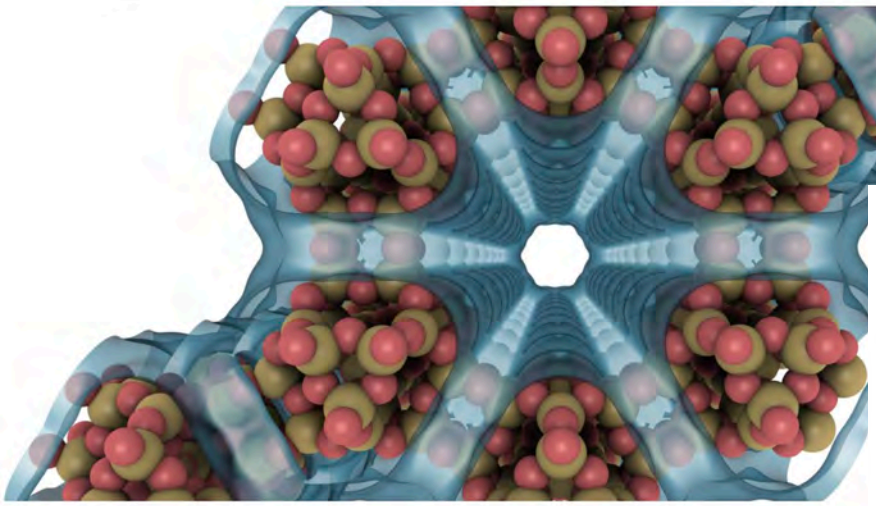
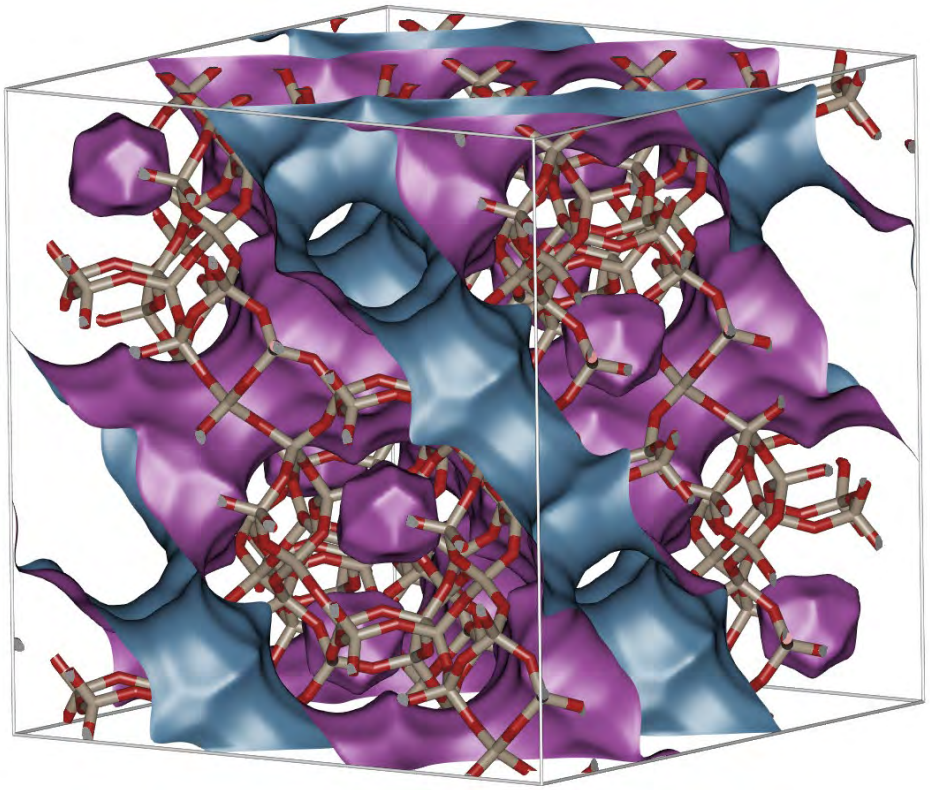
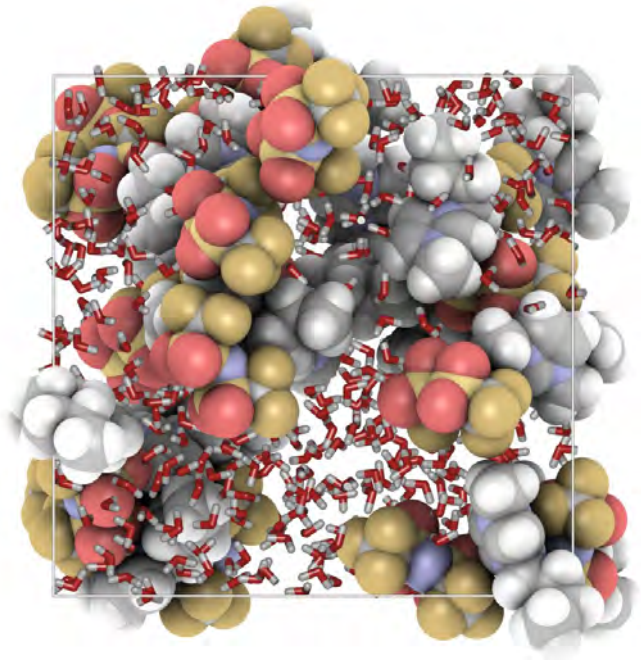
(i)RASPA

David Dubbeldam
Randy Snurr
Sofia Calero
Thijs J.H. Vlugt

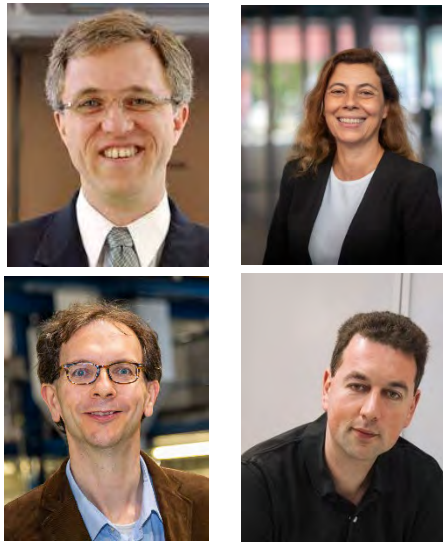


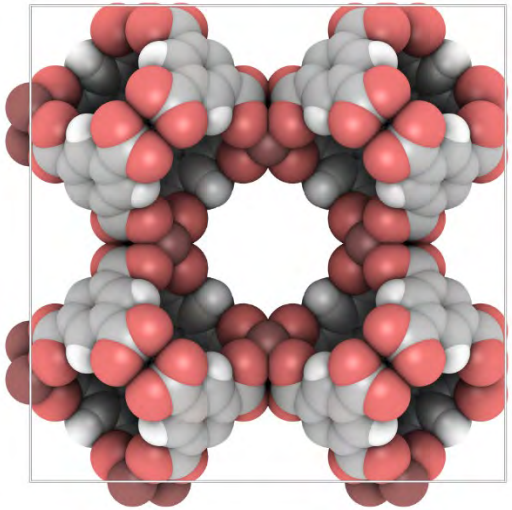


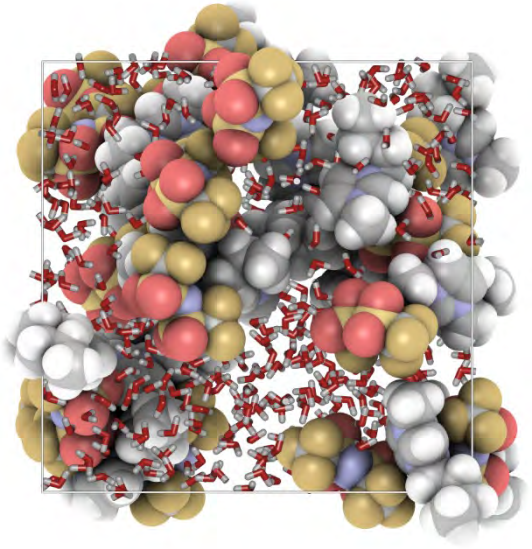


https://doi.org/10.1080/08927022.2018.1426855
www.iraspa.org
Annual workshops since 2018, >200 participants

RASPA workshop Delft (Sep. 2-4, 2024)



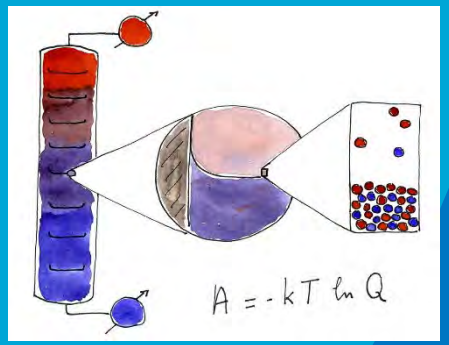




The RASPA workshop is held at Delft University of Technology (The Netherlands), 2,3,4 September 2024, just before the start of the Thermodynamics2024 conference. The workshop focuses on a practical understanding of molecular simulations of fluids, ionic liquids, and nanoporous materials and application to practical examples. The duration is 3 days, with morning-lectures and afternoon-exercises.

www.iraspa.org

The 28th Thermodynamics Conference, Delft 4-6 September 2024



https://thermodynamics2024.org

Molecular Physics Lecture

Tyler Josephson
University of Maryland

Invited speakers

Lourdes Vega
Khalifa University

André Bardow
ETH Zurich

Erika Eiser
NTNU/Porelab

Randall Snurr
Northwestern University

Hans Hasse
University of Kaiserslautern

Anton ten Kate
Nouryon

TU Delft
Delft University of Technology

TU Delft

ROYAL SOCIETY OF CHEMISTRY

TotalEnergies

IChemE ADVANCING CHEMICAL ENGINEERING WORLDWIDE

Taylor & Francis
Taylor & Francis Group

DCSE
Enabling Technology for Industry

PRO2 TECH

Configurational-Bias Monte Carlo

Thijs J.H. Vlugt

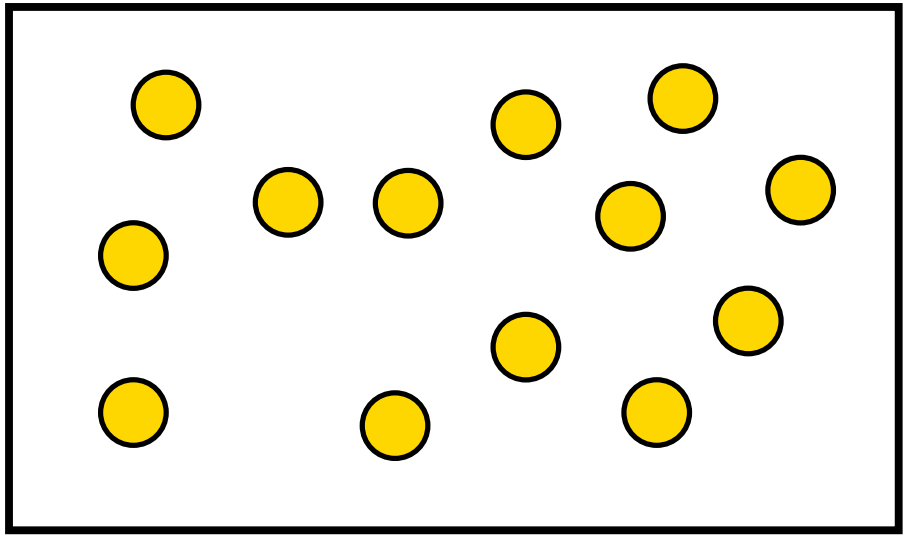
Professor and Chair Engineering Thermodynamics
Delft University of Technology
Delft, The Netherlands

t.j.h.vlugt@tudelft.nl

January 7, 2021

Random Sampling versus Metropolis Sampling (1)

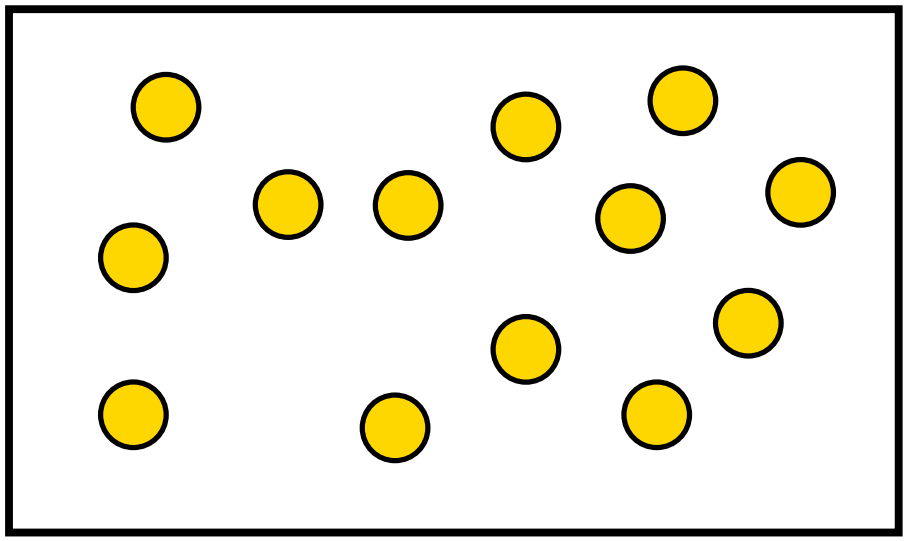
N interacting particles in volume V at temperature T



- vector representing positions of all particles in the system: $\mathbf{r}^N$
- total energy: $U(\mathbf{r}^N)$
- statistical weight of configuration $\mathbf{r}^N$ is $\exp[-\beta U(\mathbf{r}^N)]$ with $\beta = 1/(k_B T)$

Random Sampling versus Metropolis Sampling (2)

N interacting particles in volume V at temperature T
pair interactions $u(r_{ij})$



$$U(\mathbf{r}^N) = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} u(r_{ij}) = \sum_{i<j} u(r_{ij})$$

$$Q(N,V,T) = \frac{1}{\Lambda^{3N} N!} \int d\mathbf{r}^N \exp\left[-\beta U(\mathbf{r}^N)\right]$$

$$F(N,V,T) = -k_B T \ln Q(N,V,T)$$

Random Sampling versus Metropolis Sampling (3)

Computing the ensemble average $\langle \cdots \rangle$ of a certain quantity $A(\mathbf{r}^N)$

- Random Sampling of $\mathbf{r}^N$:

$$\langle A \rangle = \lim_{n \to \infty} \frac{\sum_{i=1}^{n} A(\mathbf{r}_i^N) \exp\left[-\beta U(\mathbf{r}_i^N)\right]}{\sum_{i=1}^{n} \exp\left[-\beta U(\mathbf{r}_i^N)\right]}$$

  Usually this leads to $\langle A \rangle =$ "0" / "0" $= ???$

- Metropolis sampling; generate n configurations $\mathbf{r}^N$ with probability proportional to $\exp\left[-\beta U(\mathbf{r}_i^N)\right]$, therefore:

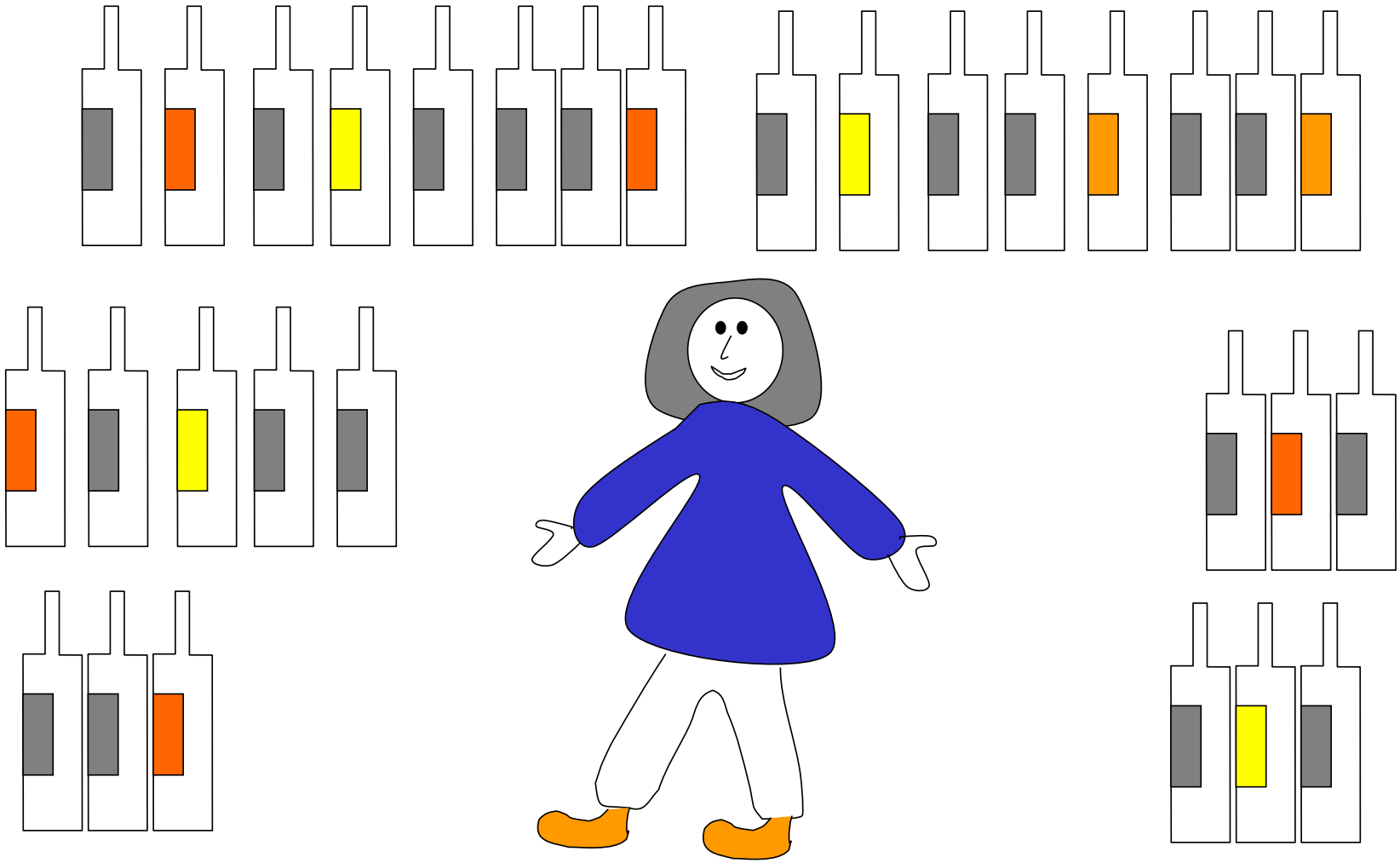
$$\langle A \rangle = \lim_{n \to \infty} \frac{\sum_{i=1}^{n} A(\mathbf{r}_i^N)}{n}$$

Simulation Technique (1)

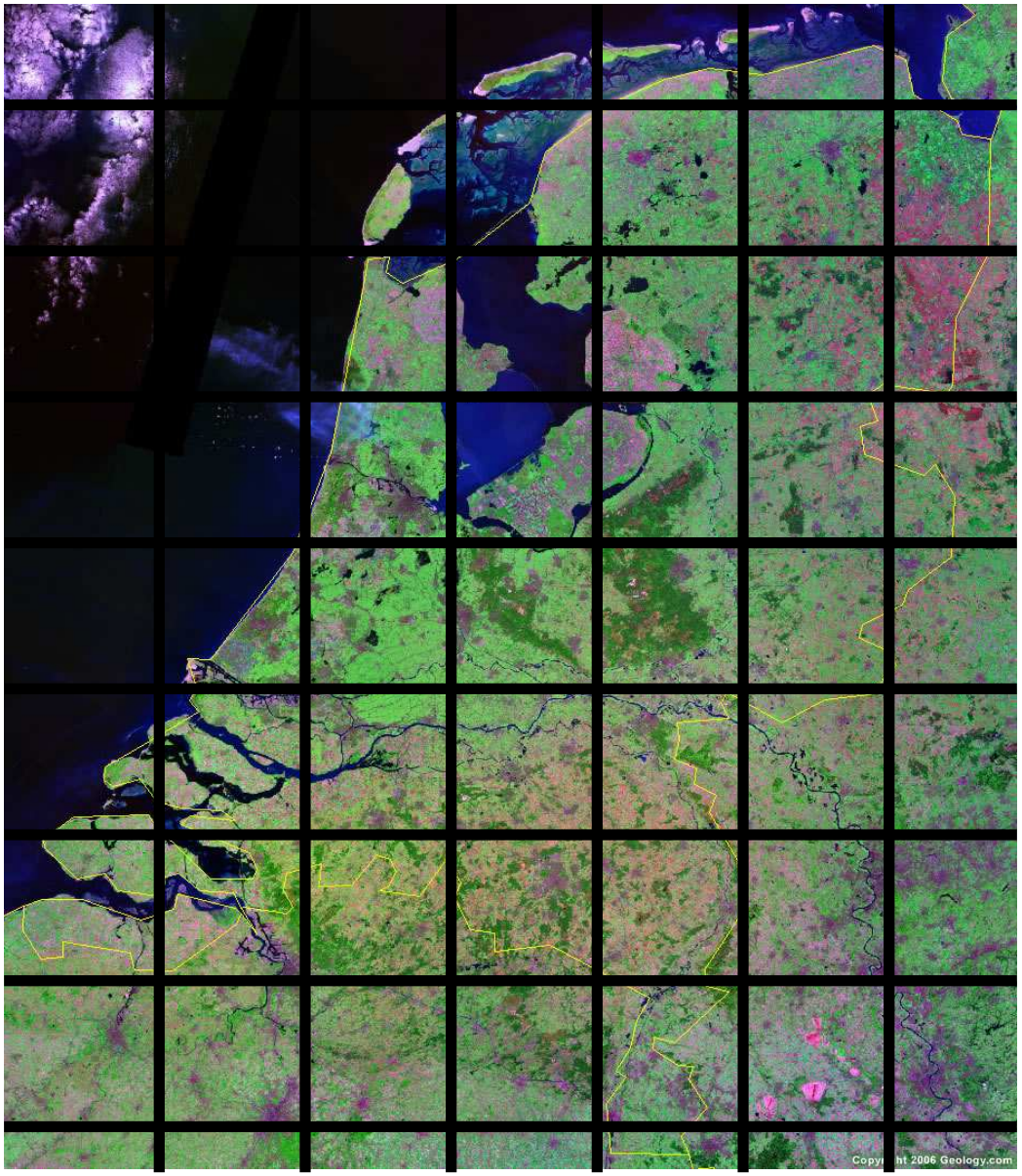
What is the ratio of red wine/white wine in the Netherlands?

Simulation Technique (2)

Bottoms up



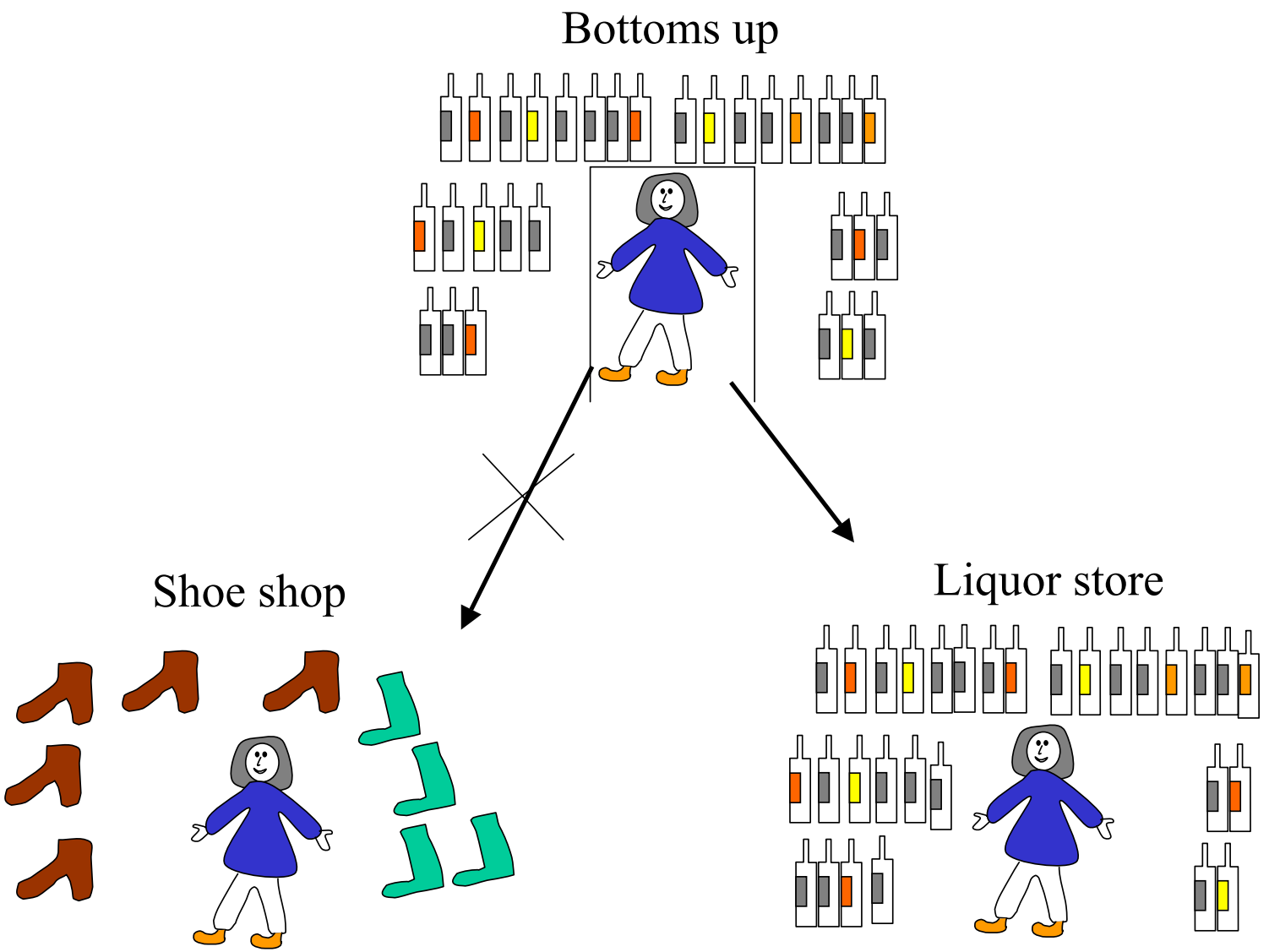
Simulation Technique (3)



Simulation Technique (4)

Simulation Technique (5)

Simulation Technique (6)



Metropolis Monte Carlo (1)

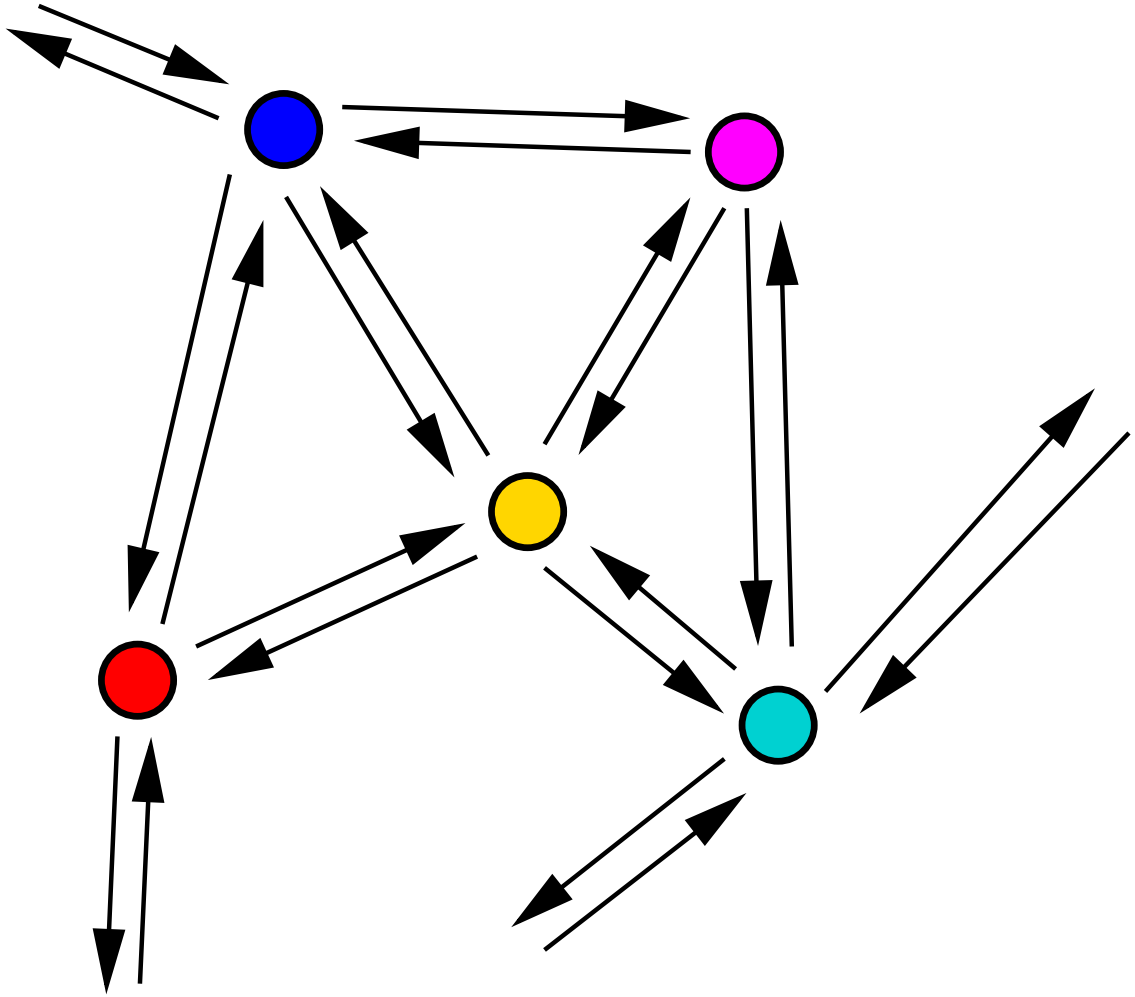
How to generate configurations $\mathbf{r}_i$ with a probability proportional to $\mathcal{N}(\mathbf{r}_i) = \exp[-\beta U(\mathbf{r}_i)]$???

Nick Metropolis

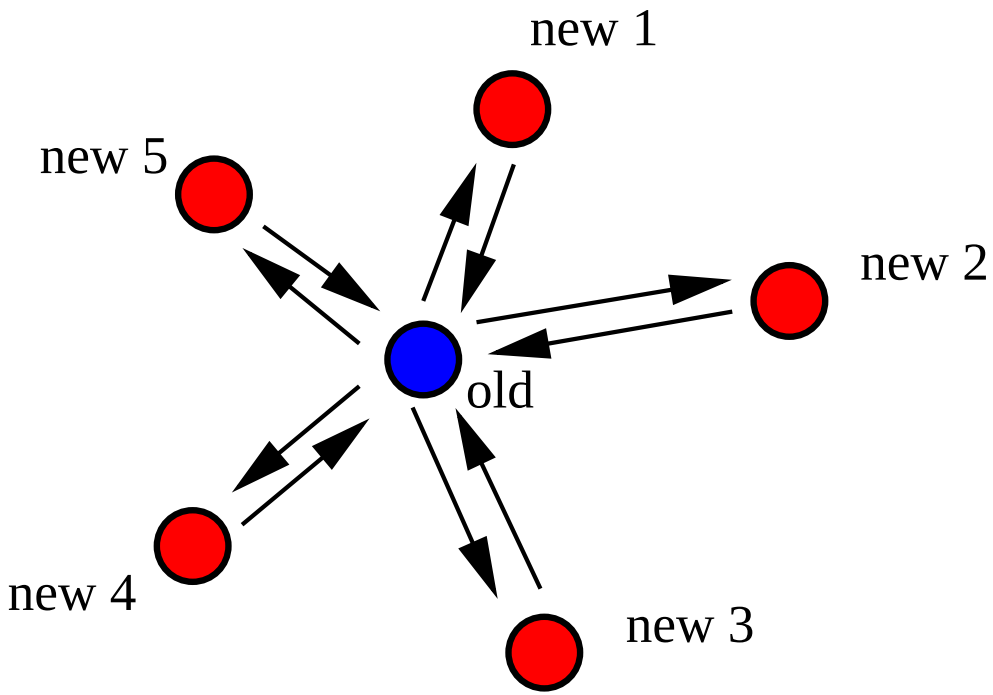
N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth. A.H. Teller and E. Teller, "Equation of State Calculations by Fast Computing Machines," J. Chem. Phys., 1953, 21, 1087-1092.

Metropolis Monte Carlo (2)

Whatever our rule is to move from one state to the next, the equilibrium distribution should not be destroyed



Move from the old state (o) to a new state (n) and back



$$\text{leaving state o} = \text{entering state o}$$

$$\mathcal{N}(o)\sum_n [\alpha(o\to n)\mathrm{acc}(o\to n)] = \sum_n [\mathcal{N}(n)\alpha(n\to o)\mathrm{acc}(n\to o)]$$

$\mathcal{N}(i)$: probability to be in state i (here: proportional to $\exp[-\beta U(\mathbf{r}_i)]$)
$\alpha(x\to y)$: probability to attempt move from state x to state y
$\mathrm{acc}(x\to y)$: probability to accept move from state x to state y

Detailed Balance (1)

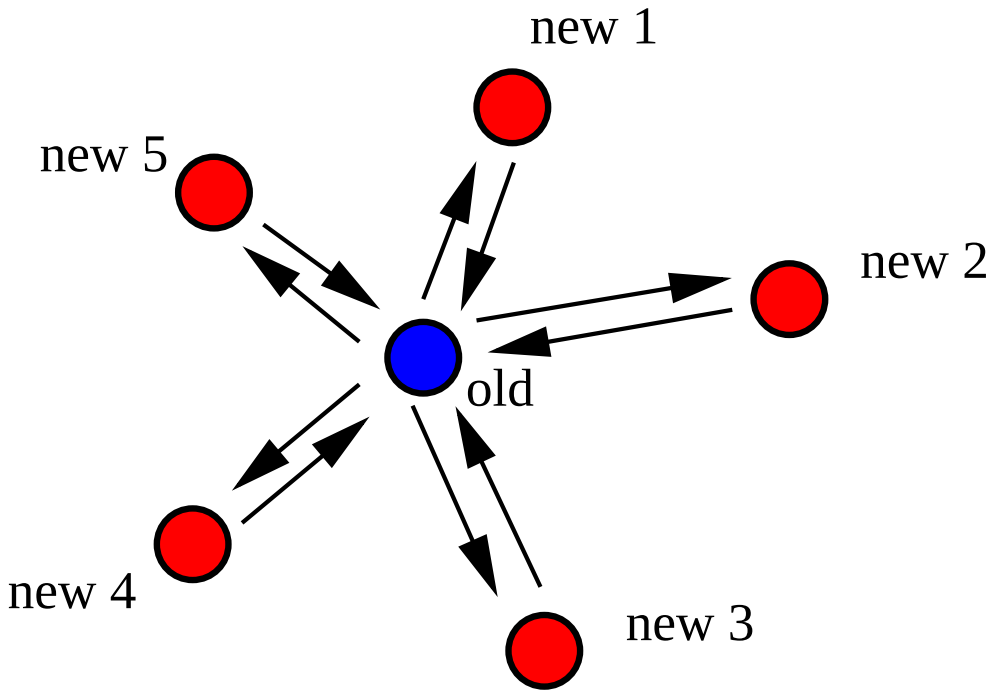
Requirement (balance):

$$\mathcal{N}(o)\sum_n [\alpha(o\to n)\mathrm{acc}(o\to n)] = \sum_n [\mathcal{N}(n)\alpha(n\to o)\mathrm{acc}(n\to o)]$$

Detailed balance: much stronger condition

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

for every pair o,n



Detailed Balance (2)

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

- $\alpha(x \to y)$; probability to select move from x to y
- $\mathrm{acc}(x \to y)$; probability to accept move from x to y
- often (but not always); $\alpha(o \to n) = \alpha(n \to o)$

Therefore (note that $\Delta U = U(n) - U(o)$):

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

In case that $\alpha(o \to n) = \alpha(n \to o)$

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

Metropolis Acceptance Rule

General:

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

Choice made by Metropolis (note: infinite number of other possibilities)

$$\mathrm{acc}(o \to n) = \min(1, X)$$

Note than $\min(a, b) = a$ if $a < b$ and b otherwise

- always accept when $X \geq 1$
- when $X < 1$, generate uniformly distributed random number between 0 and 1 and accept or reject according to $\mathrm{acc}(o \to n)$

Monte Carlo Casino

Smart Monte Carlo: $\alpha(o \to n) \neq \alpha(n \to o)$

Not a random displacement Δr uniformly from $[-\delta, \delta]$, but instead

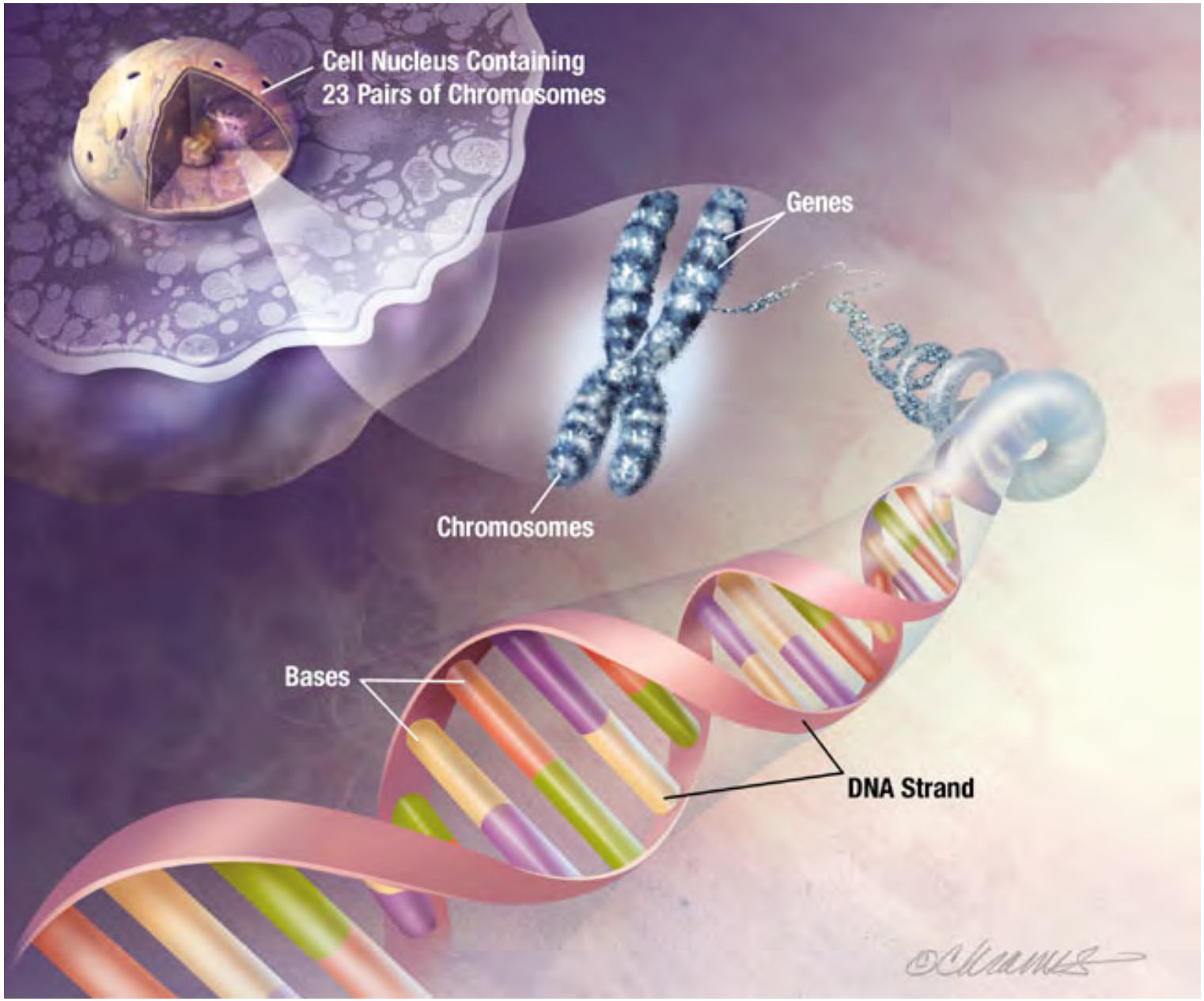
$$\Delta r = r(\text{new}) - r(\text{old}) = A \times F + \delta r$$

F : force on particle
A : constant
δr : taken from Gaussian distribution with width $2A$
so $P(\delta r) \sim \exp[-(\delta r^2)/4A]$

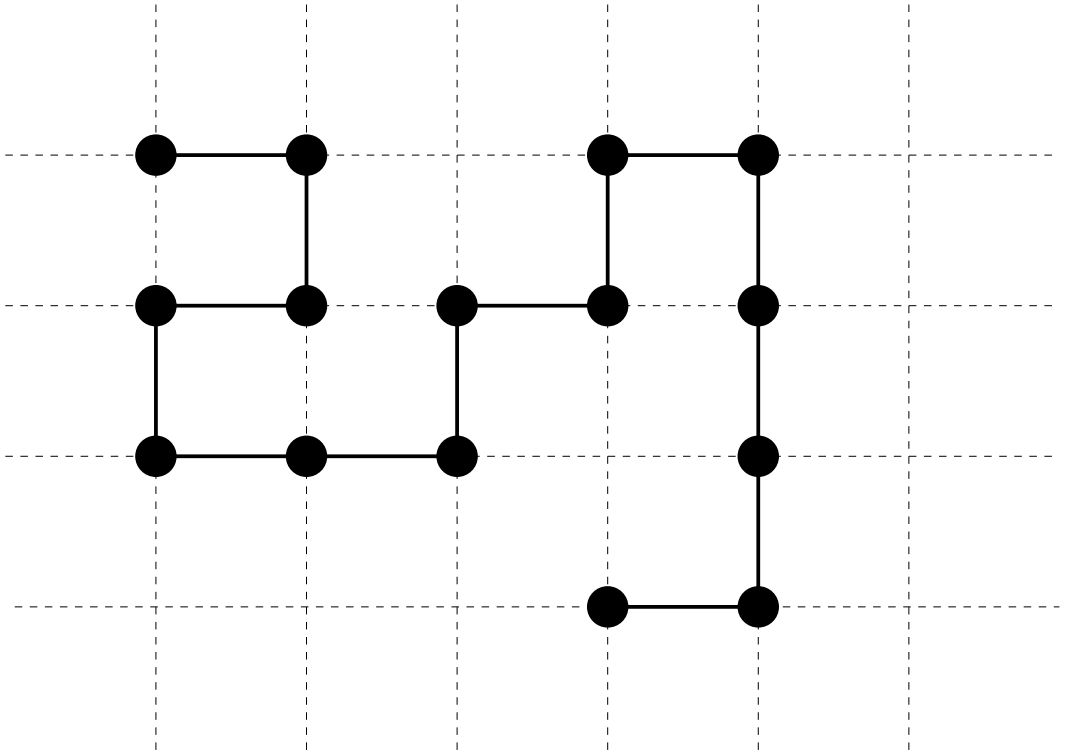
$$P(r_{\text{new}}) \sim \exp\left[-\frac{(r_{\text{new}} - (r_{\text{old}} + A \times F(o)))^2}{4A}\right]$$

$$\frac{\alpha(o \to n)}{\alpha(n \to o)} = \frac{\exp\left[-\frac{(\Delta r - A \times F(o))^2}{4A}\right]}{\exp\left[-\frac{(\Delta r + A \times F(n))^2}{4A}\right]}$$

Chain Molecules



Self-Avoiding Walk on a Cubic Lattice



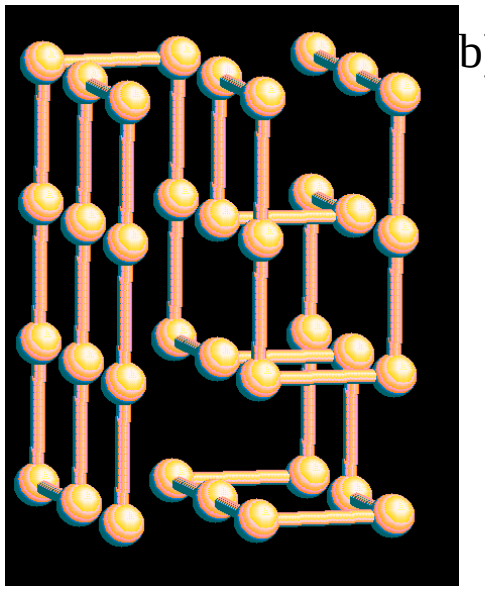
- 3D lattice; 6 lattice directions
- only 1 monomer per lattice site (otherwise $U = \infty$)
- interactions only when $|r_{ij}| = 1$ and $|i - j| > 1$

Simple Model for Protein Folding

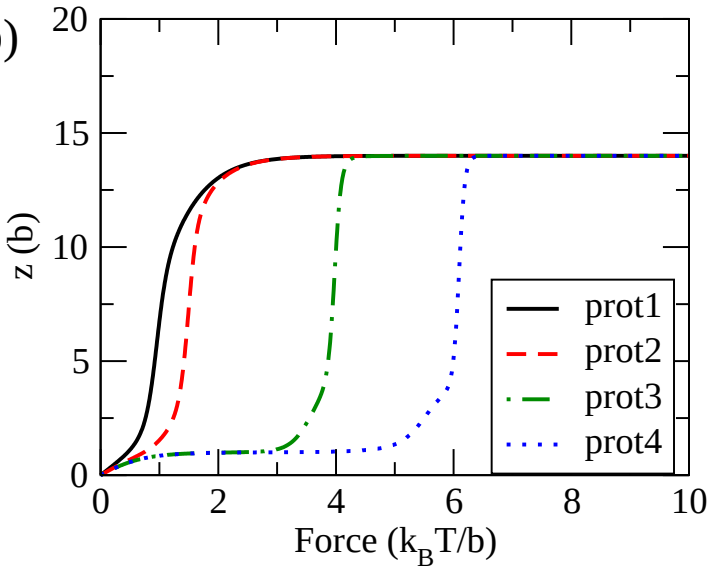
20 by 20 interaction matrix Δ_{ij}

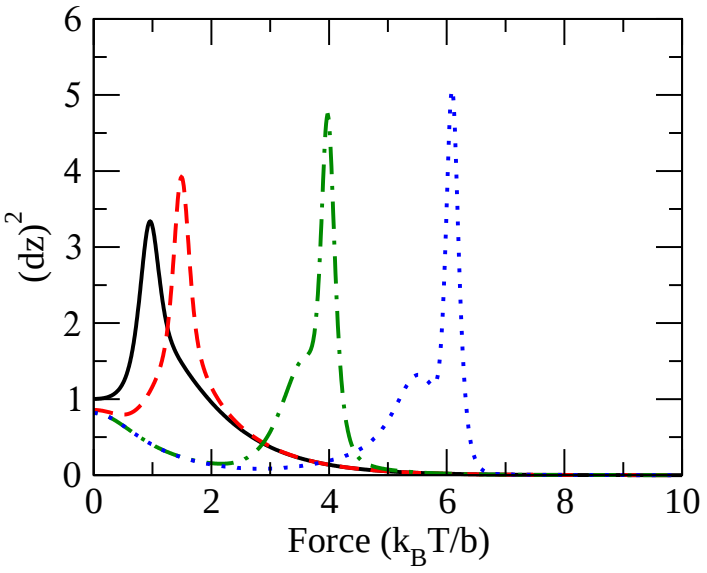
YPDLTKWHAMEAGKIRFSVPDACLNGEGIRQVTLSN

(E. Jarkova, T.J.H. Vlugt, N.K. Lee, J. Chem. Phys., 2005, 122, 114904)



b)





Number of Configurations without Overlap

Random Chains:

$$\langle R \rangle = \lim_{n \to \infty} \frac{\sum_{i=1}^{n} R_i \exp[-\beta U_i]}{\sum_{i=1}^{n} \exp[-\beta U_i]}$$

Fraction of chains without overlap decreases exponentially as a function of chainlength (N)

N	total ($= 6^{N-1}$)	without overlap	fraction no overlap
2	6	6	1
6	7776	3534	0.454
8	279936	81390	0.290
10	10077696	1853886	0.183
12	362797056	41934150	0.115
13	2176782336	198842742	0.091
14	13060694016	943974510	0.072
15	78364164096	4468911678	0.057
16	470184984576	21175146054	0.045
50	$\cdots$	$\cdots$	1.3×10^{-5}

Rosenbluth Sampling (1)

1. Place first monomer at a random position

2. For the next monomer (i), generate k trial directions ($j = 1, 2, \cdots, k$) each with energy u_{ij}

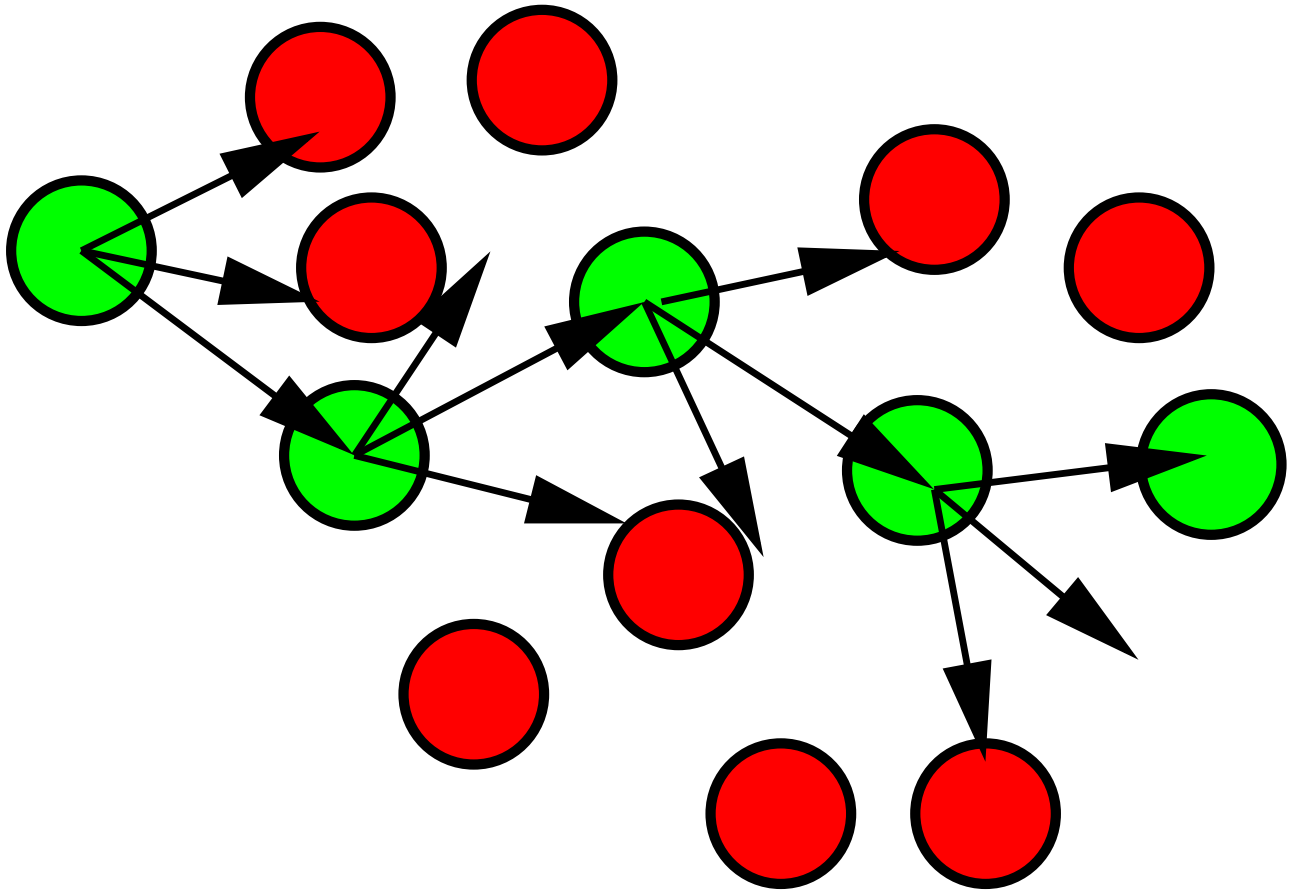
3. Select trial direction $j^\star$ with a probability

$$P_{j^\star} = \frac{\exp[-\beta u_{ij^\star}]}{\sum_{j=1}^{k} \exp[-\beta u_{ij}]}$$

4. Continue with step 2 until the complete chain is grown (N monomers)

Rosenbluth Sampling (2)

$$P_{j^\star} = \frac{\exp[-\beta u_{ij^\star}]}{\sum_{j=1}^{k} \exp[-\beta u_{ij}]}$$



Rosenbluth Sampling (3)

Probability to choose trial direction $j^\star$ for the i th monomer

$$P_{j^\star} = \frac{\exp[-\beta u_{ij^\star}]}{\sum_{j=1}^{k} \exp[-\beta u_{ij}]}$$

Probability to grow this chain (N monomers, k trial directions)

$$P_{\text{chain}} = \prod_{i=1}^{N} P_{j^\star(i)} = \frac{\prod_{i=1}^{N} \exp[-\beta u_{ij^\star(i)}]}{\prod_{i=1}^{N} \sum_{j=1}^{k} \exp[-\beta u_{ij}]} = \frac{\exp[-\beta U_{\text{chain}}]}{W_{\text{chain}}}$$

Rosenbluth Sampling (4)

Probability to grow this chain (N monomers, k trial directions)

$$P_{\text{chain}} = \frac{\prod_{i=1}^{N} \exp[-\beta u_{ij^\star(i)}]}{\prod_{i=1}^{N} \sum_{j=1}^{k} \exp[-\beta u_{ij}]} = \frac{\exp[-\beta U_{\text{chain}}]}{W_{\text{chain}}}$$

Therefore, weightfactor for each chain i is the Rosenbluth factor W_i:

$$\langle R \rangle_{\text{Boltzmann}} = \lim_{n\to\infty} \frac{\sum_{i=1}^{n} W_i \times R_i}{\sum_{i=1}^{n} W_i}$$

The unweighted distribution is called the Rosenbluth distribution:

$$\langle R \rangle_{\text{Rosenbluth}} = \lim_{n\to\infty} \frac{\sum_{i=1}^{n} R_i}{n}$$

Of course: $\langle R \rangle_{\text{Rosenbluth}} \neq \langle R \rangle_{\text{Boltzmann}}$

Intermezzo: Ensemble Averages at Different Temperatures

Ensemble averages at $\beta^\star$ can (in principle) be computed from simulations at β:

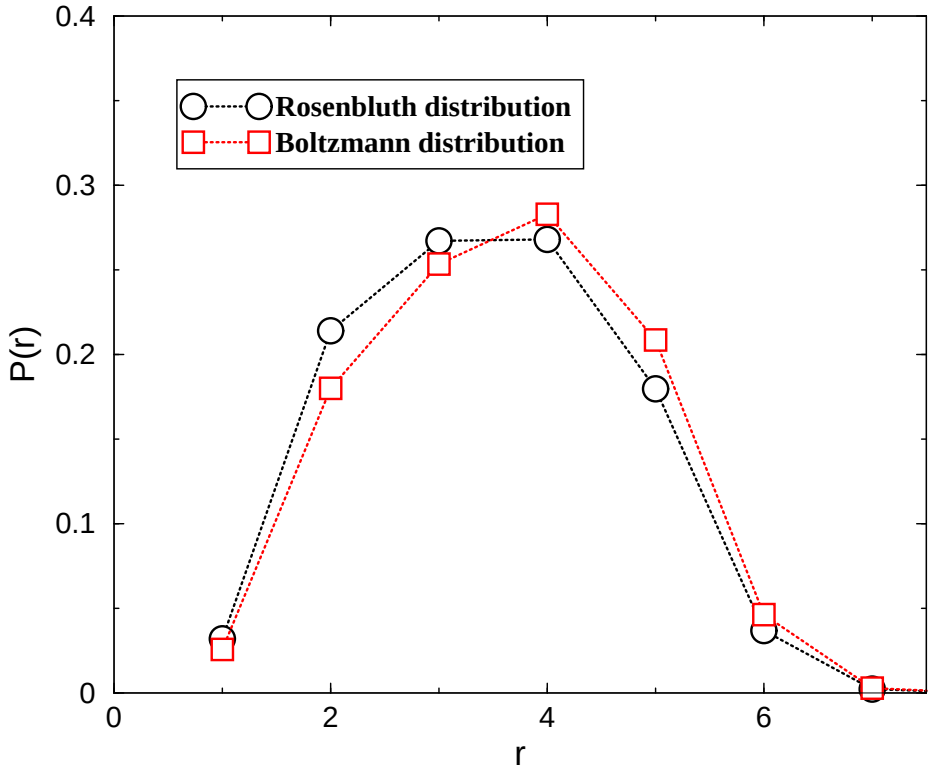
$$
\begin{aligned}
\langle U \rangle_\beta &= \frac{\int d\mathbf{r}^N U(\mathbf{r}^N) \exp[-\beta U(\mathbf{r}^N)]}{\int d\mathbf{r}^N \exp[-\beta U(\mathbf{r}^N)]} \\
&= \frac{\int d\mathbf{r}^N U(\mathbf{r}^N) \exp[-\beta^\star U(\mathbf{r}^N)] \exp[(\beta^\star - \beta) \times U(\mathbf{r}^N)]}{\int d\mathbf{r}^N \exp[-\beta^\star U(\mathbf{r}^N)] \exp[(\beta^\star - \beta) \times U(\mathbf{r}^N)]} \\
&= \frac{\left\langle U(\mathbf{r}^N) \exp[(\beta^\star - \beta) \times U(\mathbf{r}^N)] \right\rangle_{\beta^\star}}{\left\langle \exp[(\beta^\star - \beta) \times U(\mathbf{r}^N)] \right\rangle_{\beta^\star}} \\
&= \frac{\left\langle U(\mathbf{r}^N) \exp[\Delta\beta \times U(\mathbf{r}^N)] \right\rangle_{\beta^\star}}{\left\langle \exp[\Delta\beta \times U(\mathbf{r}^N)] \right\rangle_{\beta^\star}}
\end{aligned}
$$

Useful or not???

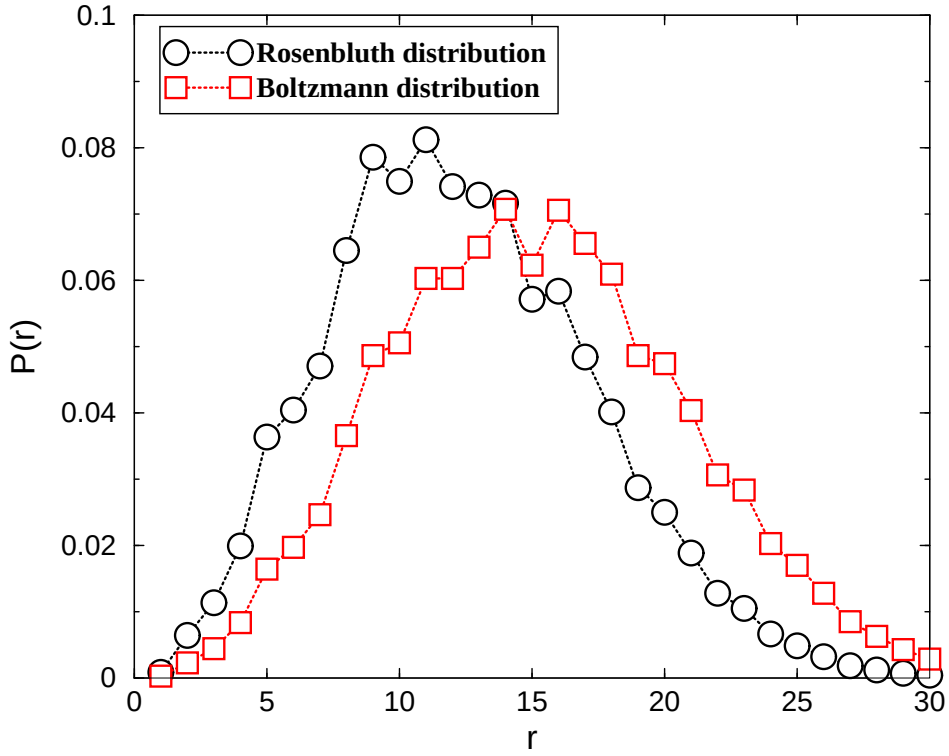
Rosenbluth Distribution Differs from Boltzmann Distribution

Probability distribution for the *end-to-end distance* r

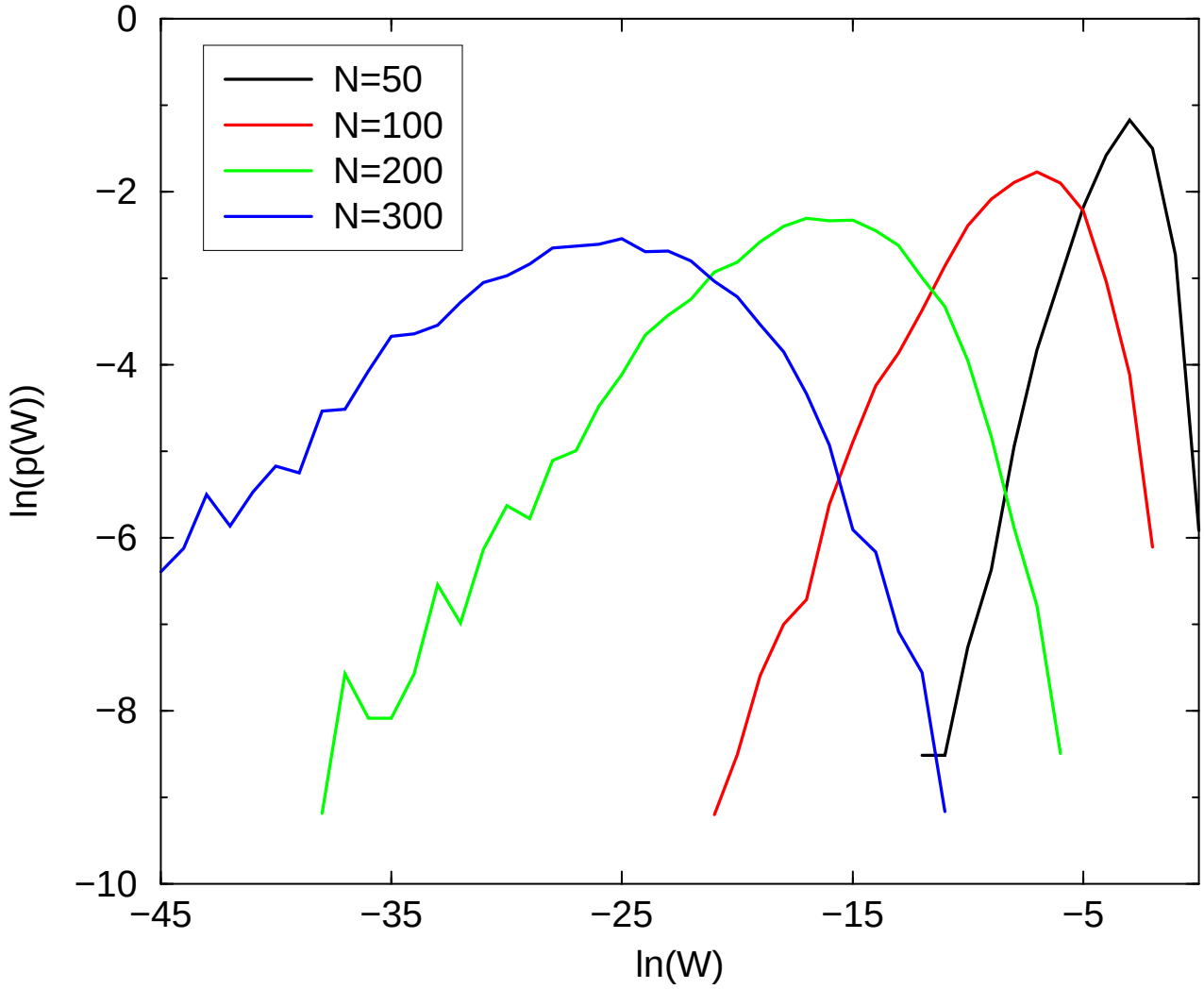
$N = 10$



$N = 100$



Distribution of Rosenbluth Weights



Of course, $\mathrm{Probability}(W = 0) \neq 0$ (not shown in this figure)

Pruned-Enriched Rosenbluth Method (1)

Grassberger (1997); grow chains using Rosenbluth Method:

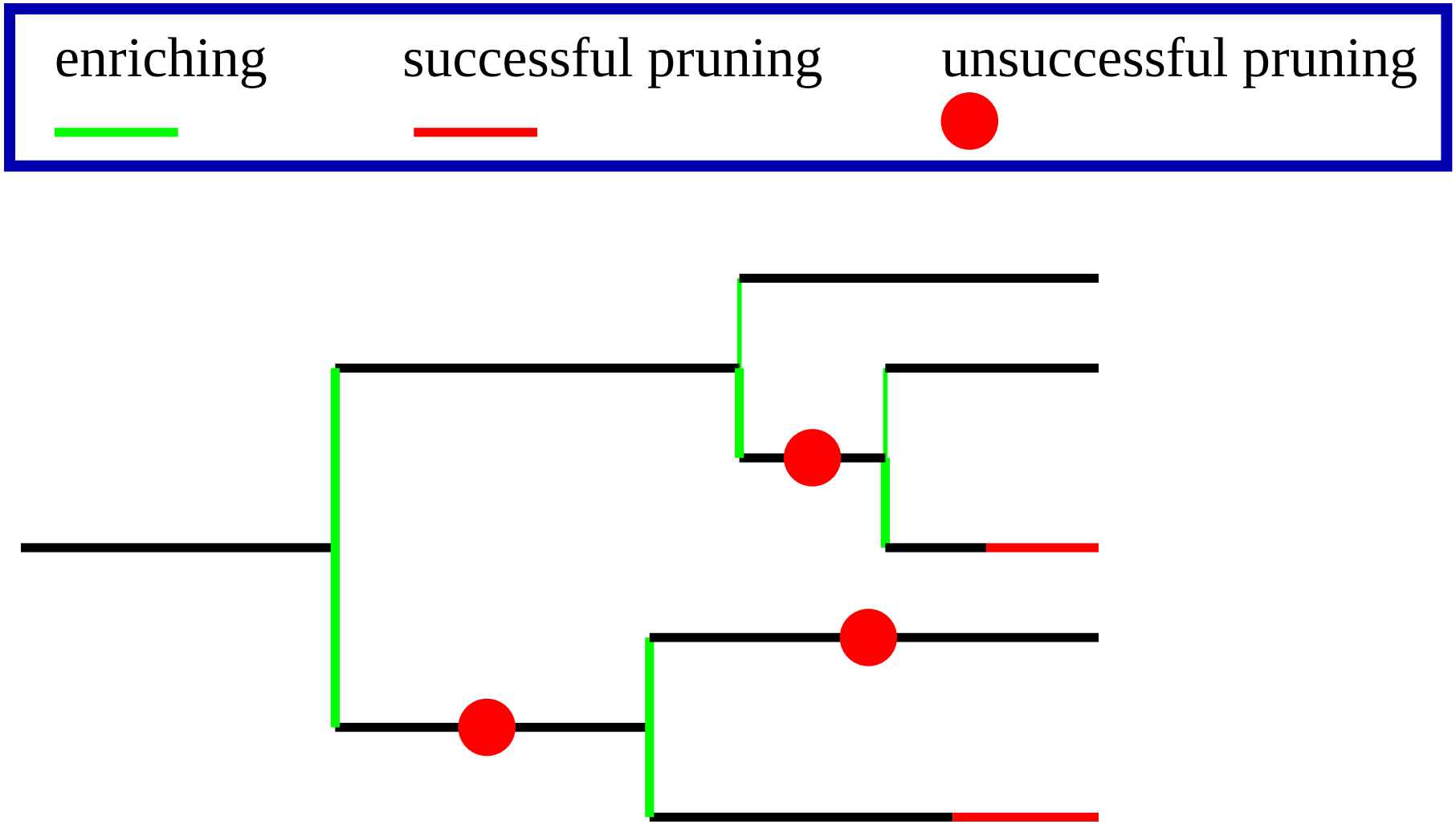
$$W = \sum_{j=1}^{6} \frac{\exp\left[-\beta u_{2j}\right]}{6} \times \prod_{i=3}^{N} \sum_{j=1}^{5} \frac{\exp\left[-\beta u_{ij}\right]}{5} = \prod_{i=3}^{N} \sum_{j=1}^{5} \frac{\exp\left[-\beta u_{ij}\right]}{5}$$

Two additional elements:

- **Enriching**
  If $W > W_{\max}$ during the construction of the chain, k copies of the chain are generated, each with a weight of W/k. This is a deterministic process. The growth of those k chains is continued.

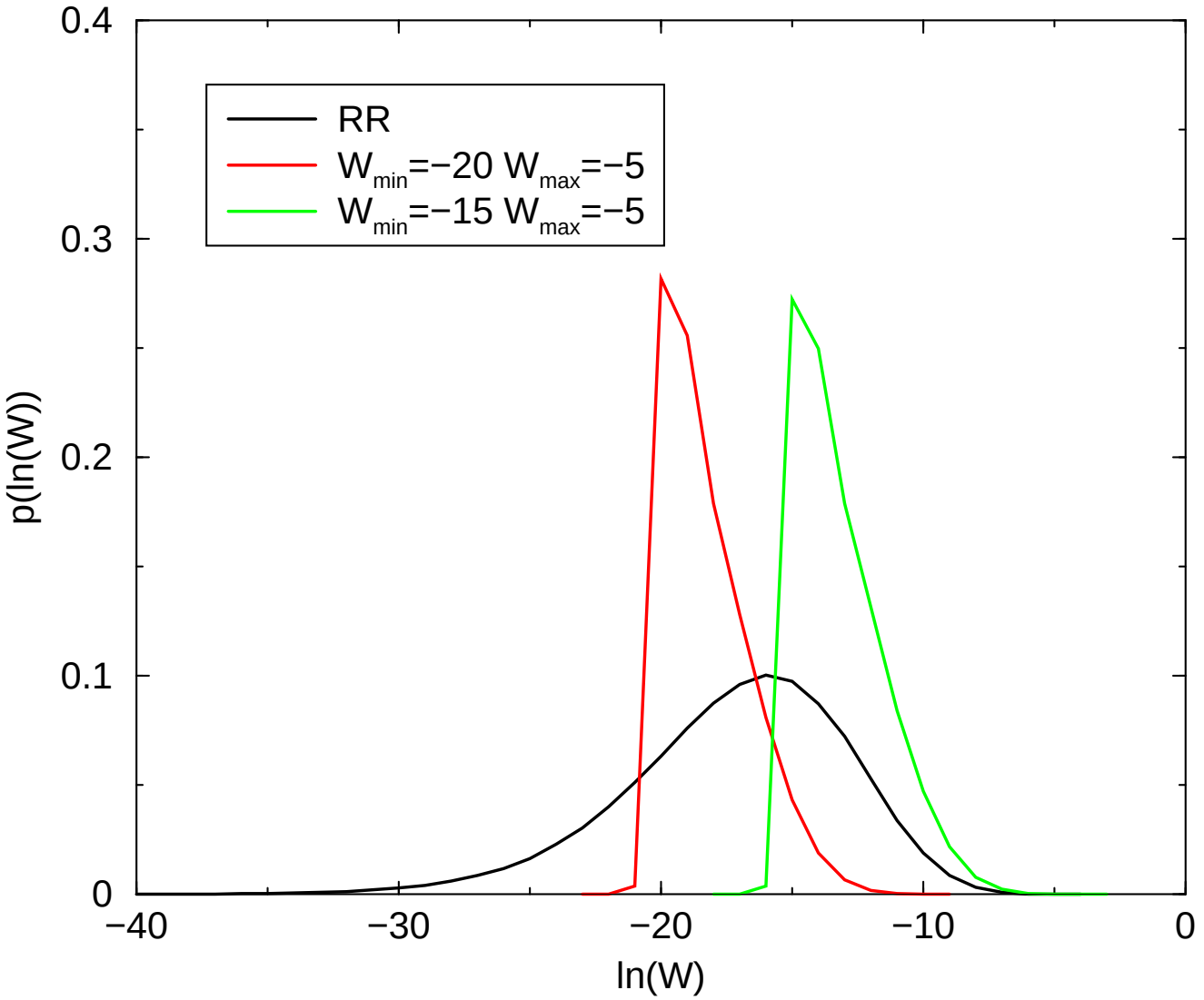
- **Pruning**
  If $W < W_{\min}$ during the construction of a chain, with a probability of $1/2$ the chain is pruned resulting in $W = 0$. If the chain survives, the Rosenbluth weight is multiplied by 2 and the growth of the chain is continued.

Pruned-Enriched Rosenbluth Method (2)



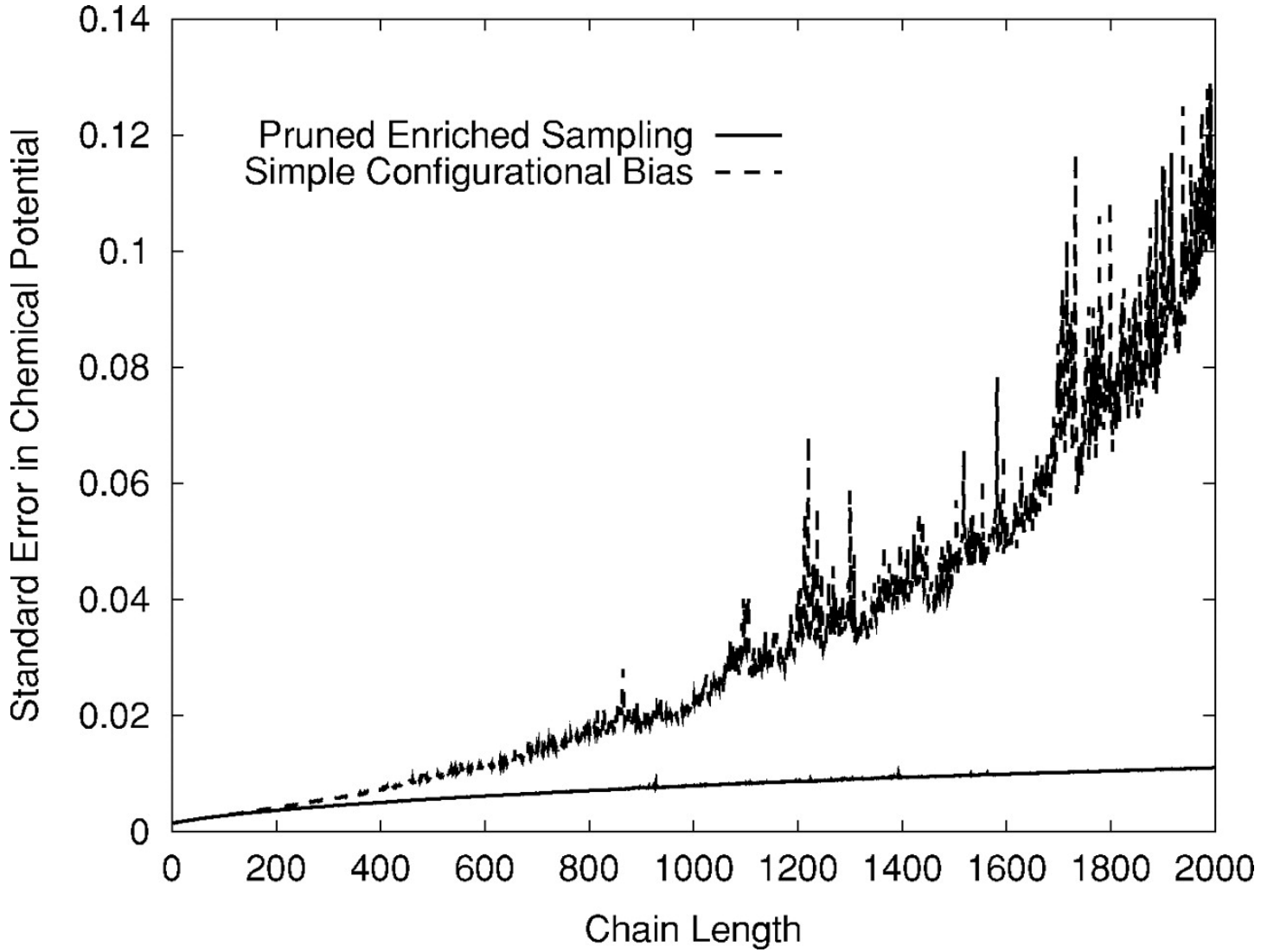
Pruned-Enriched Rosenbluth Method (3)

Example: $N = 200$, $k = 2$, $\beta\mu_{\rm ex} = -\ln\langle W\rangle = -12.14$



Pruned-Enriched Rosenbluth Method (4)

$\beta\mu_{\rm ex} = -\ln\langle W\rangle$, Ann. Rev. of Phys. Chem. 1999, 50, 377-411



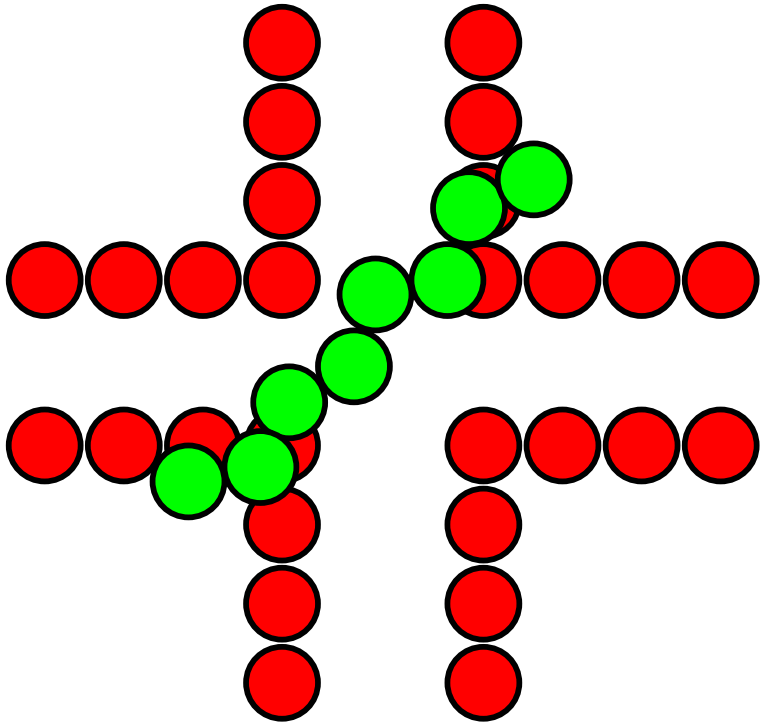
Static versus Dynamic Monte Carlo

- **Static Monte Carlo**
  - create single chain conformations, and use correct weight factor (random insertion, Rosenbluth scheme, PERM) to compute single chain averages
  - **single chain only; no Markov chain**

- **Dynamic Monte Carlo**
  - create Markov chain, accept/reject new configuration, acceptance rules should obey detailed balance
  - **multi-chain system, usable for all ensembles (e.g. Gibbs, μVT)**
  - Configurational-Bias Monte Carlo (CBMC), Recoil Growth (RG), Dynamic PERM (DPERM)

Random Insertion of Chains is Useless

Chain Length	Probability without overlaps
1	10^{-2}
2	10^{-4}
3	10^{-6}
...	...
8	10^{-16}



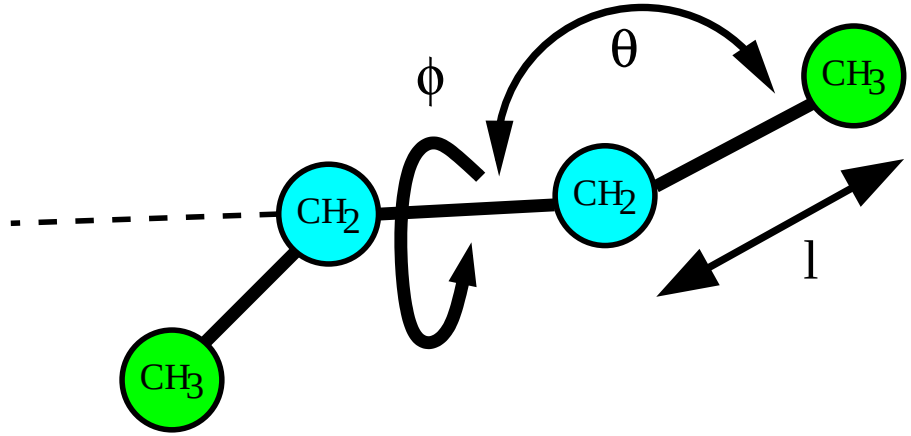
Configurational-Bias Monte Carlo

- Generate configurations using the Rosenbluth scheme
- Accept/Reject these configurations in such a way that detailed balance is obeyed
- Split potential energy into "**bonded**" (bond-stretching, bending, torsion) and "**non-bonded**" (i.e. Lennard-Jones and/or Coulombic) interactions
- Generate (k) trial positions according to **bonded** interactions (unbranched chain: l, θ, ϕ are independent)

$$U_{\rm bonded} = U_{\rm stretch}(l) + U_{\rm bend}(\theta) + U_{\rm tors}(\phi)$$

$$\begin{aligned} P(l) &\sim dl\, l^2 \exp[-\beta u(l)] \\ P(\theta) &\sim d\theta\, \sin(\theta) \exp[-\beta u(\theta)] \\ P(\phi) &\sim d\phi\, \exp[-\beta u(\phi)] \end{aligned}$$

Generate Trial Configurations: Linear Alkane



$$
\begin{aligned}
u(l) &= (k_l/2)\,(l-l_0)^2 \\
u(\theta) &= (k_\theta/2)\,(\theta-\theta_0)^2 \\
u(\phi) &= \sum_{i=0}^{5} c_i \cos^i(\phi)
\end{aligned}
$$

$$
\begin{aligned}
P(l) &\sim dl\, l^2 \exp[-\beta u(l)] \\
P(\theta) &\sim d\theta\, \sin(\theta) \exp[-\beta u(\theta)] \\
P(\phi) &\sim d\phi\, \exp[-\beta u(\phi)]
\end{aligned}
$$

Generate a Random Number from a Distribution (1)

$$F(y) = \int_0^y p(y')dy'$$

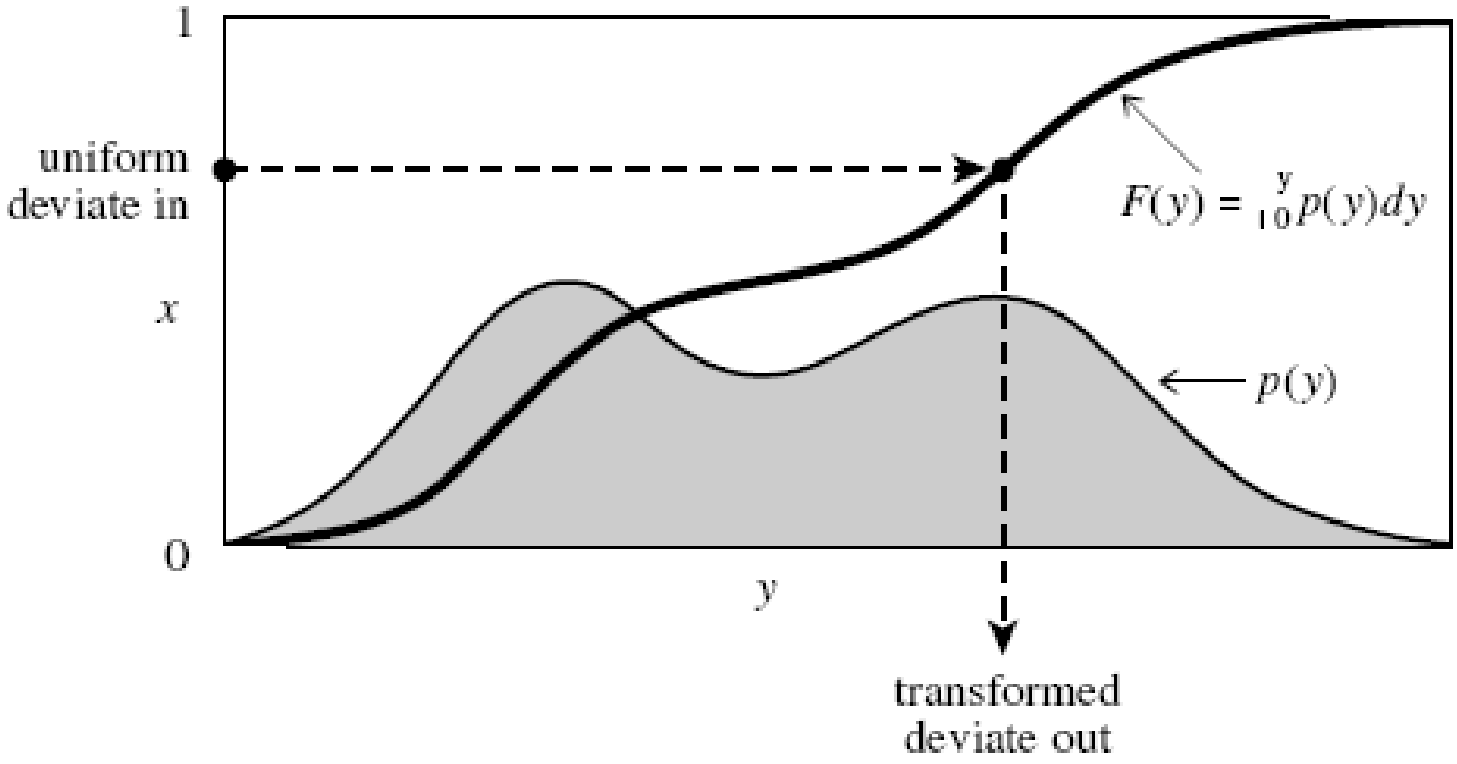


Figure 7.2.1. Transformation method for generating a random deviate y from a known probability distribution $p(y)$. The indefinite integral of $p(y)$ must be known and invertible. A uniform deviate x is chosen between 0 and 1. Its corresponding y on the definite-integral curve is the desired deviate.

Generate a Random Number from a Distribution (2)

$$F(y) = \int_0^y f(y')dy' \qquad f(x) > p(x)$$

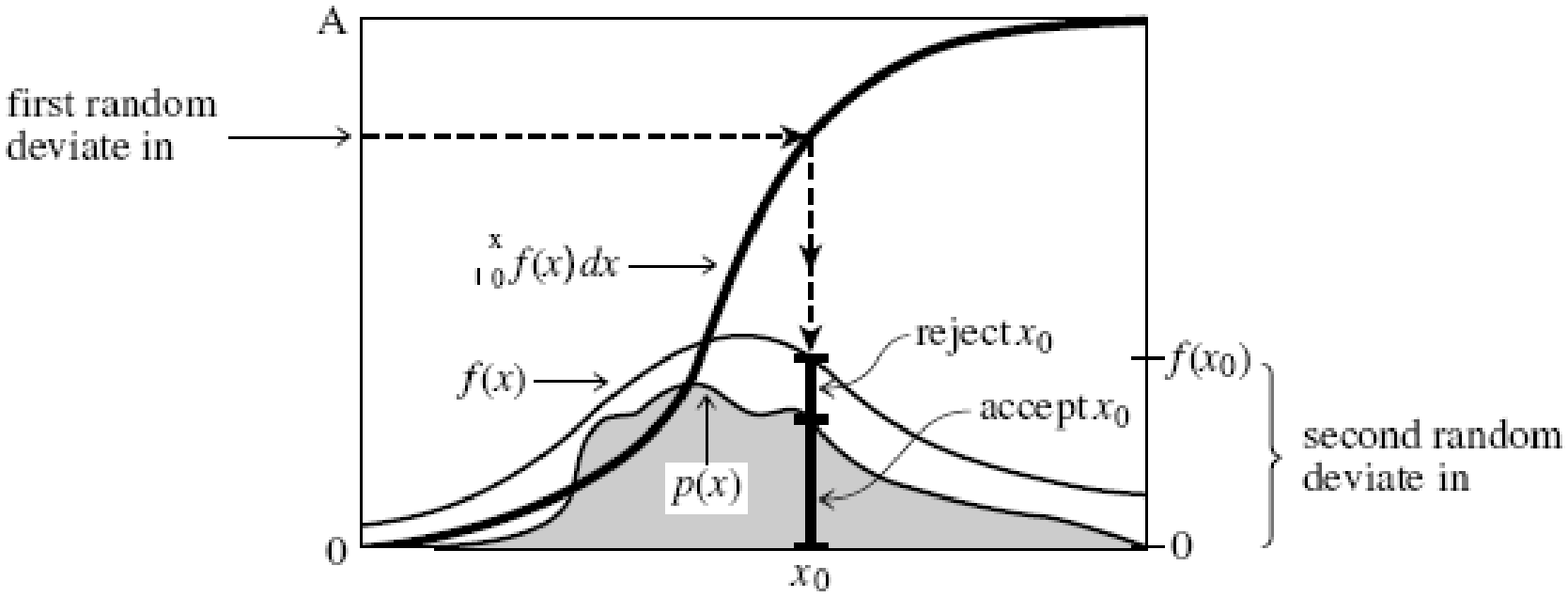


Figure 7.3.1. Rejection method for generating a random deviate x from a known probability distribution $p(x)$ that is everywhere less than some other function $f(x)$. The transformation method is first used to generate a random deviate x of the distribution f (compare Figure 7.2.1). A second uniform deviate is used to decide whether to accept or reject that x. If it is rejected, a new deviate of f is found; and so on. The ratio of accepted to rejected points is the ratio of the area under p to the area between p and f.

Generate a Random Number from a Distribution (3)

```
subroutine bend-tors                              generate appropriate θ and φ

lready=.false
do while (.not.lready)
  call ransphere(dx,dy,dz)                        random vector on unit sphere
  x = xold + dx                                   monomer position
  y = yold + dy
  z = zold + dz
  call bend(ubend,x,y,z)                          bending energy
  call tors(utors,x,y,z)                          torsion energy
  if(ranf().lt.exp(-beta*(ubend+utors)))          accept or reject
 +  lready=.true
enddo
```

CBMC Algorithm

- Generate a trial configuration using the Rosenbluth scheme. k trial segments $\{\mathbf{b}\}_k = \{\mathbf{b}_1 \cdots \mathbf{b}_k\}$, each trial segment is generated according to

$$P(\mathbf{b}) \sim \exp[-\beta u_{\mathrm{bonded}}(\mathbf{b})]$$

- Compute non-bonded energy, select configuration i with a probability

$$P(\mathbf{b}_i) = \frac{\exp[-\beta u_{\mathrm{non-b}}(\mathbf{b}_i)]}{\sum_{j=1}^{k} \exp[-\beta u_{\mathrm{non-b}}(\mathbf{b}_j)]} = \frac{\exp[-\beta u_{\mathrm{non-b}}(\mathbf{b}_i)]}{w_l}$$

- Continue until chain is grown, $W(n) = \prod_{l=1}^{n} w_l$

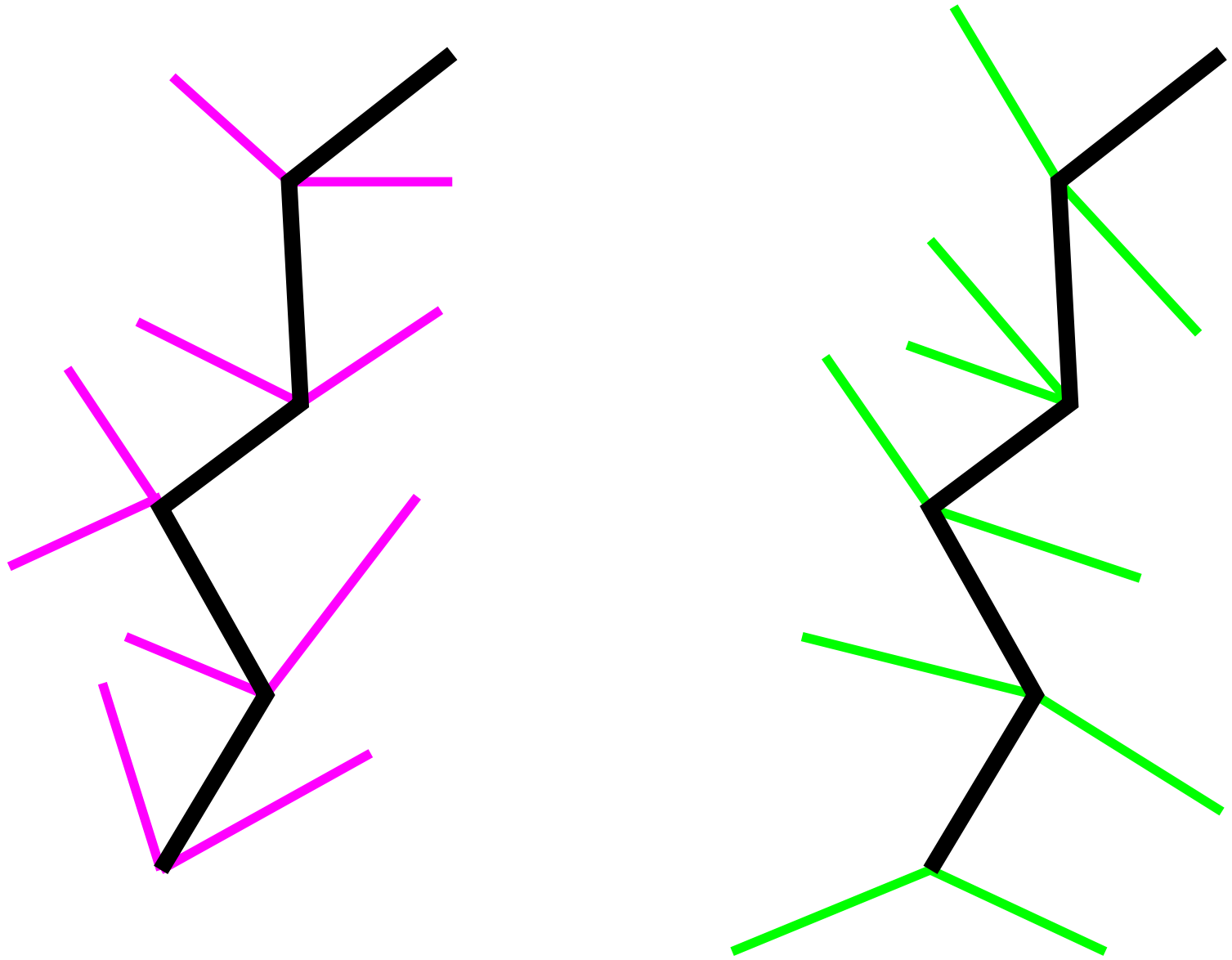
- Similar procedure for old configuration, generate $k-1$ trial positions (trial position 1 is the old configuration itself), leading to $W(o)$

- Accept/reject according to

$$\mathrm{acc}(o \rightarrow n) = \min(1, W(n)/W(o))$$

Super-Detailed Balance (1)

Same chain can be grow for different sets of trial directions...



Super-Detailed Balance (2)

Flux of configurations

$$\begin{aligned} K(o \to n) &= \sum_{\mathbf{b}_n \mathbf{b}_o} K(o \to n | \mathbf{b}_n \mathbf{b}_o) \\ K(n \to o) &= \sum_{\mathbf{b}_n \mathbf{b}_o} K(n \to o | \mathbf{b}_n \mathbf{b}_o) \end{aligned}$$

Detailed balance requires

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

Possible solution (**super-detailed balance**):

$$K(o \to n | \mathbf{b}_n \mathbf{b}_o) = K(n \to o | \mathbf{b}_n \mathbf{b}_o)$$

for each $\mathbf{b}_n \mathbf{b}_o$.

Super-Detailed Balance (3)

Detailed balance for each set $\mathbf{b}_n\mathbf{b}_o$:

$$
\begin{aligned}
K(o \to n|\mathbf{b}_n\mathbf{b}_o) &= N(o) \times \alpha(o \to n|\mathbf{b}_n\mathbf{b}_o) \times \mathrm{acc}(o \to n|\mathbf{b}_n\mathbf{b}_o) \\
&= \exp[-\beta U(o)] \times C \exp[-\beta u_{\mathrm{bonded}}(n)] \times \\
&\quad \frac{\exp[-\beta u_{\mathrm{non-b}}(n)]}{W(n)} \times P(\mathbf{b}_n\mathbf{b}_o) \times \mathrm{acc}(o \to n|\mathbf{b}_n\mathbf{b}_o) \\
K(n \to o|\mathbf{b}_n\mathbf{b}_o) &= \exp[-\beta U(n)] \times C \exp[-\beta u_{\mathrm{bonded}}(o)] \times \\
&\quad \frac{\exp[-\beta u_{\mathrm{non-b}}(o)]}{W(o)} \times P(\mathbf{b}_n\mathbf{b}_o) \times \mathrm{acc}(n \to o|\mathbf{b}_n\mathbf{b}_o)
\end{aligned}
$$

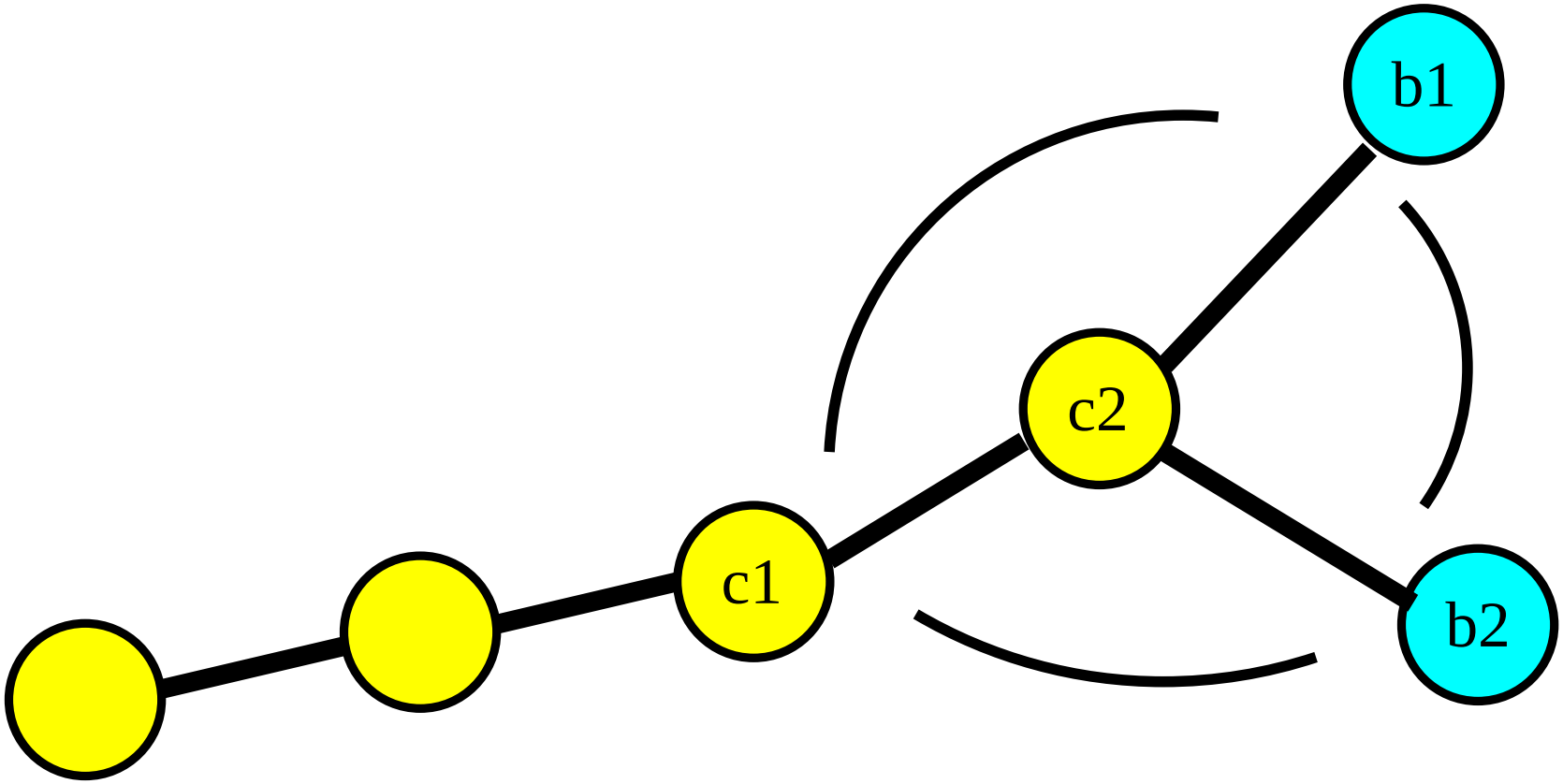
As

$$U = u_{\mathrm{non-b}} + u_{\mathrm{bonded}}$$

therefore

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

Branched Molecules (1)



$$P(\mathbf{b}_1, \mathbf{b}_2) \sim \exp[-\beta[u_{\text{bend}}(\mathbf{c}_1, \mathbf{c}_2, \mathbf{b}_1)] + u_{\text{bend}}(\mathbf{c}_1, \mathbf{c}_2, \mathbf{b}_2)] + u_{\text{bend}}(\mathbf{b}_1, \mathbf{c}_2, \mathbf{b}_2)]]$$

Branched Molecules (2)

Use CBMC to generate internal configurations:

- Generate n_t *random* trial positions and select one (i) with a probability

$$P^{\text{int}}(i) = \frac{\exp[-\beta U_{\text{bonded}}(i)]}{\sum_{j=1}^{n_t} \exp[-\beta U_{\text{bonded}}(j)]} = \frac{\exp[-\beta U_{\text{bonded}}(i)]}{W^{\text{int}}(n)}$$

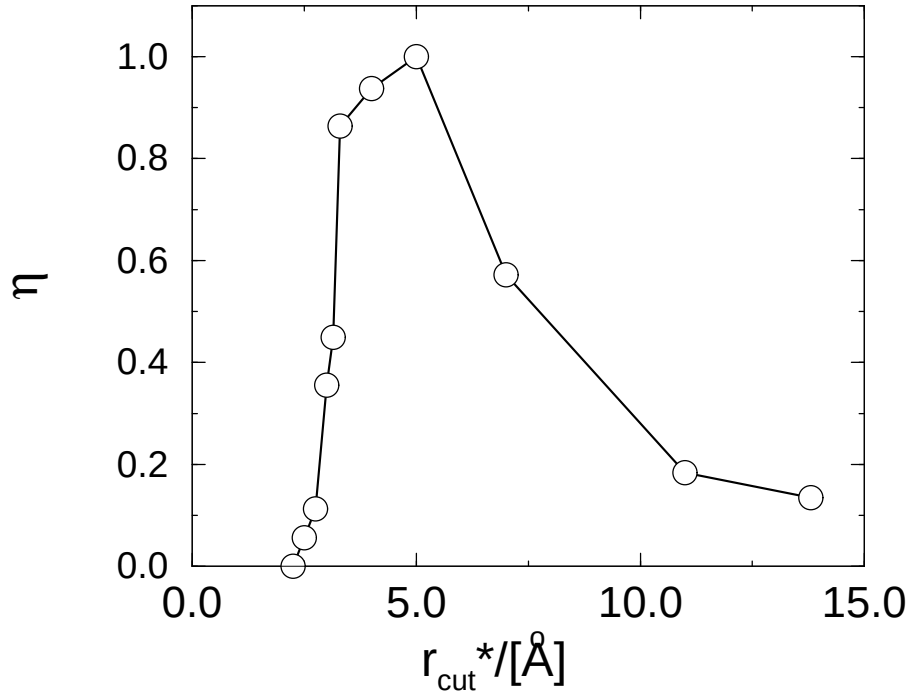
- Repeat until k trial orientations are found; these are fed into CBMC leading to $W(n)$

- Repeat procedure for old configuration, leading to $W^{\text{int}}(o)$ and $W(o)$.

- Accept or reject according to

$$\text{acc}(o \rightarrow n) = \min\left(1, \frac{W(n) \times W^{\text{int}}(n)}{W(o) \times W^{\text{int}}(o)}\right)$$

Significant Speedup: Dual-Cutoff CBMC



- Grow chain with approximate (cheaper) potential; $W^\star$
- Correct for difference later
  (δu, difference real and approximate potential for $selected$ configuration)

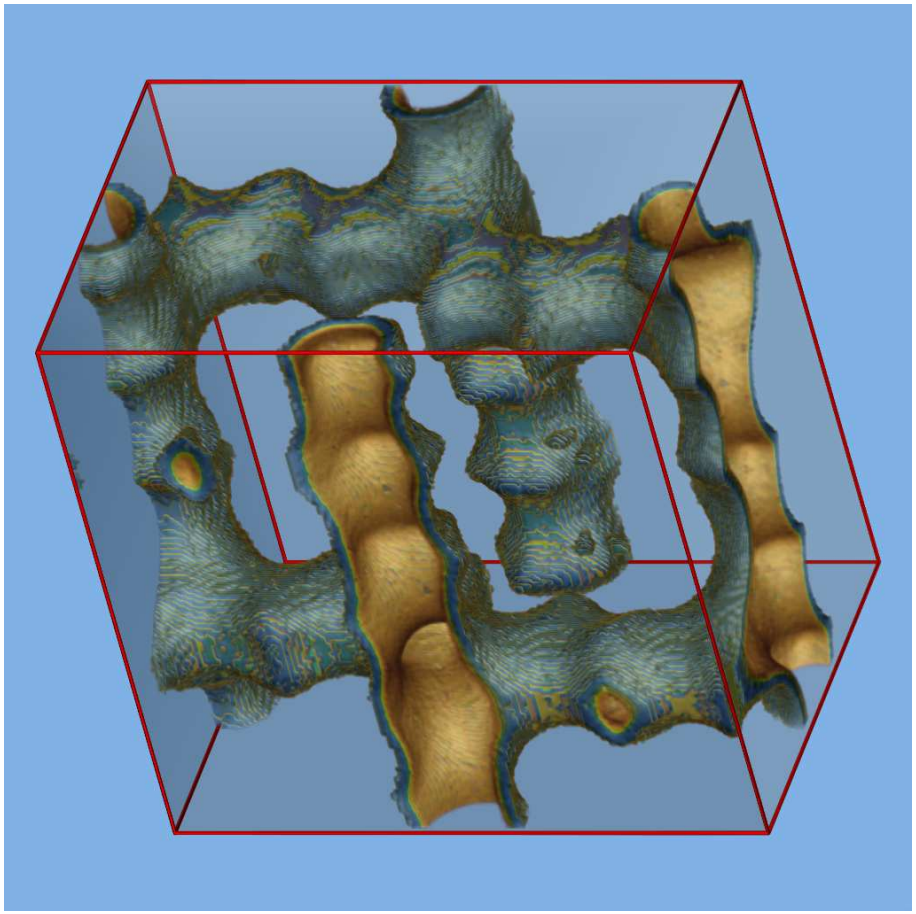
$$\mathrm{acc}(o \to n) = \min\left(1, \frac{W^\star(n)}{W^\star(o)} \times \exp[-\beta[\delta u(n) - \delta u(o)]]\right)$$

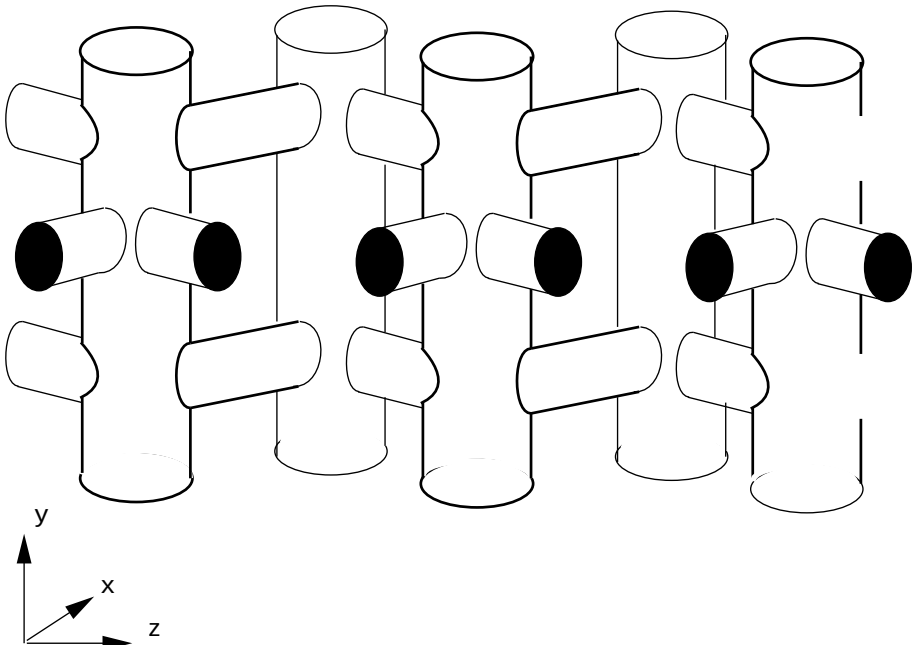
Application 1: Adsorption of Alkanes in MFI-type zeolite (1)

Zeolites:

- microporous channel structure
- crystalline, SiO_2 building blocks
- substitution of Si^{4+} by Al^{3+} and a cation (Na^+ or H^+)
- typical poresize: $4 - 12\mathring{A}$
- synthetic and natural; $>$190 framework types

Application 1: Adsorption of Alkanes in MFI-type zeolite (2)



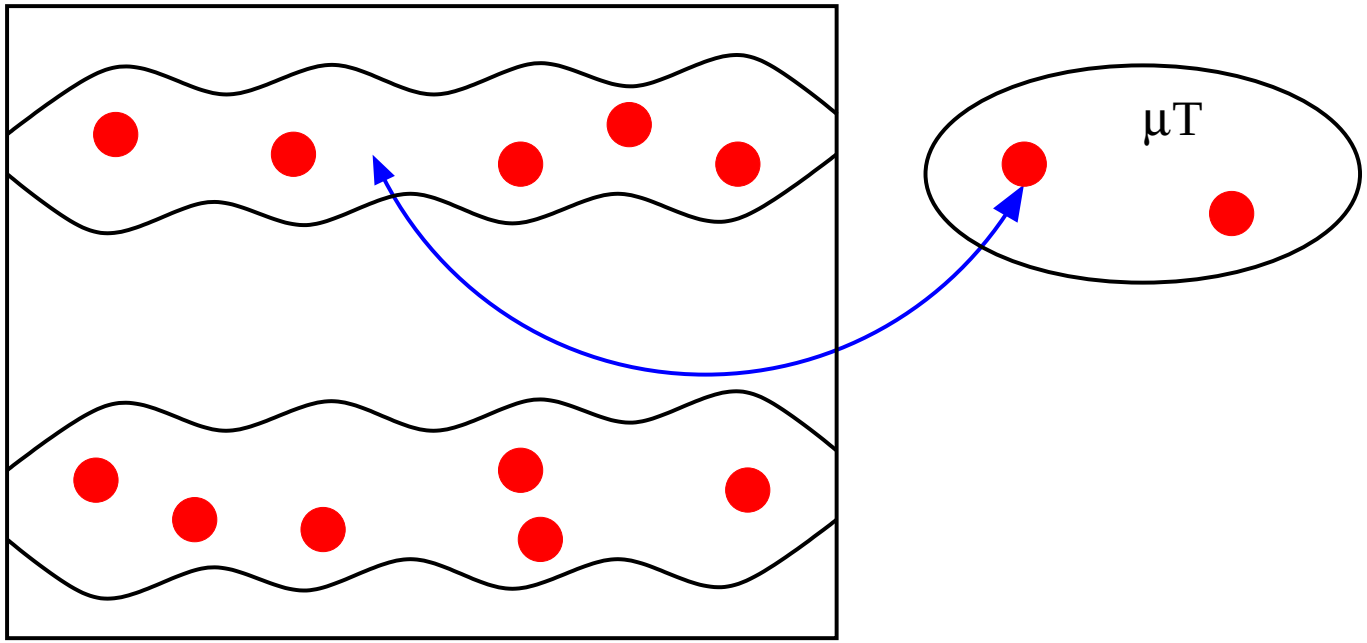


straight channels (y direction), zig-zag channels (xz plane) and intersections

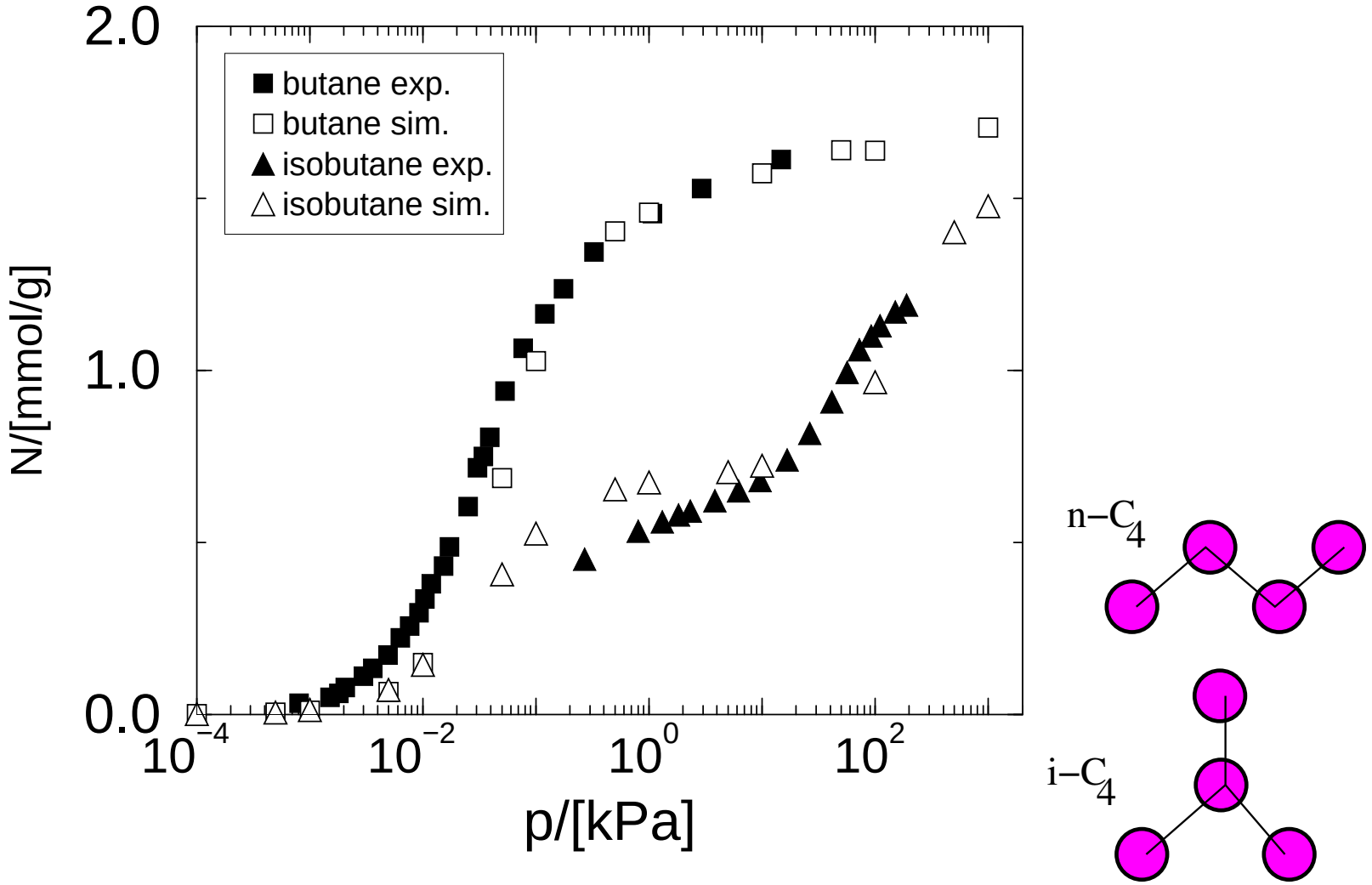
Application 1: Adsorption of Alkanes in MFI-type zeolite (3)

Grand-canonical (μVT) ensemble; number of particles (N) fluctuates

- system is coupled to particle reservoir at chemical potential μ and temperature T, statistical weight $\sim V^N \exp[\beta\mu N - \beta U(\mathbf{r}^N)] / \left(\Lambda^{3N} N!\right)$
- trial moves to exchange particles between zeolite and reservoir (using CBMC)
- equilibrium: $\mu_{\rm gas} = \mu_{\rm zeolite}$; measure average $\langle N \rangle$ for given μ and β



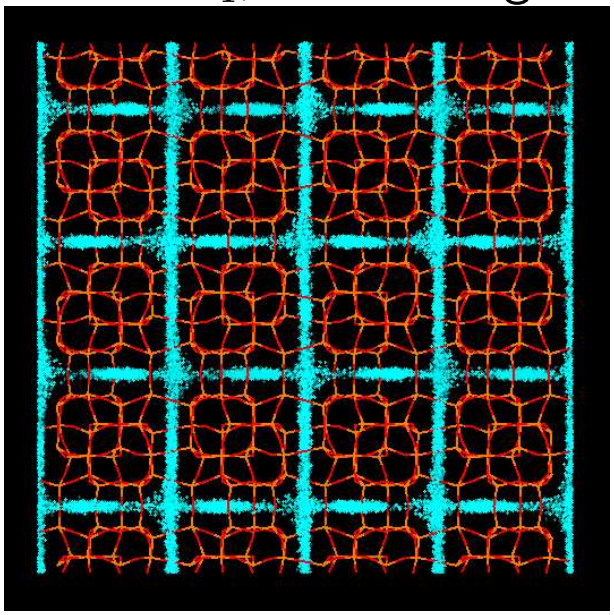
Application 1: Adsorption of Alkanes in MFI-type zeolite (4)



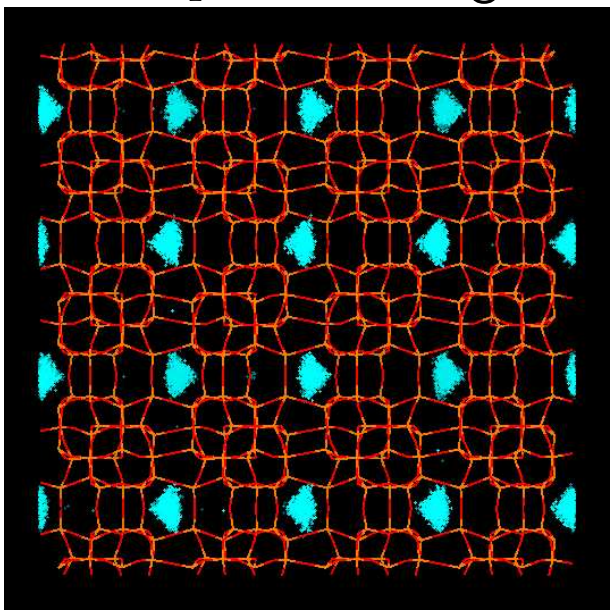
Vlugt et al, J. Am. Chem. Soc., 1998, 120, 5599

Application 1: Adsorption of Alkanes in MFI-type zeolite (5)

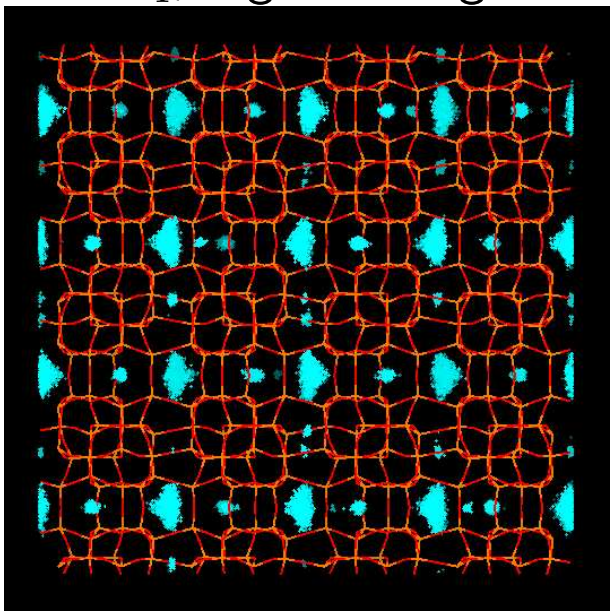
n-C_4, low loading



i-C_4, low loading



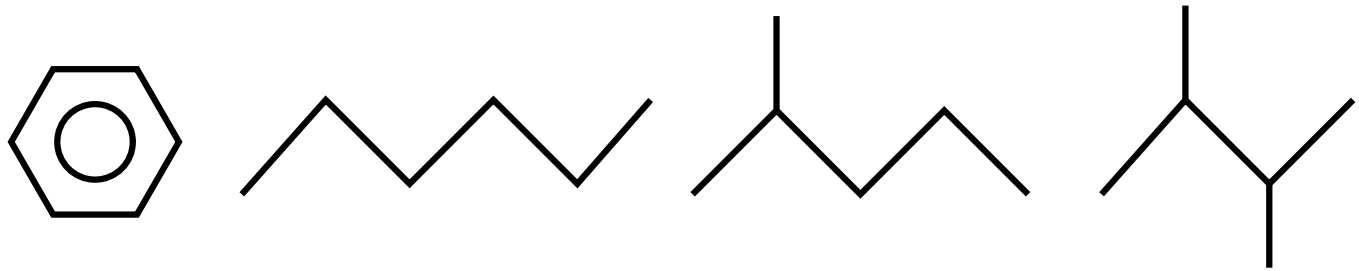
i-C_4, high loading



Vlugt et al, J. Am. Chem. Soc., 1998, 120, 5599

Application 1: Adsorption of Alkanes in MFI-type zeolite (6)

Research Octane Number (RON)

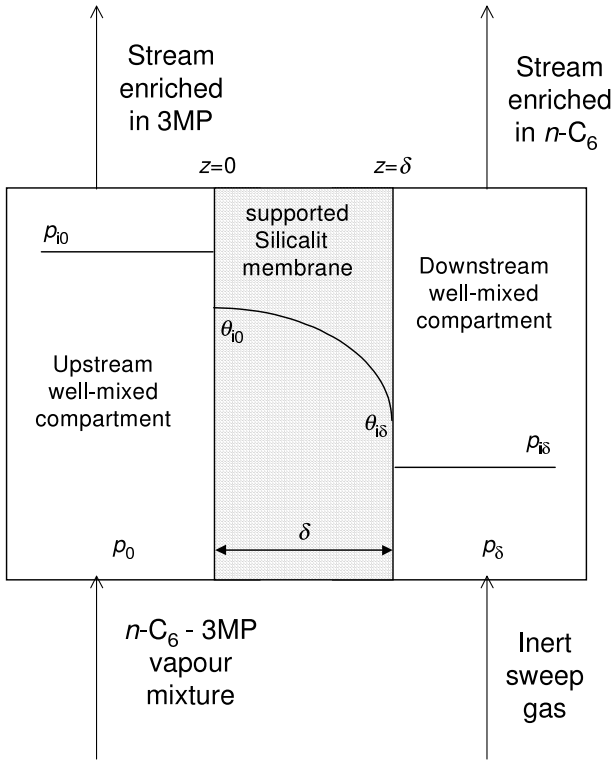


>100	25	73	92

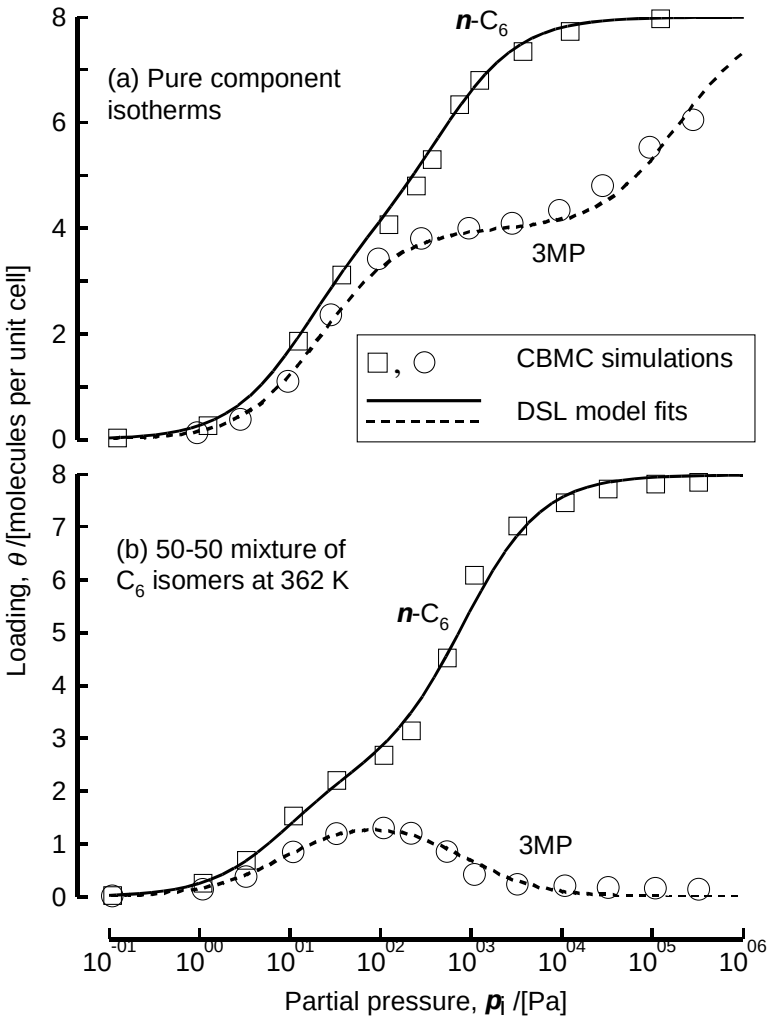
Application 1: Adsorption of Alkanes in MFI-type zeolite (7)

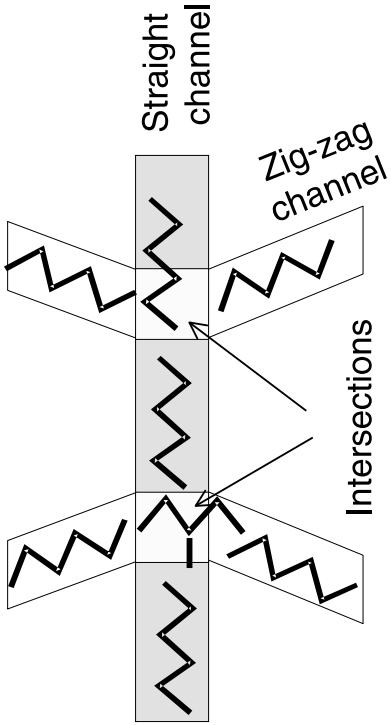
Flux	$n-C_6$	$i-C_6$	selectivity
pure	179	136	1.3
50%-50%	46	1.9	24

Experiments by J. Falconer, Univ. Colorado



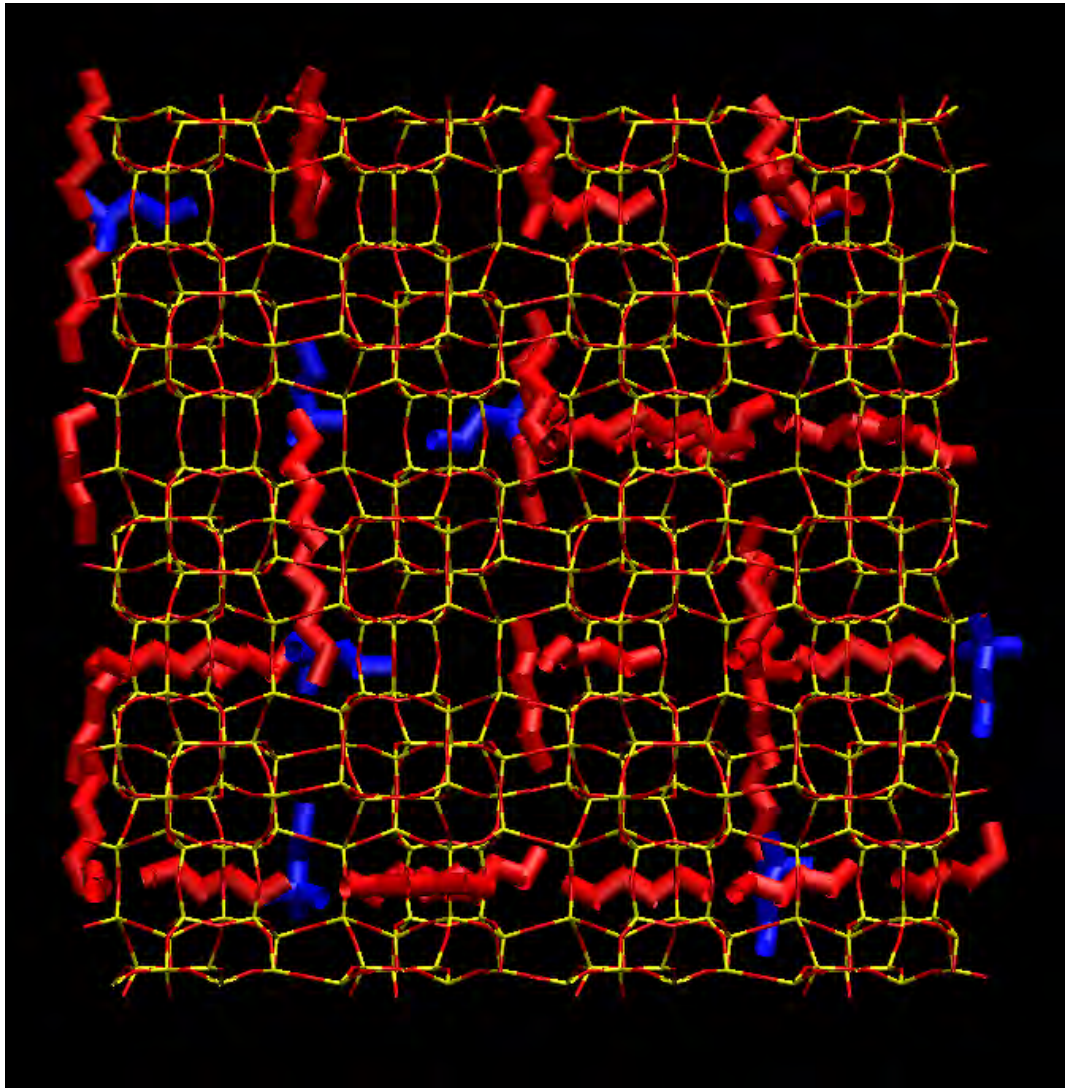
Application 1: Adsorption of Alkanes in MFI-type zeolite (8)



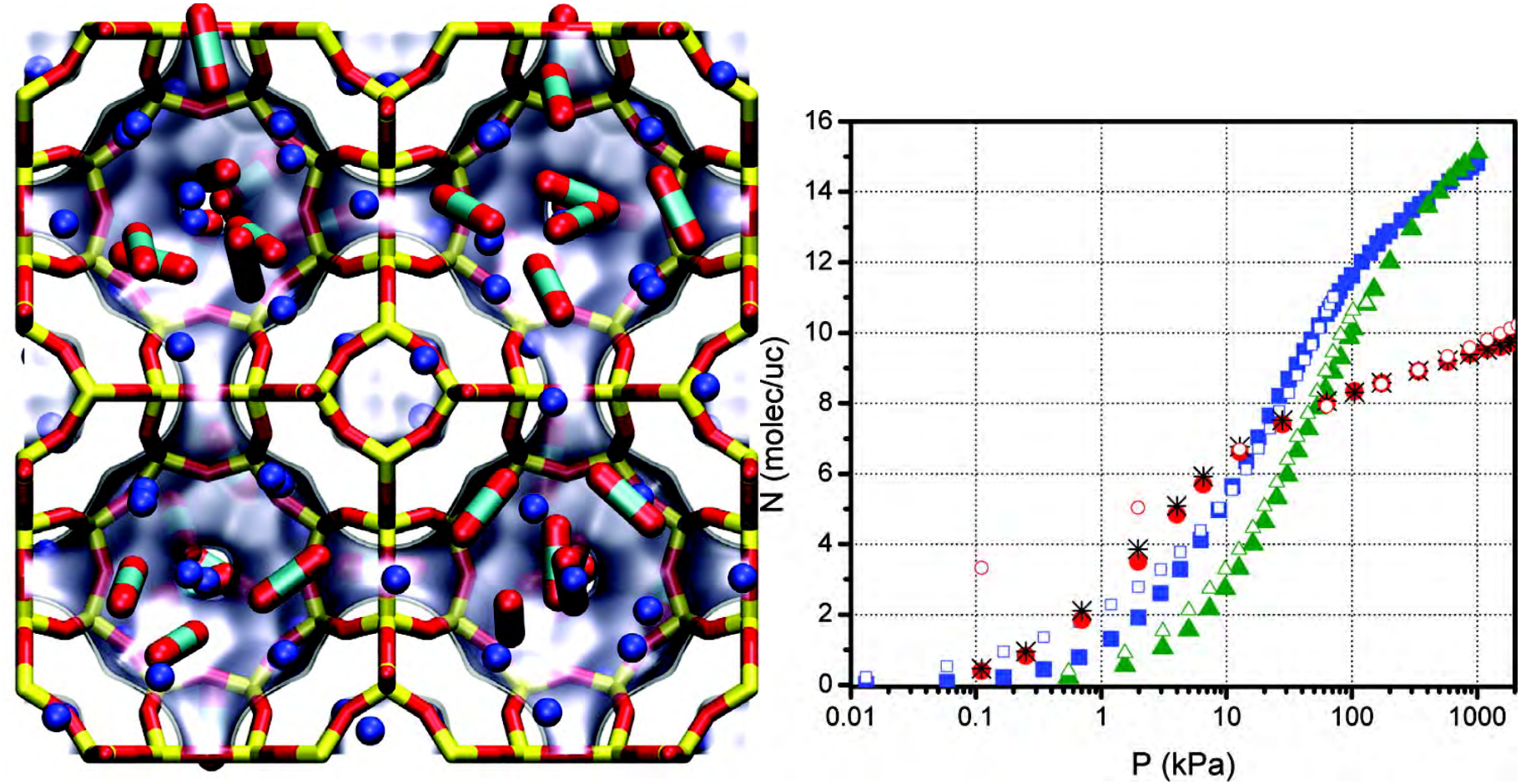


Application 1: Adsorption of Alkanes in MFI-type zeolite (9)

blue = branched (i-C_6) red = linear (n-C_6)

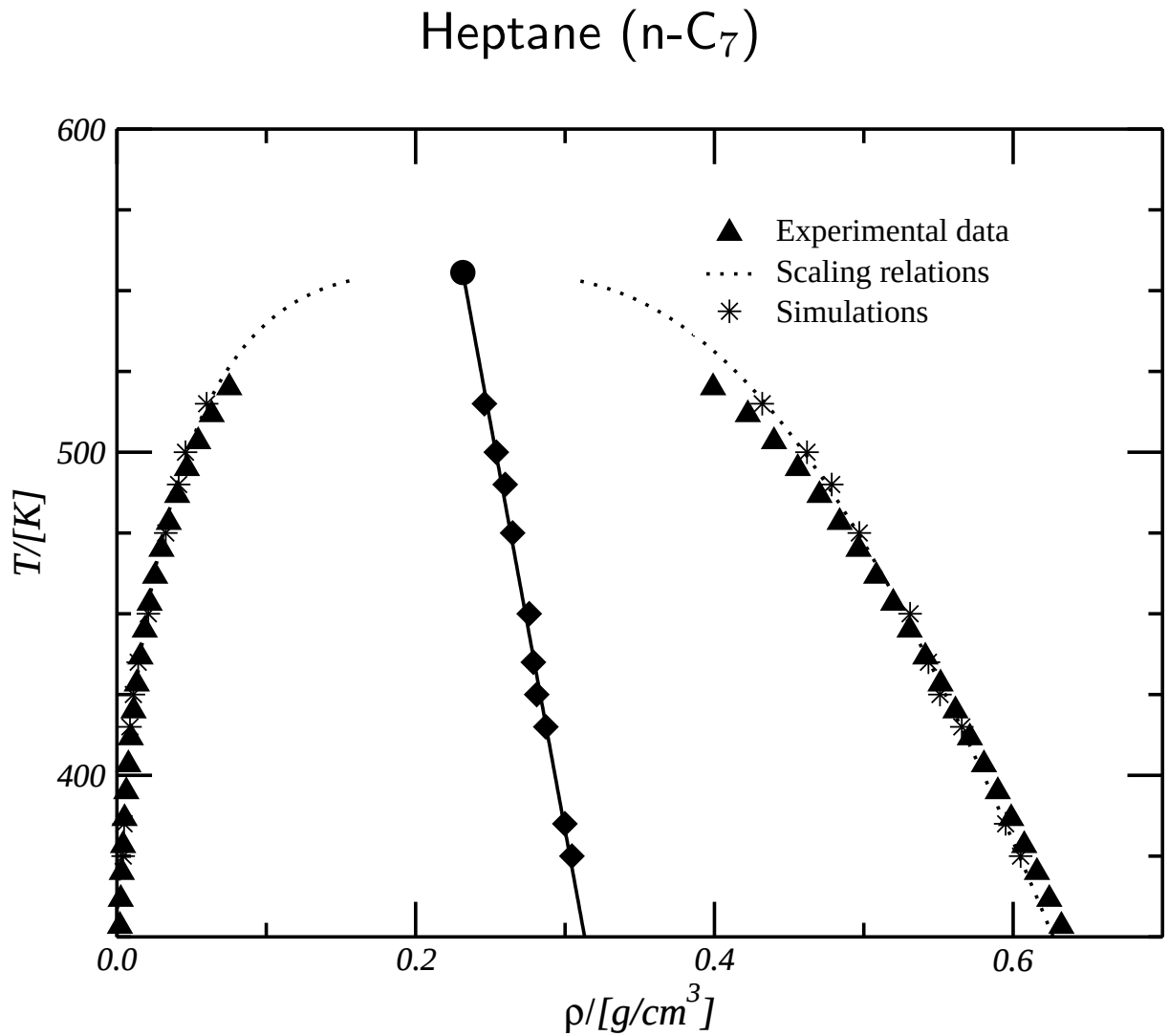


Application 2: Adsorption of CO_2 in Na^+ containing zeolites



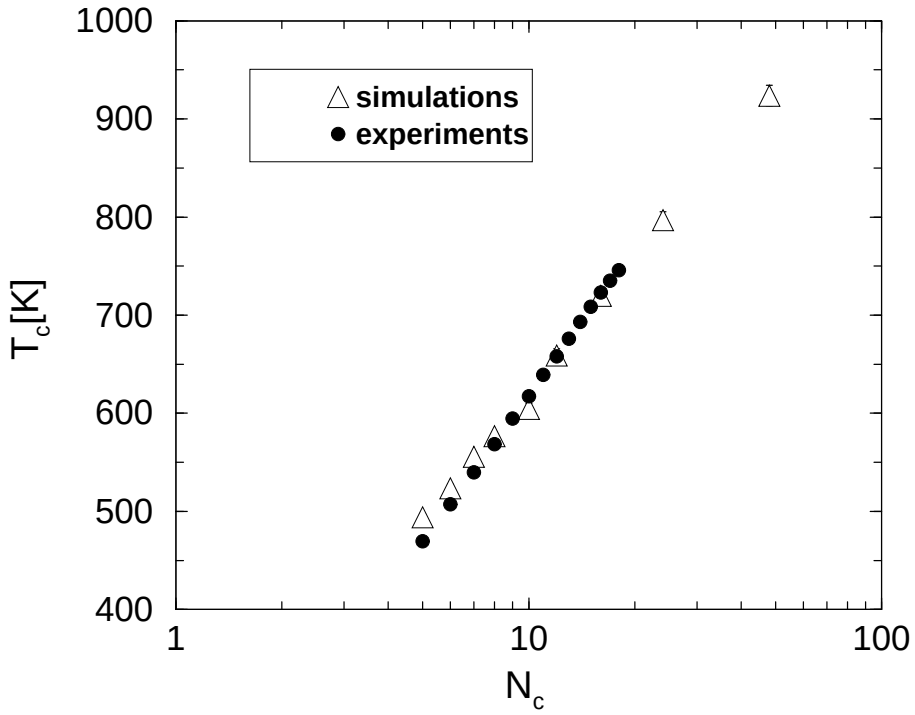
Sofia Calero et al., J. Phys. Chem. C, 2009, 113, 8814-8820

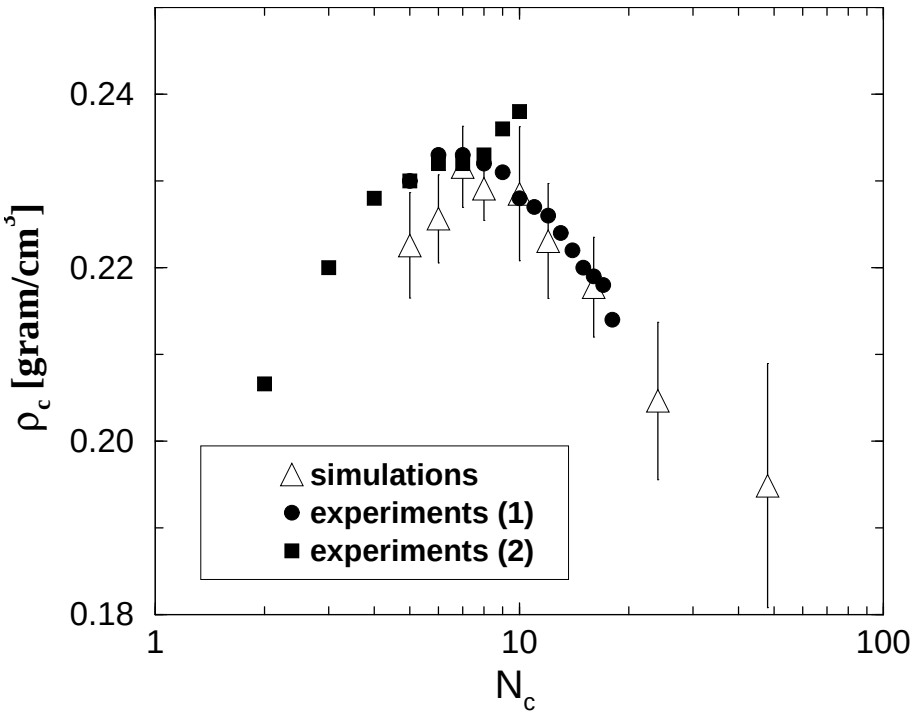
Application 3: Gibbs Ensemble Monte Carlo (1)



B. Smit, S. Karaborni, and J.I. Siepmann, J. Chem. Phys. 102, 2126 (1995)

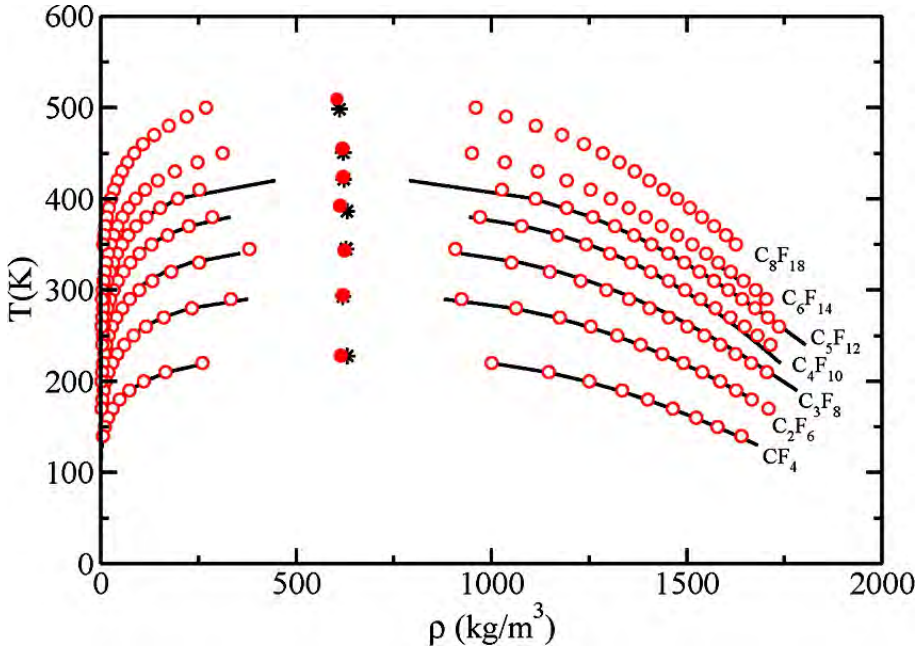
Application 3: Gibbs Ensemble Monte Carlo (2)

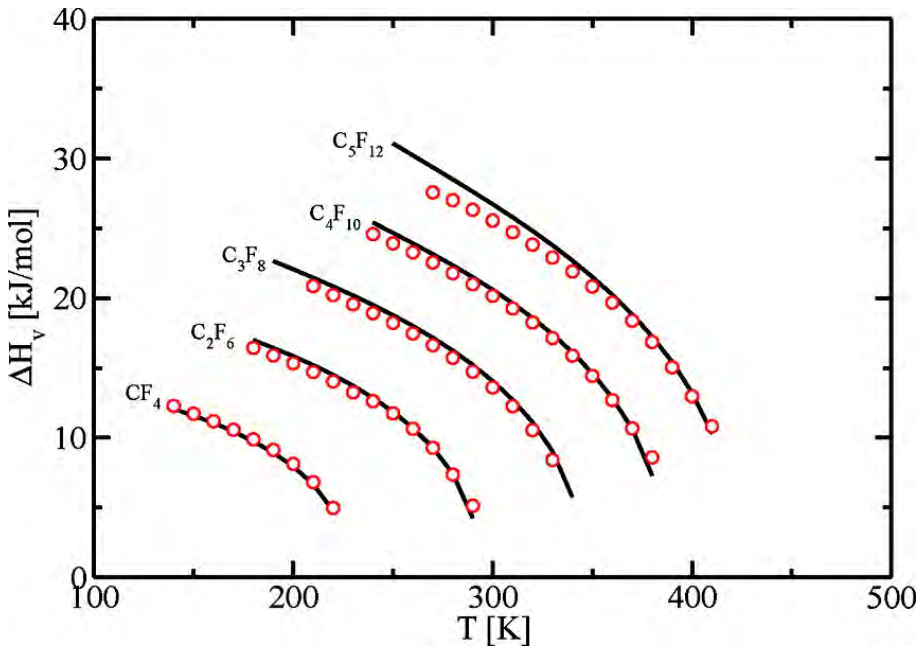




B. Smit, S. Karaborni, and J.I. Siepmann, J. Chem. Phys. 102, 2126 (1995)

Application 4: GCMC Histogram Reweighting





J.J. Potoff et al., J. Phys. Chem. B, 2009, 113, pp 14725-14731

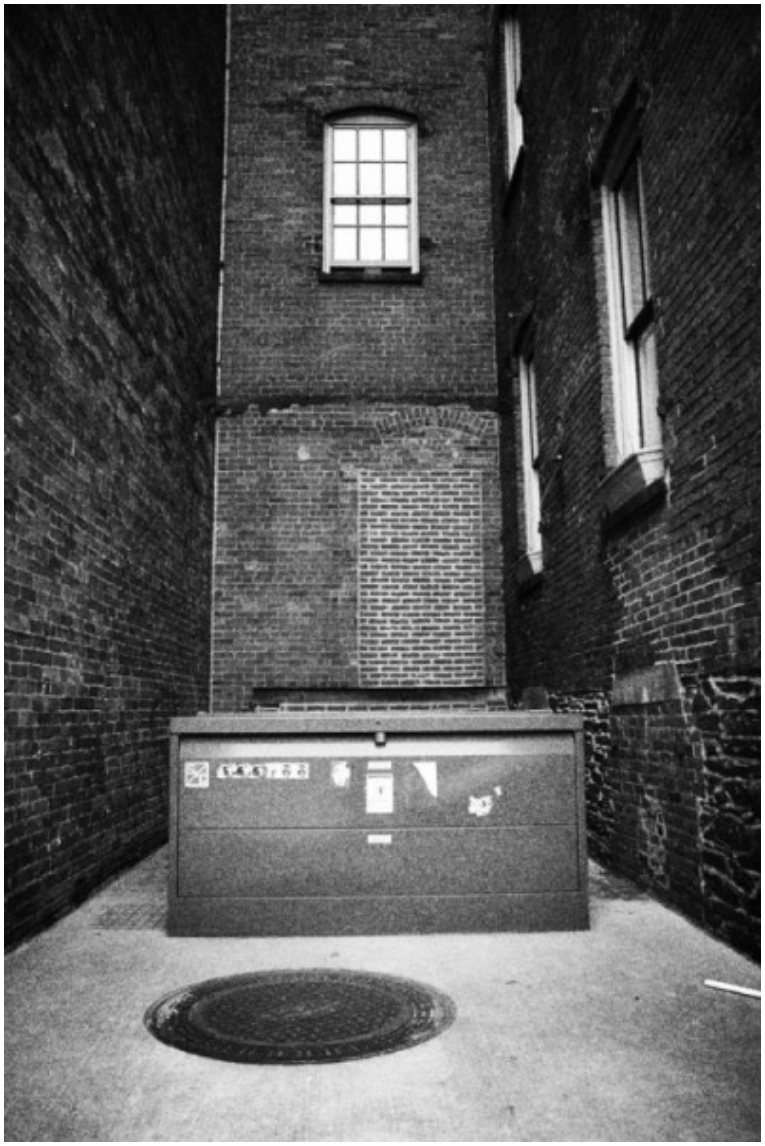
Efficiency of CBMC

- k: number of trial directions
- a: probability that trial direction has a "favorable" energy
- growth can continue as long as at least 1 trial direction is "favorable"
- generate chain of length N successfully

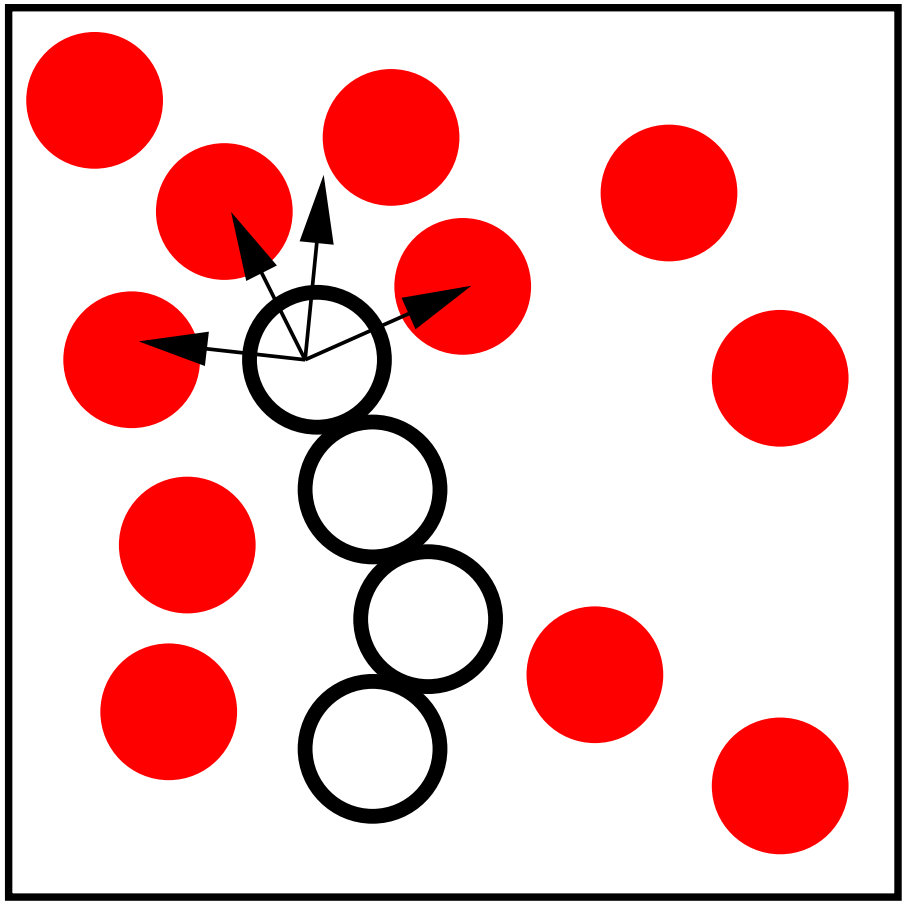
$$P_{\text{success}} = (1 - (1 - a)^k)^N = \exp[-cN]$$

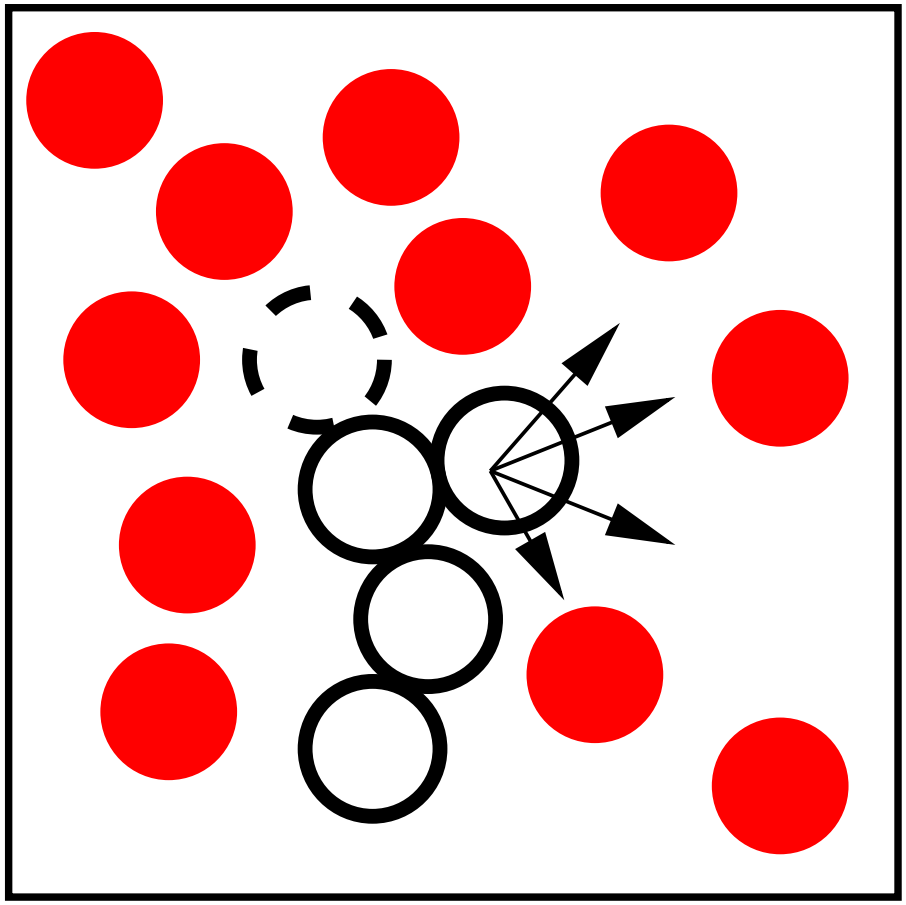
- increasing k means increasing CPU time.

Dead-End Alley



Recoil Growth: Avoiding Dead-End Alley



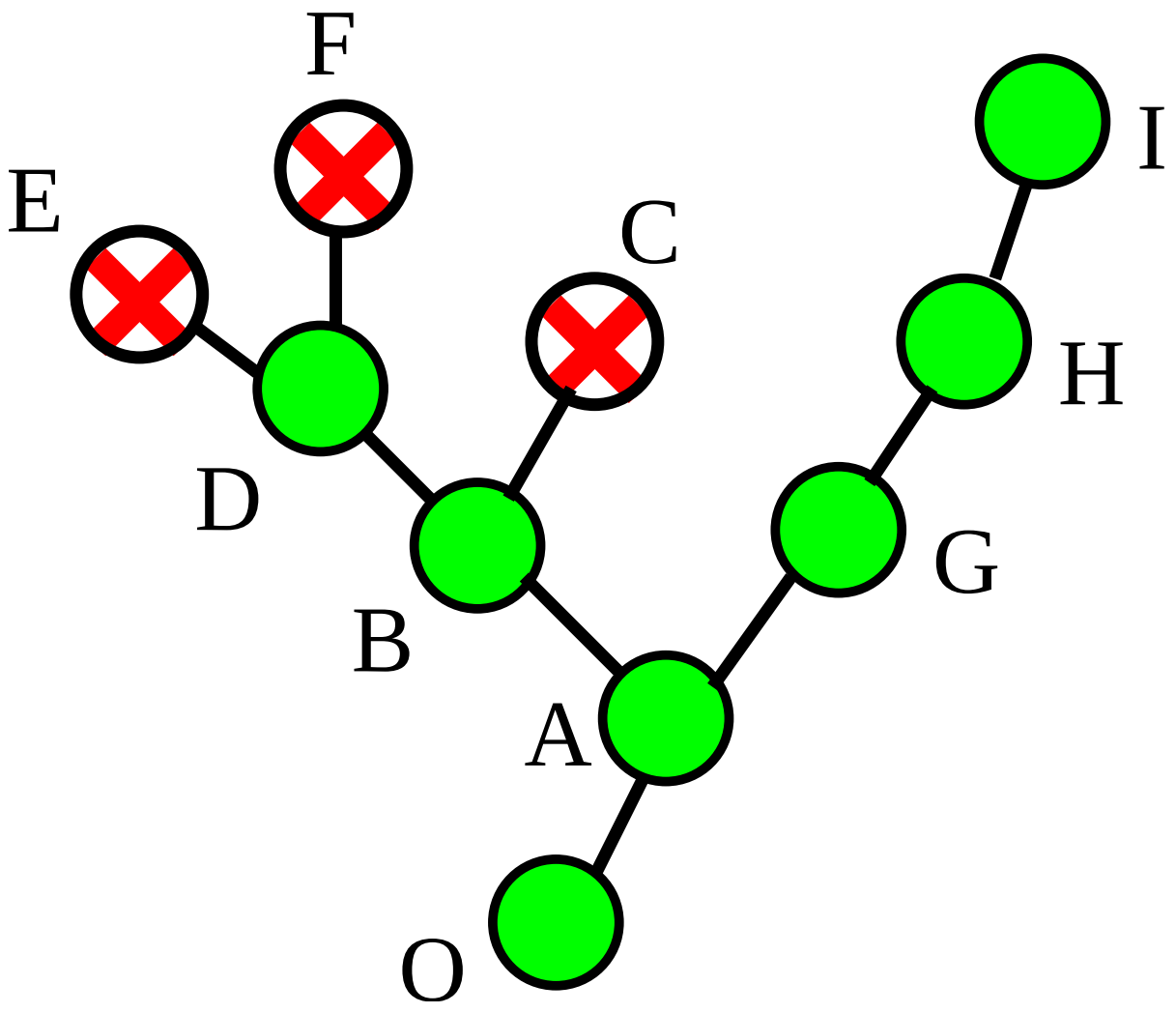


Recoil Growth Algorithm (1)

- Place first bead at random position
- Position (i) can be "open" or closed" depending on the environment (energy u_i); here we use $p_{\rm open} = \min(1, \exp[-\beta u_i])$ and toss a coin.
- If "open", continue with next segment
- If "closed", try another trial direction up to a maximum of k
- If all k directions are closed, retract by one step
- Maximum retraction length: $l_{\rm max} - l + 1$
- l: recoil length, $l_{\rm max}$: maximum length obtained during the growth of the chain
- Computed weight $W(n)$ and repeat procedure for old configuration
- Accept or reject using ${\rm acc}(o \rightarrow n) = \min(1, W(n)/W(o))$

Recoil Growth Algorithm (2)

Example for $k = 2$ and $l = 3$



Super Detailed Balance

In general,

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

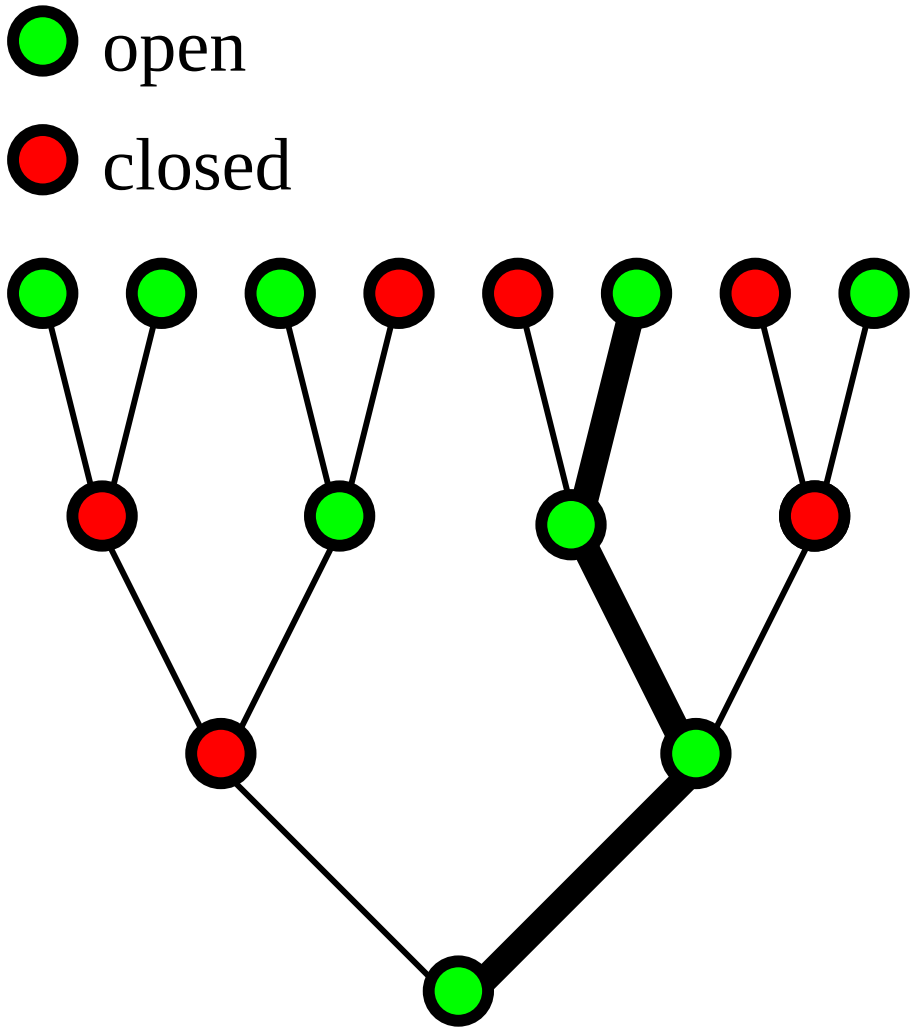
Therefore,

$$\mathrm{acc}(o \to n) = \min\left(1, \exp[-\beta \Delta U] \times \frac{\alpha(n \to o)}{\alpha(o \to n)}\right)$$

What about $\alpha(o \to n)$?

- Generate a tree t_n.
- Decide which parts of the tree are "open" or "closed" (O_n).
- Make a random walk on the tree (rw_n)

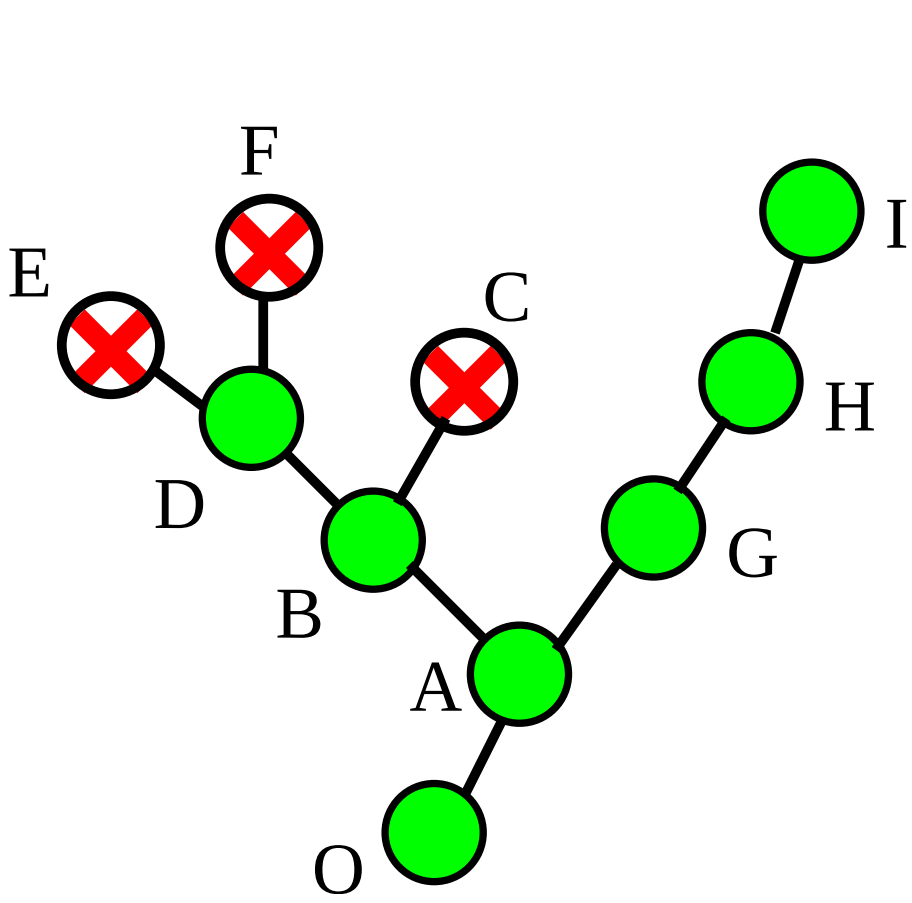
Random Walk on a Tree ($k = 2$, $l = 3$)

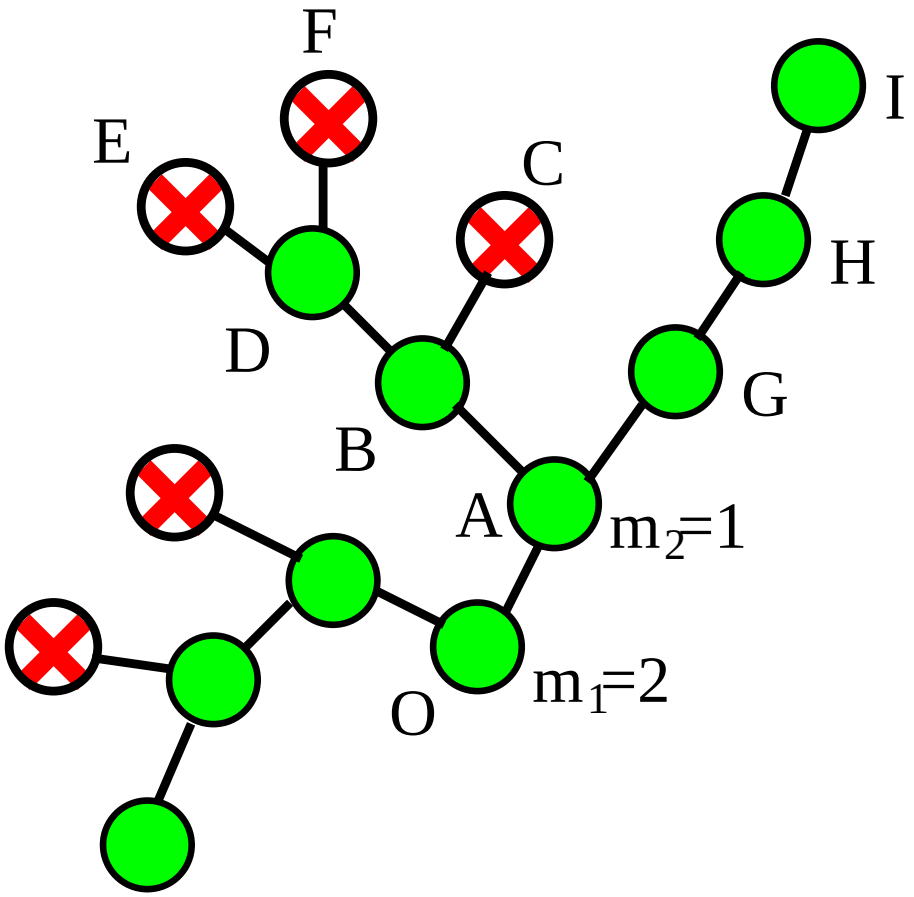


Super-Detailed Balance

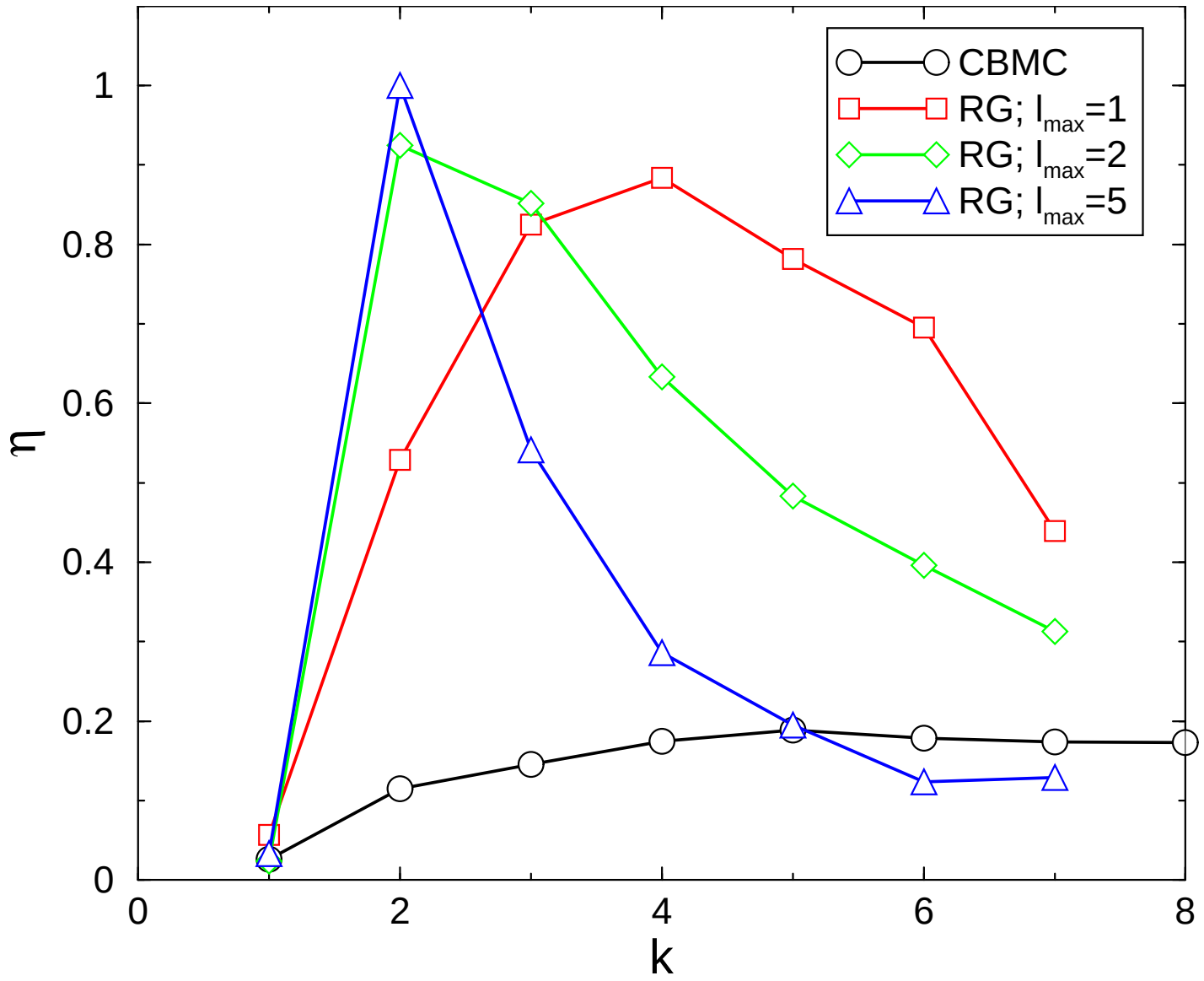
$$
\begin{aligned}
K(o \to n|t_n t_o O_n O_o) &= K(n \to o|t_n t_o O_n O_o) \\
\alpha(o \to n|t_n t_o O_n O_o) &= P(t_n)P(O_n|t_n)P(rw_n|t_n O_n) \times \\
&\quad P(t_o|rw_o)P(O_o|t_o rw_o) \\
\alpha(n \to o|t_n t_o O_n O_o) &= P(t_o)P(O_o|t_o)P(rw_o|t_o O_o) \times \\
&\quad P(t_n|rw_n)P(O_n|t_n rw_n) \\
\frac{P(O_n|t_n)}{P(O_n|t_n rw_n)} &= \prod_{i=1}^{n} p_i \\
P(rw_n|t_n O_n) &= \frac{1}{\prod_{i=1}^{n} m_i} \\
\mathrm{acc}(o \to n) &= \min\left(1, \exp[-\beta\Delta U] \times \frac{\prod_{i=1}^{n} \frac{m_i(n)}{p_i(n)}}{\prod_{i=1}^{n} \frac{m_i(o)}{p_i(o)}}\right)
\end{aligned}
$$

Computing the Weight ($k = 2$, $l = 3$)

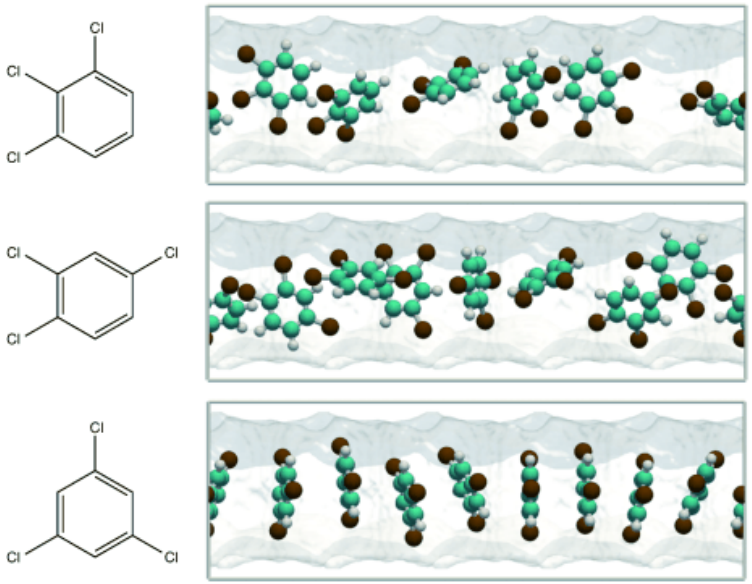


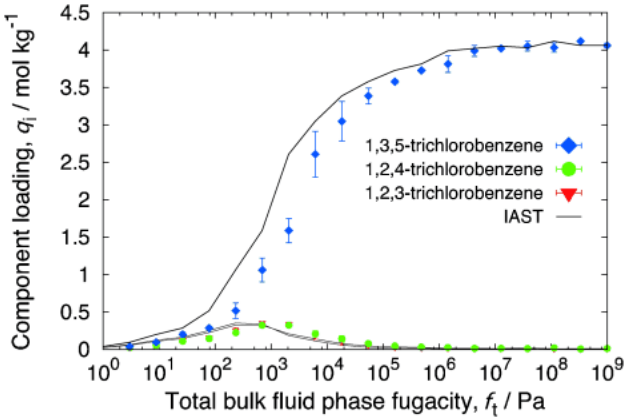


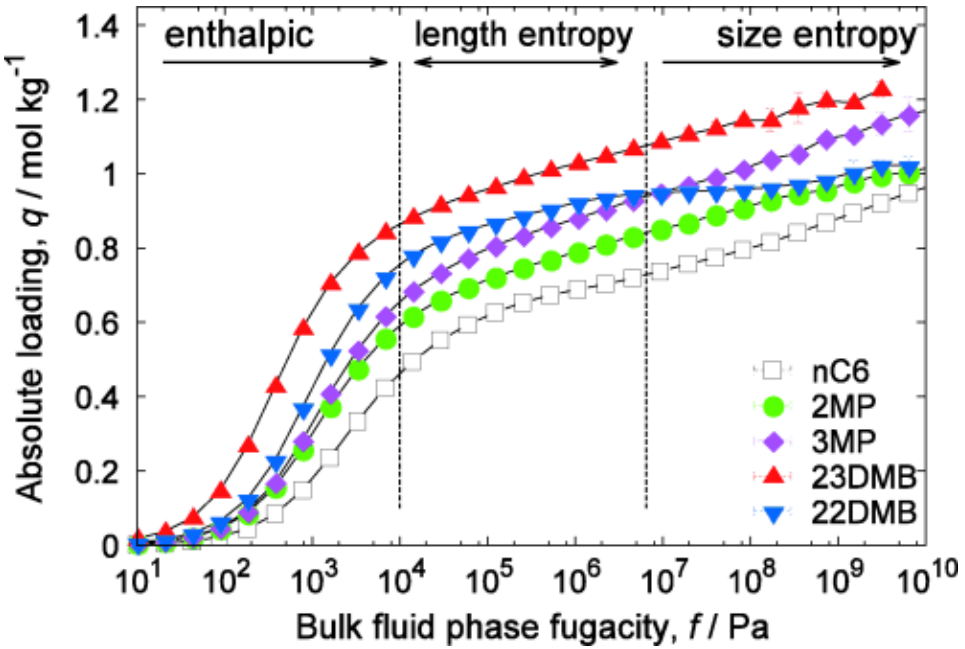
Efficiency of RG Compared to CBMC



High loadings are relevant!



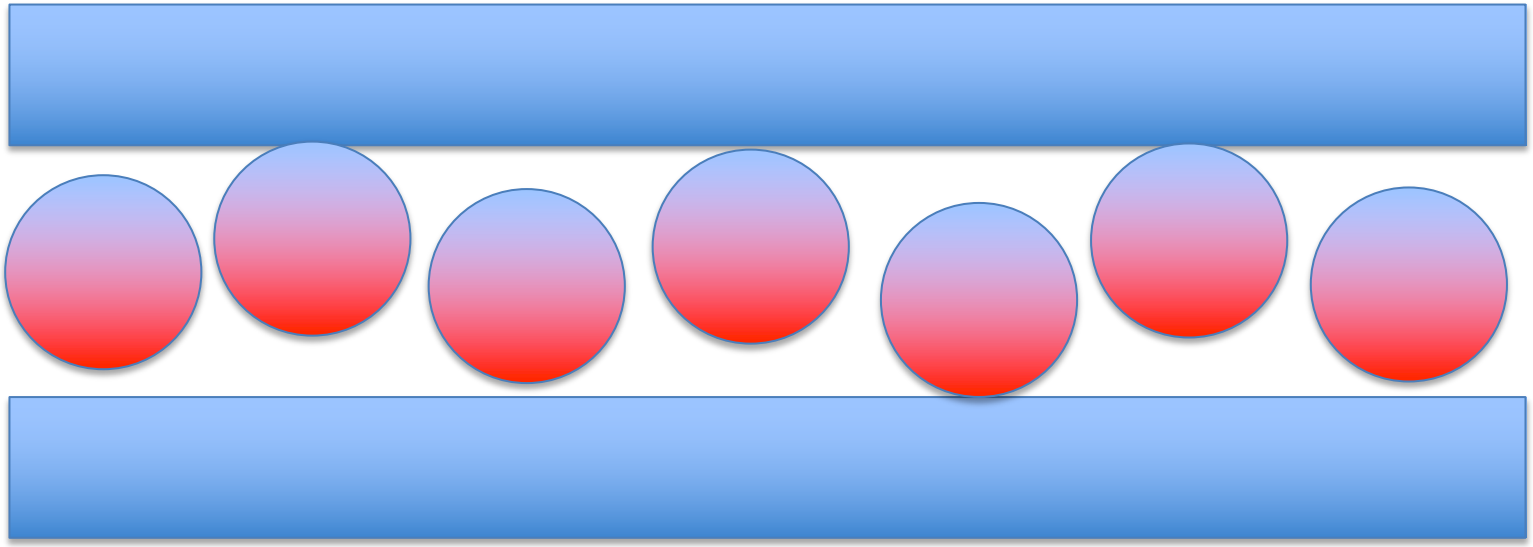




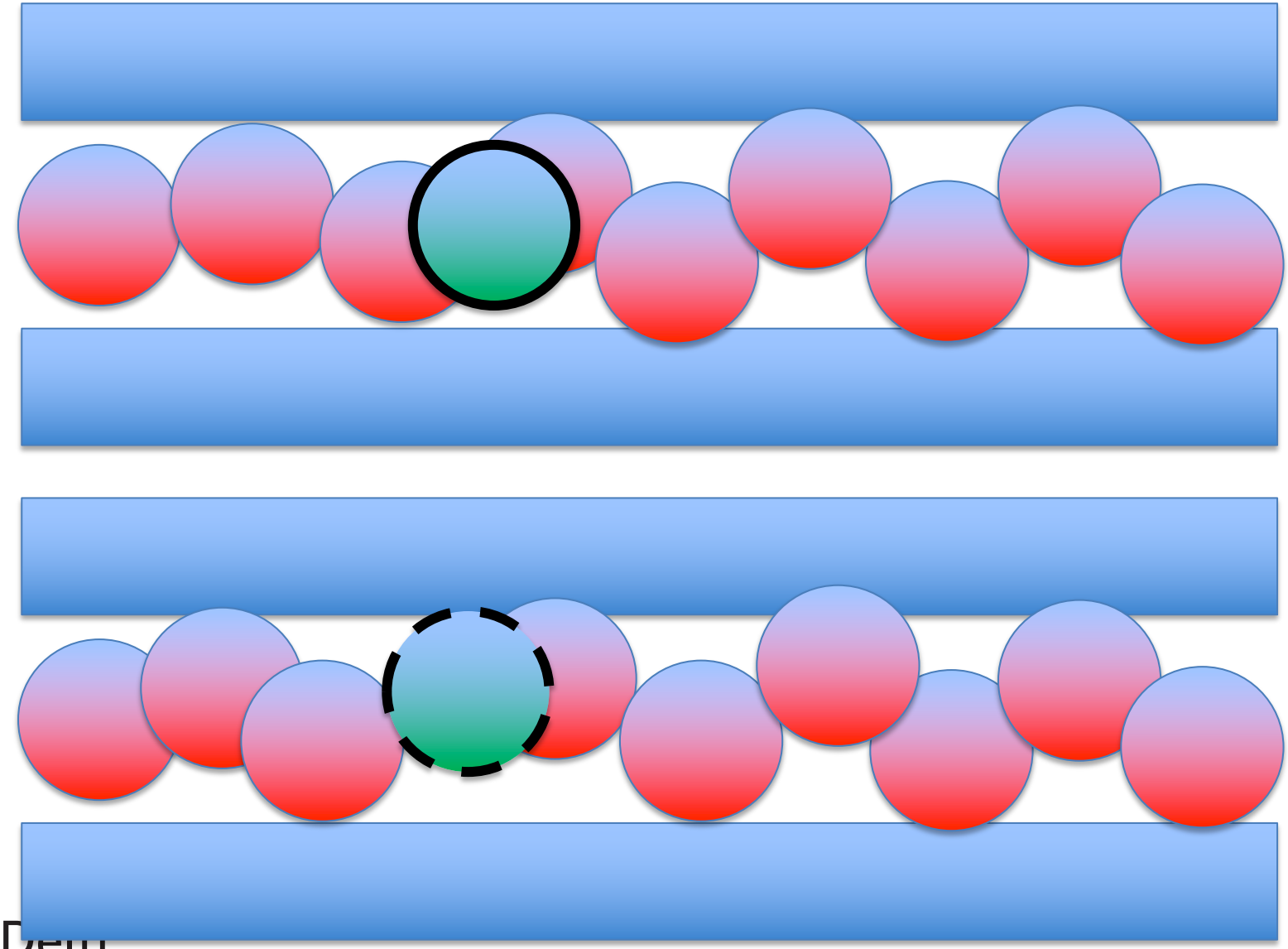
David Dubbeldam and co-workers
ChemPhysChem, *16* (3), 532-535
ChemPhysChem, *16* (10), 2046-2067

TU Delft

What is the maximum pore loading?

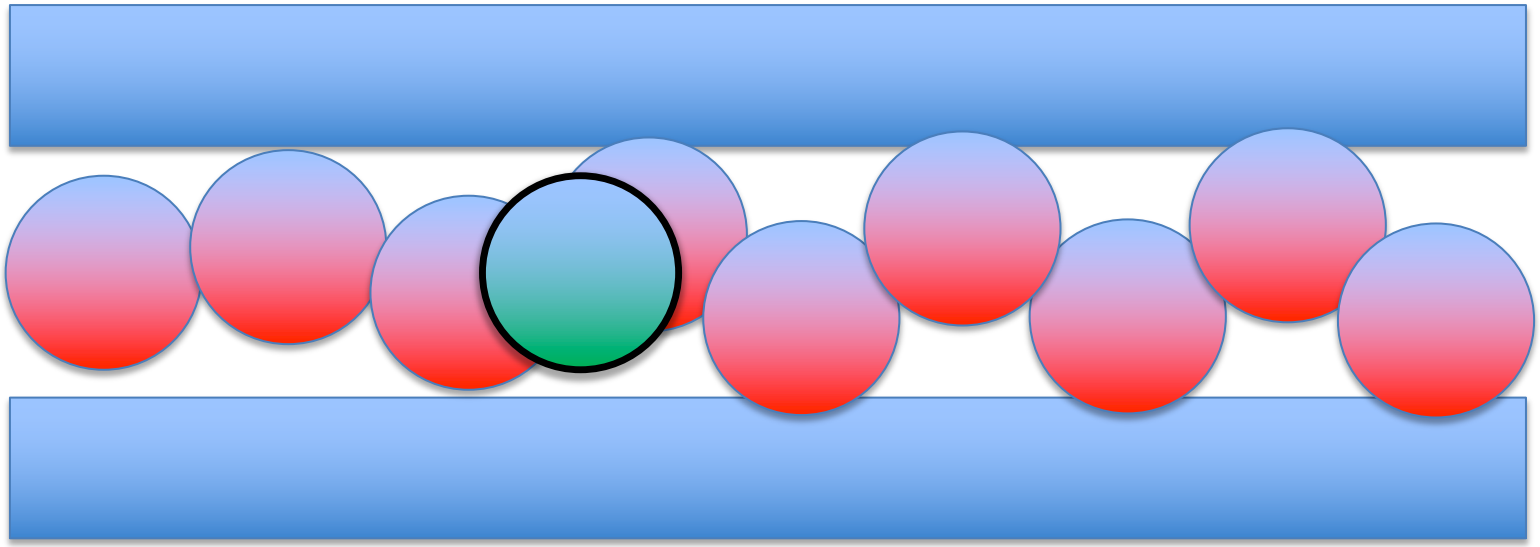


Insertion of a whole vs fractional molecule

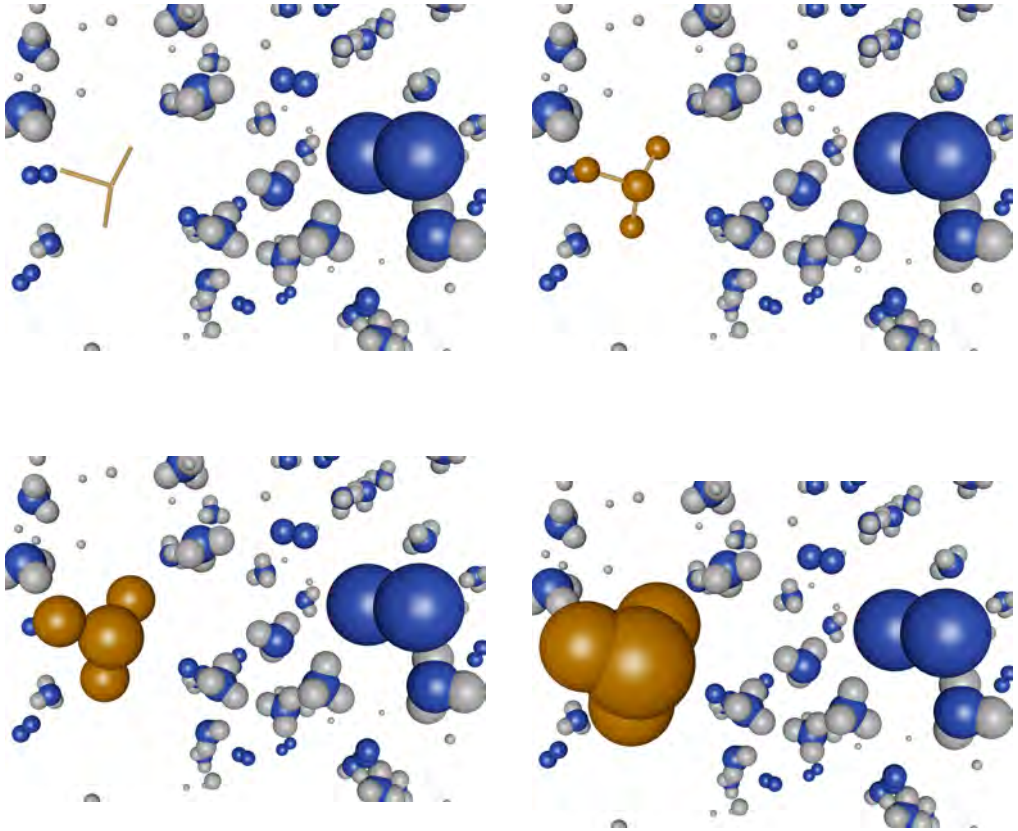


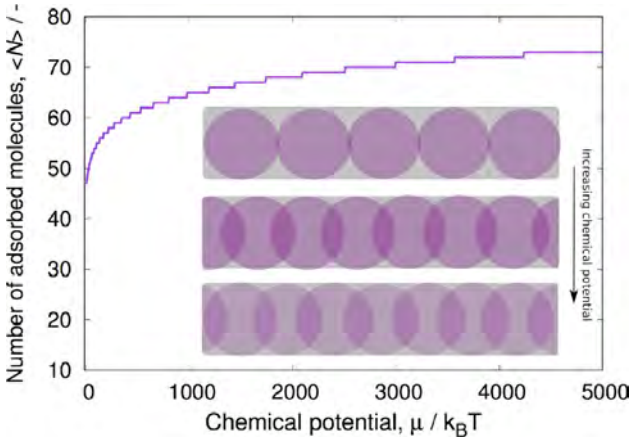
Zero times infinity problem → bad statistics

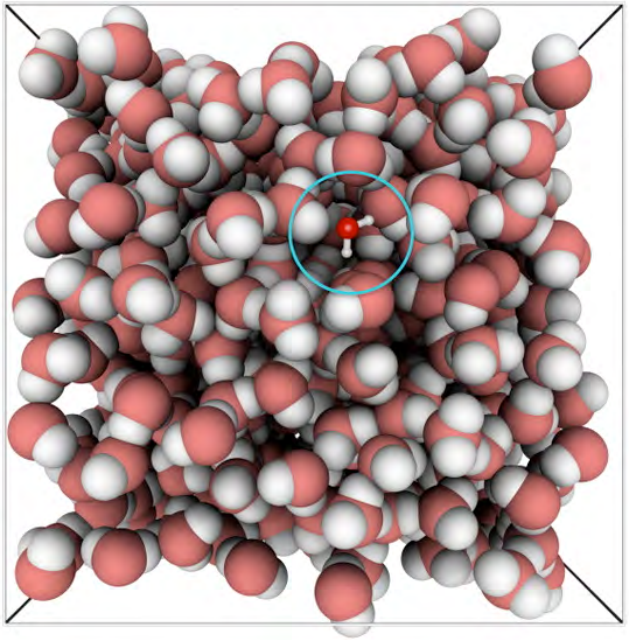
p(25M€) = (#wins)/(#tickets)



Insertions made easy using fractional molecules



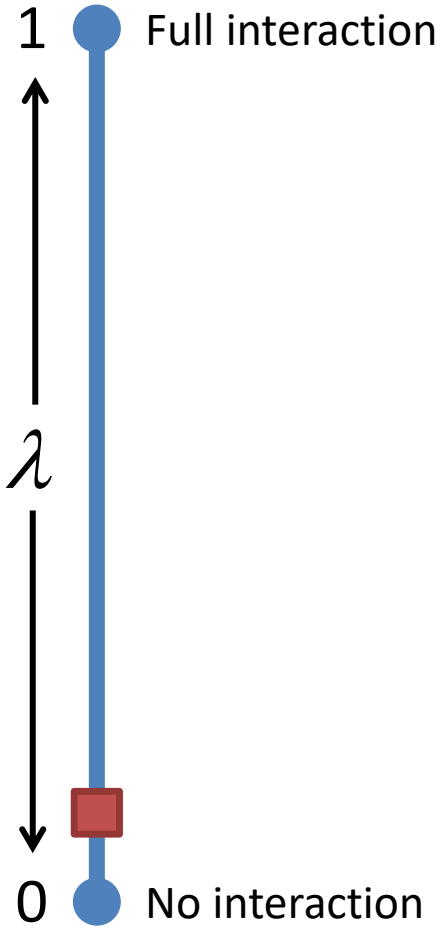


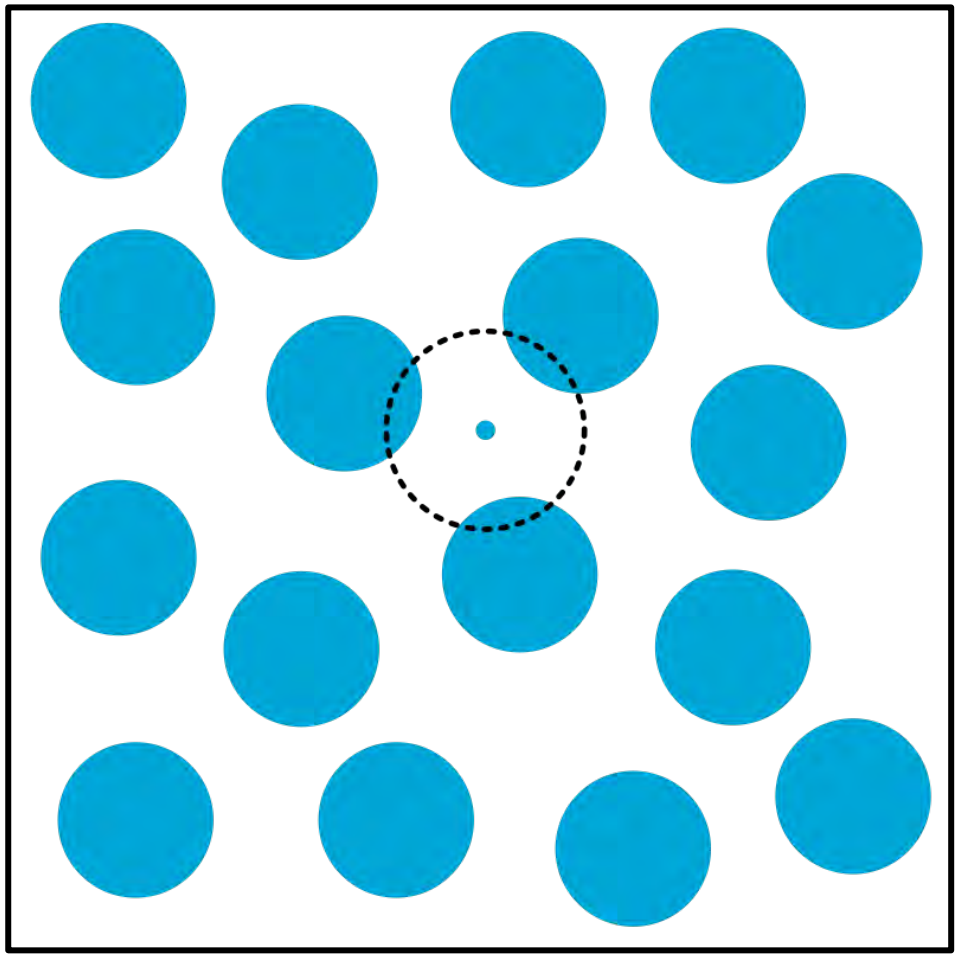


Shi & Maginn, *J. Chem. Theory Comput.*, 2007, 3, 1451–1463

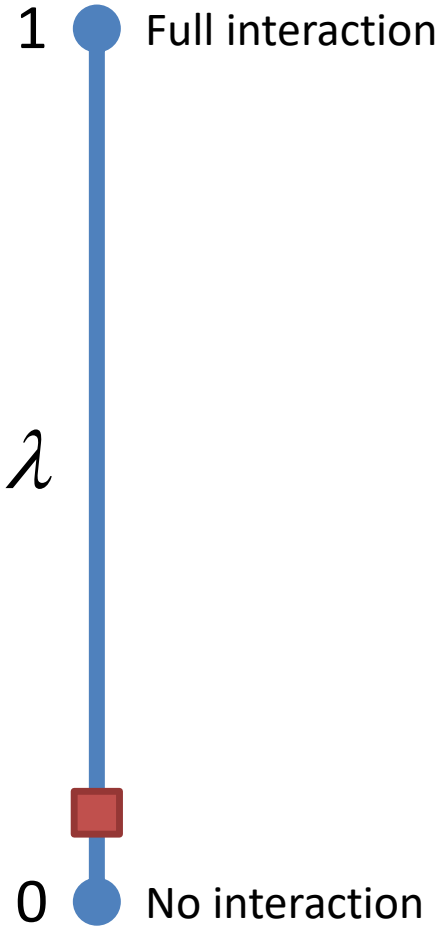
TU Delft

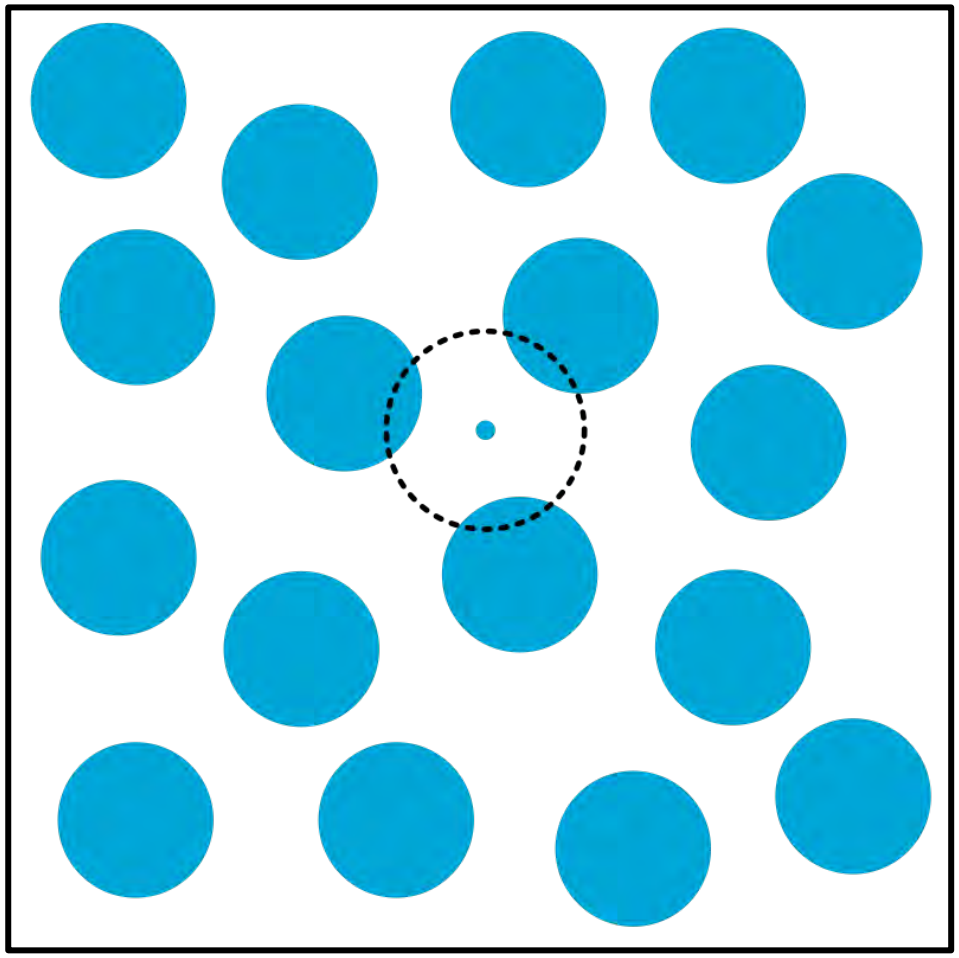
CFCMC



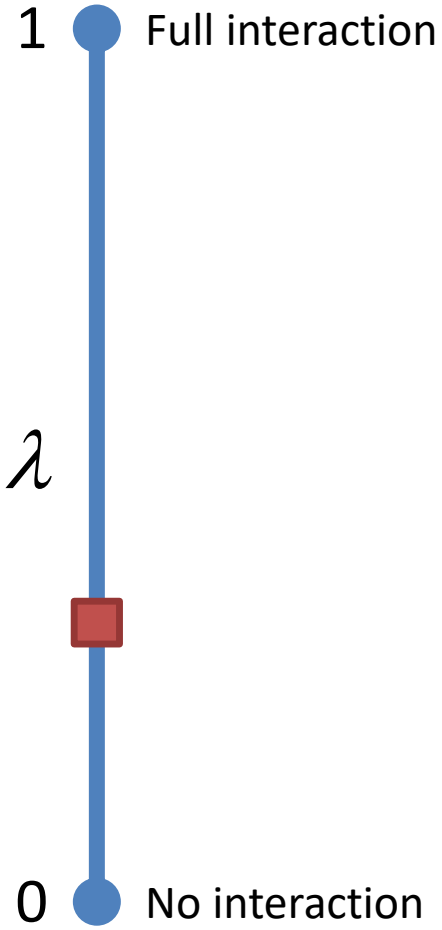


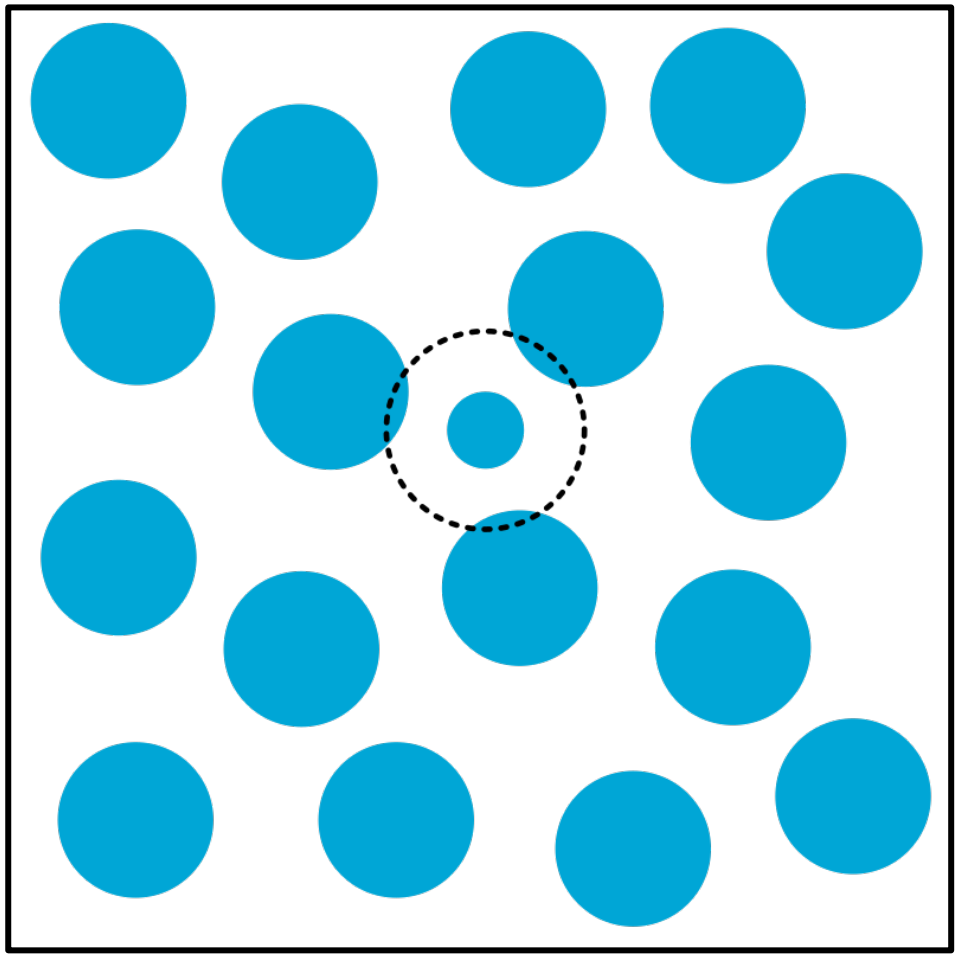
CFCMC



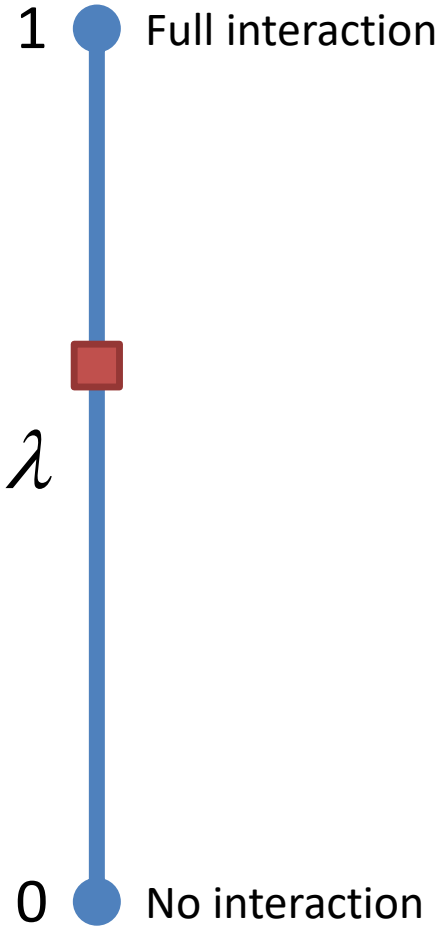


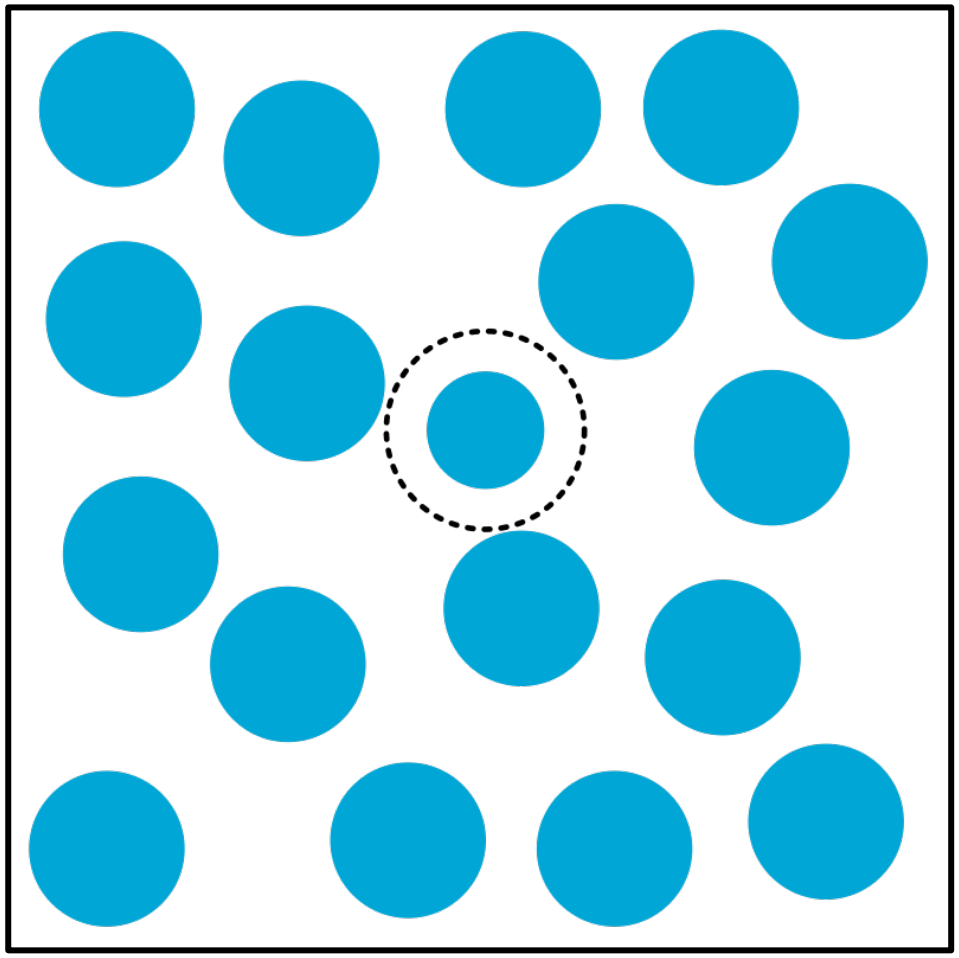
CFCMC



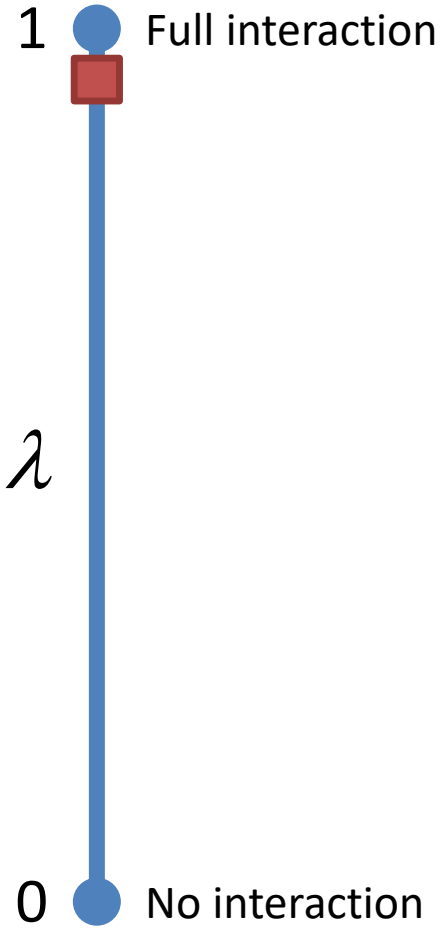


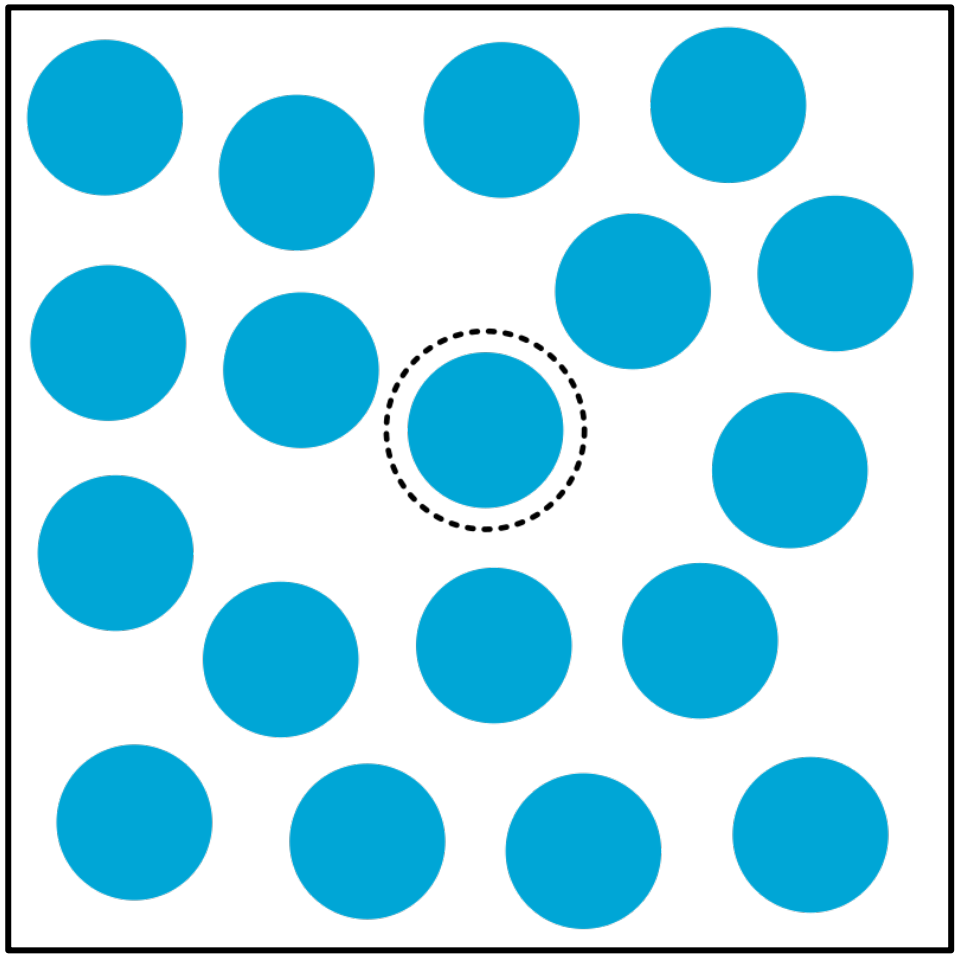
CFCMC



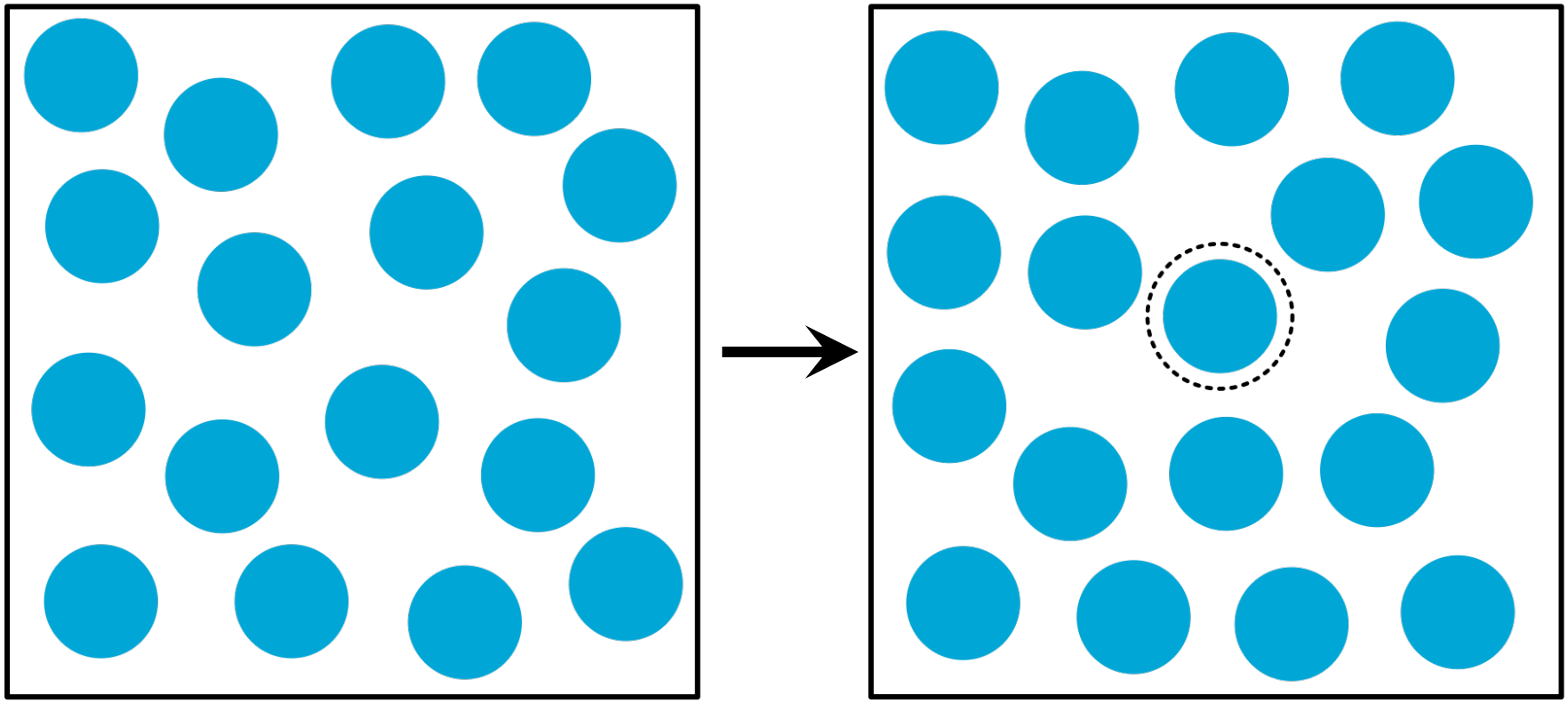


CFCMC



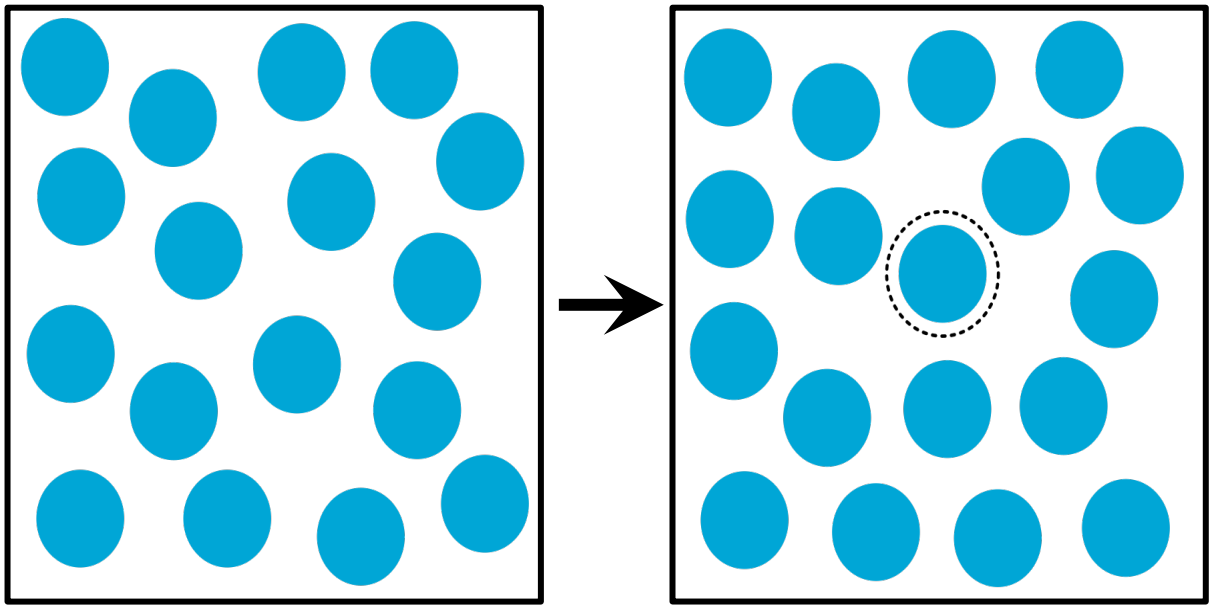


CFCMC



$\lambda = 0$ $\lambda = 1$

Avoid singularities



$$u_{\mathrm{LJ}}(r) = \lambda 4\varepsilon \left[\frac{1}{\left[\frac{1}{2}(1-\lambda)^2 + \left(\frac{r}{\sigma}\right)^6\right]^2} - \frac{1}{\left[\frac{1}{2}(1-\lambda)^2 + \left(\frac{r}{\sigma}\right)^6\right]} \right]$$

CFCMC in the grand-canonical ensemble

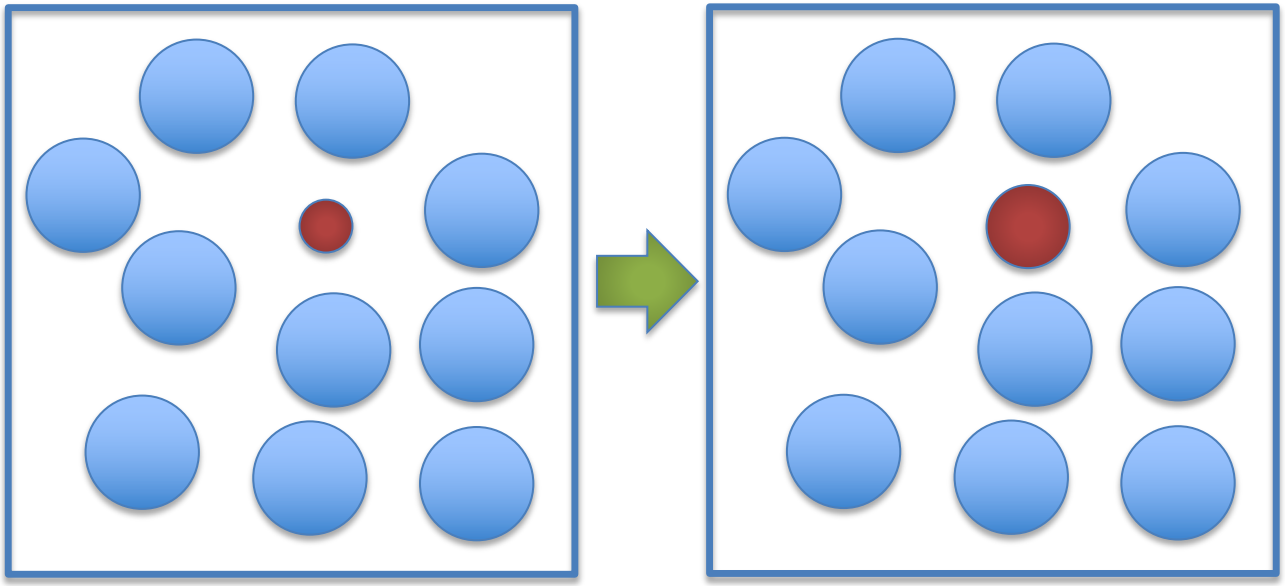
- *N* whole molecules, 1 fractional molecule
- Fractional molecule has scaled interactions (λ)
  - λ=0: no interactions with surrounding molecules
  - λ=1: full interactions with surrounding molecules
- Usual trial moves (translation, rotation, volume change, etc.)
- New types of trial moves:
  - Change value of λ
  - Identity change of the fractional molecule

Shi & Maginn, *J. Chem. Theory Comput.*, 2007, 3, 1451–1463

Changes in lambda

- λ(new) between 0 and 1

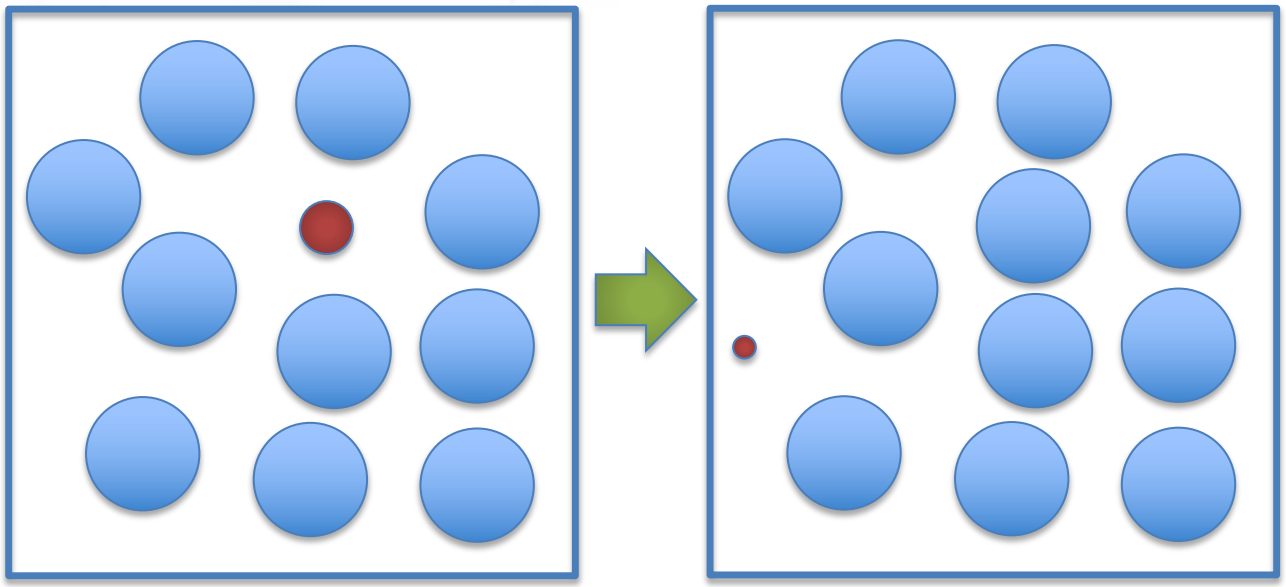
$$P_{acc} = \min(1, \exp[-\beta(U_{inter}(n) - U_{inter}(o)) + \eta(\lambda(n)) - \eta(\lambda(o))])$$



Changes in lambda

- λ(new) larger than 1: fractional becomes whole and insert new fractional with λ(new) -1

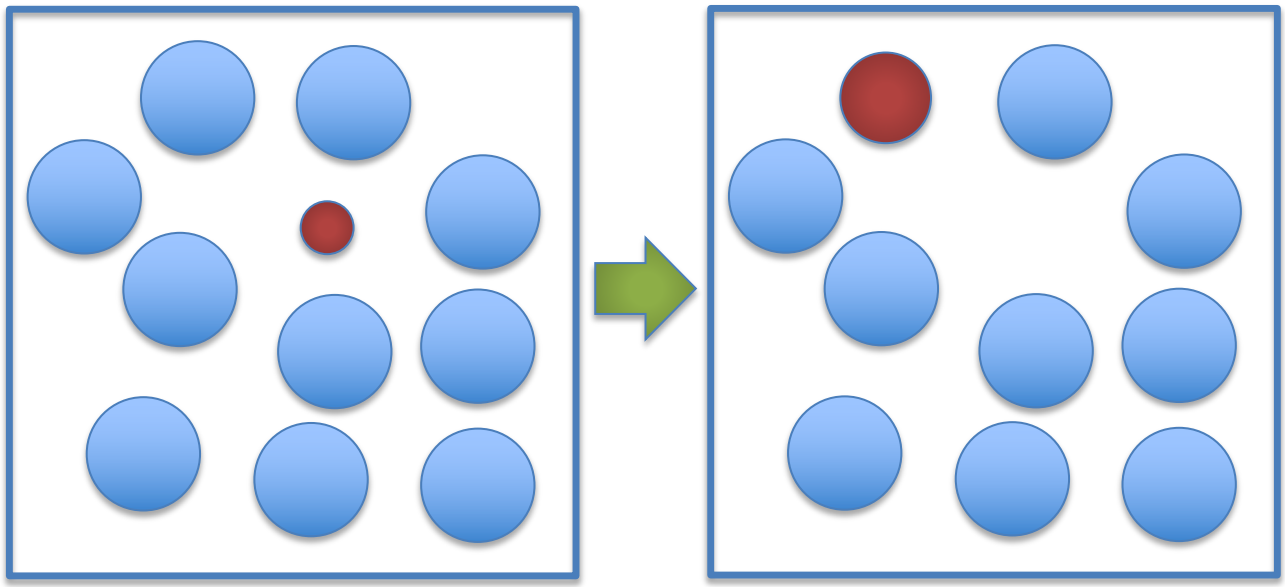
$$P_{\text{acc}}(N \to N+1) = \min\left(1, \frac{f\beta V}{N+1}\exp[\eta(\lambda(n)) - \eta(\lambda(o))] \exp[-\beta(U_{\text{inter}}(n) - U_{\text{inter}}(o))]\right)$$



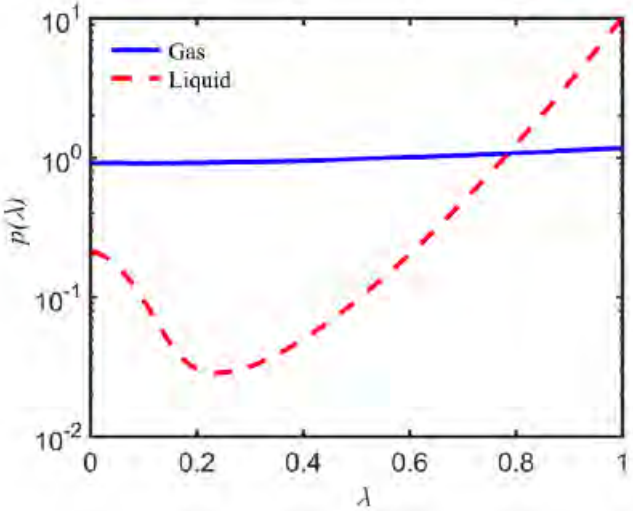
Changes in lambda

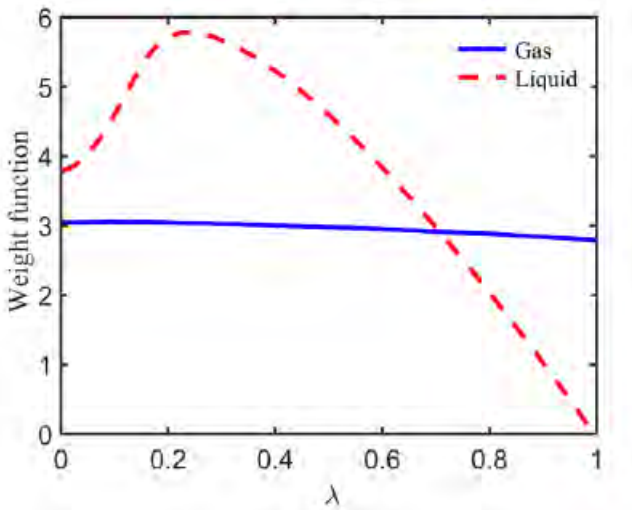
λ(new) smaller than 0: delete fractional molecule, pick random new whole molecule and turn it into a fractional one, and insert new fractional with lambda equals 1+λ(new)

$$P_{\mathrm{acc}}(N \to N-1) = \min\left(1, \frac{N}{f\beta V}\exp[\eta(\lambda(n)) - \eta(\lambda(o))]\right.$$
$$\left.\exp[-\beta(U_{\mathrm{inter}}(n) - U_{\mathrm{inter}}(o))]\right)$$

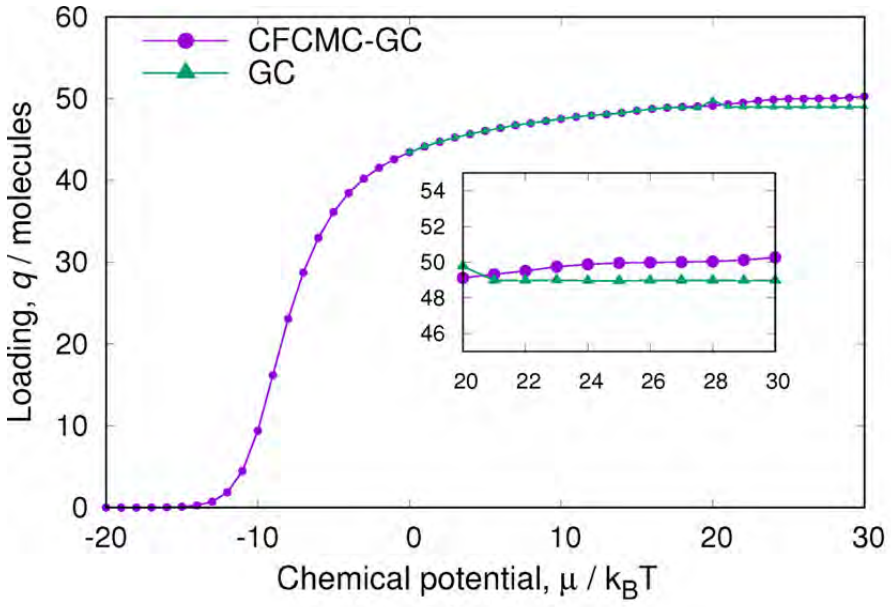


Add weight function W(λ) (or η(λ)) for flat sampling of λ

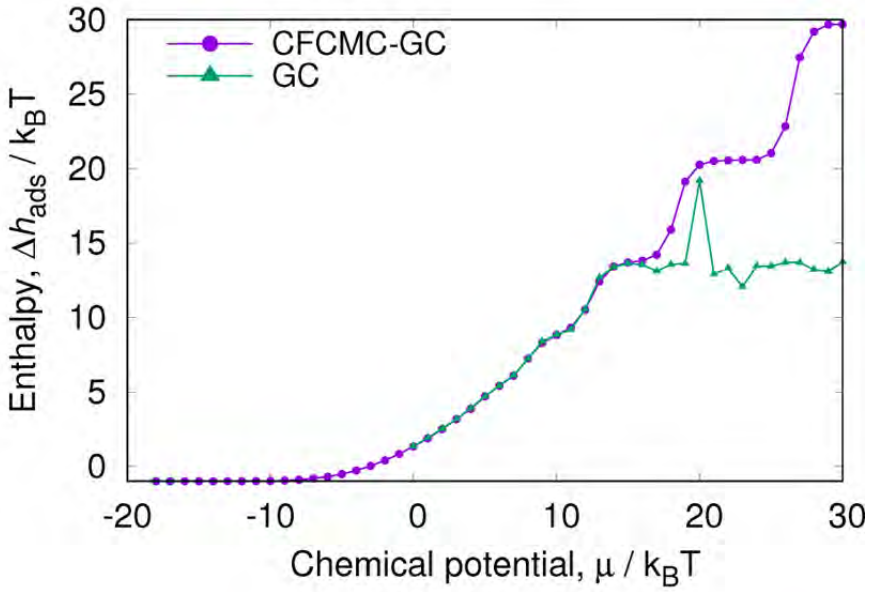




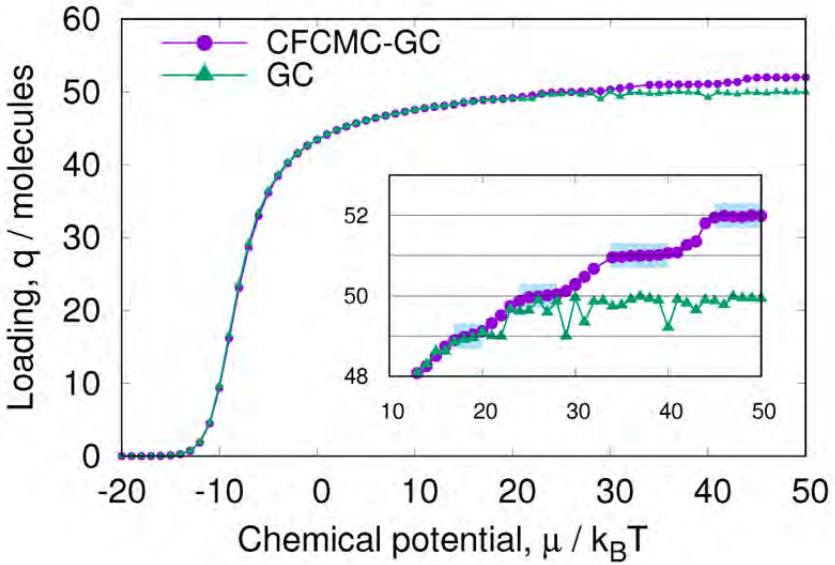
- Avoid getting stuck in λ-space
- Add weight fractor exp[W(λ)] to the partition function
- Choose W(λ) such that the observed $p_{obs}(\lambda)$ is flat (e.g. using the Wang-Landau method, Phys. Rev. Lett., 2001, 86, 2050-2053)
- Obtain Boltzmann averages by multiplying with exp[-W(λ)] for each sampled configuration



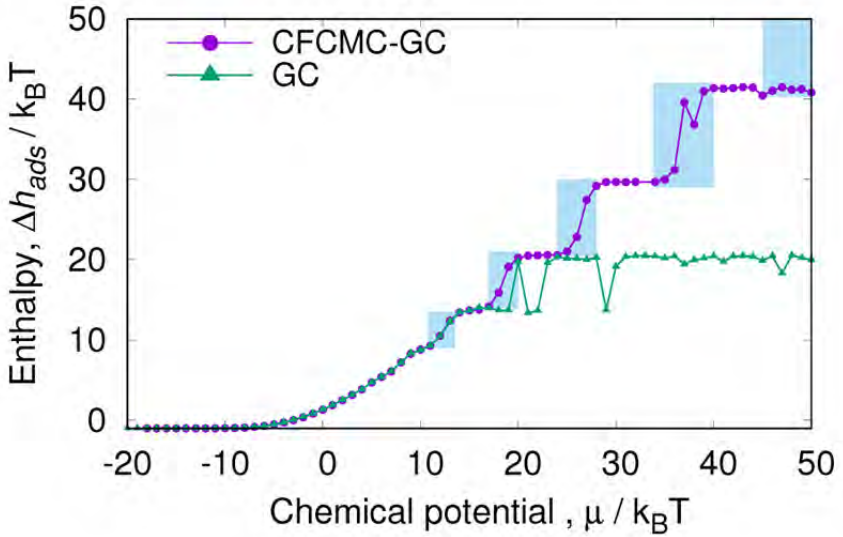
(a)



(b)



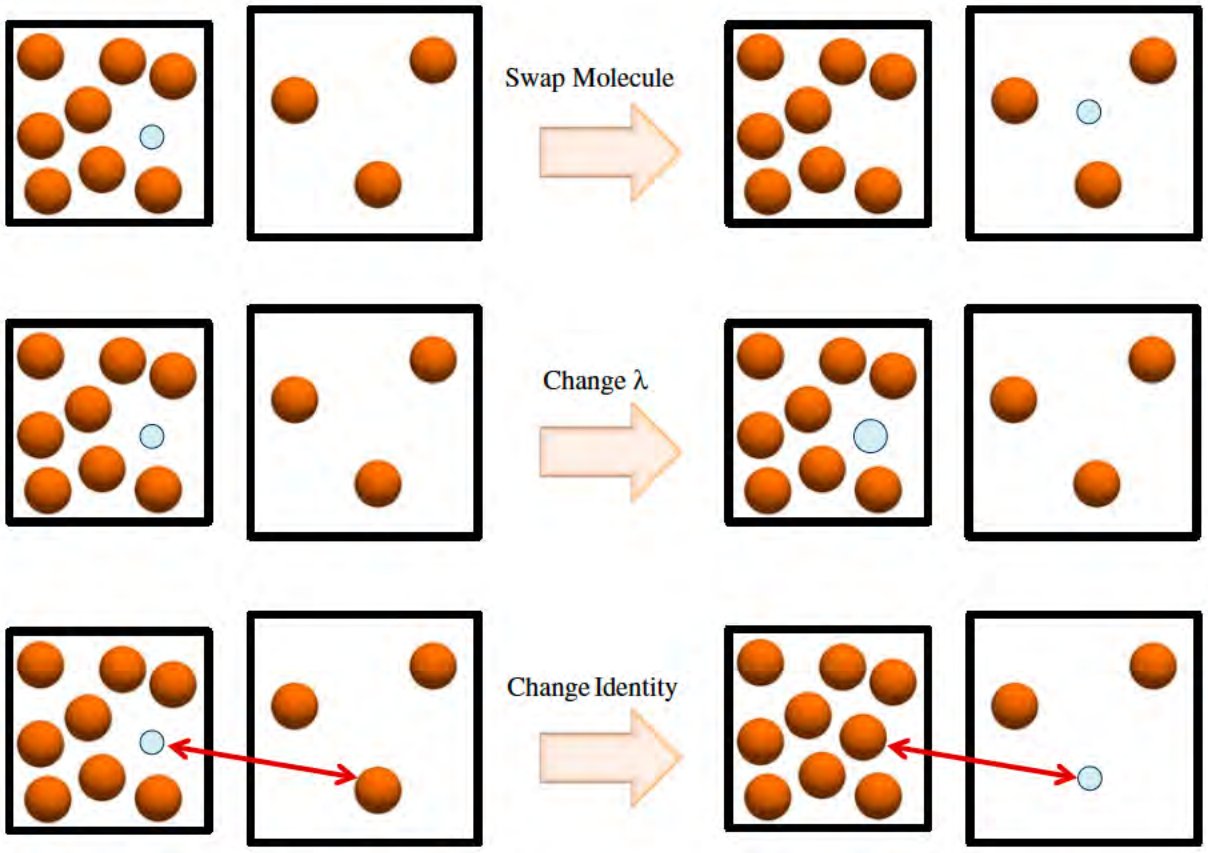
(c)



(d)

TU Delft

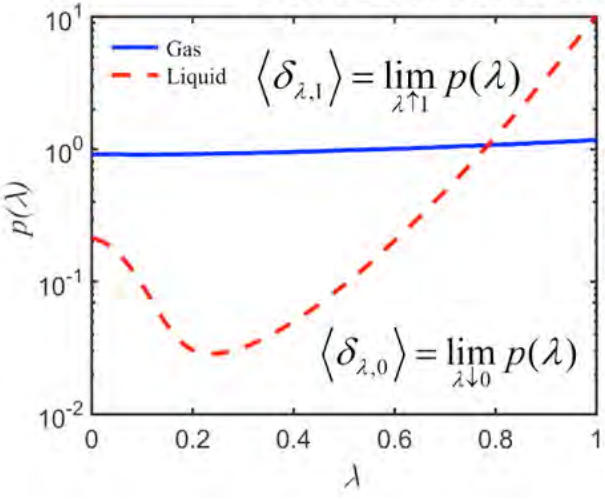
Journal of Chemical Theory and Computation, 2017, 13, 3326-3339.

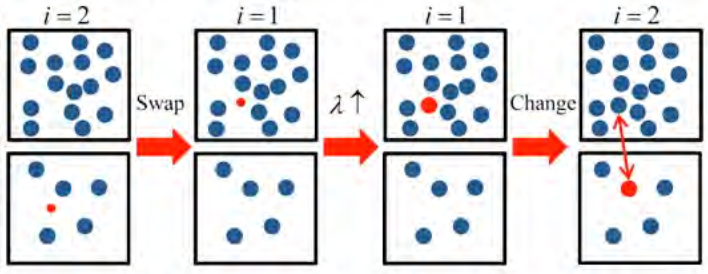


New CFCMC formulation for the Gibbs Ensemble

Journal of Chemical Theory and Computation, 2016, 12, 1481-1490.

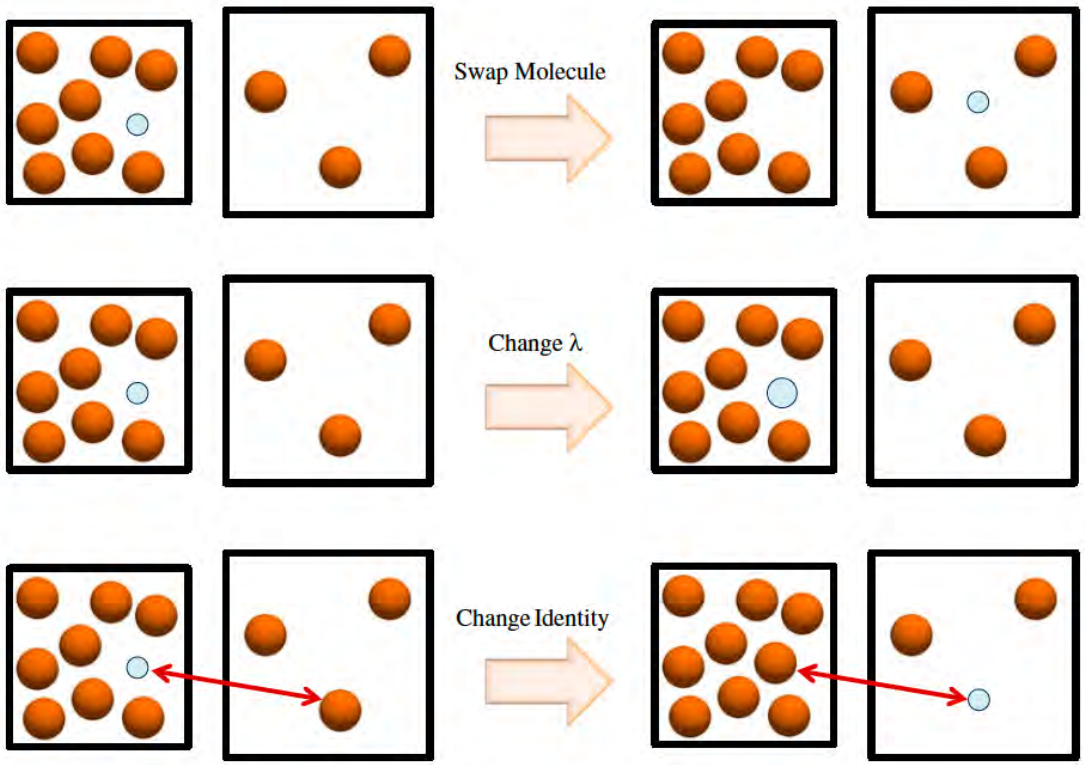
Efficient Molecule Exchange + Free Energy Calculation





$$\mu_1 = -k_B T \ln \frac{\left\langle \frac{\delta_{\lambda,1}\delta_{i,1}}{N_1+1} \right\rangle}{\left\langle \frac{\delta_{\lambda,0}\delta_{i,1}}{V_1/\Lambda^3} \right\rangle}$$

$$Q_{\mathrm{CFCMC}} = \frac{1}{\Lambda^{3(N_T+1)}\,(N_T)!} \sum_{i=1}^{2} \sum_{N_1=0}^{N_T} \int_0^1 \mathrm{d}\lambda$$
$$\times \int_0^{V_T} \mathrm{d}V_1\, {V_1}^{N_1+\delta_{i,1}} (V_T - V_1)^{N_T-N_1+\delta_{i,2}} \frac{(N_T)!}{(N_1)!\,(N_T-N_1)!}$$
$$\times \int \mathrm{d}s^{N_1} \exp\left[-\beta U_{\mathrm{int},1}\left(s^{N_1}, V_1\right)\right]$$
$$\times \int \mathrm{d}s^{N_T-N_1} \exp\left[-\beta U_{\mathrm{int},2}\left(s^{N_T-N_1}, V_T - V_1\right)\right]$$
$$\times \begin{pmatrix} \delta_{i,1} \int \mathrm{d}s^1_{\mathrm{frac}} \exp\left[-\beta U_{\mathrm{frac},1}\left(s^1_{\mathrm{frac}}, s^{N_1}, \lambda, V_1\right)\right] \\ + \delta_{i,2} \int \mathrm{d}s^2_{\mathrm{frac}} \exp\left[-\beta U_{\mathrm{frac},2}\left(s^2_{\mathrm{frac}}, s^{N_T-N_1}, \lambda, V_T - V_1\right)\right] \end{pmatrix}$$



Trial moves:
-volume changes
-translation, rotation etc.
-change lambda
-swap molecule
-change identity

Compute the chemical potential “for free”

$$\mu_{i,\,\mathrm{CFCGE\,MC}} = -k_{\mathrm{B}}T \ln\left\langle \frac{V_i/\Lambda^3}{N_i+1} \right\rangle - k_{\mathrm{B}}T \ln\left\langle \frac{p_i\,(\lambda \uparrow 1)}{p_i\,(\lambda \downarrow 0)} \right\rangle$$

Journal of Chemical Theory and Computation, 2016, 12, 1481-1490.

Molecular Simulation, 2018, 44, 405-414

Molecular Simulation, 2017, 43, 189-195.

Add weight function

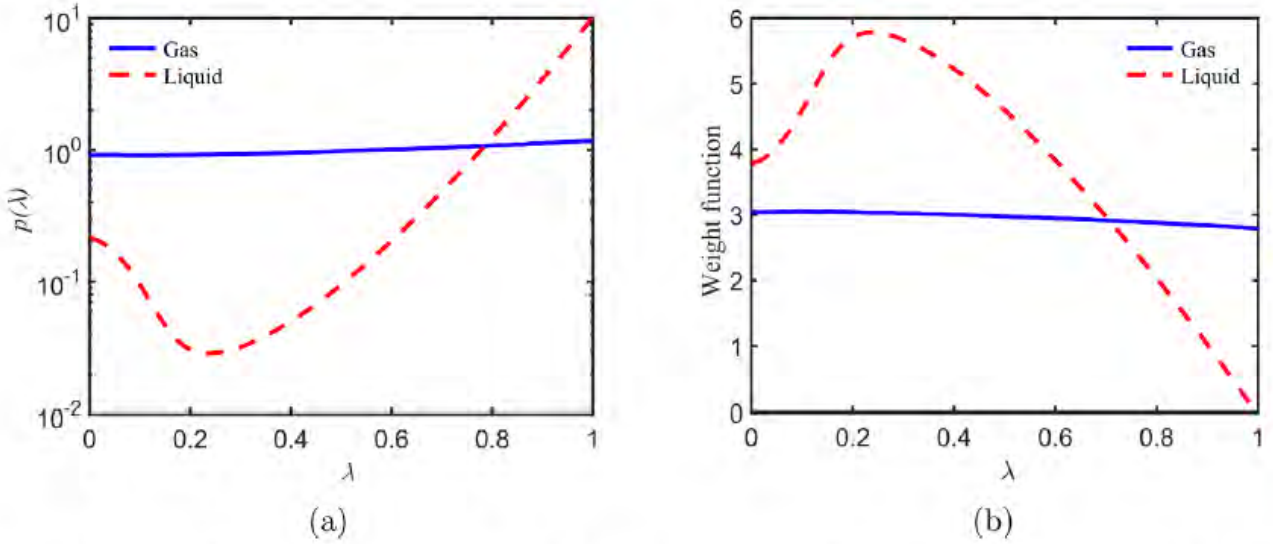
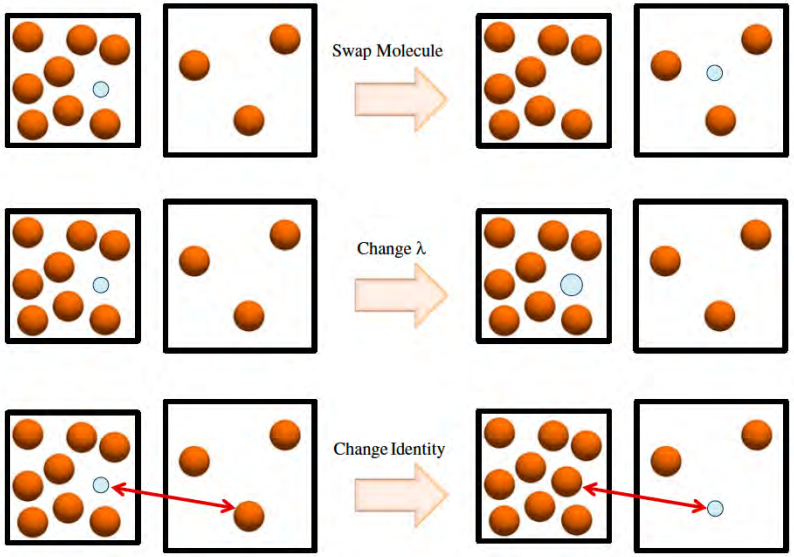
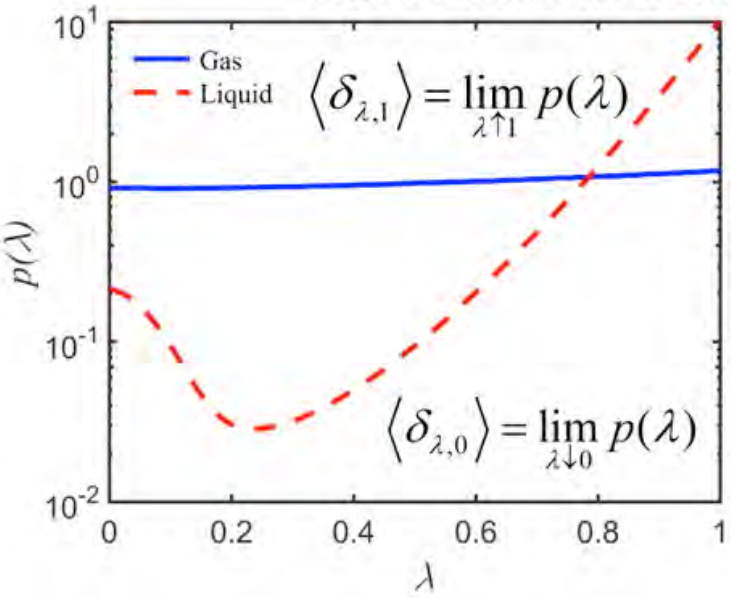
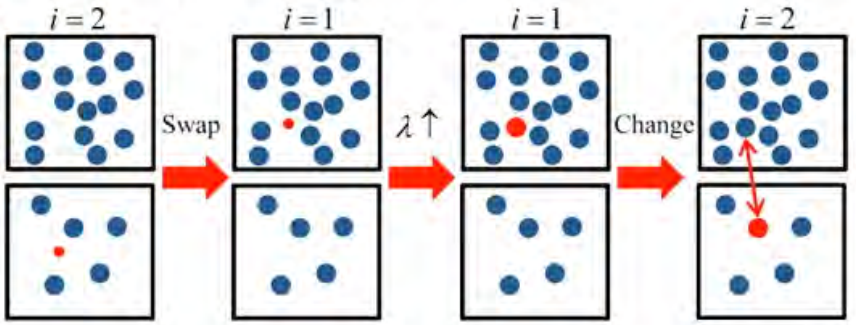


Figure 5. (a) Probability distribution of λ for the gas and the liquid phases as used in CFCMC GE of LJ particles at $T = 0.8$; (b) weight functions to flatten the corresponding probability distributions of λ (as in panel a) and to ensure that the fractional molecule is equally likely to be in both simulation boxes.



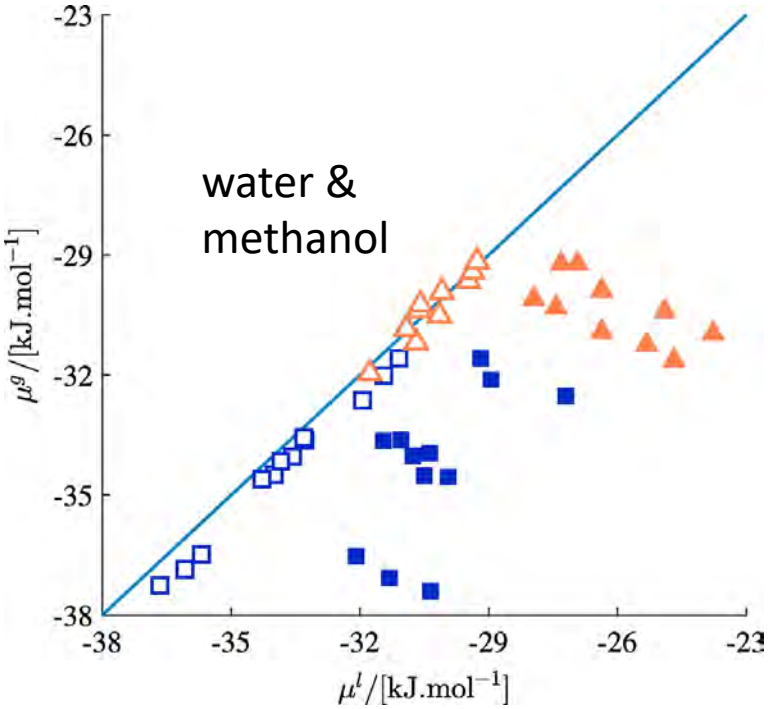
Efficient Molecule Exchange + Free Energy Calculation

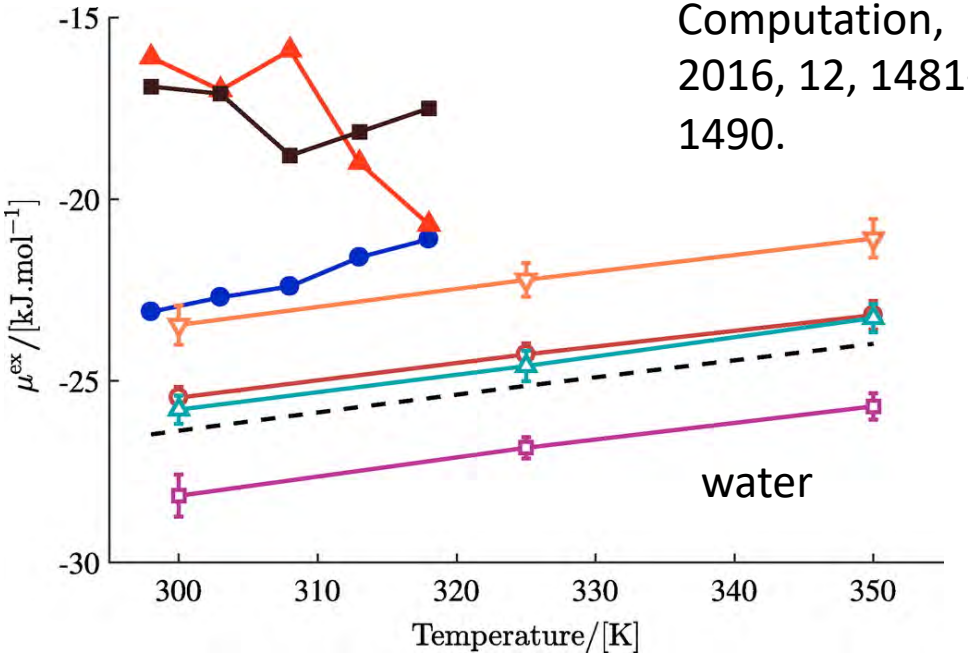




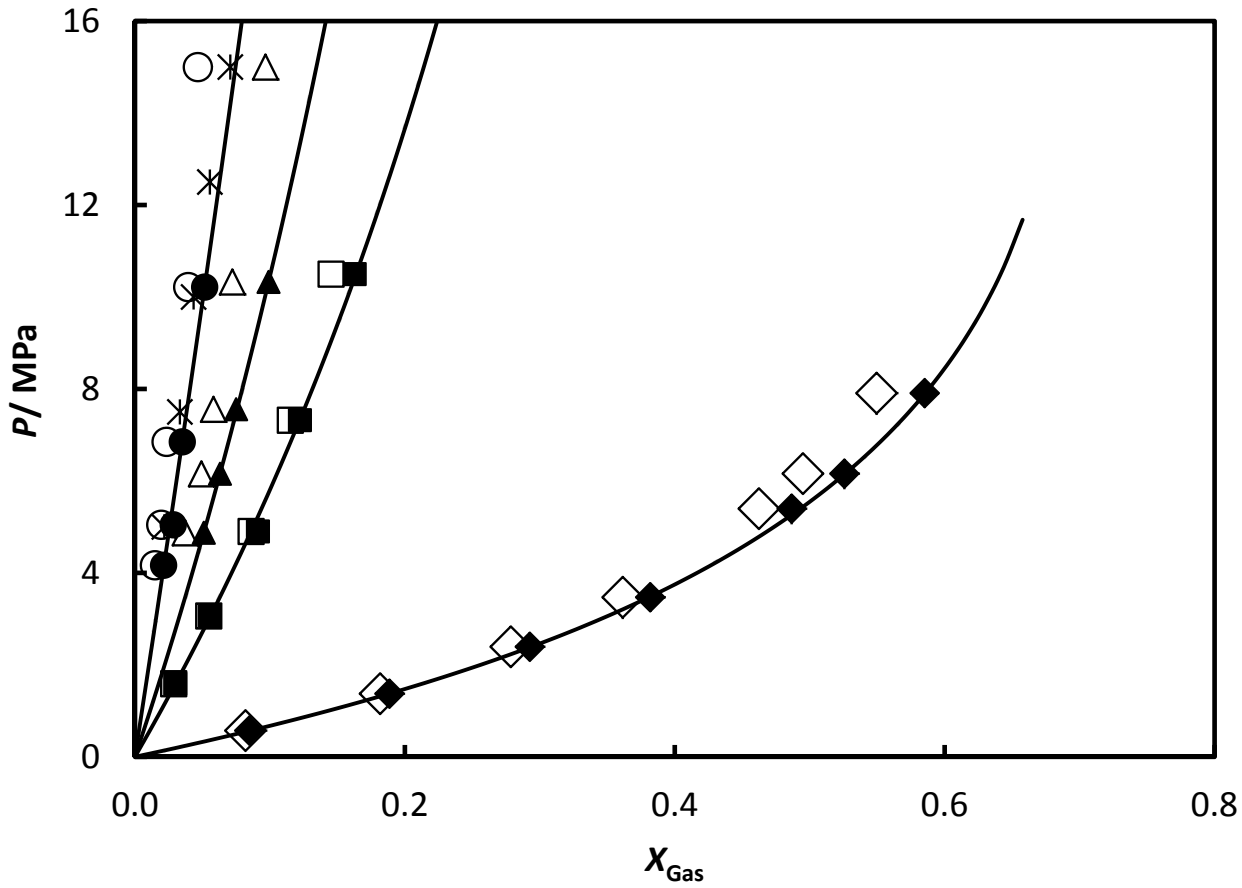
$$\mu_1 = -k_B T \ln \frac{\left\langle \frac{\delta_{\lambda,1}\delta_{i,1}}{N_1 + 1} \right\rangle}{\left\langle \frac{\delta_{\lambda,0}\delta_{i,1}}{V_1/\Lambda^3} \right\rangle}$$

Journal of Chemical Theory and Computation, 2016, 12, 1481-1490.





Solubility of precombustion gasses in [bmim][Tf2N]



Solubility of CO_2, CH_4, CO, H_2 and N_2 in [bmim][Tf_2N] from MC simulations (open symbols) and experiments (filled symbols) at 333.15 K. CO_2 experiments (filled diamonds) and MC data (open diamonds); CH_4 experiments (filled squares) and MC data (open squares); CO experiments (filled triangles) and MC data (open triangles); H_2 experiments (filled circles) and MC data (open circles), and MC data of N_2 (stars). Lines: PR-EOS modeling

Henry constants of CO_2, CH_4, CO, H_2, N_2, and H_2S in Selexol and [bmim][Tf_2N] at 333.15 K

solute	$H^{\text{exp.}}_{\text{Selexol}}$/MPa	$H^{\text{exp.}}_{\text{IL}}$/MPa	$H^{\text{sim.}}_{\text{IL}}$/MPa	difference/%
CO_2	6.81[a]	6.56	7.10	8.2
CH_4	40.13[a]	52.4	53.7	2.5
CO	-	95	125.9	33.0
H_2	193[b]	199	271.7	36.3
N_2	151[b]	-	225.7	-
H_2S (3S)	1.01[c]	2.17	1.15	47.0
H_2S (4S)	1.01	2.17	1.16	46.5
H_2S (5S)	1.01	2.17	1.17	46.1

[a] Taken from Rayer *et al.* [b] Calculated from Gainar *et al.* [c] Taken from Xu *et al.*

The End

Questions ??