Configurational-Bias Monte Carlo

Thijs J.H. Vlugt

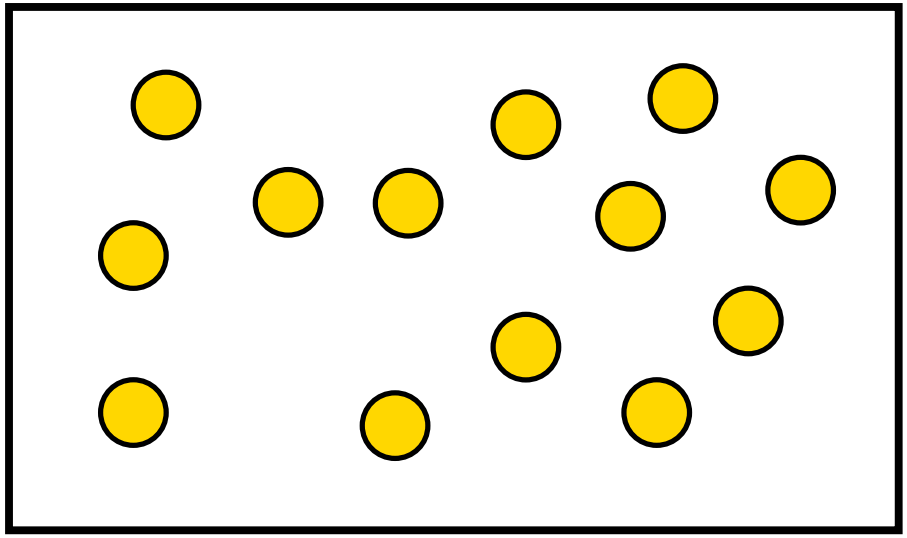
Professor and Chair Engineering Thermodynamics
Delft University of Technology
Delft, The Netherlands

t.j.h.vlugt@tudelft.nl

January 10, 2011

Random Sampling versus Metropolis Sampling (1)

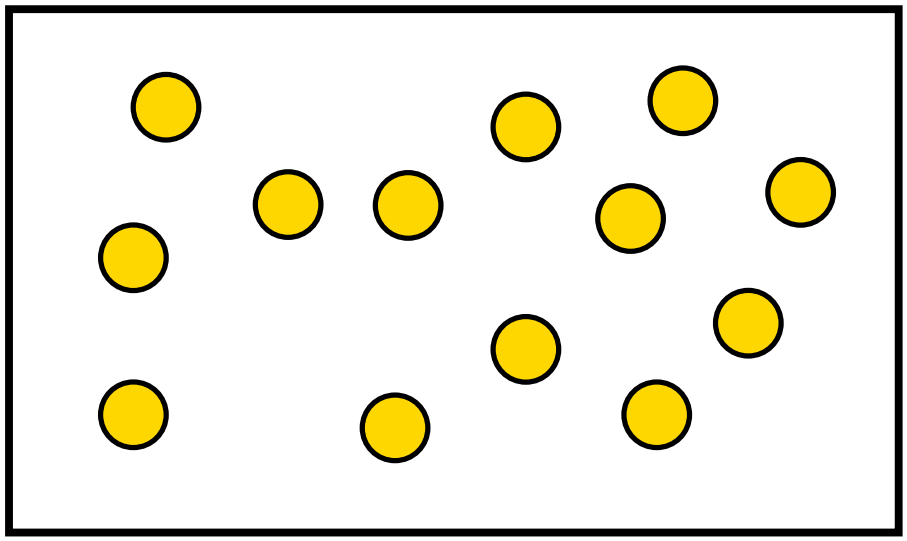
N interacting particles in volume V at temperature T



- 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:

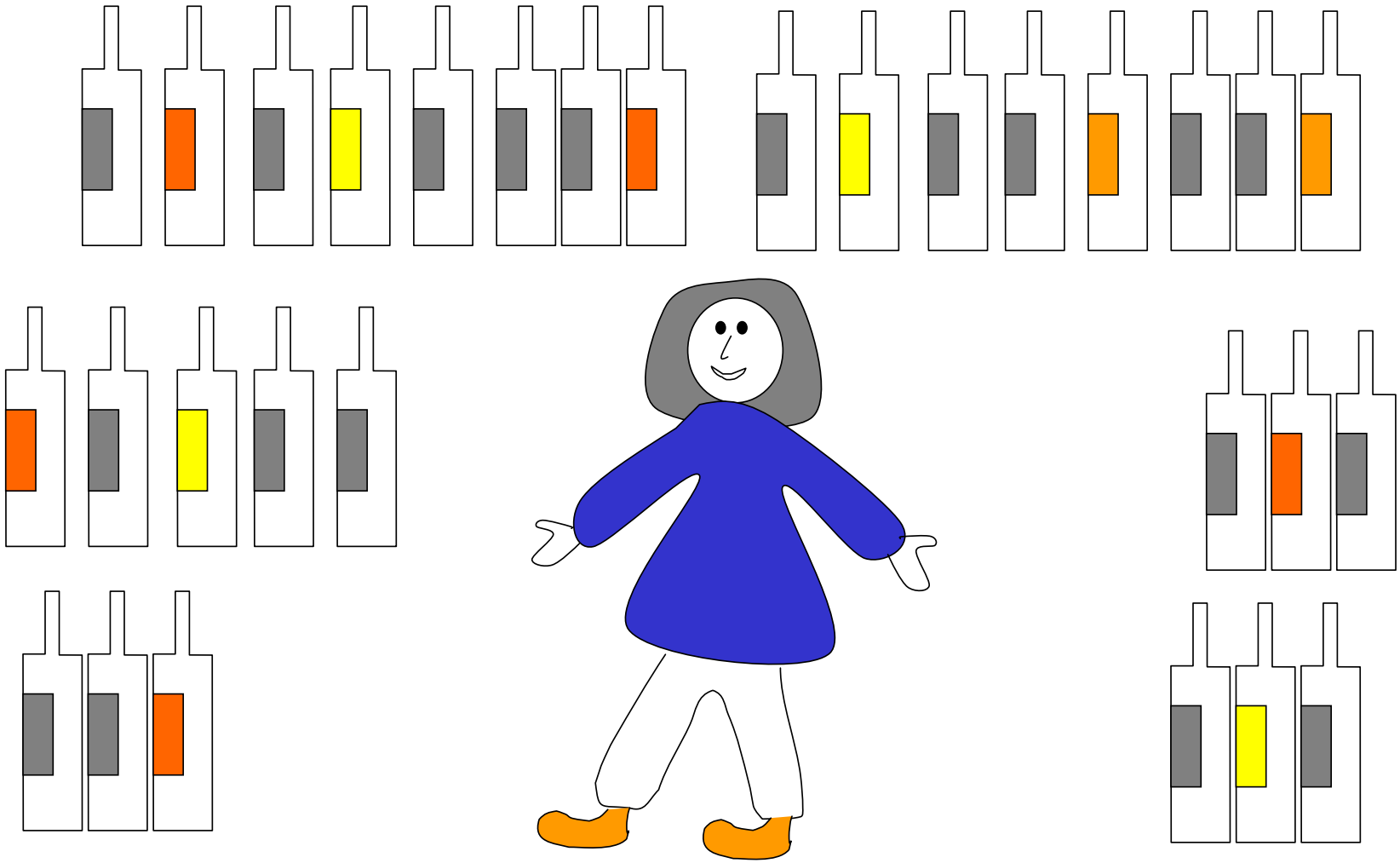
$$\langle A \rangle = \lim_{n \to \infty} \frac{\sum_{i=1}^{n} A(\mathbf{r}_i^N)}{n}$$

Simulation Technique (1)

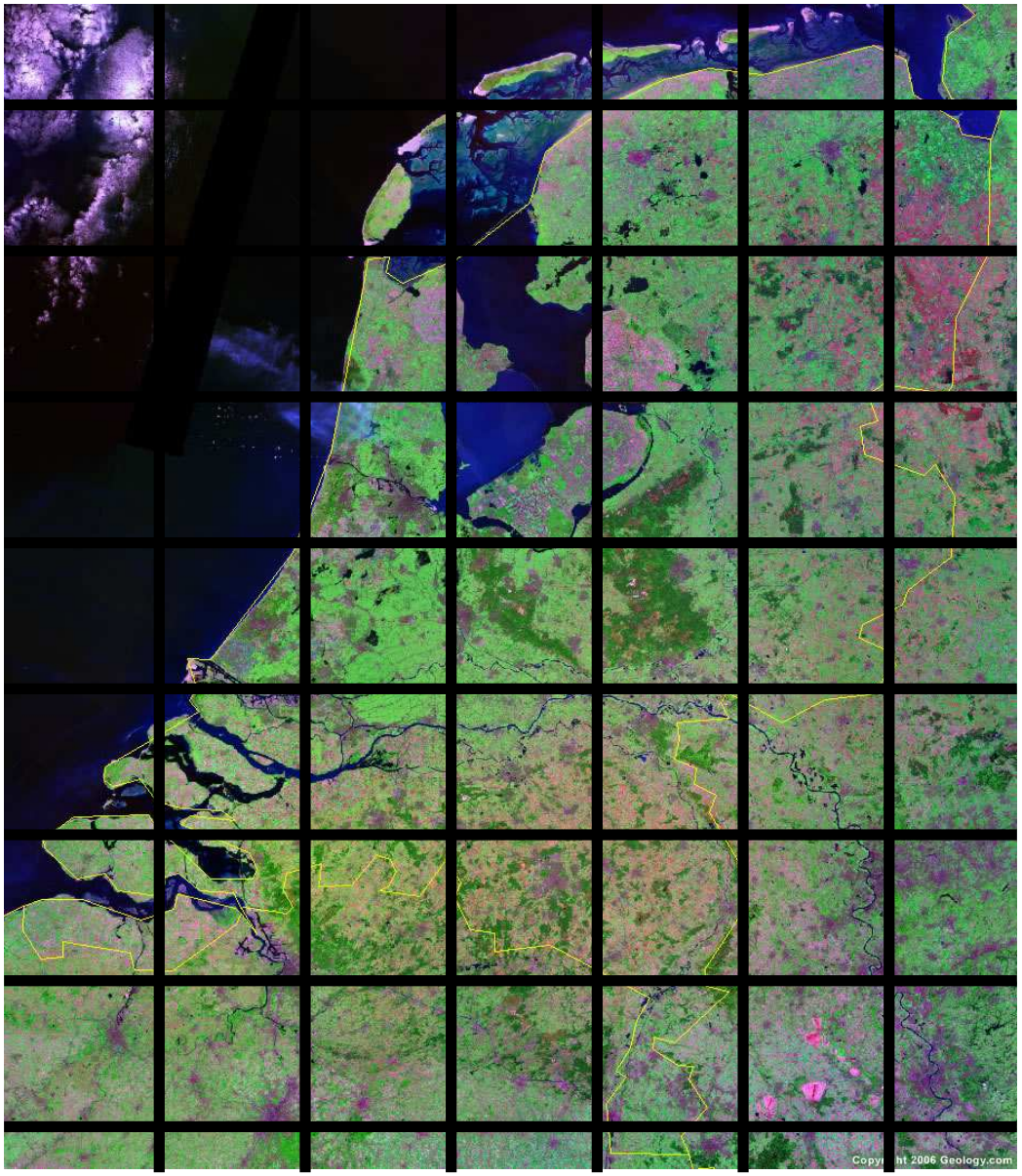
What is the ratio of red wine/white wine in the Netherlands?

Simulation Technique (2)

Bottoms up



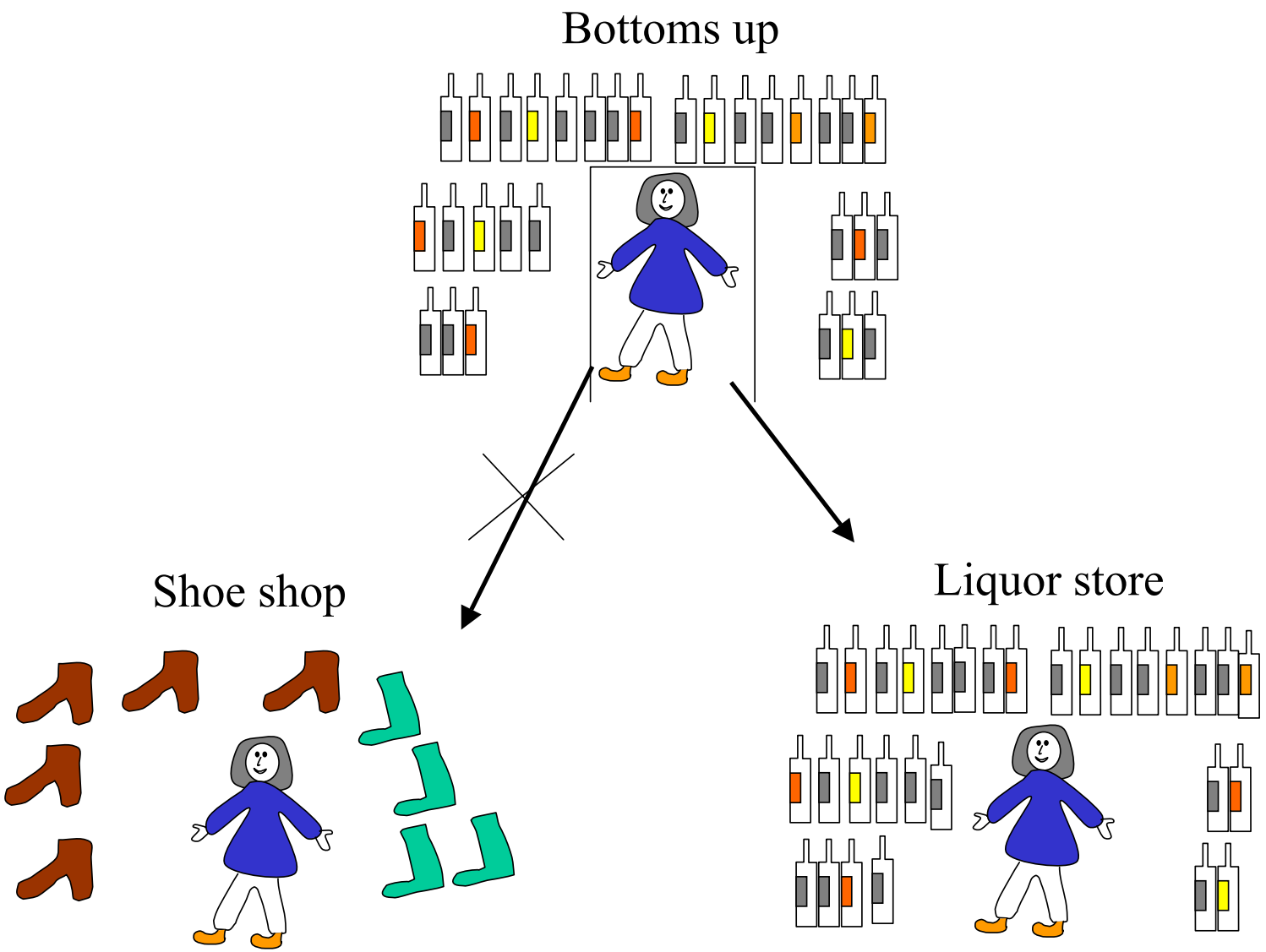
Simulation Technique (3)



Simulation Technique (4)

Simulation Technique (5)

Simulation Technique (6)



Metropolis Monte Carlo (1)

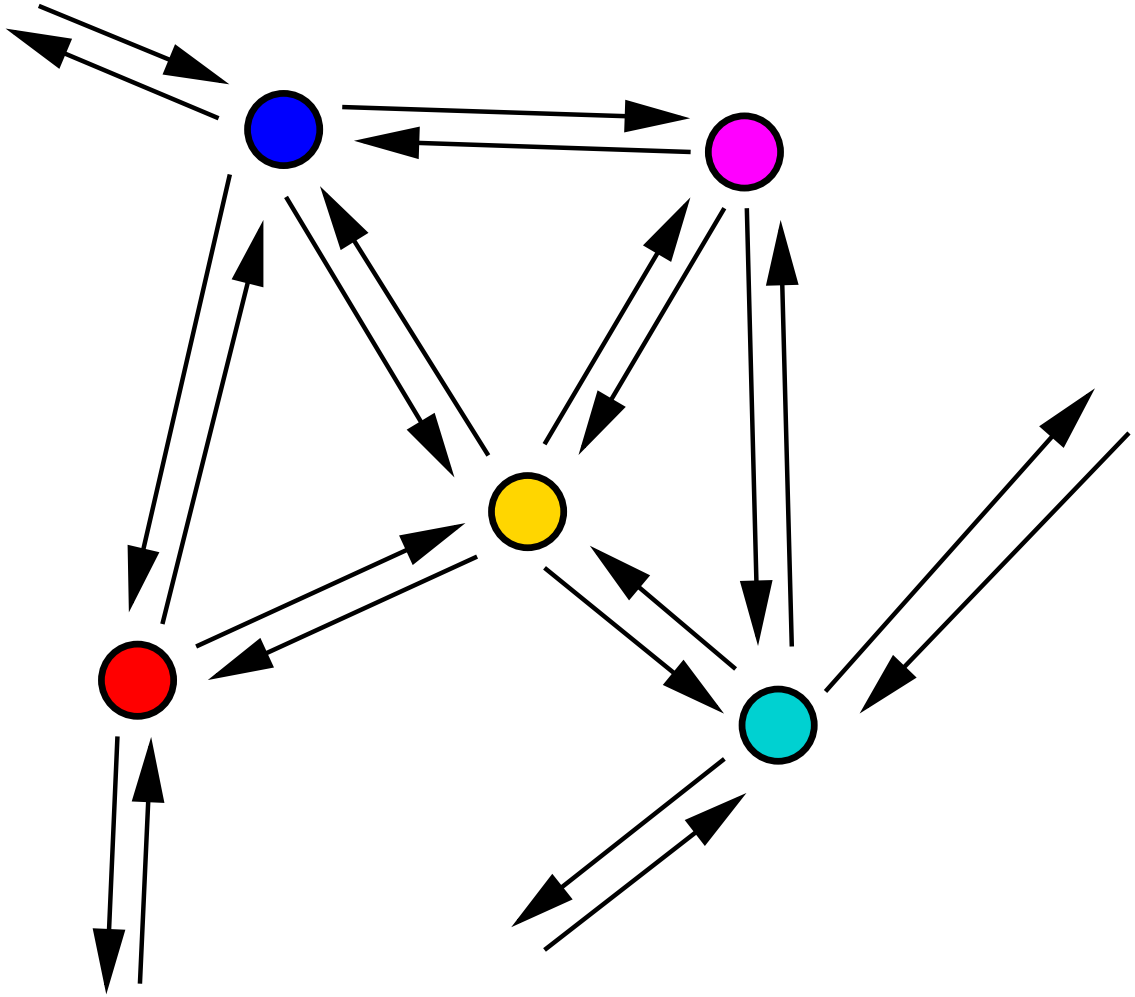
How to generate configurations $\mathbf{r}_i$ with a probability proportional to $\mathcal{N}(\mathbf{r}_i) = \exp[-\beta U(\mathbf{r}_i)]$???

Nick Metropolis

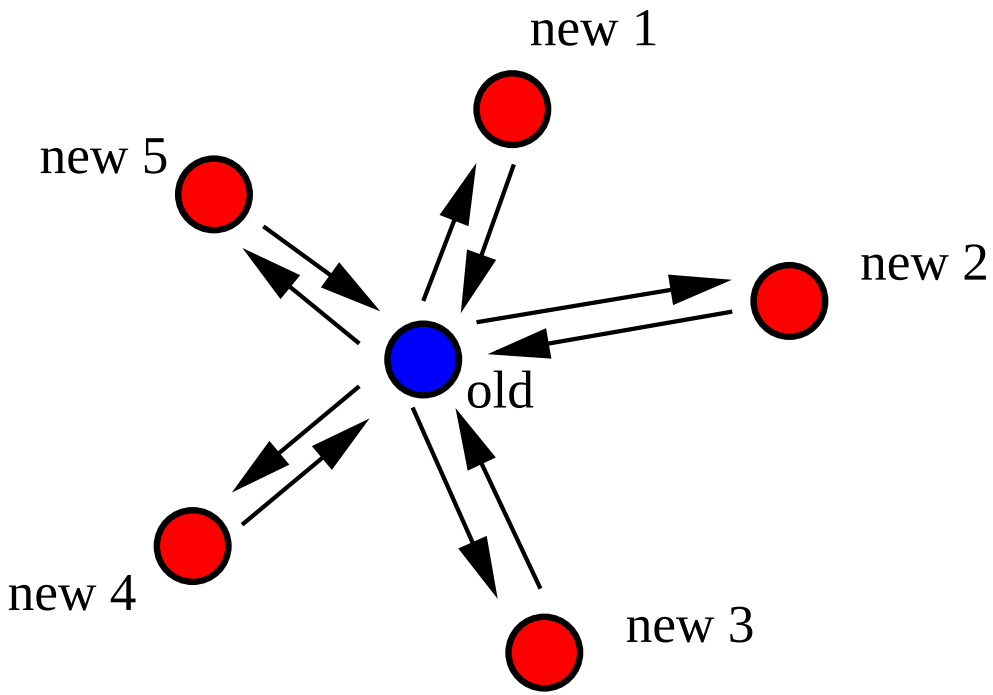
N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth. A.H. Teller and E. Teller, "Equation of State Calculations by Fast Computing Machines," J. Chem. Phys., 1953, 21, 1087-1092.

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)

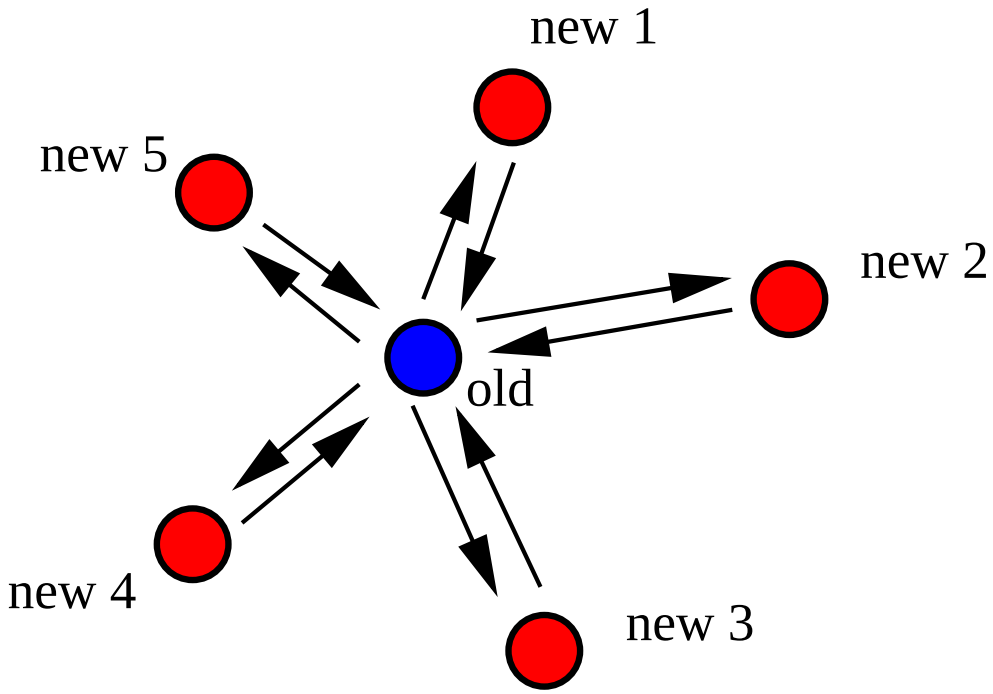
Requirement (balance):

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

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

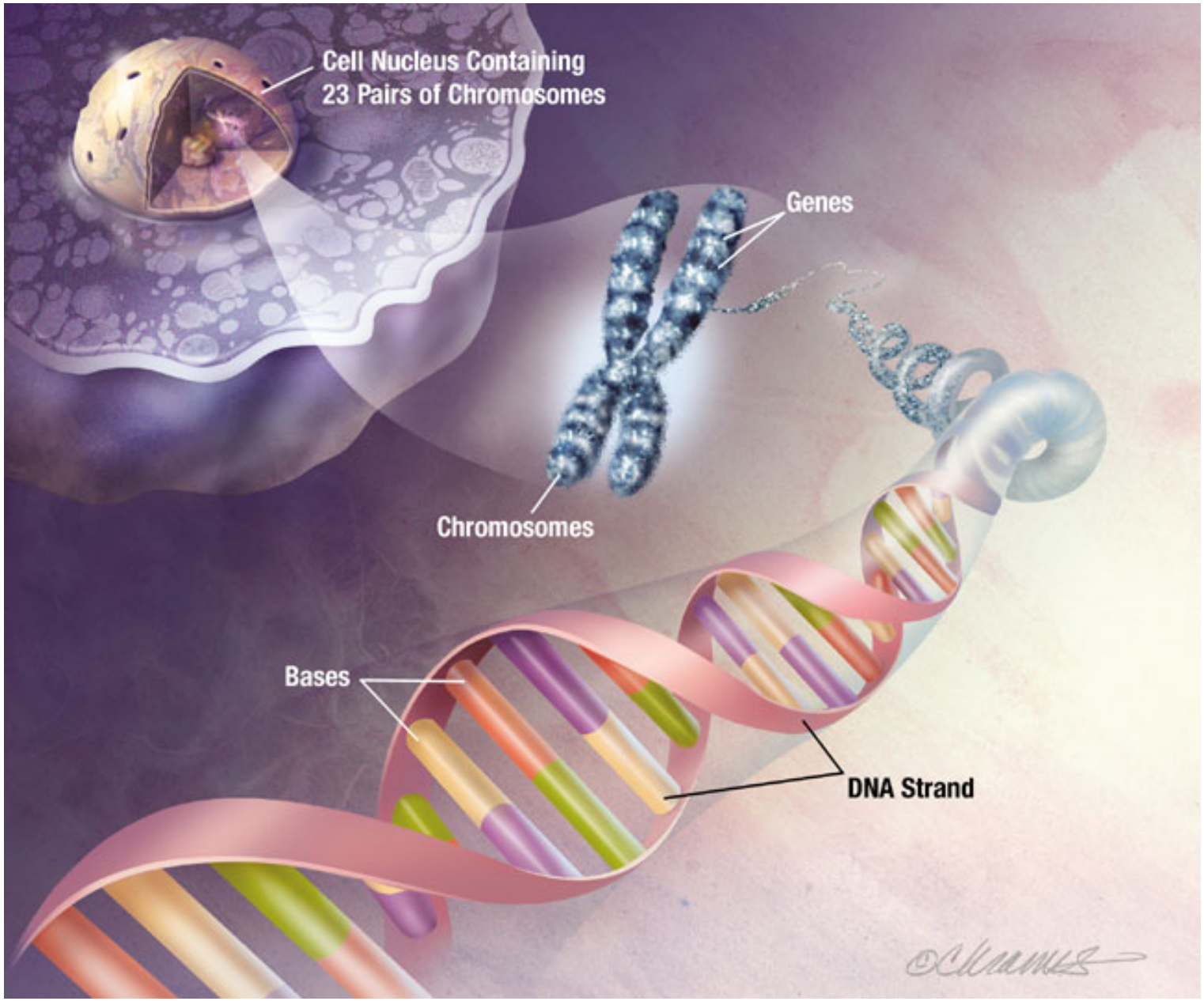
$$\Delta r = r(\text{new}) - r(\text{old}) = A \times F + \delta r$$

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

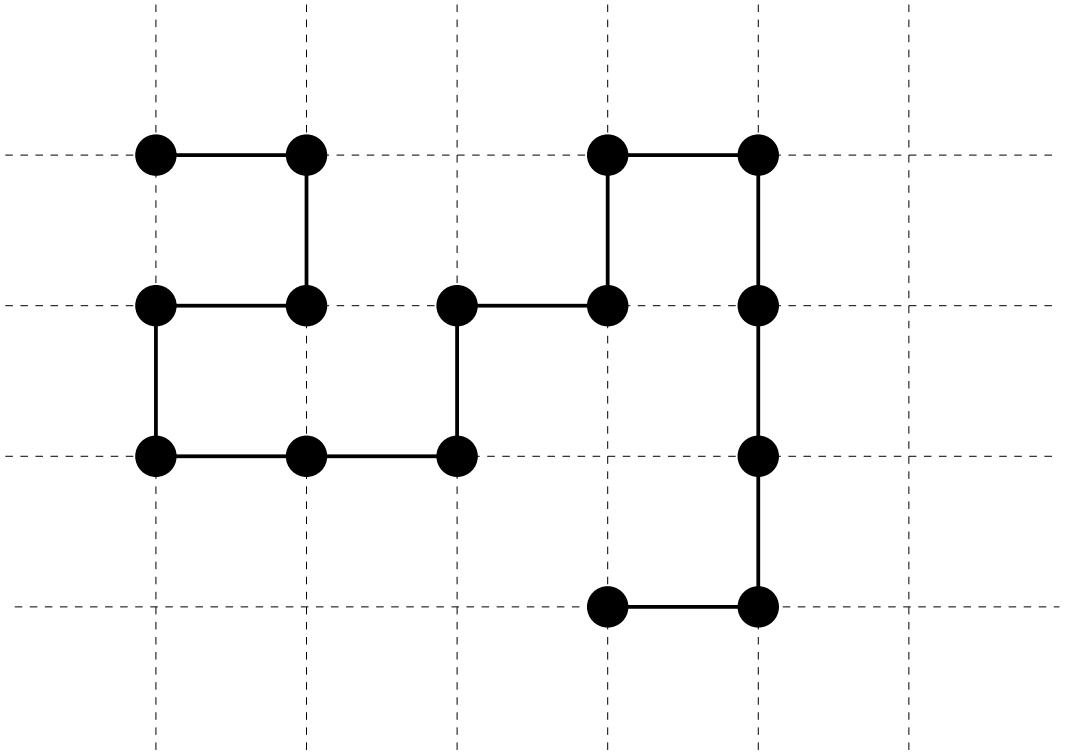
$$P(r_{\text{new}}) \sim \exp\left[-\frac{(r_{\text{new}} - (r_{\text{old}} + A \times F(o)))^2}{4A}\right]$$

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

Chain Molecules



Self-Avoiding Walk on a Cubic Lattice



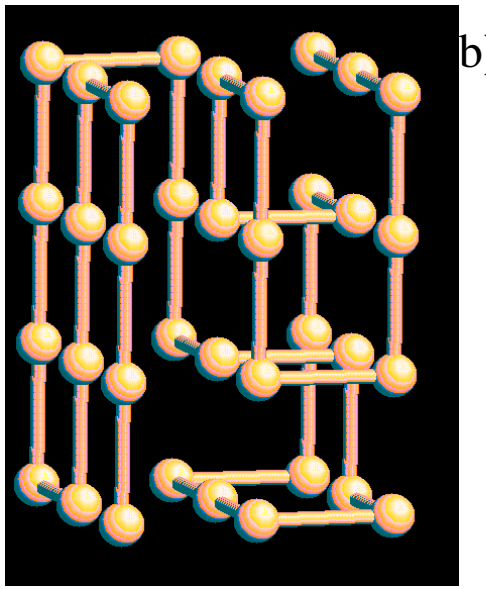
- 3D lattice; 6 lattice directions
- only 1 monomer per lattice site (otherwise $U = \infty$)
- interactions only when $|r_{ij}| = 1$ and $|i - j| > 1$

Simple Model for Protein Folding

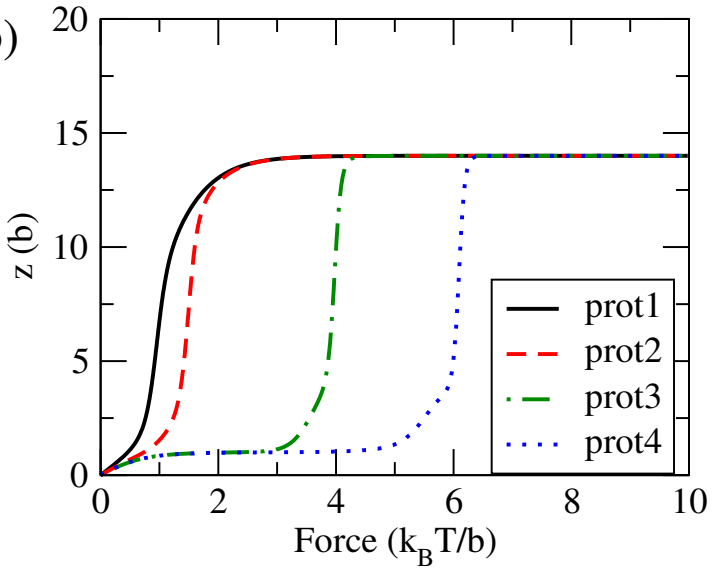
20 by 20 interaction matrix Δ_{ij}

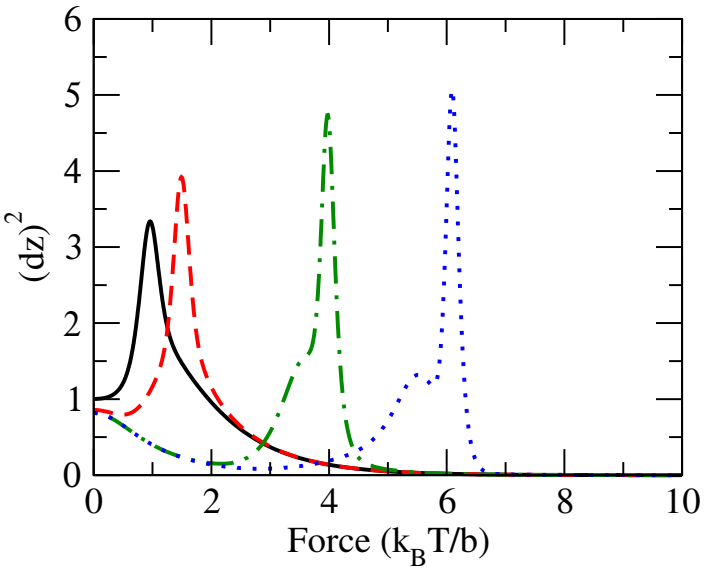
YPDLTKWHAMEAGKIRFSVPDACLNGEGIRQVTLSN

(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}

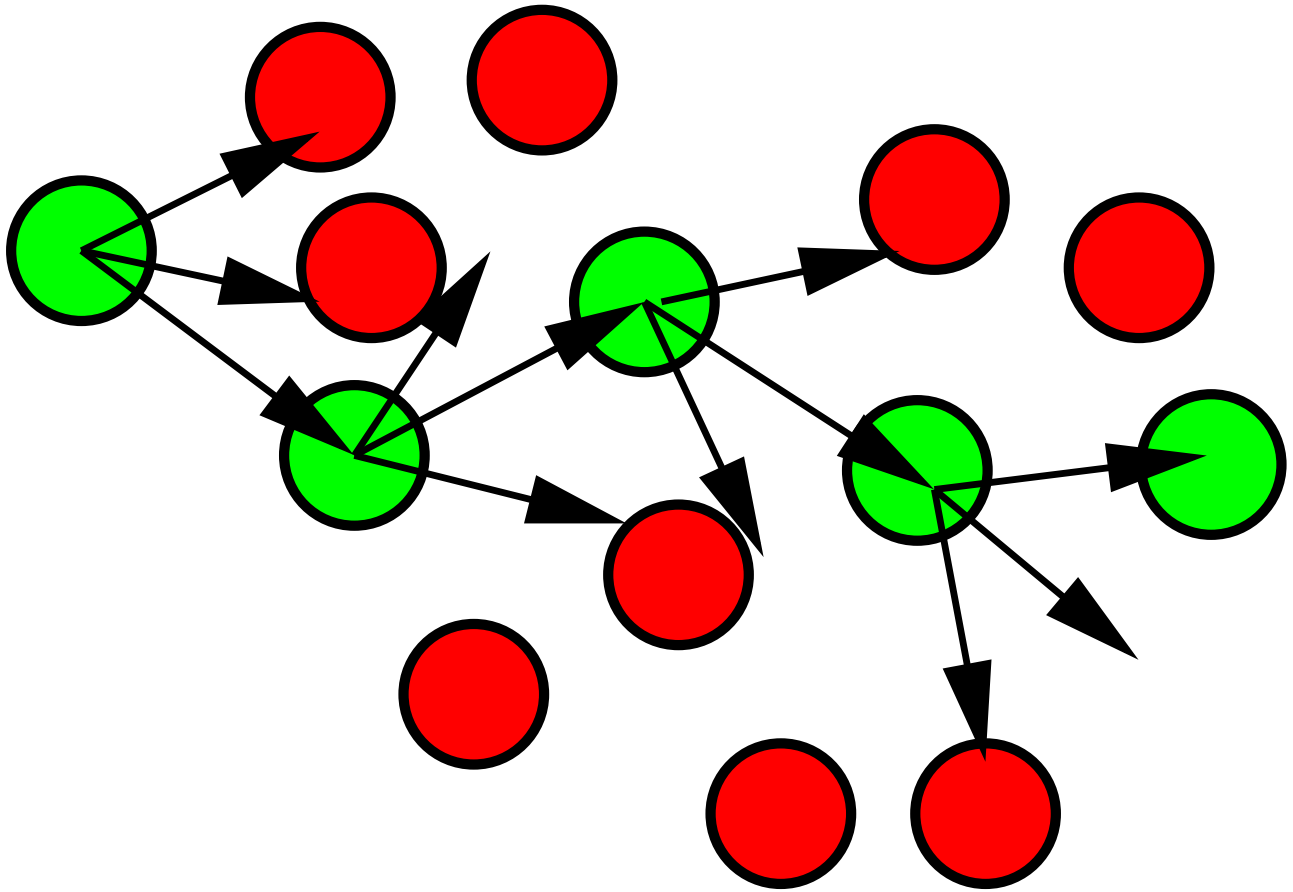
3. Select trial direction $j^\star$ with a probability

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

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:

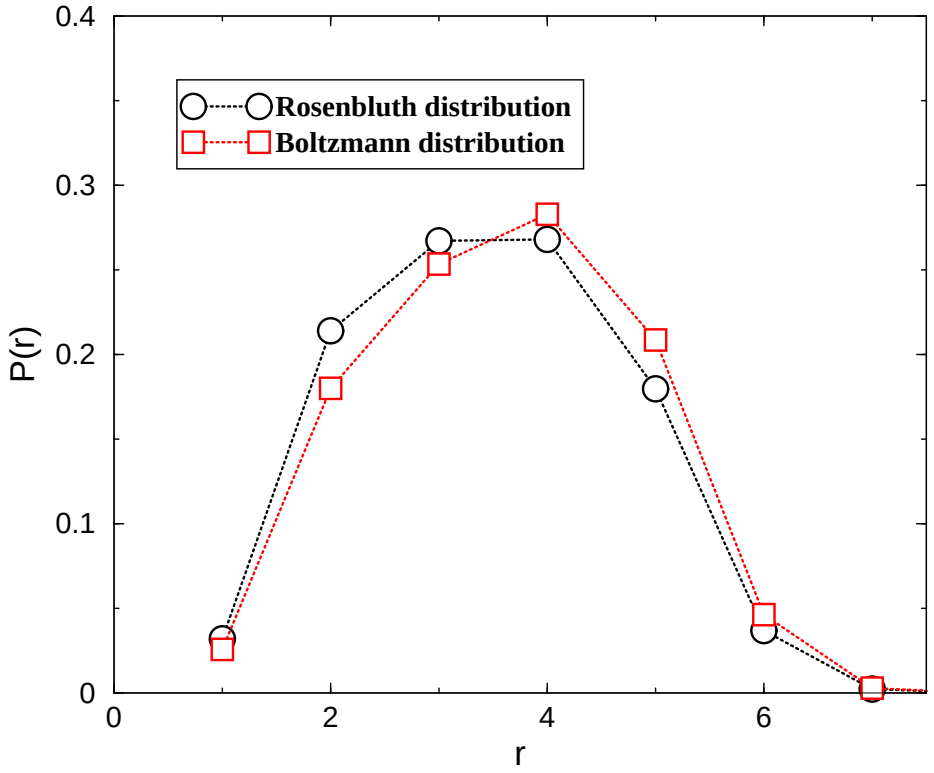
$$
\begin{aligned}
\langle U \rangle_\beta &= \frac{\int d\mathbf{r}^N U(\mathbf{r}^N) \exp[-\beta U(\mathbf{r}^N)]}{\int d\mathbf{r}^N \exp[-\beta U(\mathbf{r}^N)]} \\
&= \frac{\int d\mathbf{r}^N U(\mathbf{r}^N) \exp[-\beta^\star U(\mathbf{r}^N)] \exp[(\beta^\star - \beta) \times U(\mathbf{r}^N)]}{\int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

Useful or not???

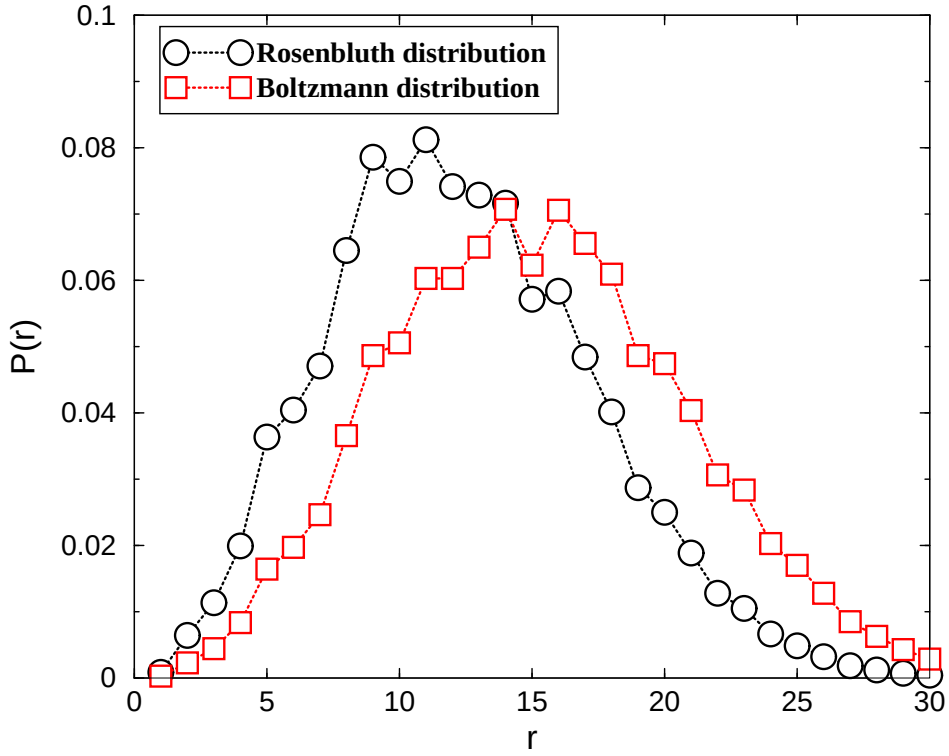
Rosenbluth Distribution Differs from Boltzmann Distribution

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

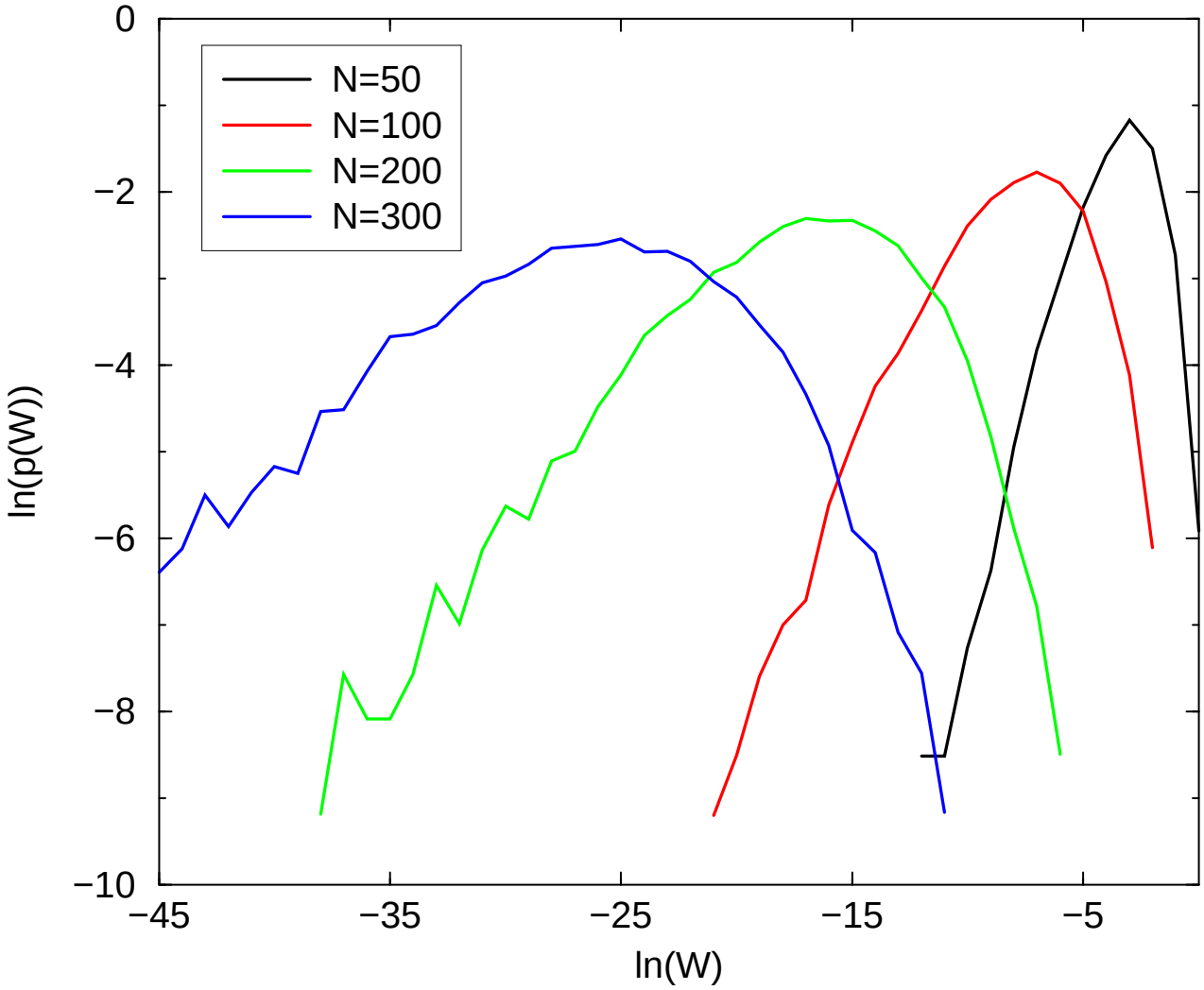
$N = 10$



$N = 100$



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:

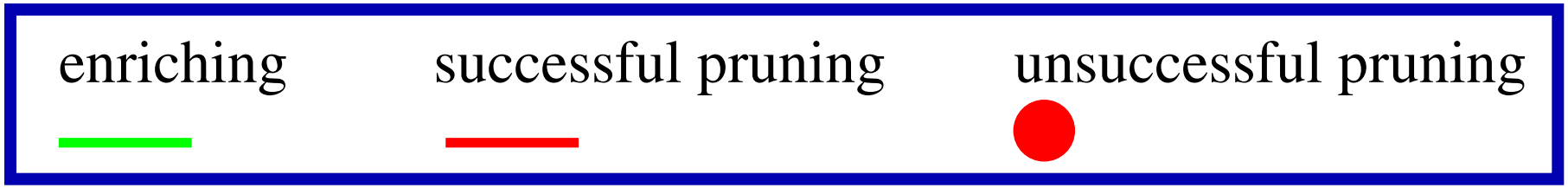
$$W = \sum_{j=1}^{6} \frac{\exp\left[-\beta u_{2j}\right]}{6} \times \prod_{i=3}^{N} \sum_{j=1}^{5} \frac{\exp\left[-\beta u_{ij}\right]}{5} = \prod_{i=3}^{N} \sum_{j=1}^{5} \frac{\exp\left[-\beta u_{ij}\right]}{5}$$

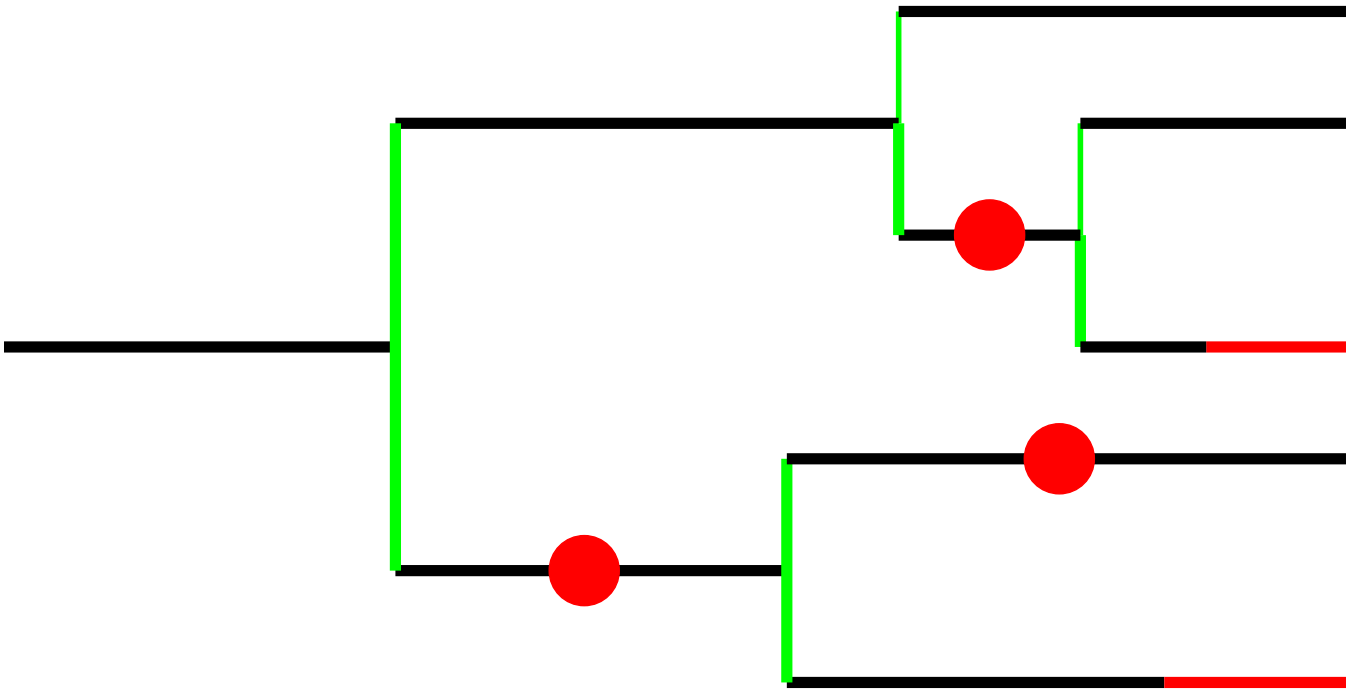
Two additional elements:

- **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.

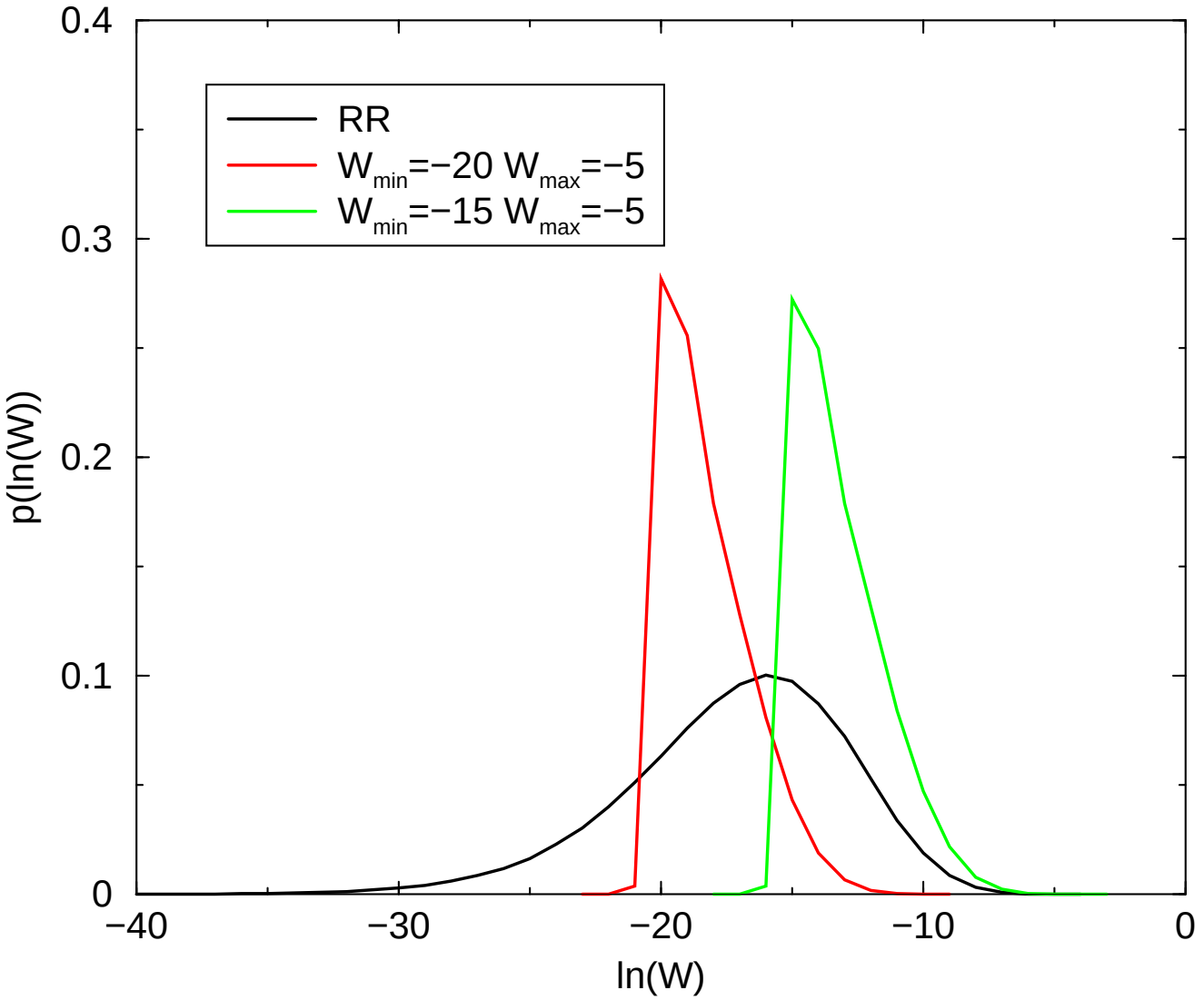
Pruned-Enriched Rosenbluth Method (2)





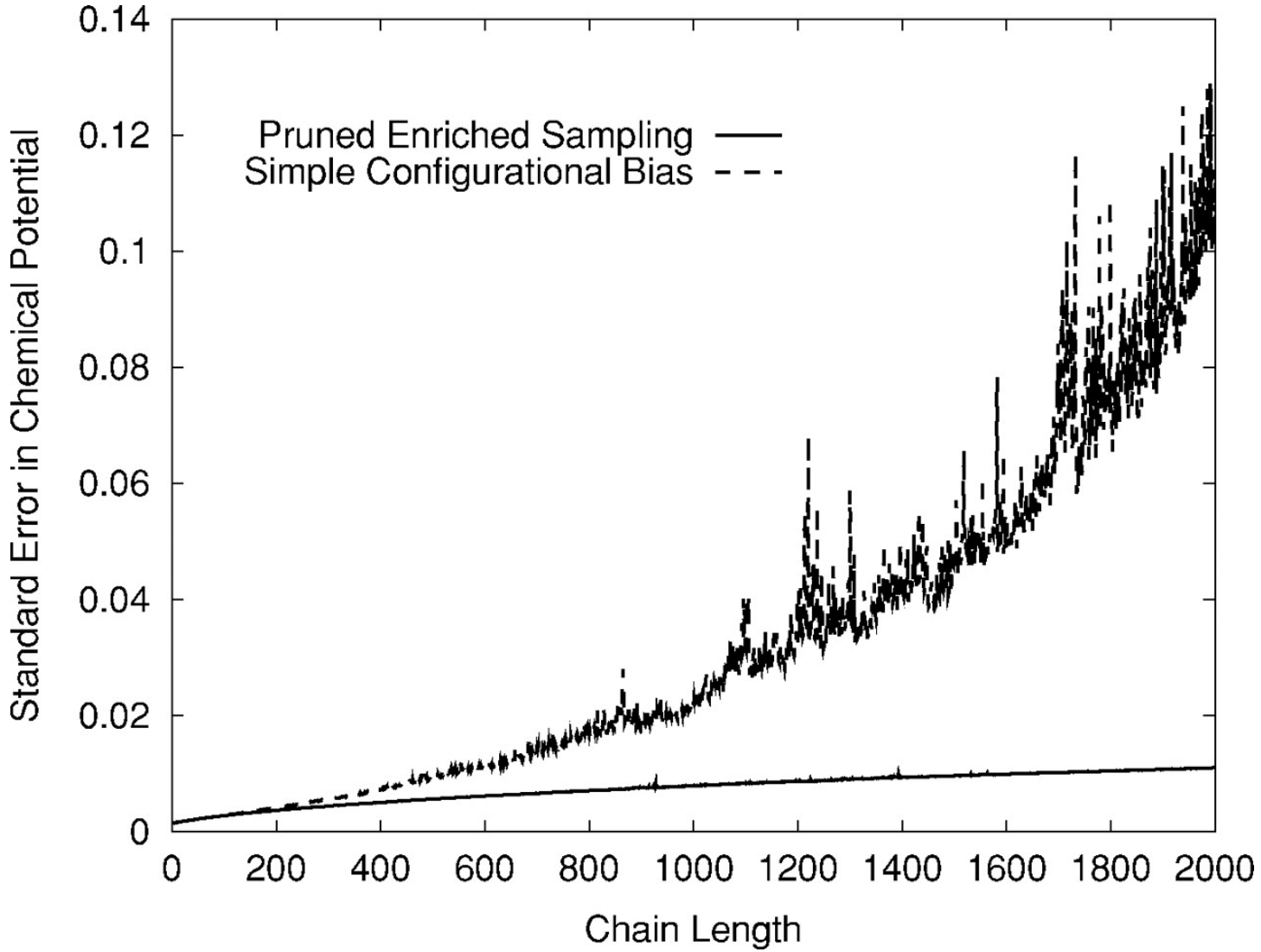
Pruned-Enriched Rosenbluth Method (3)

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



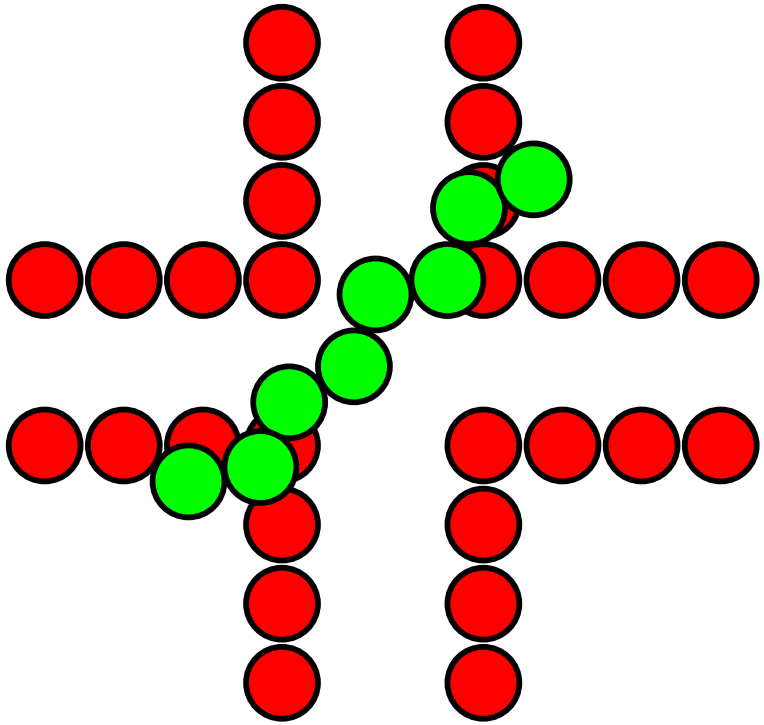
Static versus Dynamic Monte Carlo

- **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}



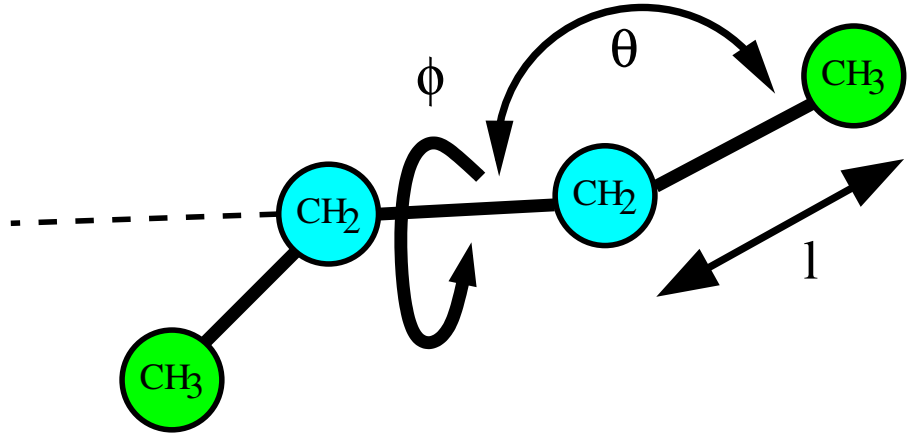
Configurational-Bias Monte Carlo

- 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)

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

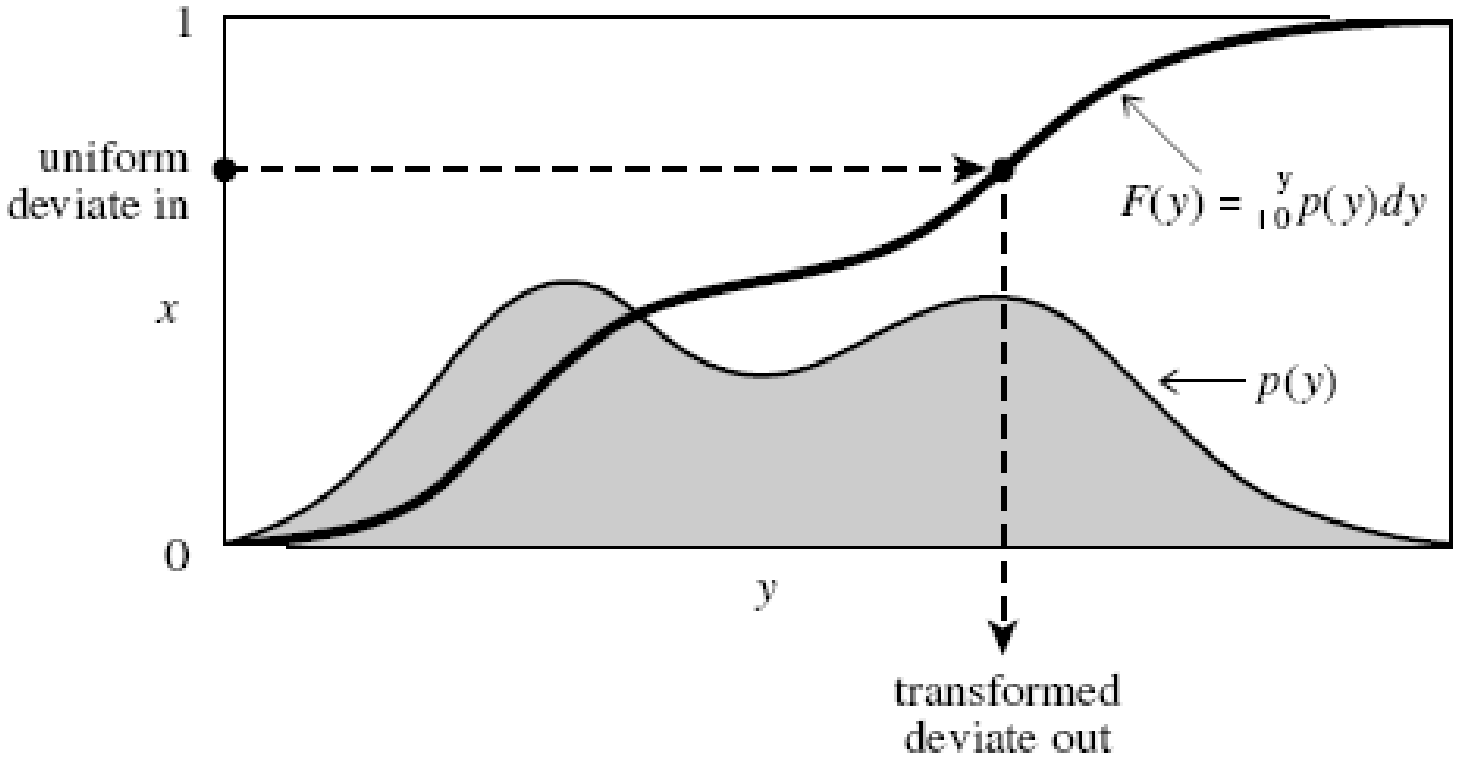


Figure 7.2.1. Transformation method for generating a random deviate y from a known probability distribution $p(y)$. The indefinite integral of $p(y)$ must be known and invertible. A uniform deviate x is chosen between 0 and 1. Its corresponding y on the definite-integral curve is the desired deviate.

Generate a Random Number from a Distribution (2)

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

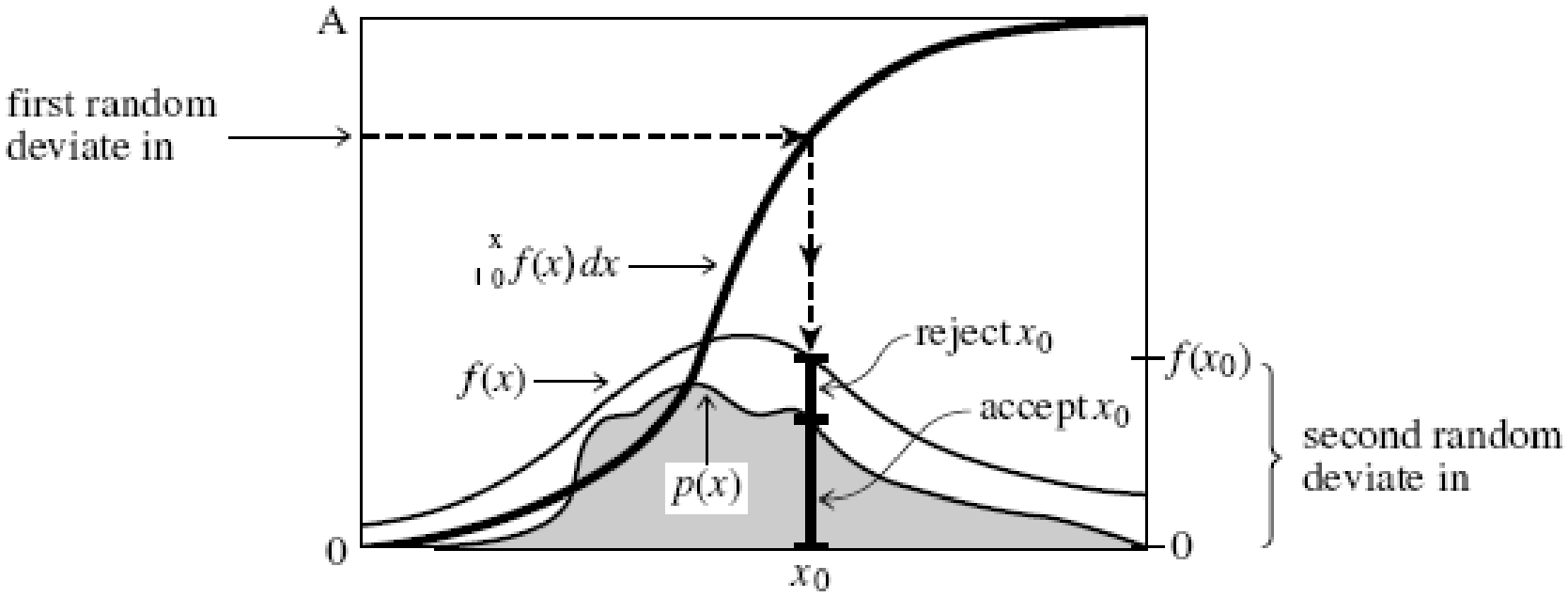


Figure 7.3.1. Rejection method for generating a random deviate x from a known probability distribution $p(x)$ that is everywhere less than some other function $f(x)$. The transformation method is first used to generate a random deviate x of the distribution f (compare Figure 7.2.1). A second uniform 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, $W(n) = \prod_{l=1}^{n} w_l$

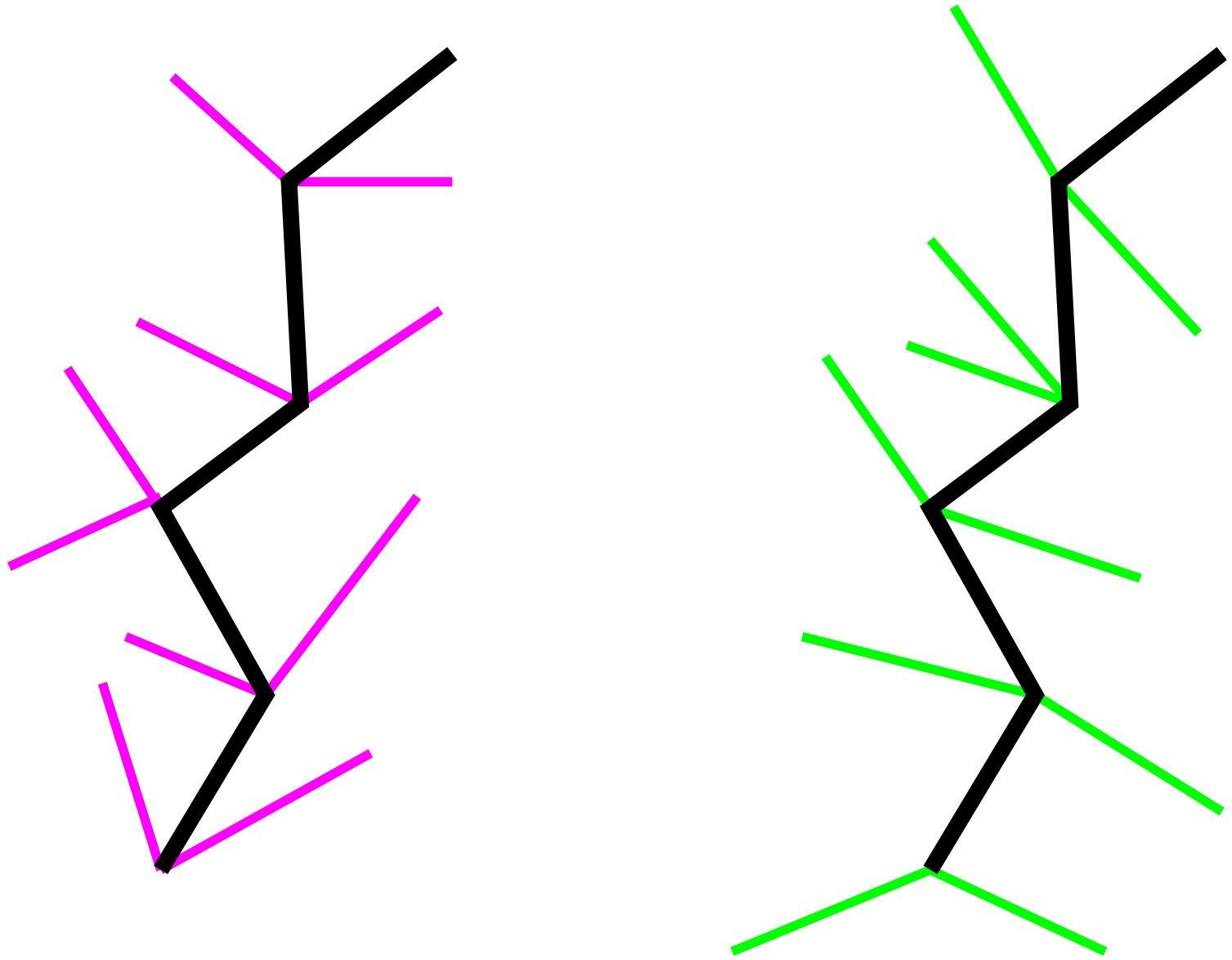
- Similar procedure for old configuration, generate $k-1$ trial positions (trial position 1 is the old configuration itself), leading to $W(o)$

- 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 $\mathbf{b}_n\mathbf{b}_o$:

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

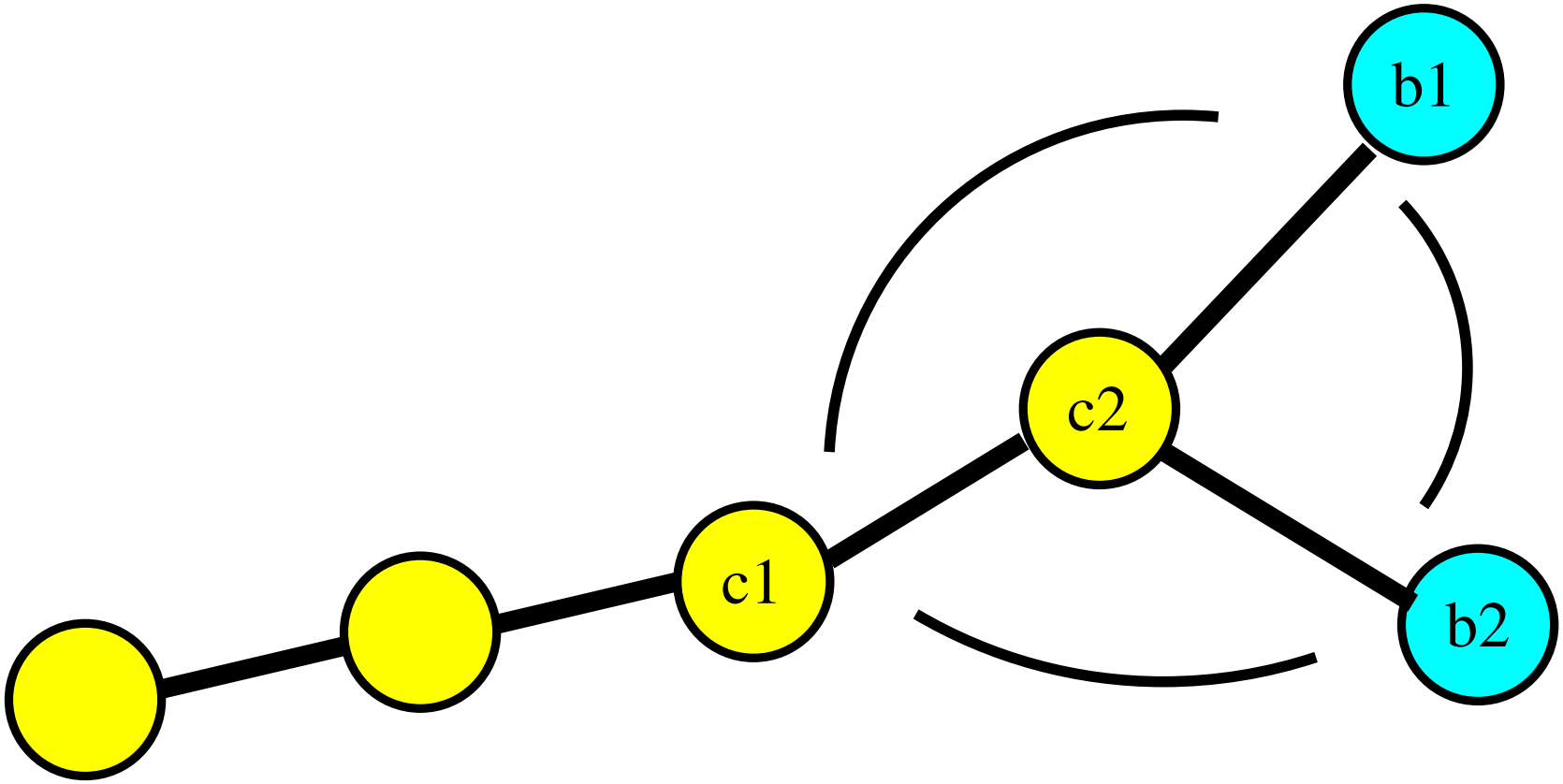
As

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

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

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

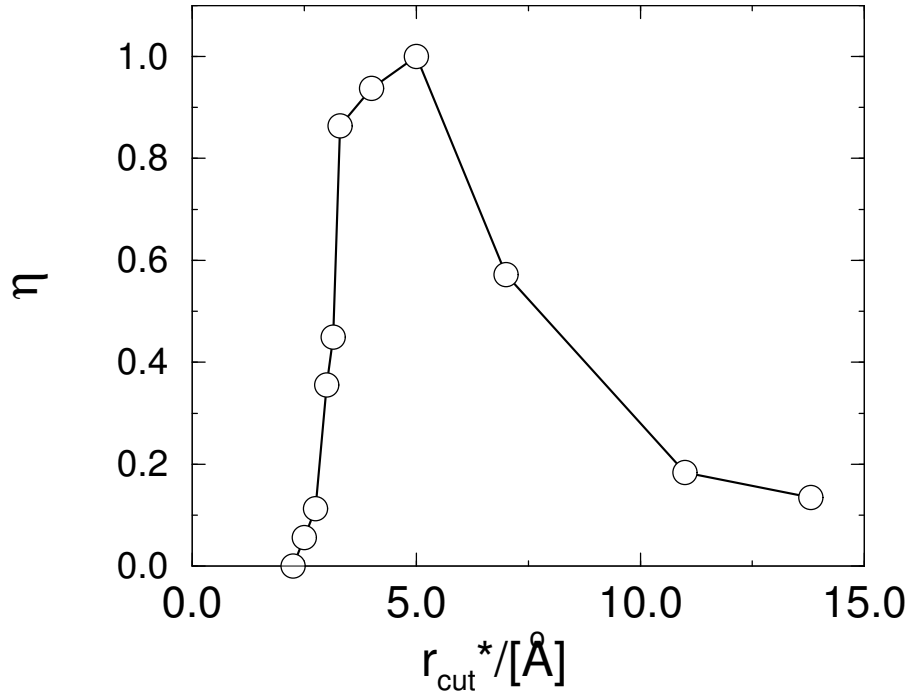
- Repeat until k trial orientations are found; these are fed into CBMC leading to $W(n)$

- 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)

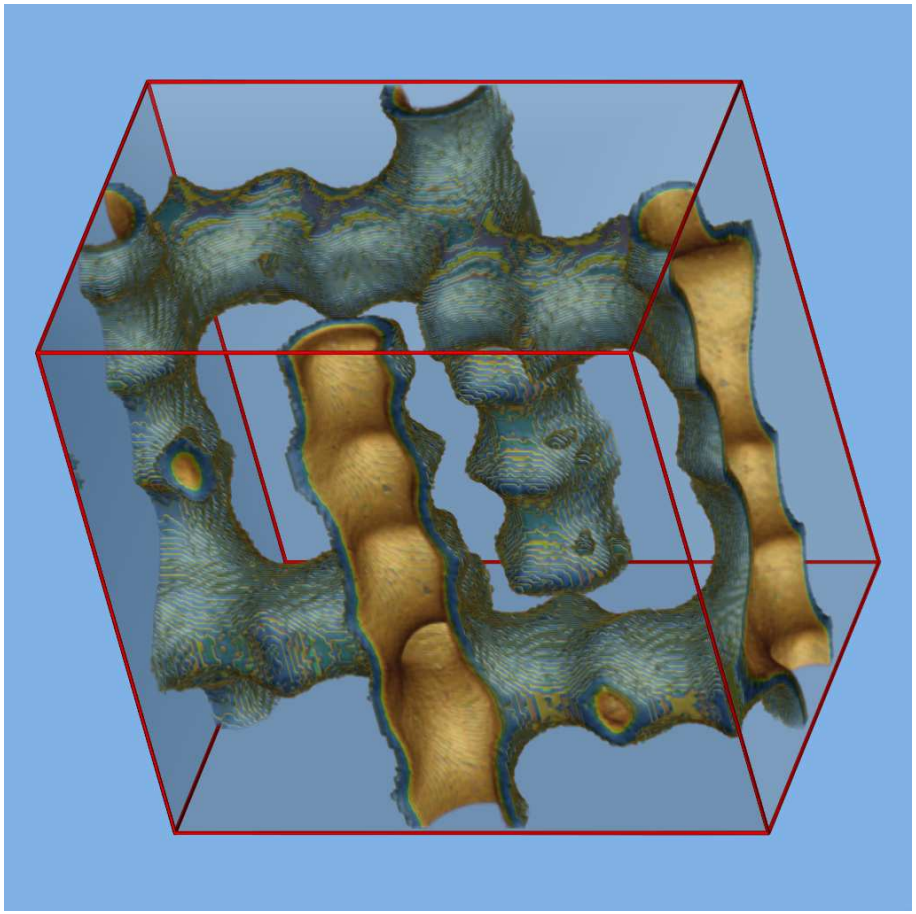
$$\mathrm{acc}(o \to n) = \min\left(1, \frac{W^\star(n)}{W^\star(o)} \times \exp[-\beta[\delta u(n) - \delta u(o)]]\right)$$

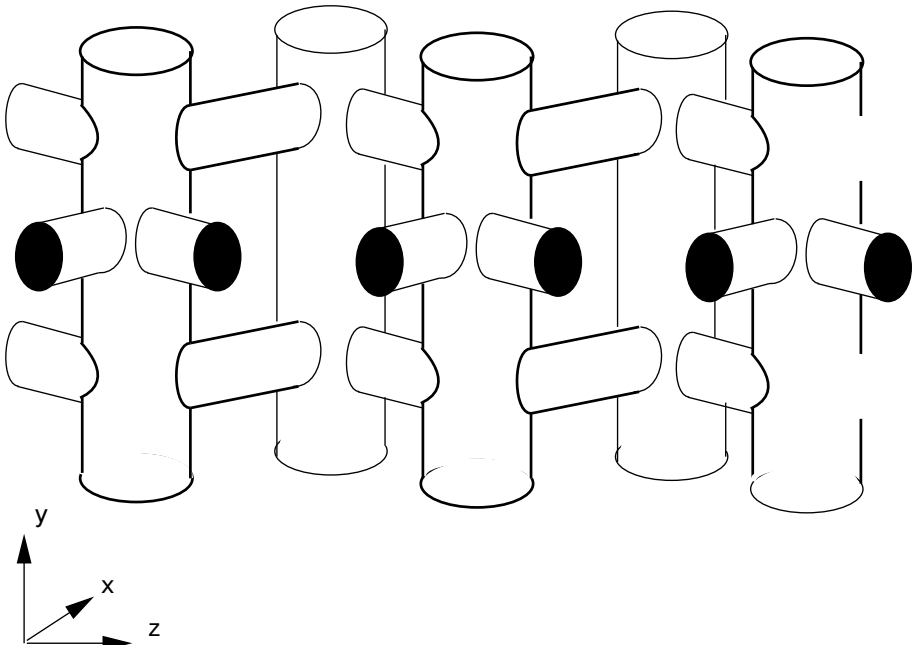
Application 1: Adsorption of Alkanes in MFI-type zeolite (1)

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)



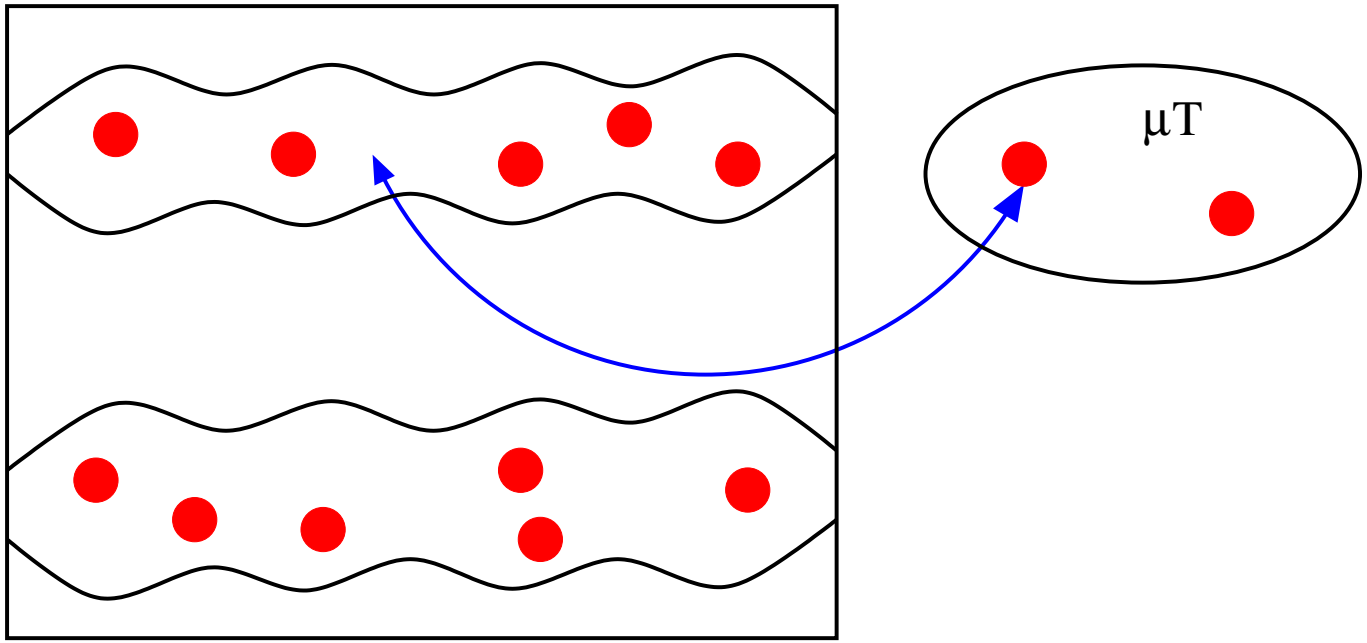


straight channels (y direction), zig-zag channels (xz plane) and intersections

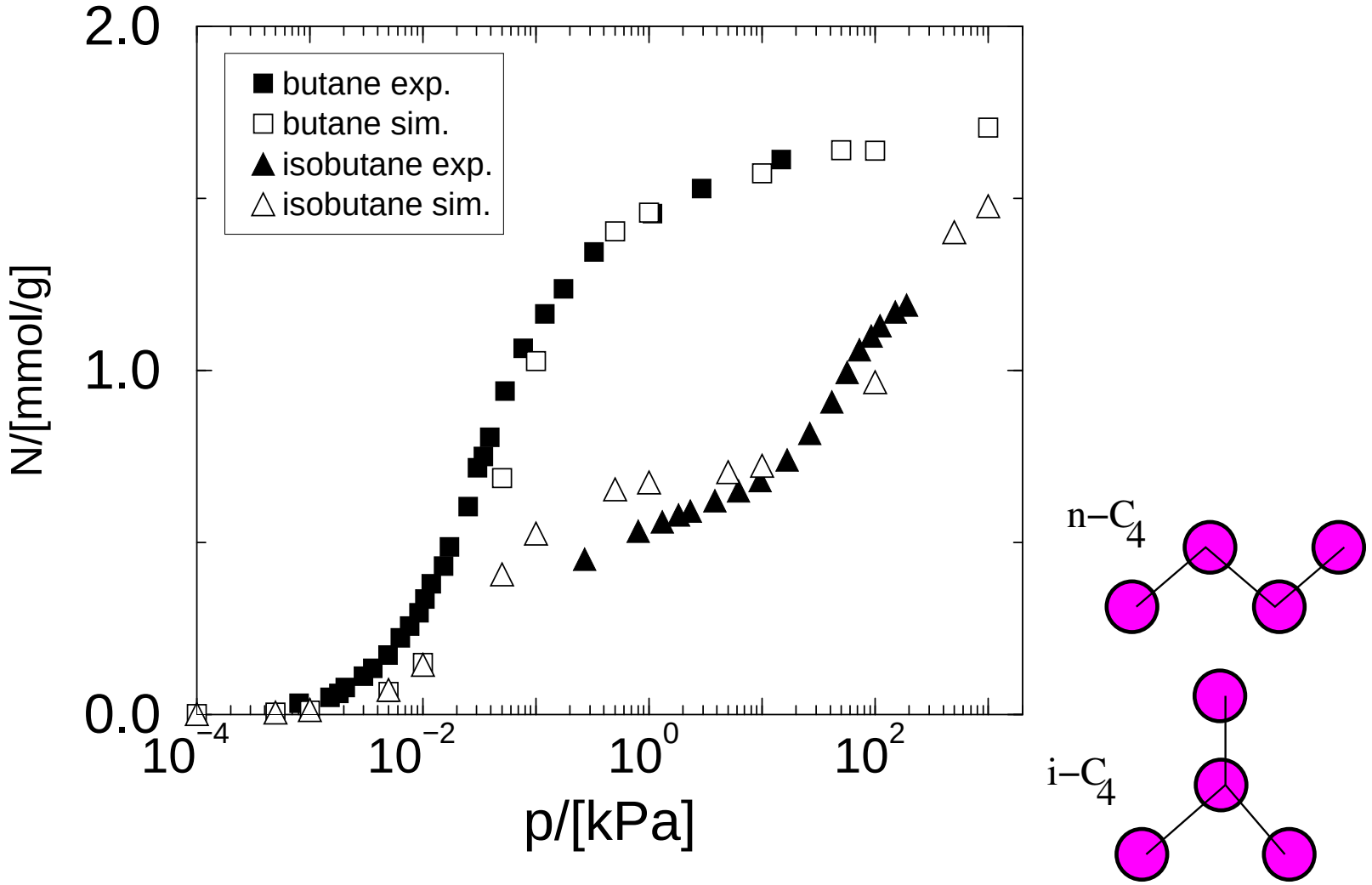
Application 1: Adsorption of Alkanes in MFI-type zeolite (3)

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: $\mu_{\mathrm{gas}} = \mu_{\mathrm{zeolite}}$; measure average $\langle N \rangle$ for given μ and β



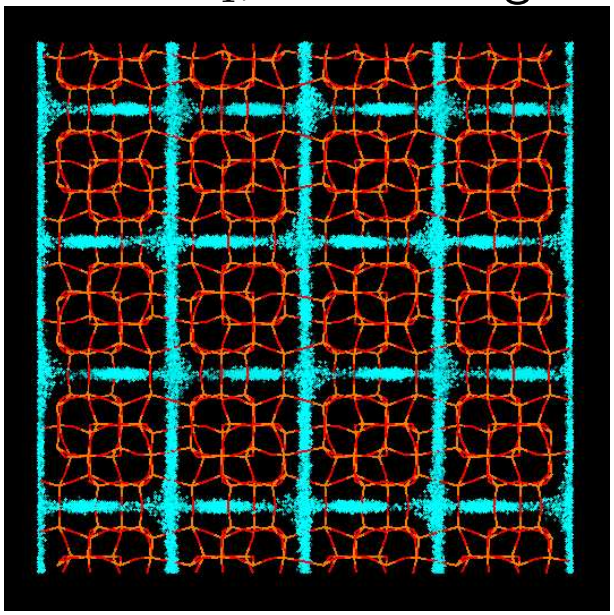
Application 1: Adsorption of Alkanes in MFI-type zeolite (4)



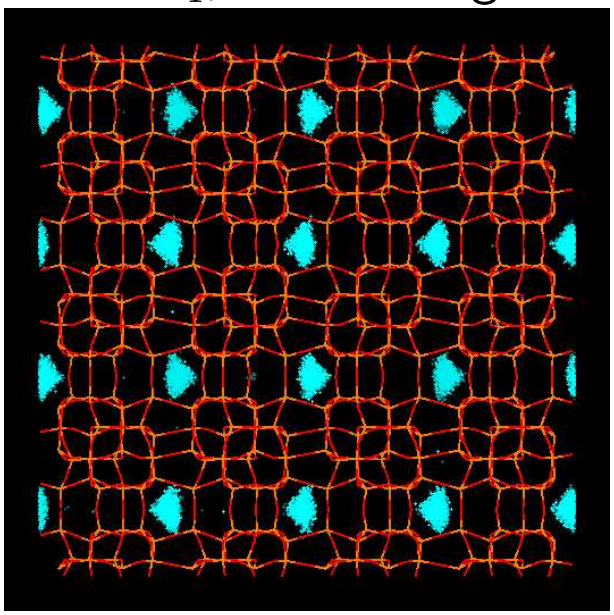
Vlugt et al, J. Am. Chem. Soc., 1998, 120, 5599

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

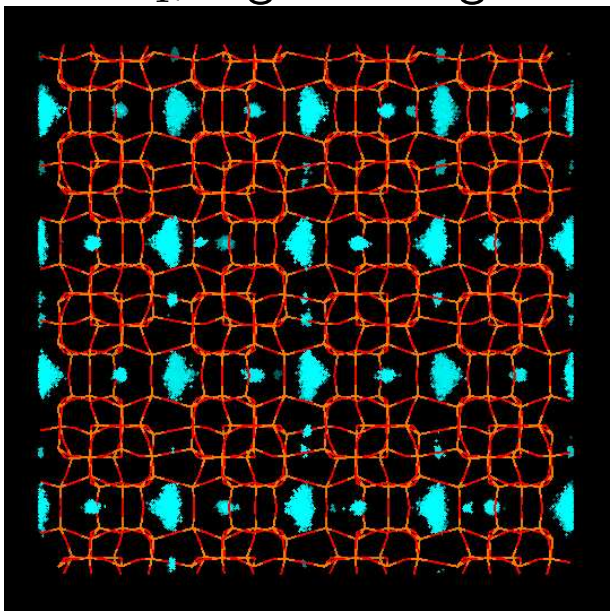
n-C_4, low loading



i-C_4, low loading



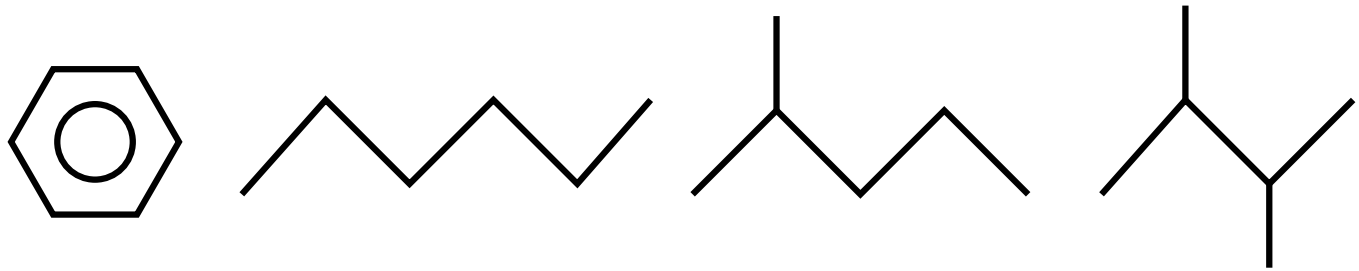
i-C_4, high loading



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

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

Research Octane Number (RON)

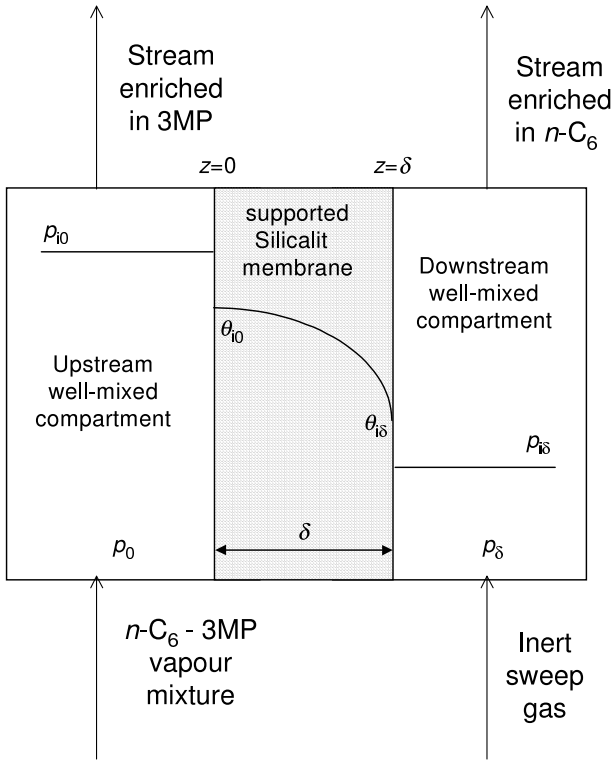


>100 25 73 92

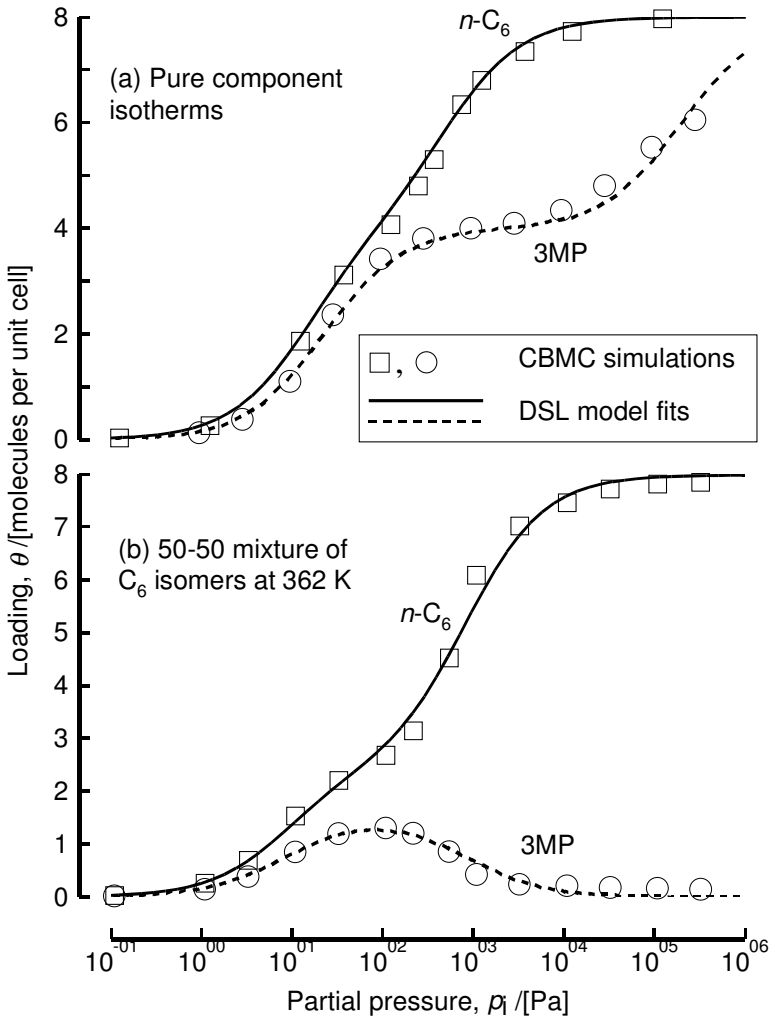
Application 1: Adsorption of Alkanes in MFI-type zeolite (7)

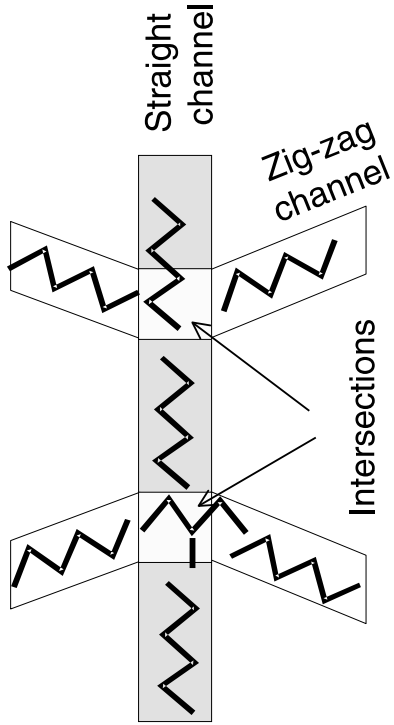
Flux	$n-C_6$	$i-C_6$	selectivity
pure	179	136	1.3
50%-50%	46	1.9	24

Experiments by J. Falconer, Univ. Colorado



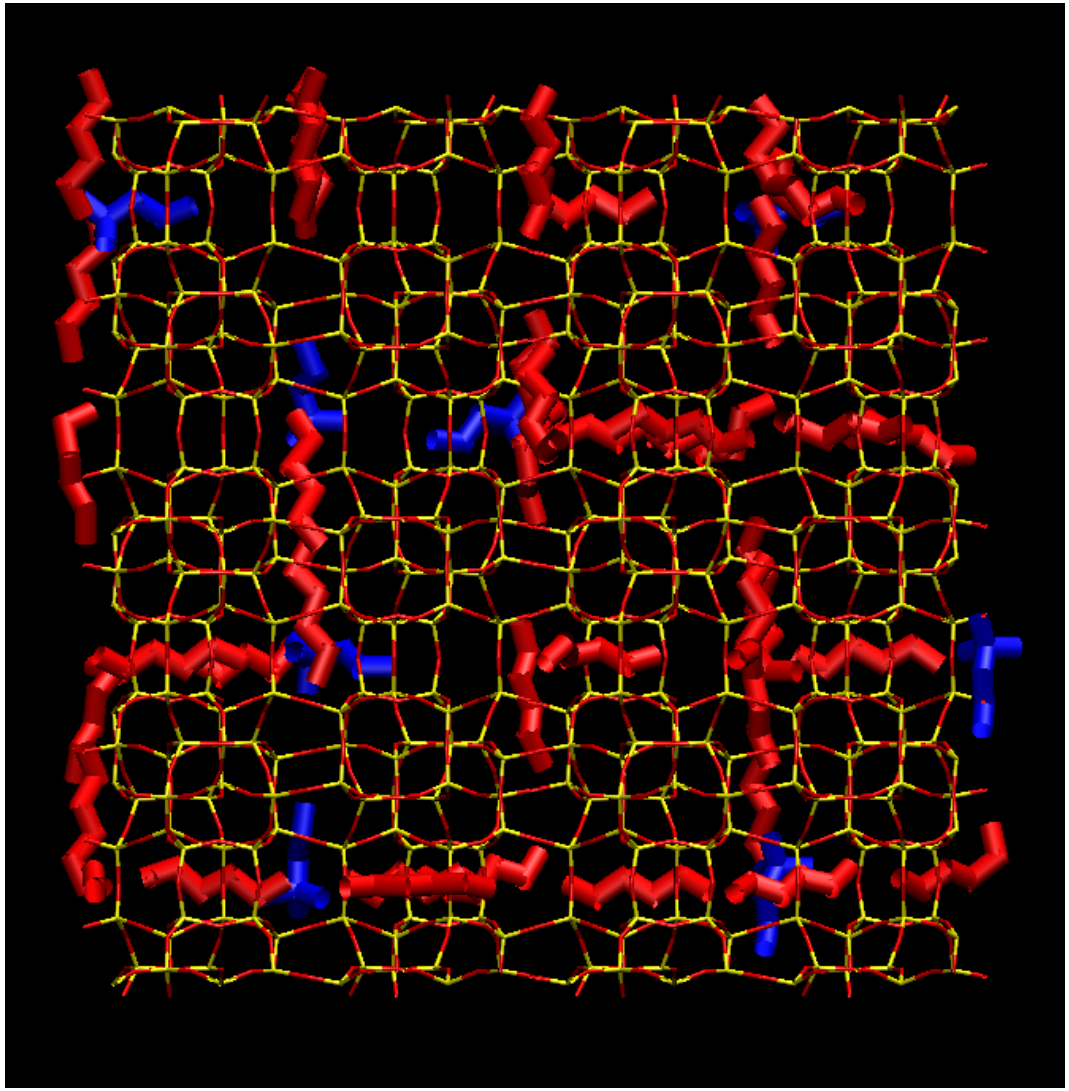
Application 1: Adsorption of Alkanes in MFI-type zeolite (8)



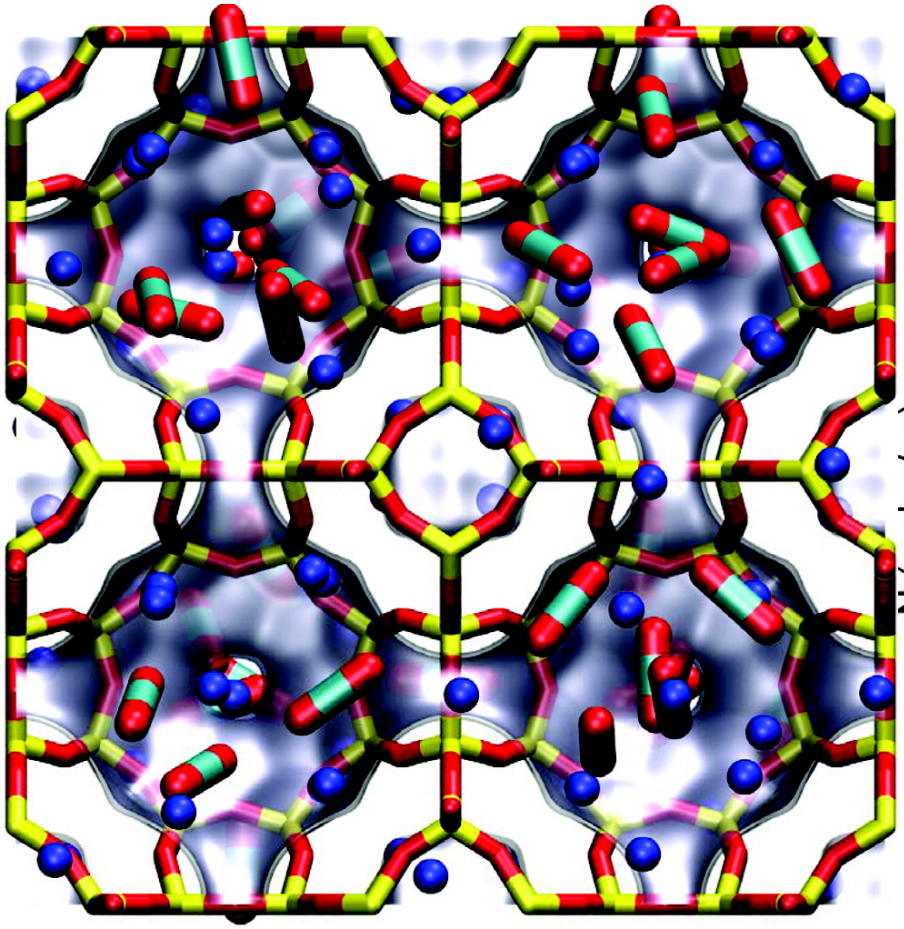


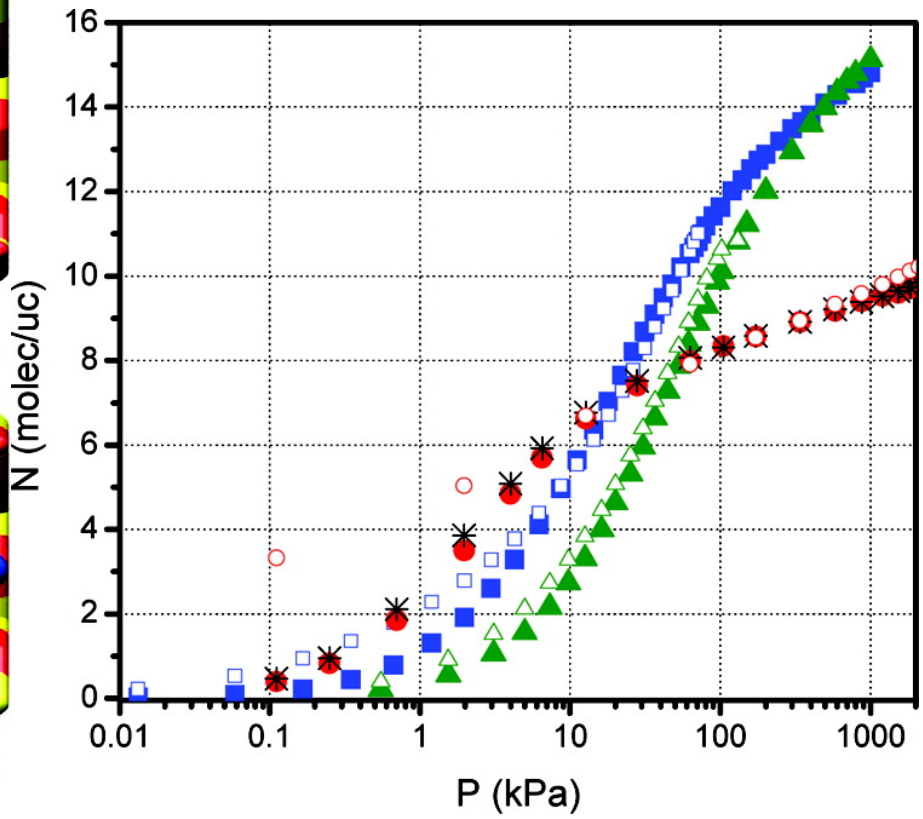
Application 1: Adsorption of Alkanes in MFI-type zeolite (9)

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



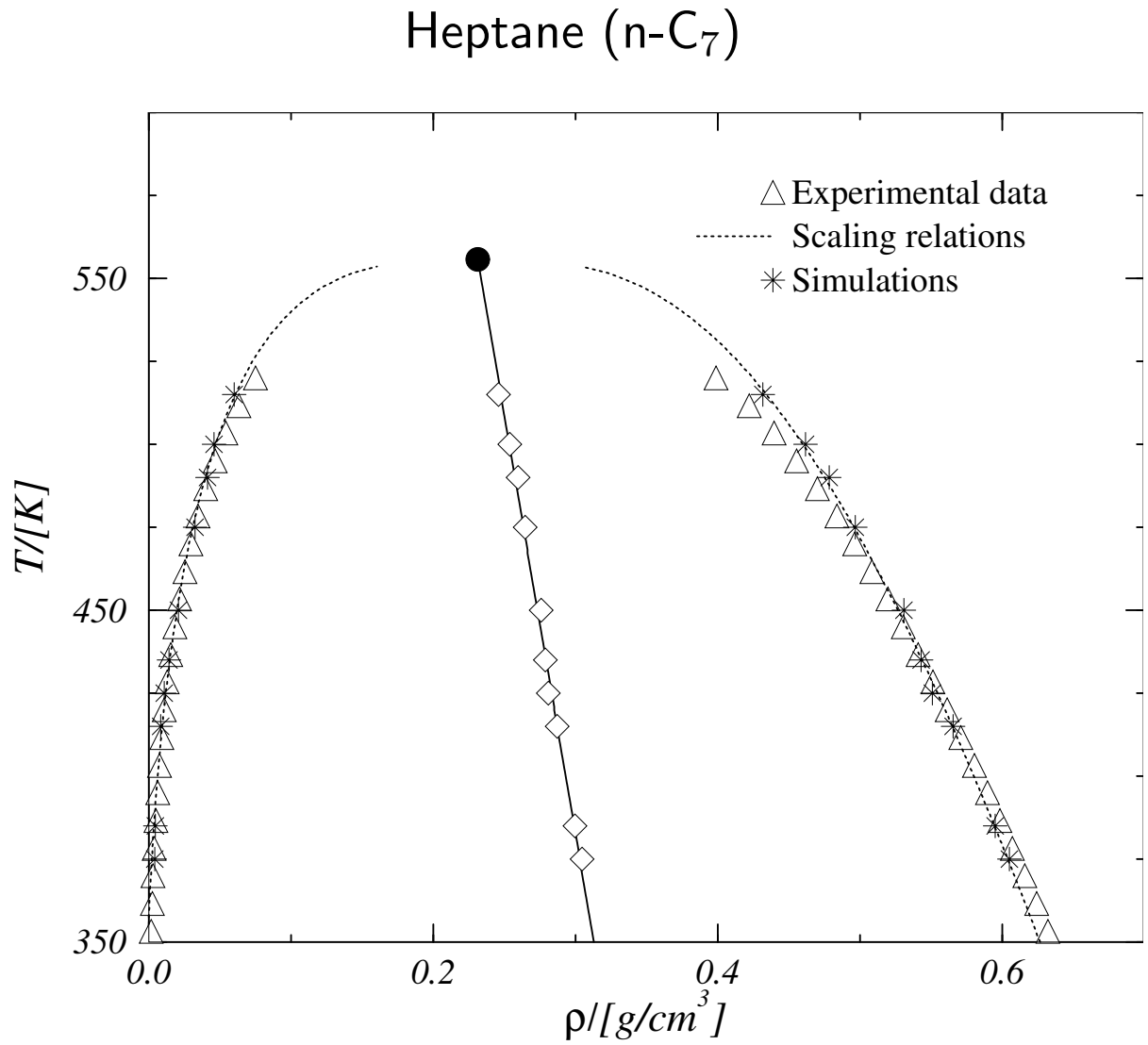
Application 2: Adsorption of CO_2 in Na^+ containing zeolites





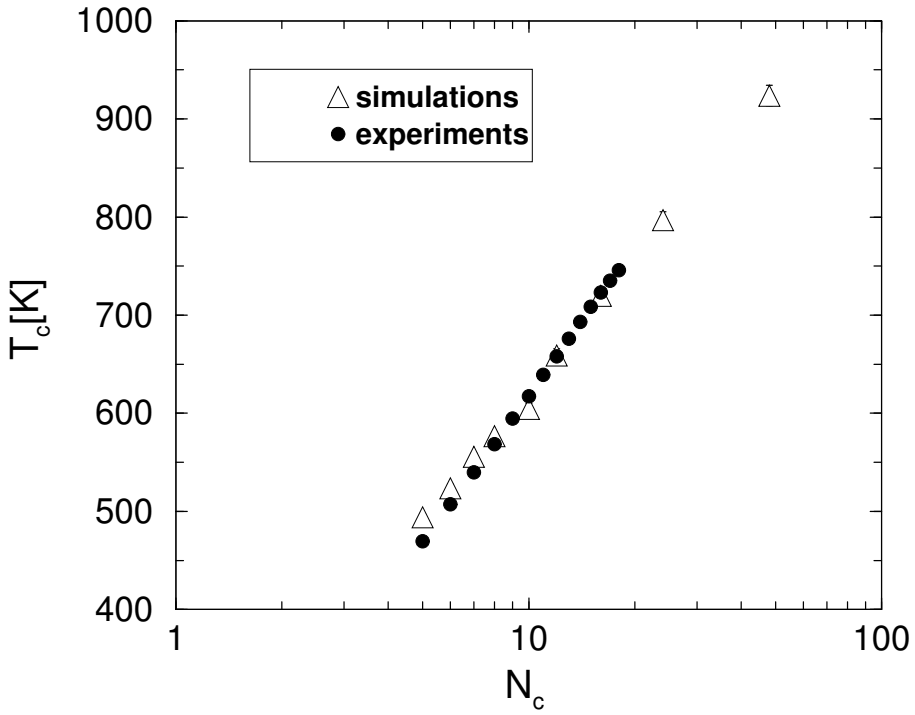
Sofia Calero et al., J. Phys. Chem. C, 2009, 113, 8814-8820

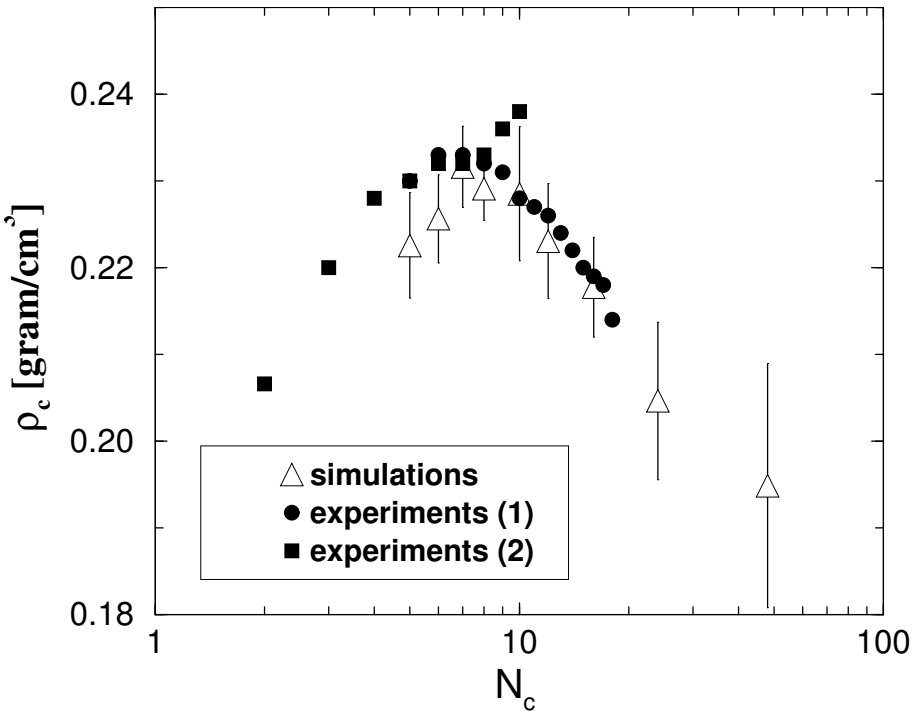
Application 3: Gibbs Ensemble Monte Carlo (1)



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

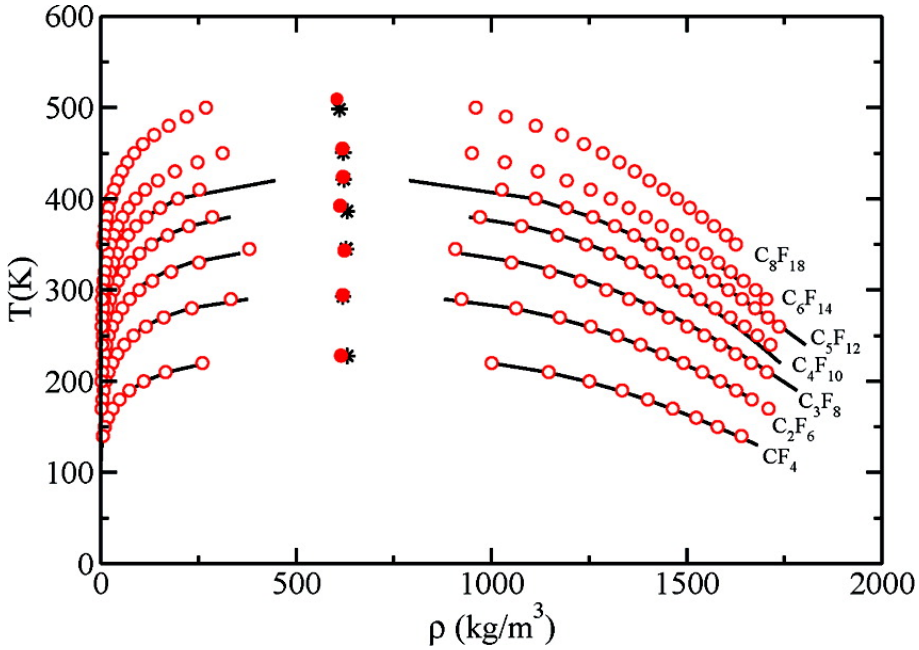
Application 3: Gibbs Ensemble Monte Carlo (2)

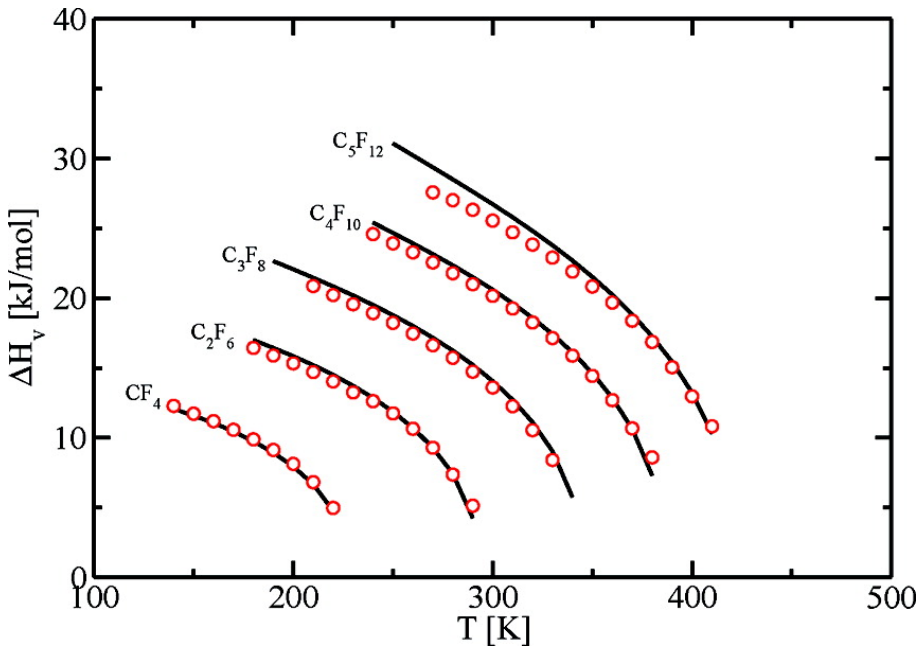




B. Smit, S. Karaborni, and J.I. Siepmann, J. Chem. Phys. 102, 2126 (1995)

Application 4: GCMC Histogram Reweighting





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

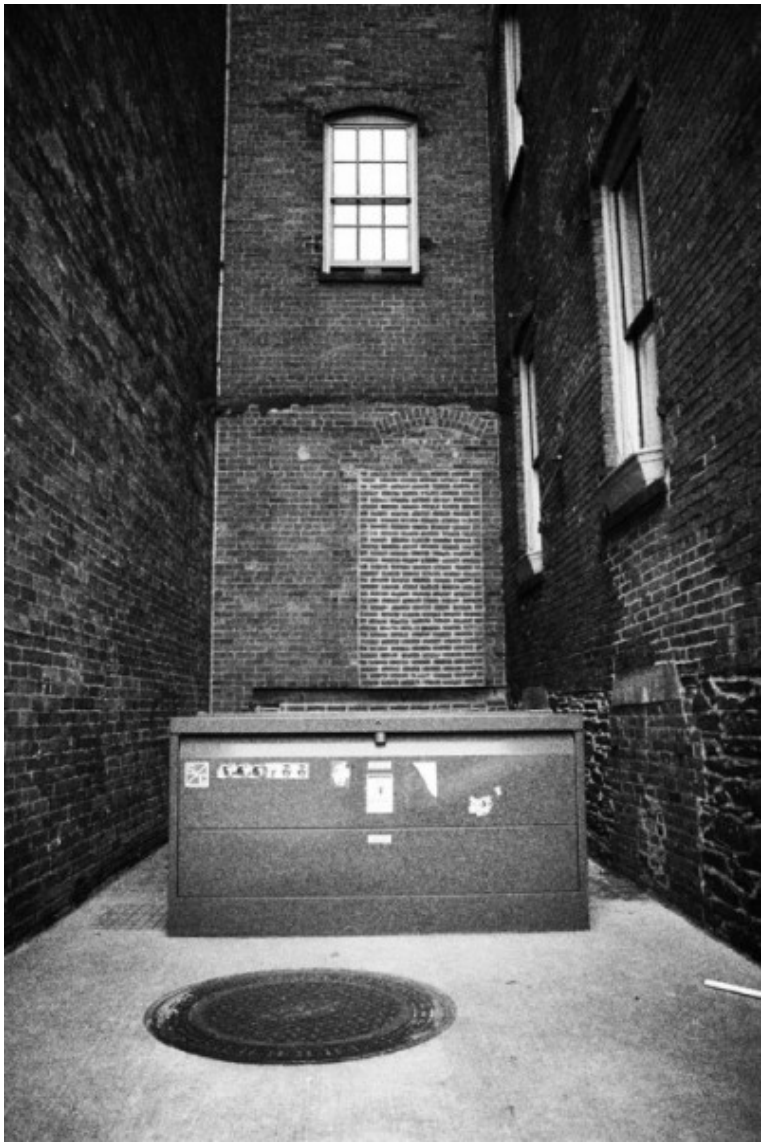
Efficiency of CBMC

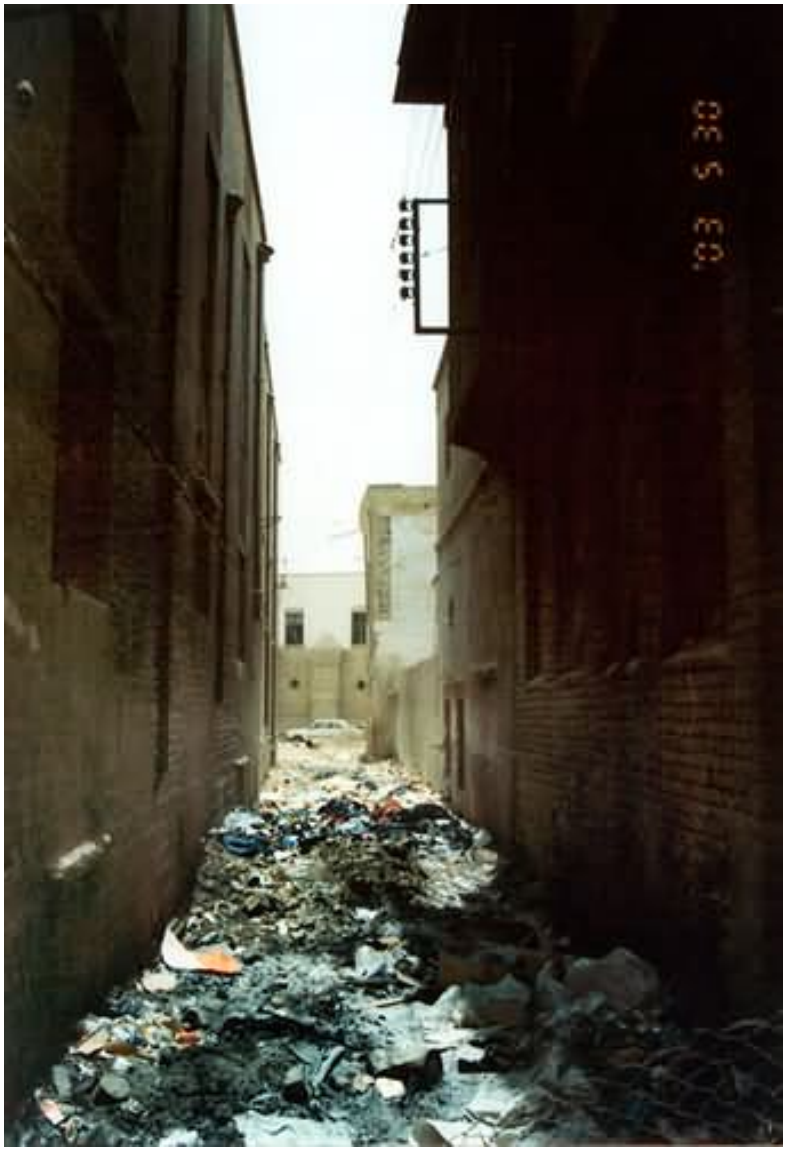
- k: number of trial directions
- a: probability that trial direction has a "favorable" energy
- growth can continue as long as at least 1 trial direction is "favorable"
- generate chain of length N successfully

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

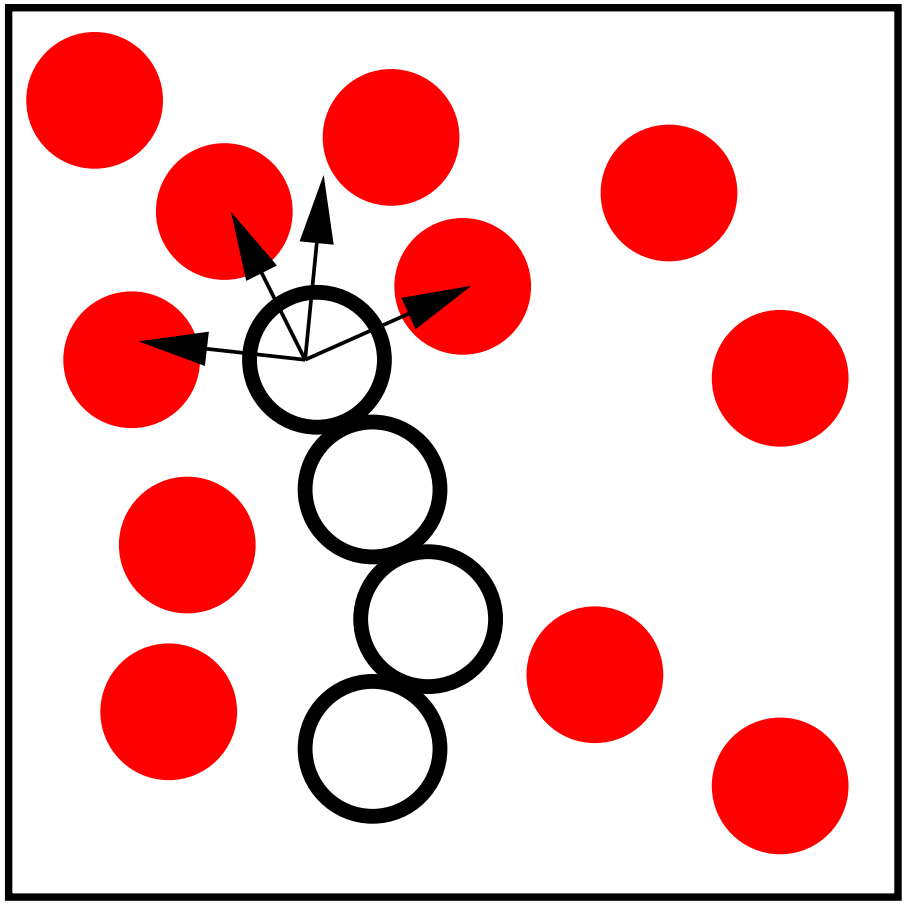
- increasing k means increasing CPU time.

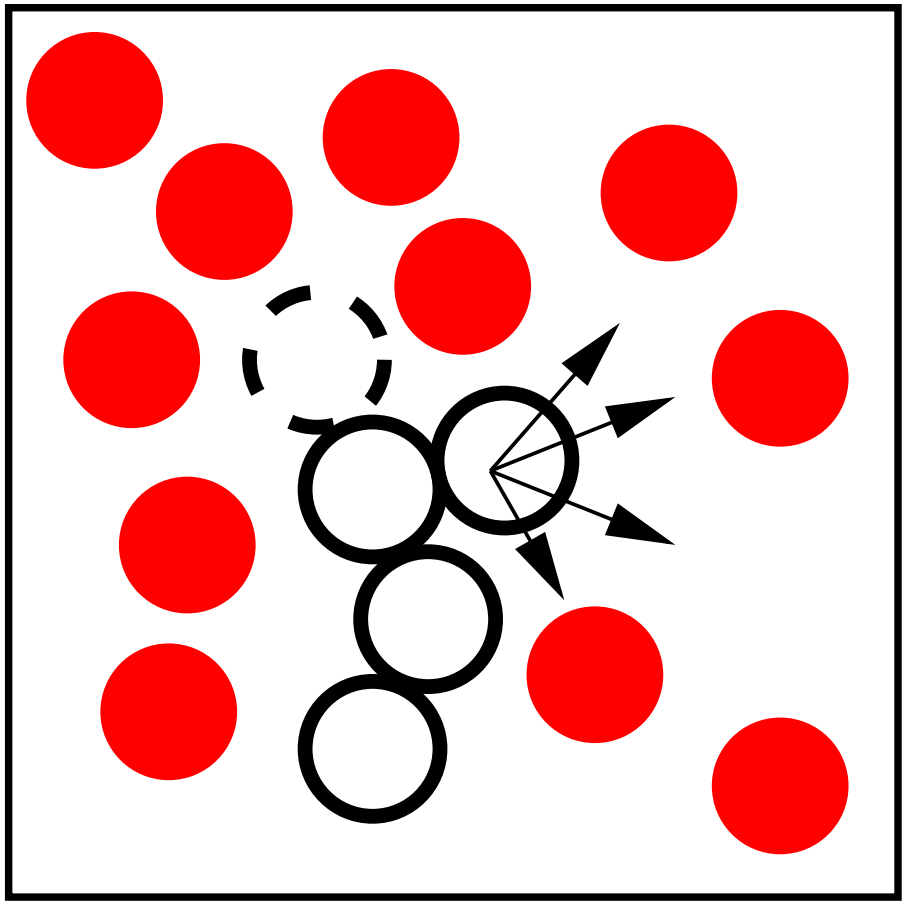
Dead-End Alley





Recoil Growth: Avoiding Dead-End Alley



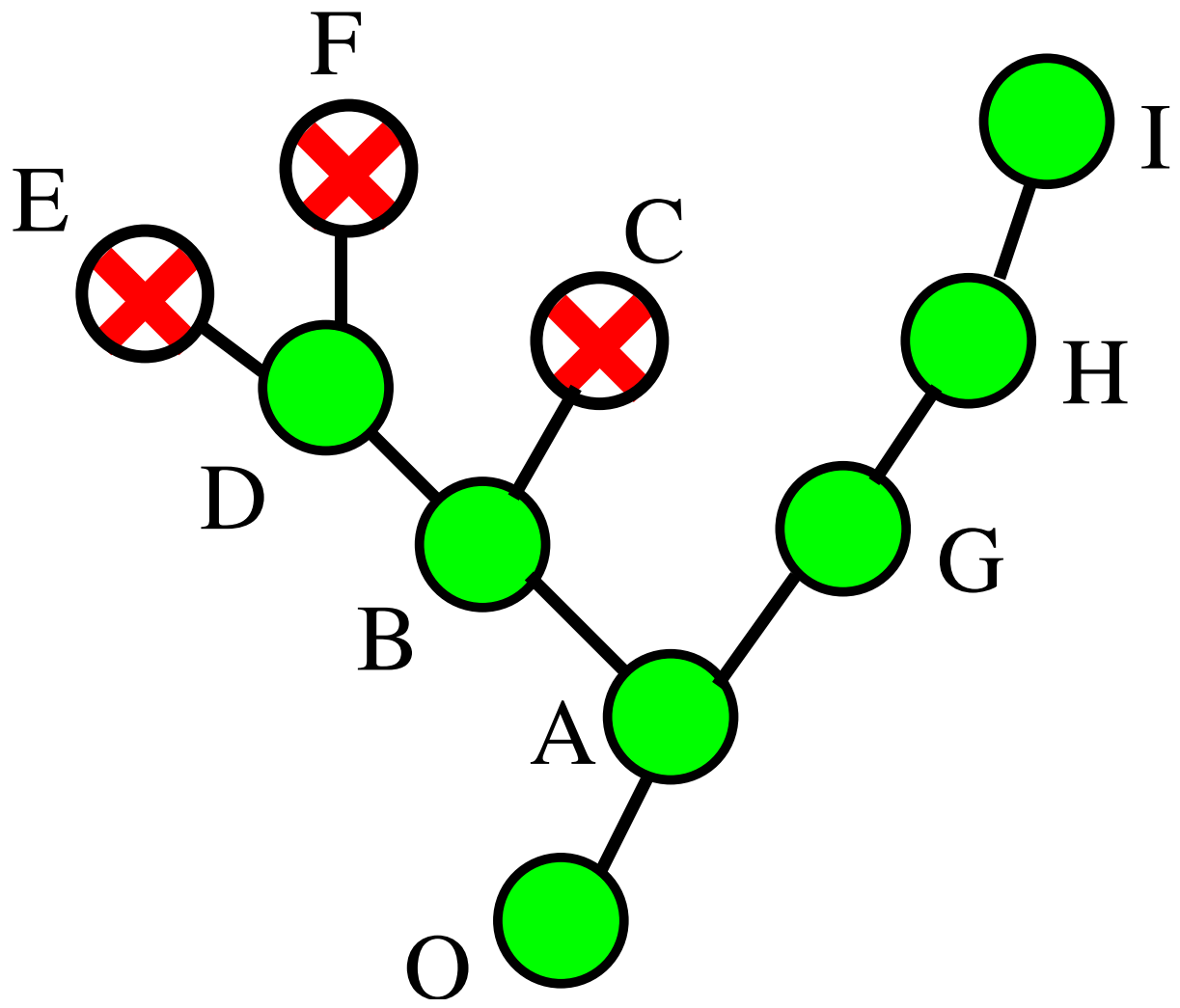


Recoil Growth Algorithm (1)

- 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,

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

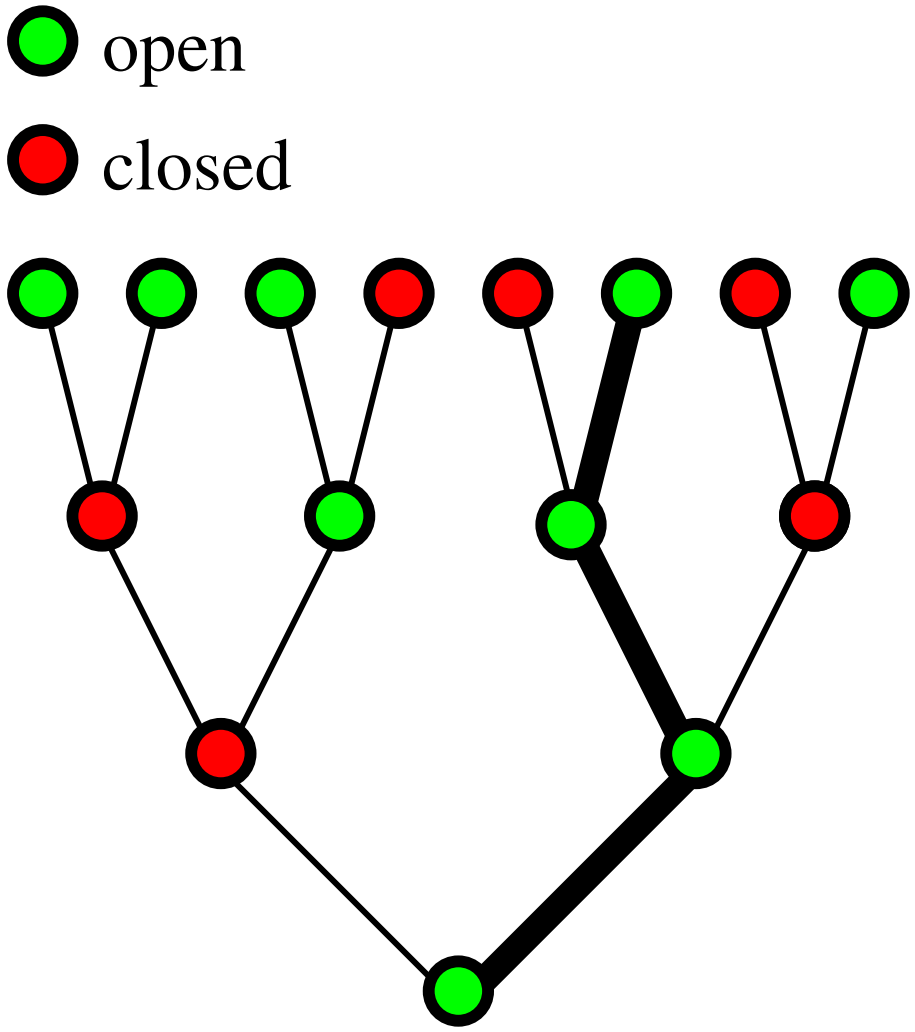
Therefore,

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

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)

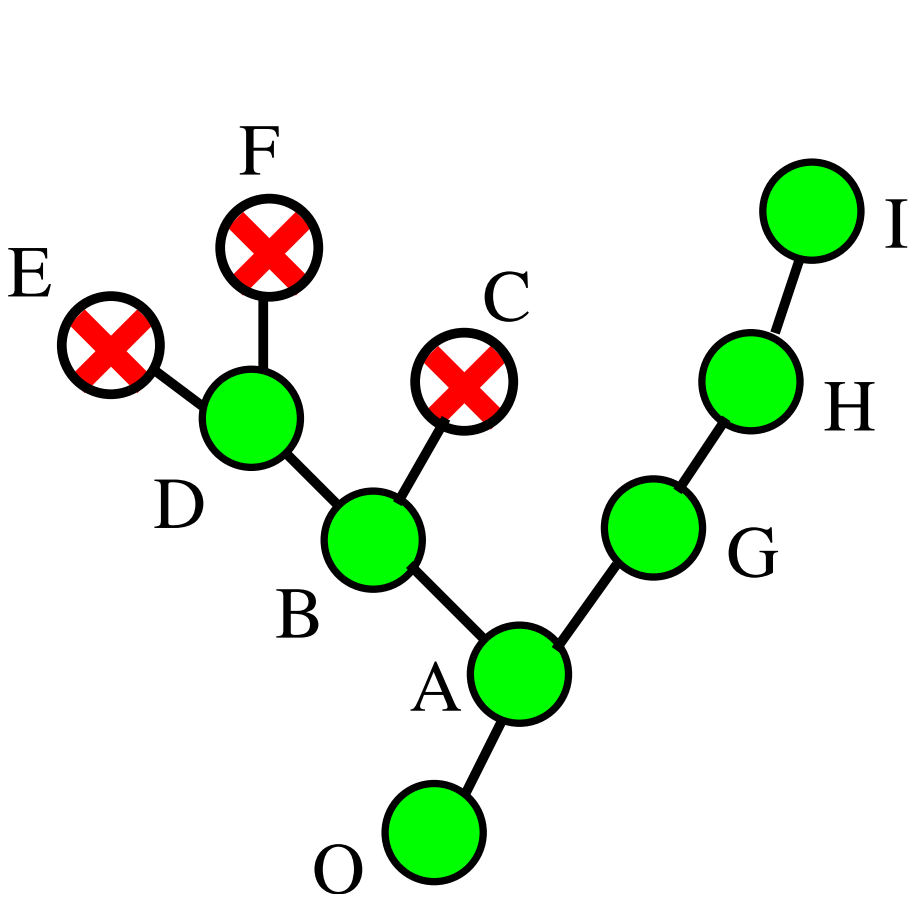
Random Walk on a Tree ($k = 2$, $l = 3$)

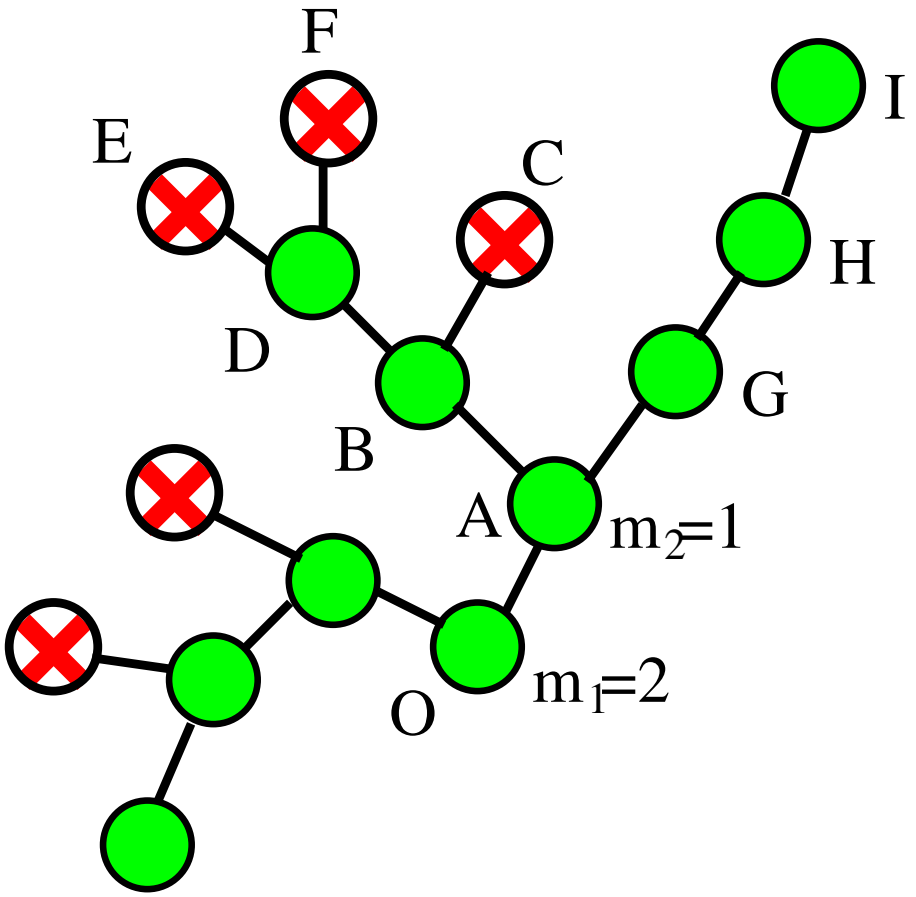


Super-Detailed Balance

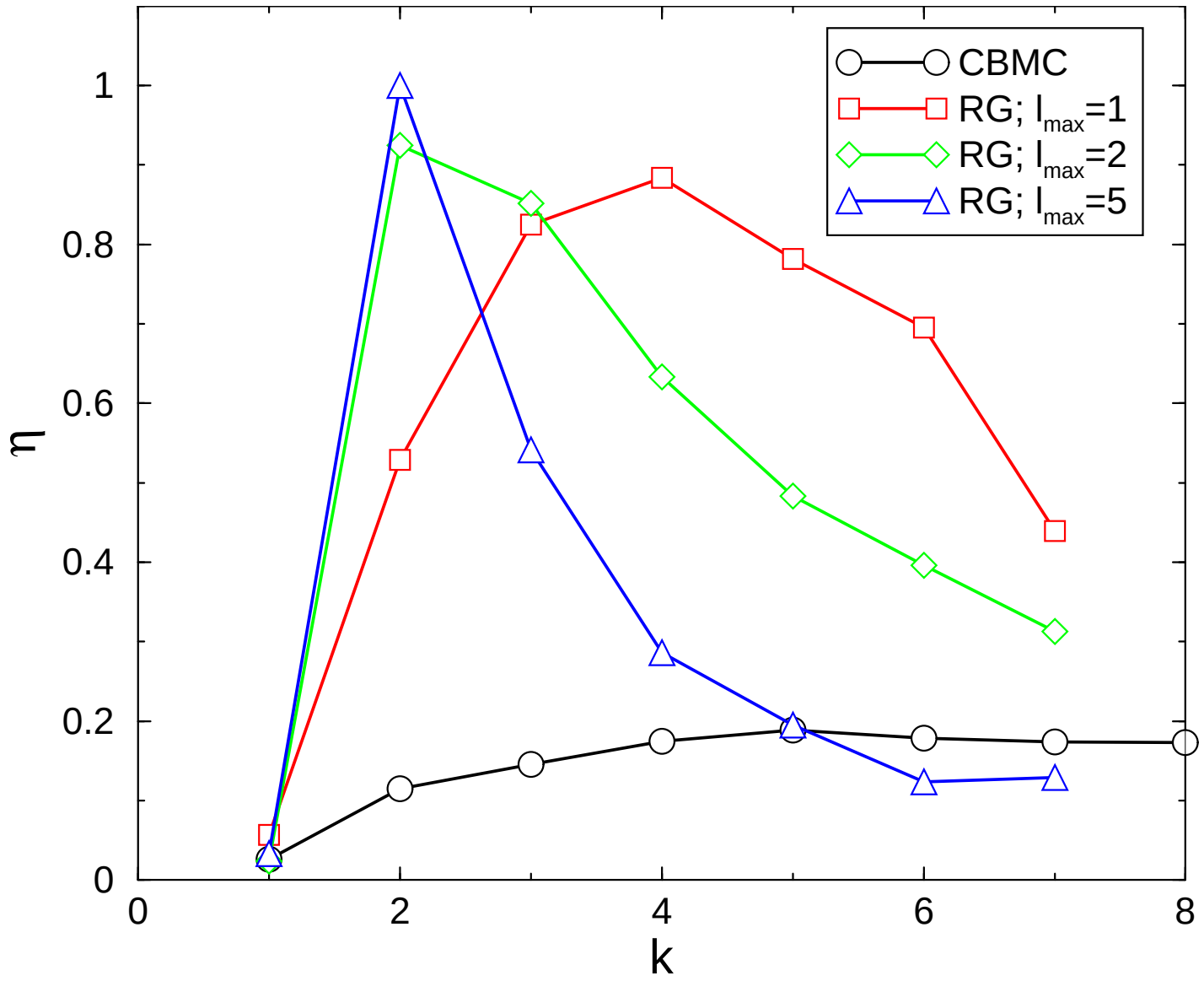
$$
\begin{aligned}
K(o \to n|t_n t_o O_n O_o) &= K(n \to o|t_n t_o O_n O_o) \\
\alpha(o \to n|t_n t_o O_n O_o) &= P(t_n)P(O_n|t_n)P(rw_n|t_n O_n) \times \\
&\quad P(t_o|rw_o)P(O_o|t_o rw_o) \\
\alpha(n \to o|t_n t_o O_n O_o) &= P(t_o)P(O_o|t_o)P(rw_o|t_o O_o) \times \\
&\quad P(t_n|rw_n)P(O_n|t_n rw_n)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

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





Efficiency of RG Compared to CBMC



Other Methods

- Continuous Fractional Component Monte Carlo
  - system contains N molecules and one "fractional" molecule
  - interactions of the fractional molecule are described by order parameter λ
  - include trial-moves to change λ
  - "fractional" molecule can become a "real" molecule or disappear
  - Maginn, J. Chem. Theory Comput., 2007, 3, 451-1463

- Wormhole move
  - create an artificial "hole" in the system
  - use reptation steps to gradually insert the chain
  - accept/reject individual reptation steps
  - move is completed when the whole chain is transferred through the hole
  - Houdayer, Journal of Chemical Physics, 2002, 116, 1783-1787

The End

Questions ??