilibrium conditions?

# Grand-canonical ensemble

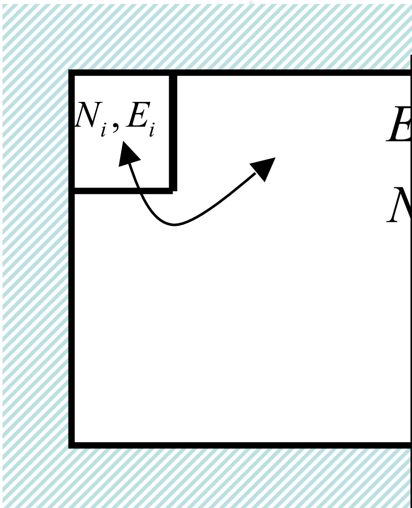



We impose:

- Temperature
- Chemical potential
- Volume

- But **NOT** pressure

# Grand-canonical simulations: $\mu V T$ ensemble




… n that ca… exchange … with a b… g reservoir

$-\mu/k_BT$

### Thermo recall (5)

First law of thermodynamics

$$\mathrm{d}E = T\mathrm{d}S - p\mathrm{d}V + \sum_i \mu_i \mathrm{d}N_i$$

Hence

$$\frac{1}{T} = \left(\frac{\partial S}{\partial E}\right)_{V,N}$$

and

$$\left(\frac{\partial S}{\partial N_i}\right)_{T,V} = -\frac{\mu_i}{T}$$

$$\ln\Omega\left(N - N_i, \ldots\right) \ldots - \left(\frac{\partial \ln\Omega}{\partial N}\right)_E N_i + \cdots$$

$$\ln\frac{\Omega\left(E - E_i, \ldots\right)}{\Omega\left(E, \ldots\right)} \ldots$$

Hence, the prob…

$$P\left(E_i, N_i\right) = \frac{\ldots}{\sum_{j,k}\Omega\left(E - E_j, N - N_k\right)} = \frac{\ldots\beta\left(E_i - \mu_i N_i\right)\big]}{\sum_{j,k}\exp\left[-\beta\left(E_j - \mu_k N_k\right)\right]}$$

$$\propto \exp\left[-\beta\left(E_i - \mu_i N_i\right)\right]$$

# MuVT Ensemble

Partition function:

$$Q_{\mu VT} = \sum_{N=0}^{N=\infty} \frac{\exp(\beta\mu N) V^N}{\Lambda^{3N} N!} \int \mathrm{d}s^N \exp\left[-\beta U\left(s^N; L\right)\right]$$

Probability to find a particular configuration:

$$N_{\mu VT}\left(N, \mathbf{s}^N\right) \propto \frac{\exp\left(\beta\mu N\right) V^N}{\Lambda \ldots} [illegible]$$

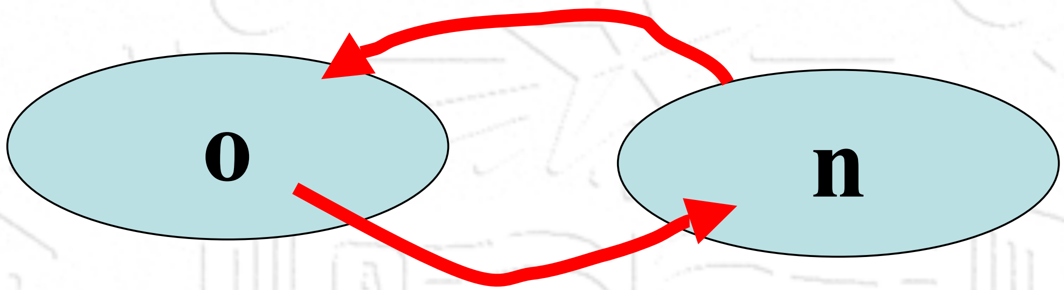
Sample a particular c[illegible]

- Change of the nu[illegible]
- Change of reduce[illegible]

Acceptance rules ??

*Detailed balance*

# Detailed balance




$$K(o \rightarrow n) = K(n \rightarrow o)$$

$$K(o \rightarrow n) = N(o) \times \alpha(o \rightarrow n) \times \mathrm{acc}(o \rightarrow n)$$

$$K(n \rightarrow o) = N(n) \times \alpha(n \rightarrow o) \times \mathrm{acc}(n \rightarrow o)$$

$$\frac{\mathrm{acc}(o \rightarrow n)}{\mathrm{acc}(n \rightarrow o)} = \frac{N(n) \times \alpha(n \rightarrow o)}{N(o) \times \alpha(o \rightarrow n)} = \frac{N(n)}{N(o)}$$

# $\mu VT$-ensemble

$$N_{\mu VT}\left(N,\mathbf{s}^N\right) \propto \frac{\exp\left(\beta\mu N\right)V^N}{\Lambda^{3N}N!}\exp\left[-\beta U\left(\mathrm{s}^N;L\right)\right]$$

$$\frac{\mathrm{acc}(o\rightarrow n)}{\mathrm{acc}(n\rightarrow o)}=\frac{N(n)}{N(o)}$$

Suppose we change the position of a randomly selected particle

$$\frac{\mathrm{acc}(o\rightarrow n)}{\mathrm{acc}(n\rightarrow o)}=\frac{\frac{\exp\left(\beta\mu N\right)V^N}{\Lambda^{3N}N!}\exp\left[-\beta U\left(\mathrm{s}_\mathrm{n}^\mathrm{N};L\right)\right]}{\frac{\exp\left(\beta\mu N\right)V^N}{\Lambda^{3N}N!}\exp\left[-\beta U\left(\mathrm{s}_\mathrm{o}^\mathrm{N};L\right)\right]}$$

$$=\exp\left\{-\beta\left[U\left(n\right)-U\left(0\right)\right]\right\}$$

# $\mu VT$-ensemble

$$N_{\mu VT}\left(N,\mathbf{s}^N\right) \propto \frac{\exp\left(\beta\mu N\right)V^N}{\Lambda^{3N}N!}\exp\left[-\beta U\left(\mathrm{s}^N;L\right)\right]$$

$$\frac{\mathrm{acc}(o\to n)}{\mathrm{acc}(n\to o)}=\frac{N(n)}{N(o)}$$

Suppose we change the ***number of particles*** of the system

$$\frac{\mathrm{acc}(o\to n)}{\mathrm{acc}(n\to o)}=\frac{\dfrac{\exp\left(\beta\mu\left(N+1\right)\right)V^{N+1}}{\Lambda^{3N+3}\left(N+1\right)!}\exp\left[-\beta U\left(\mathrm{s}^{\mathrm{N+1}};L_n\right)\right]}{\dfrac{\exp\left(\beta\mu N\right)V^N}{\Lambda^{3N}N!}\exp\left[-\beta U\left(\mathrm{s}^{\mathrm{N}};L_o\right)\right]}$$

$$=\frac{\exp\left(\beta\mu\right)V}{\Lambda^3\left(N+1\right)}\exp\left[-\beta\Delta U\right]$$

## Algorithm 12 (Basic Grand-Canonical Ensemble Simulation)

```
 PROGRAM mc_gc                              basic μVT ensemble
                                            simulation
 do icycl=1,ncycl                           perform ncycl MC cycles
  ran=int(ranf()*(npav+nexc))+1
  if (ran.le.npart) then
    call mcmove                             displace a particle
  else
    call mcexc                              exchange a particle
  endif                                     with the reservoir
  if (mod(icycl,nsamp).eq.0)
+   call sample                             sample averages
 enddo
 end
```

*Comments to this algorithm:*

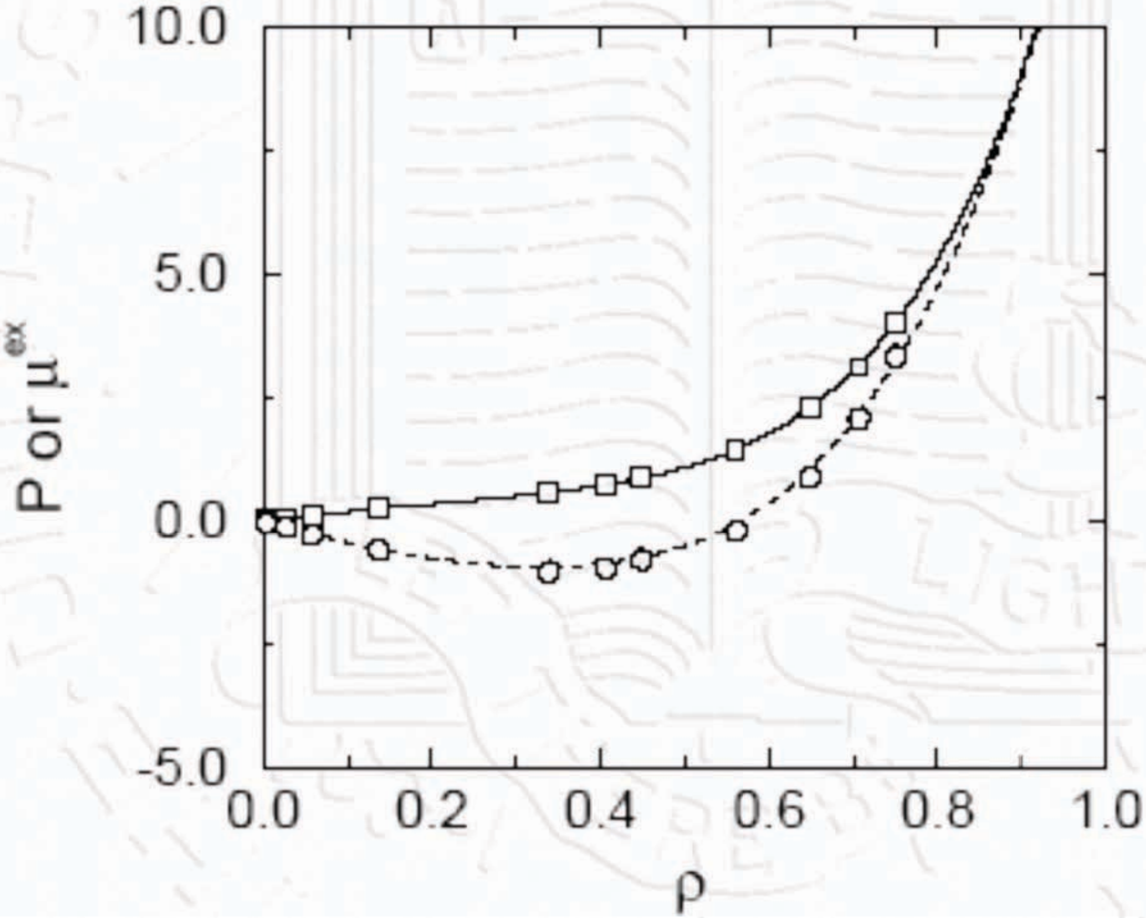
1. *This algorithm ensures that, after each MC step, detailed balance is obeyed. Per cycle we perform on average* `npav` *attempts*[6] *to displace particles and* `nexc` *attempts to exchange particles with the reservoir.*
2. *Subroutine* `mcmove` *attempts to displace a particle (Algorithm 2), subroutine* `mcexc` *attempts to exchange a particle with a reservoir (Algorithm 13), and subroutine* `sample` *samples quantities every* `nsamp` *cycle.*

## Algorithm 13 (Attempt to Exchange a Particle with a Reservoir)

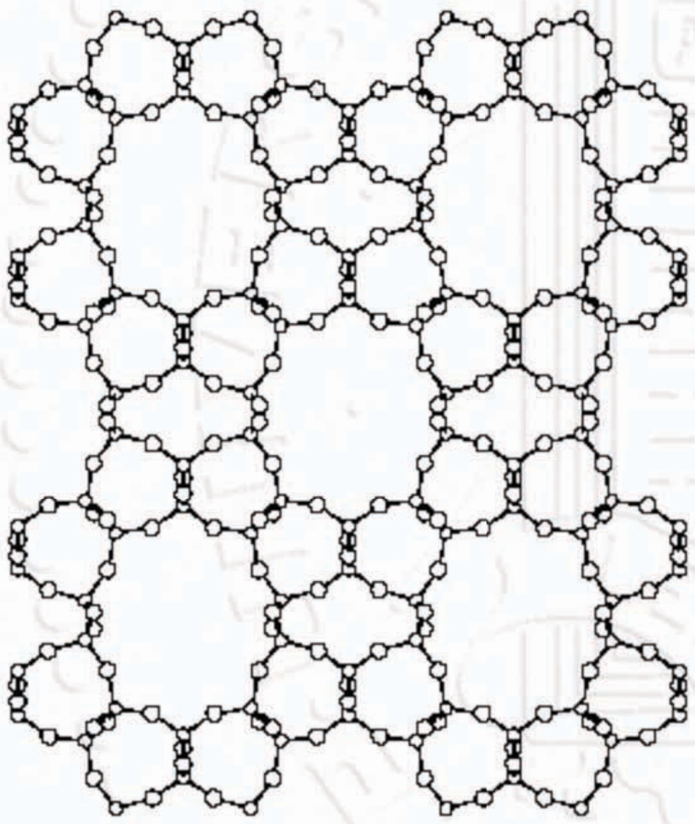
```
      SUBROUTINE mcexc                  attempt to exchange a particle
                                        with a reservoir
      if (ranf().lt.0.5) then           decide to remove or add a particle
        if (npart.eq.0) return          test whether there is a particle
        o=int(npart*ranf())+1           select a particle to be removed
        call ener(x(o),eno)             energy particle o
        arg=npart*exp(beta*eno)         acceptance rule (5.6.9)
     +        /(zz*vol)
        if (ranf().lt.arg) then
          x(o)=x(npart)                 accepted: remove particle o
          npart=npart-1
        endif
      else
        xn=ranf()*box                   new particle at a random position
        call ener(xn,enn)               energy new particle
        arg=zz*vol*exp(-beta*enn)       acceptance rule (5.6.8)
     +        /(npart+1)
        if (ranf().lt.arg) then
          x(npart+1)=xn                 accepted: add new particle
          npart=npart+1
        endif
      endif
      return
      end
```

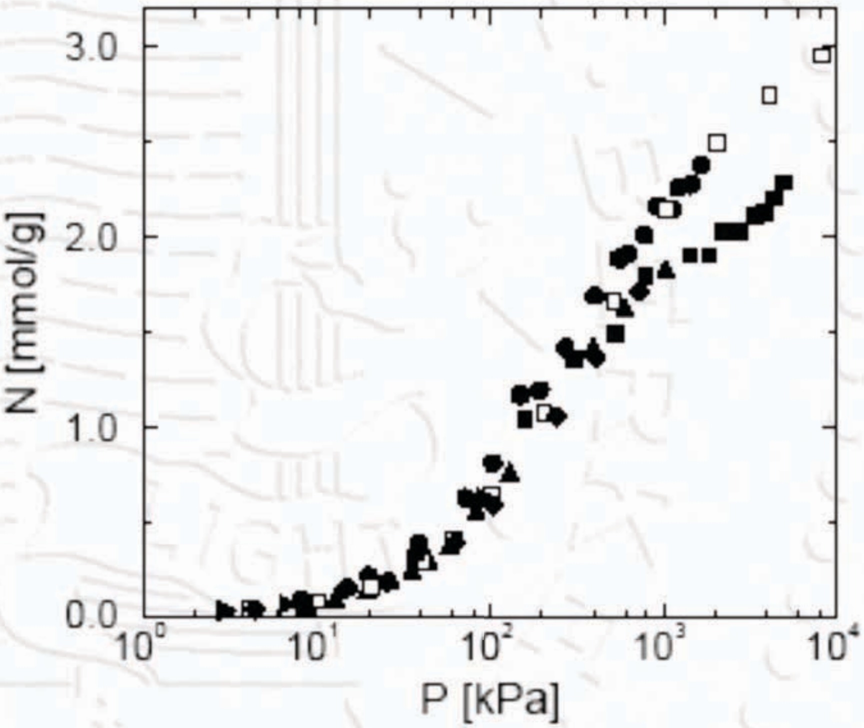
Comment to this algorithm:

# Application: equation of state of Lennard-Jones



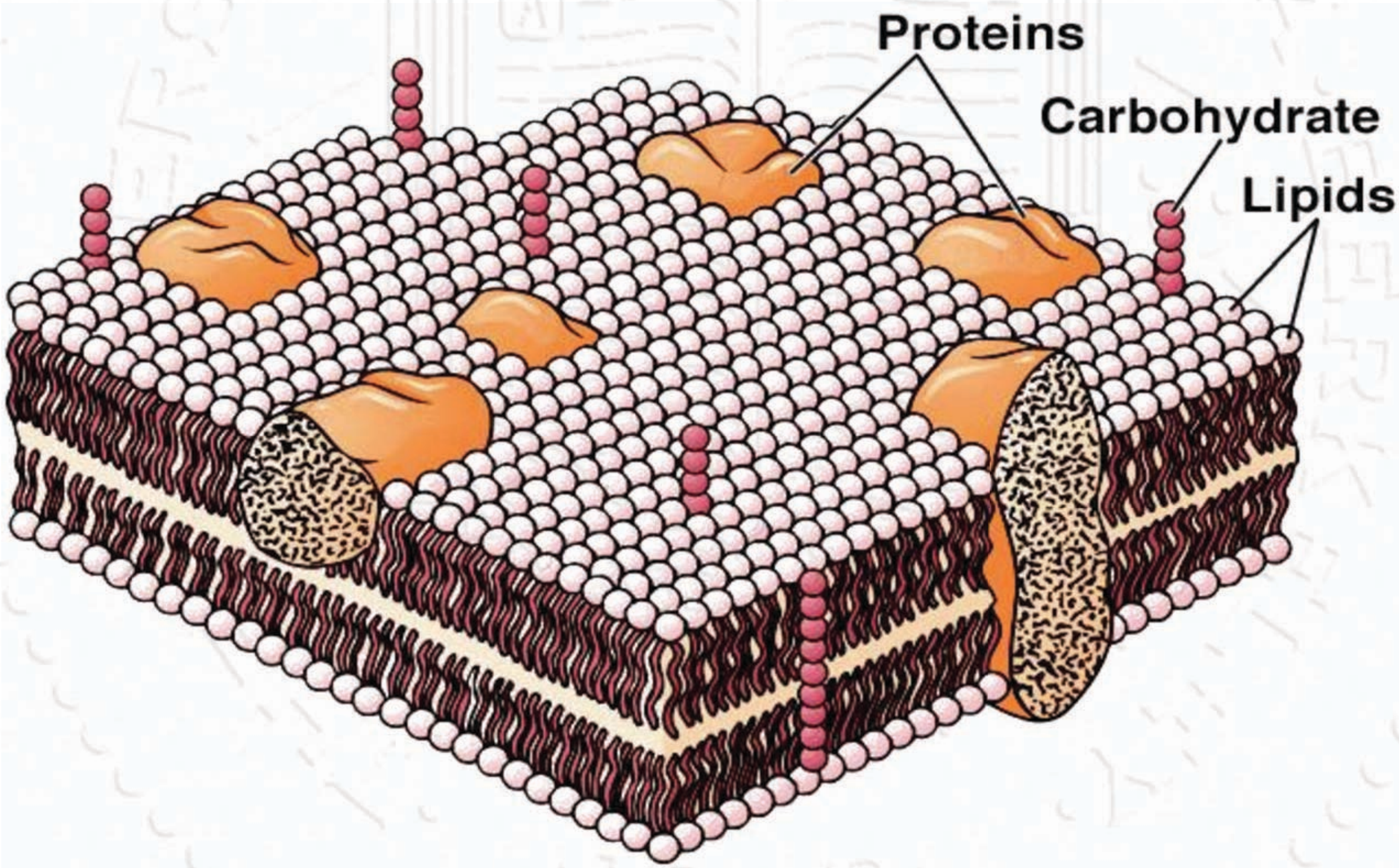
# Application: adsorption in zeolites



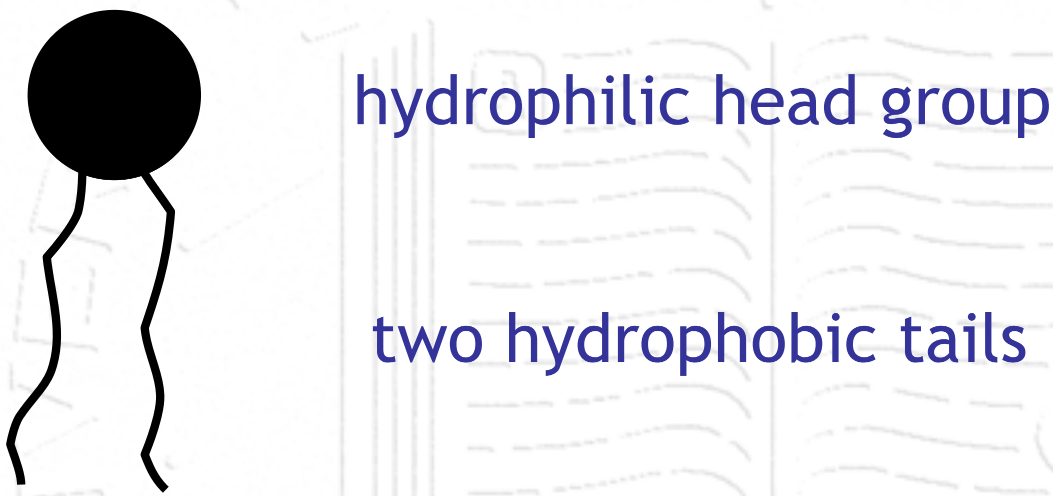


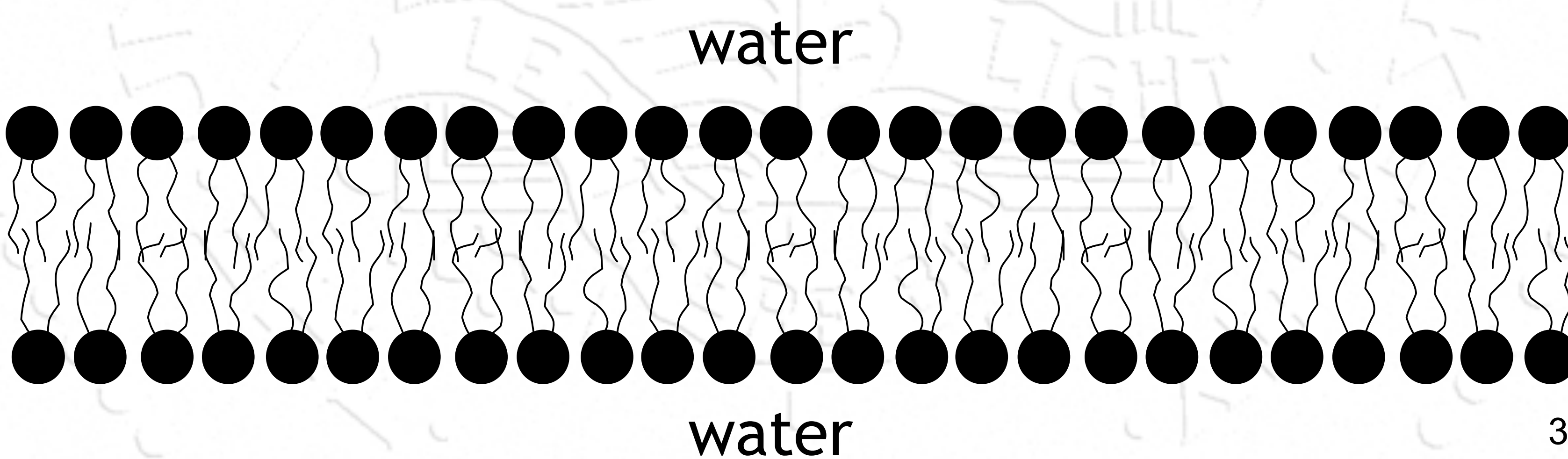
# Exotic ensembles

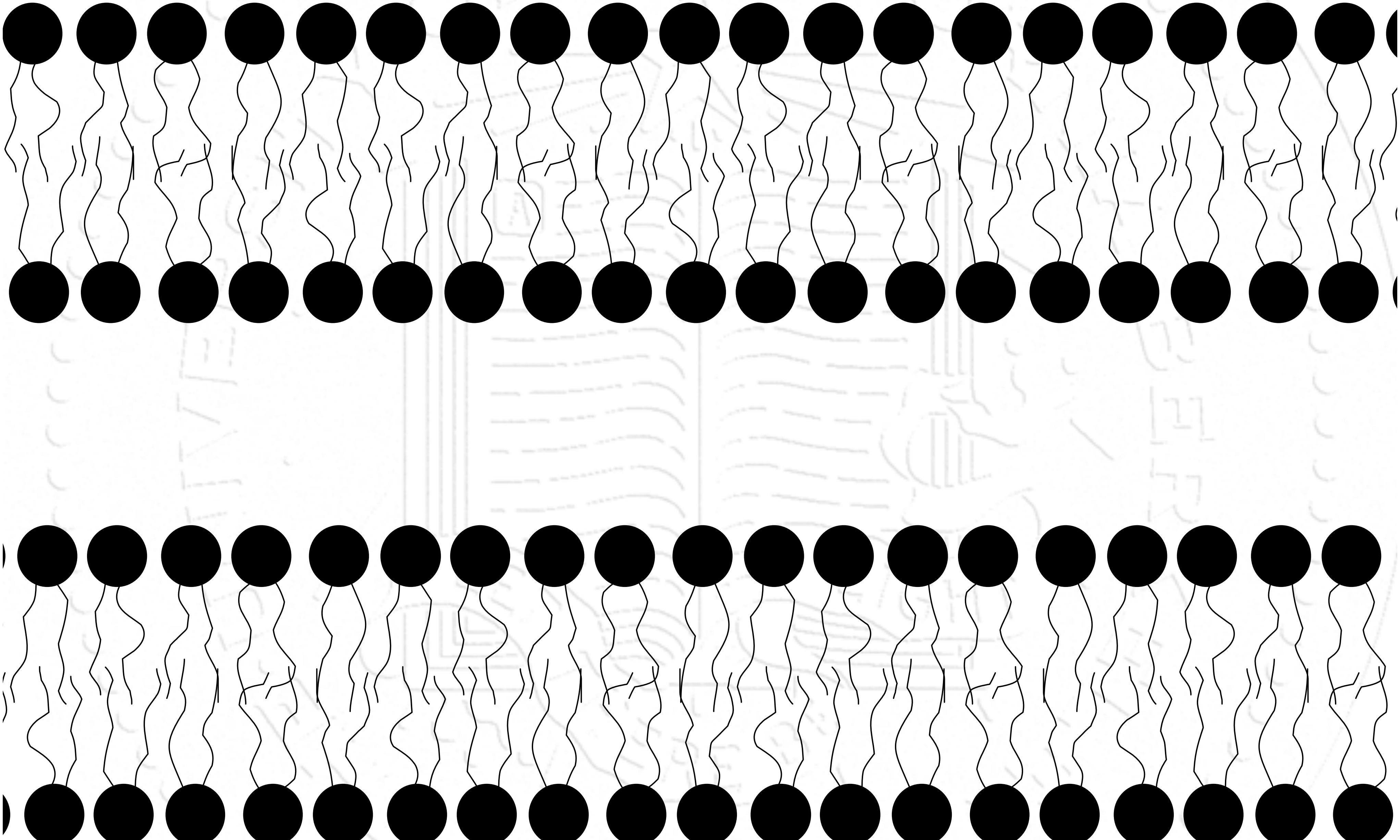
What to do with a biological membrane?



# Model membrane: Lipid bilayer



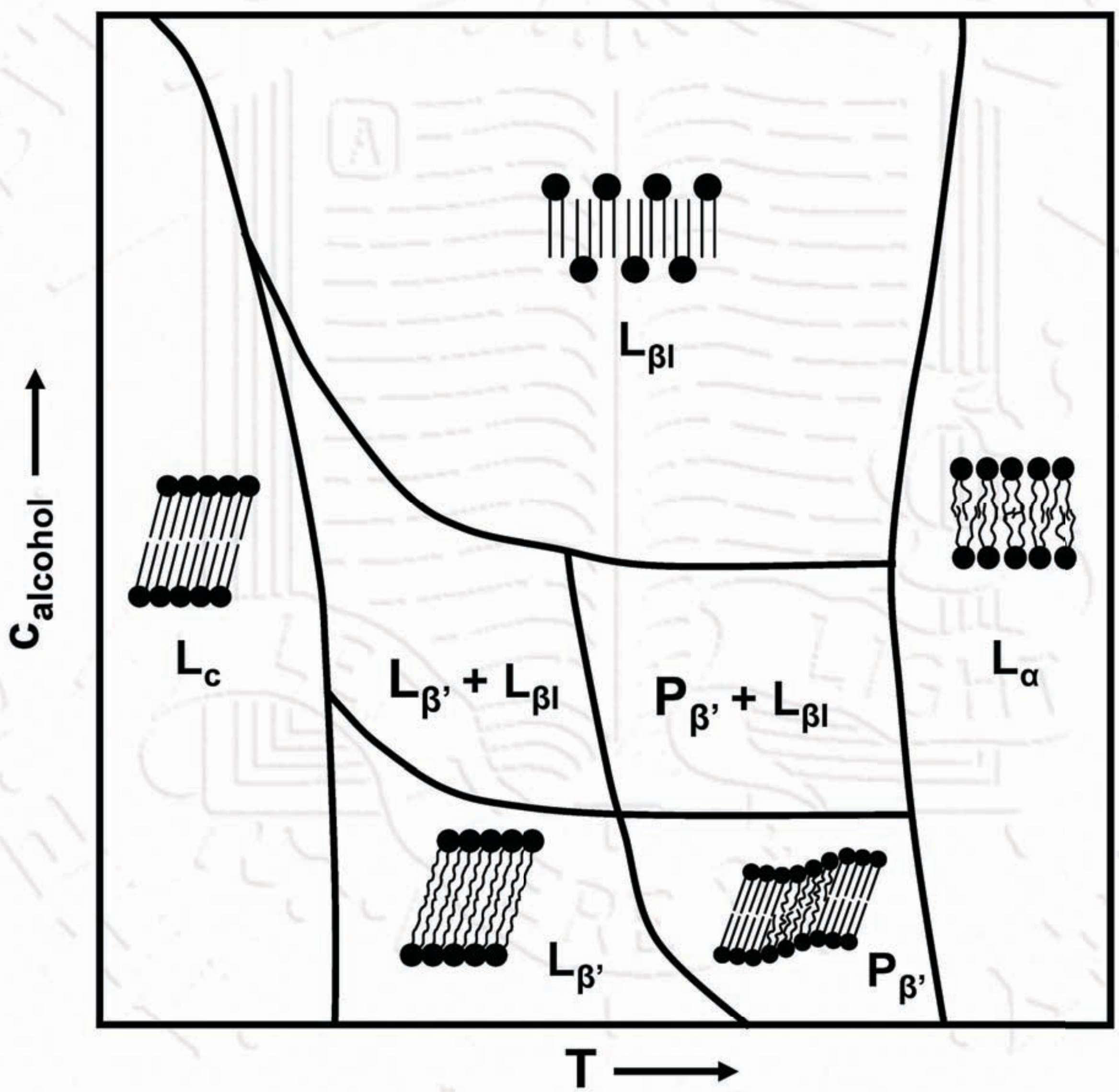





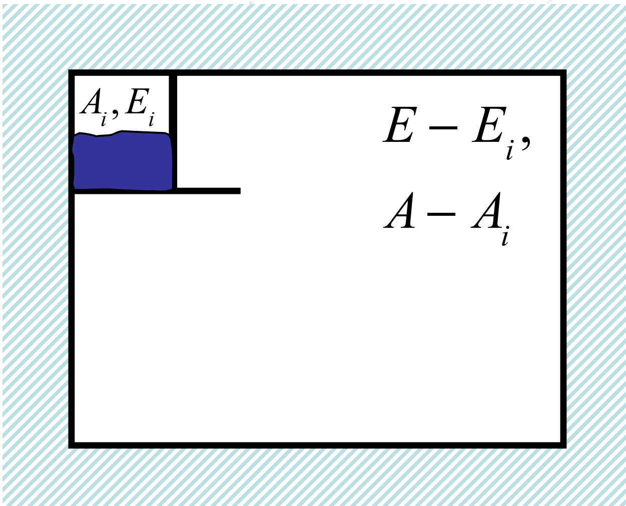
# Questions

- What is the surface tension of this system?
- What is the surface tension of a biological membrane?
- What to do about this?

# Phase diagram: alcohol



# Constant surface tension simulations: $\gamma, V, T$ ensemble



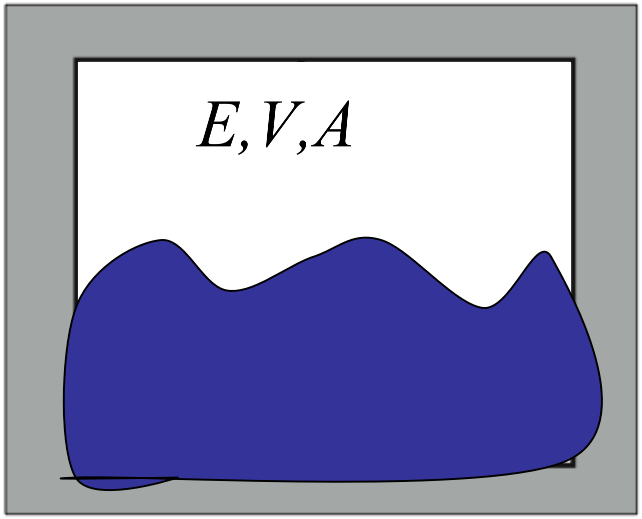

Consider a small system that can exchange
*ar* rgy with oir

$1/k_B T$

???

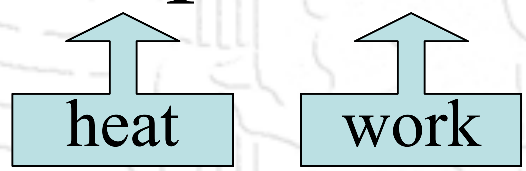
$$\ln\Omega\left(E - E_{i,} A - A_i\right) = \ln\Omega\left(E, A\right) - \left(\frac{\partial \ln\Omega}{\partial E}\right)_A E_i - \left(\frac{\partial \ln\Omega}{\partial A}\right)_E A_i + \cdots$$

# First law of thermodynamic




Conservation of energy:

$$\mathrm{d}E = \mathrm{d}q + \mathrm{d}w$$




γ is the interfacial tension of the system

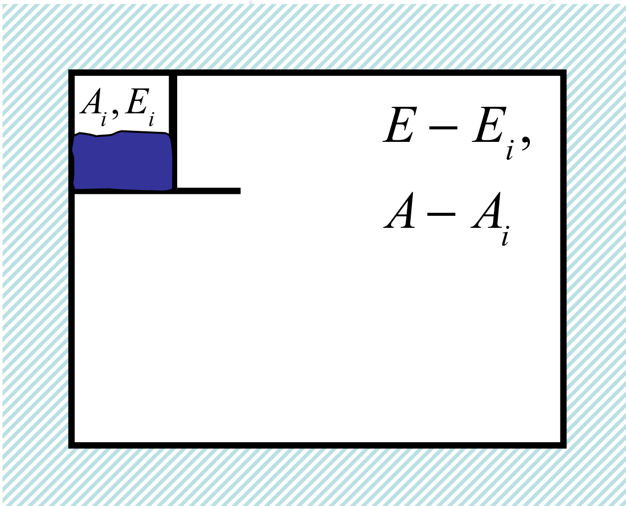
$\mathrm{d}q = T\mathrm{d}S$

$\mathrm{d}w = -p\mathrm{d}V + \gamma \mathrm{d}A$

$$\mathrm{d}E = T\mathrm{d}S + \gamma \mathrm{d}A$$

$$\left(\frac{\partial S}{\partial A}\right)_E = -\frac{\gamma}{T}$$

# Constant surface tension simulations: $\gamma, V, T$ ensemble




Consider small system that can exchange *area* and energy with a big reservoir



$$\ln\Omega\left(E-E_{i,}A-A_i\right)=\ln\Omega\left(E,A\right)-\left(\frac{\partial\ln\Omega}{\partial E}\right)_A E_i-\left(\frac{\partial\ln\Omega}{\partial A}\right)_E A_i+\cdots$$

$$\left(\frac{\partial\ln\Omega}{\partial A}\right)_N=\left(\frac{\partial S/k_B}{\partial A}\right)_N=-\frac{\gamma}{k_B T}$$

$$\ln\frac{\Omega\left(E-E_i,A-A_i\right)}{\Omega\left(E,A\right)}=-\frac{E_i}{k_B T}+\frac{\gamma A_i}{k_B T}$$

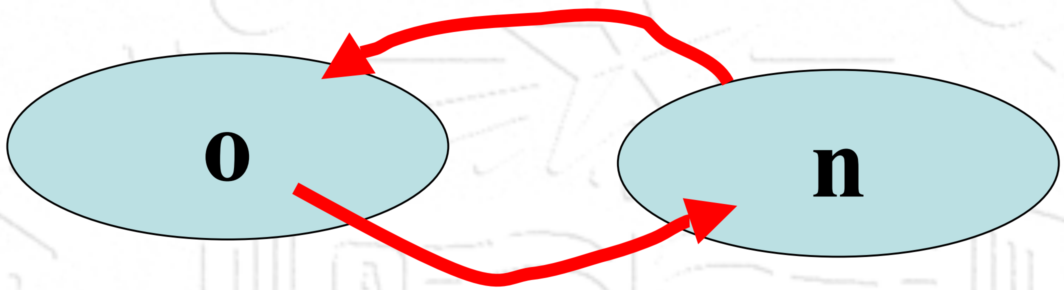
$$P\left(E_i,A_i\right)=\frac{\Omega\left(E-E_i,A-A_i\right)}{\sum_{j,k}\Omega\left(E-E_j,A-A_k\right)}=\frac{\exp\left[-\beta\left(E_i-\gamma A_i\right)\right]}{\sum_{j,k}\exp\left[-\beta\left(E_j-\gamma A_k\right)\right]}$$

$$\propto\exp\left[-\beta\left(E_i-\gamma A_i\right)\right]$$

# Simulations at imposed surface tension

- Simulation to a constant surface tension
  - Simulation box: allow the area of the bilayer to change in such a way that the volume is constant.

# Detailed balance




$$K(o \rightarrow n) = K(n \rightarrow o)$$

$$K(o \rightarrow n) = N(o) \times \alpha(o \rightarrow n) \times \mathrm{acc}(o \rightarrow n)$$

$$K(n \rightarrow o) = N(n) \times \alpha(n \rightarrow o) \times \mathrm{acc}(n \rightarrow o)$$

$$\frac{\mathrm{acc}(o \rightarrow n)}{\mathrm{acc}(n \rightarrow o)} = \frac{N(n) \times \alpha(n \rightarrow o)}{N(o) \times \alpha(o \rightarrow n)} = \frac{N(n)}{N(o)}$$

# $\gamma VT$-ensemble

$$N_{\gamma VT}\left(A,\mathbf{s}^N\right) \propto \exp\left[-\beta\left[U\left(\mathrm{s}^N;L\right)-\gamma A\right]\right]$$

$$\frac{\mathrm{acc}(o \rightarrow n)}{\mathrm{acc}(n \rightarrow o)} = \frac{N(n)}{N(o)}$$

Suppose we change: $\left.\begin{array}{l} L_o \rightarrow L_n \\ A_o \rightarrow A_n \end{array}\right\} V_o = V_n$

$$\frac{\mathrm{acc}(o \rightarrow n)}{\mathrm{acc}(n \rightarrow o)} = \frac{\exp\left[-\beta\left[U\left(\mathrm{s}_\mathrm{n}^\mathrm{N};L_n\right)-\gamma A_n\right]\right]}{\exp\left[-\beta\left[U\left(\mathrm{s}_\mathrm{o}^\mathrm{N};L_o\right)-\gamma A_o\right]\right]}$$

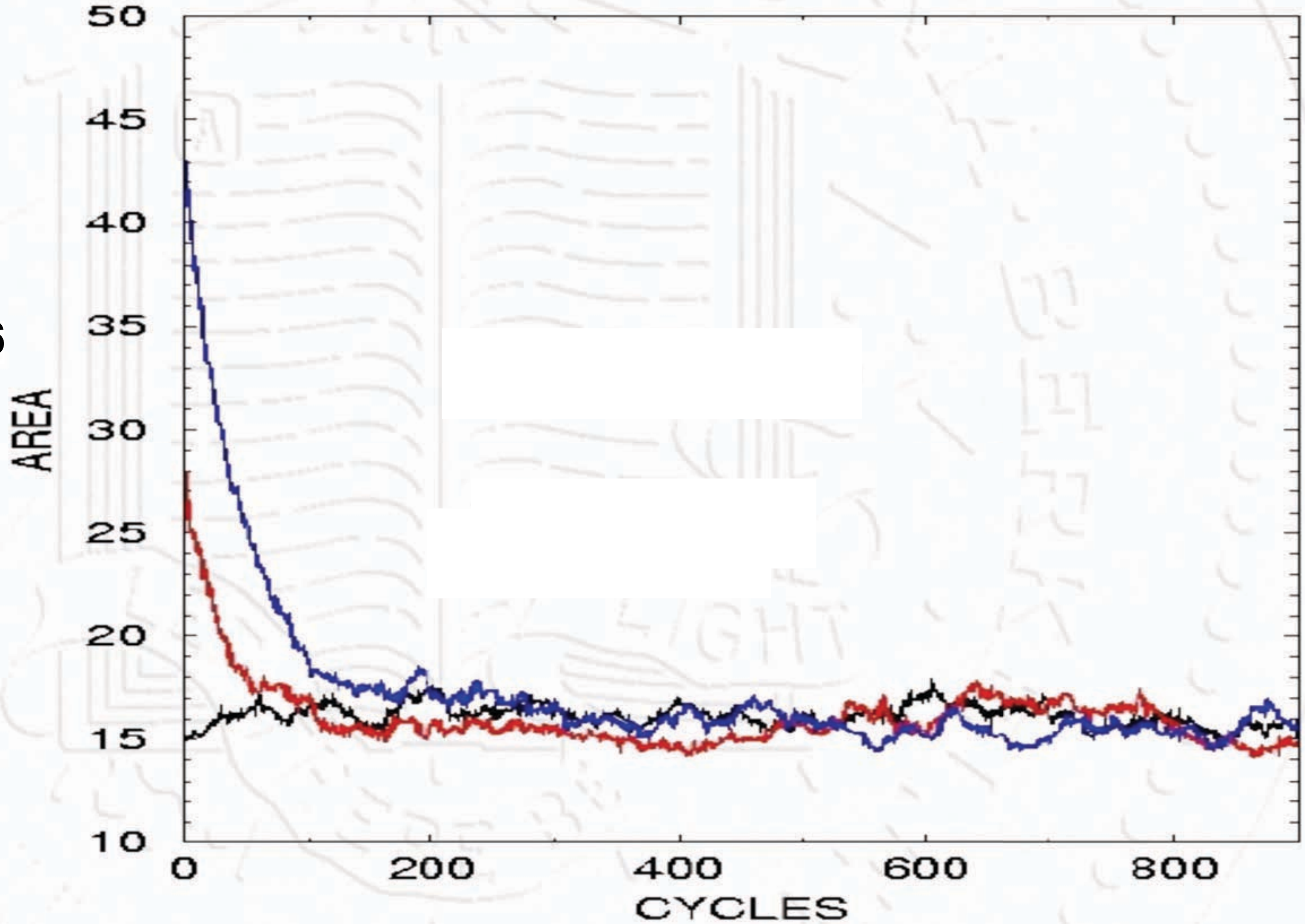
$$= \exp\left\{-\beta\left[U\left(n\right)-U\left(o\right)\right]-\beta\gamma\left(A_n - A_o\right)\right\}$$

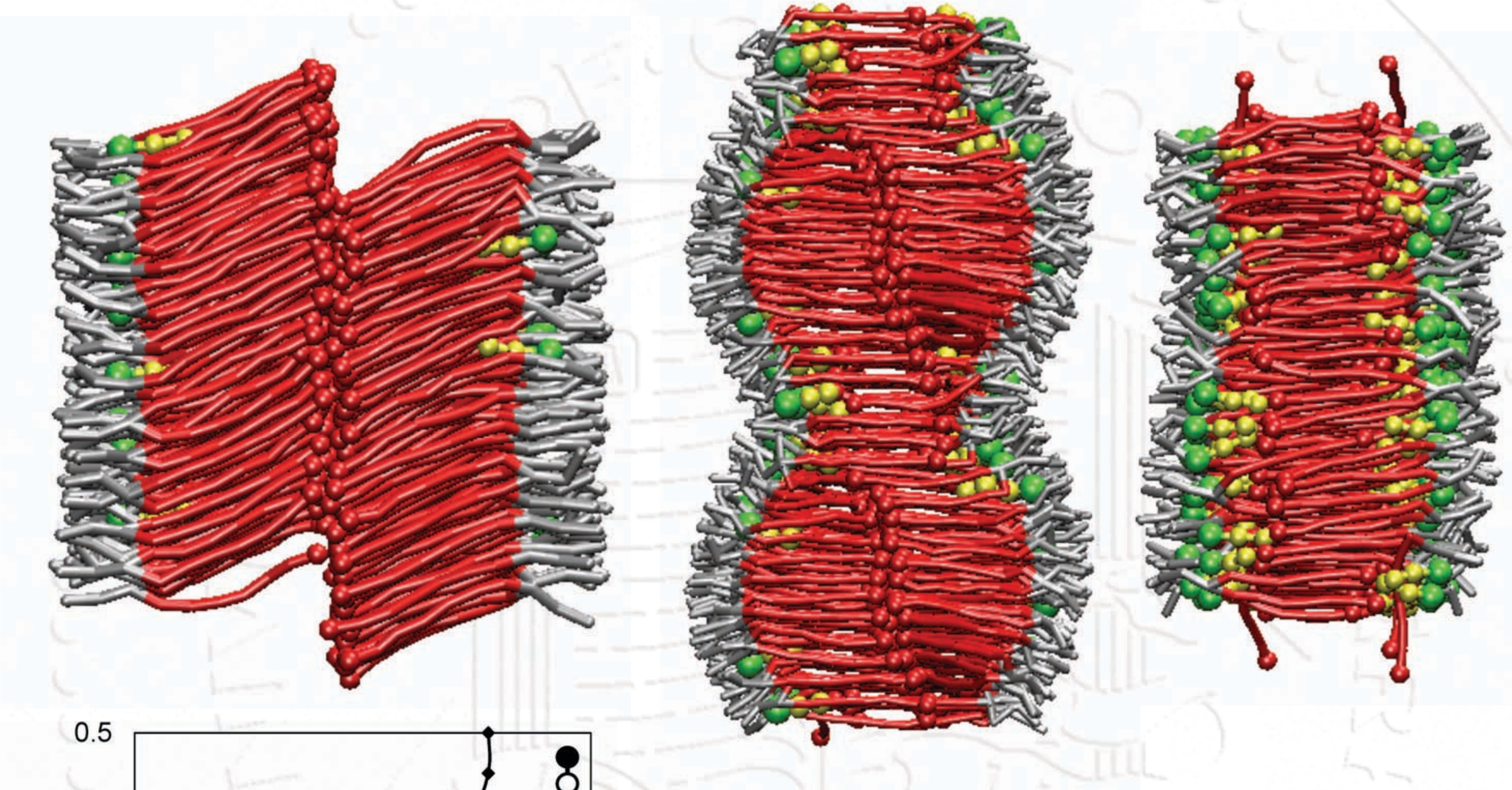
# Tensionless state: $\gamma = 0$

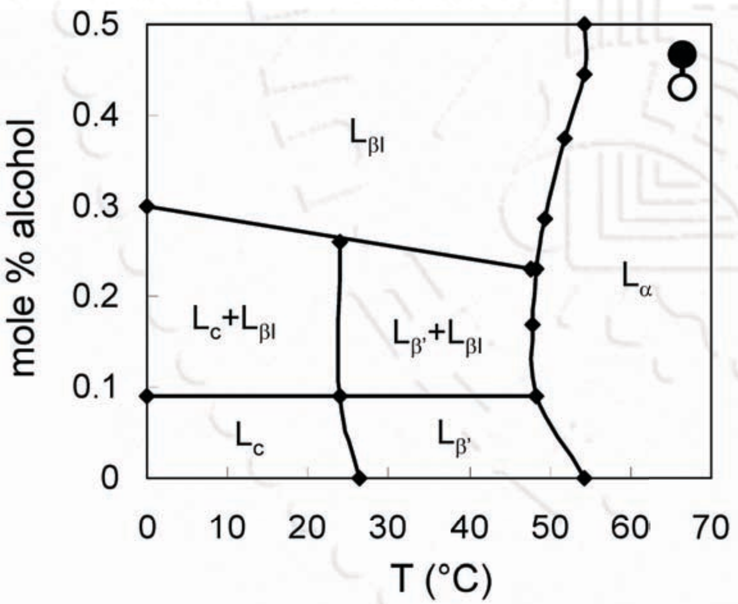
$\gamma(A_o) = -0.3 \pm 0.6$ (black)

$\gamma(A_o) = 2.5 \pm 0.3$ (red)

$\gamma(A_o) = 2.9 \pm 0.3$ (blue)







mole % alcohol
0.5
0.4
0.3
0.2
0.1
0
$L_{\beta I}$
$L_c+L_{\beta I}$
$L_{\beta'}+L_{\beta I}$
$L_\alpha$
$L_c$
$L_{\beta'}$
0 10 20 30 40 50 60 70
T (°C)