**ICTP School in Computational Condensed Matter Physics:**
**From Atomistic Simulations to Universal Model Hamiltonians**
**September 2015**

# Lecture 1
# Stochastic series expansion (SSE) method for simulations of quantum spins

**Anders W Sandvik, Boston University**

**Review article** on quantum spin systems and numerical methods (incl. SSE)**: ArXiv:1101.3281**



# Lecture outline

- Intro to quantum Heisenberg model

- Path integrals on the lattice

- Stochastic Series expansion (alternative to path integral)

- Implementation for 2D S=1/2 Heisenberg antiferromagnet

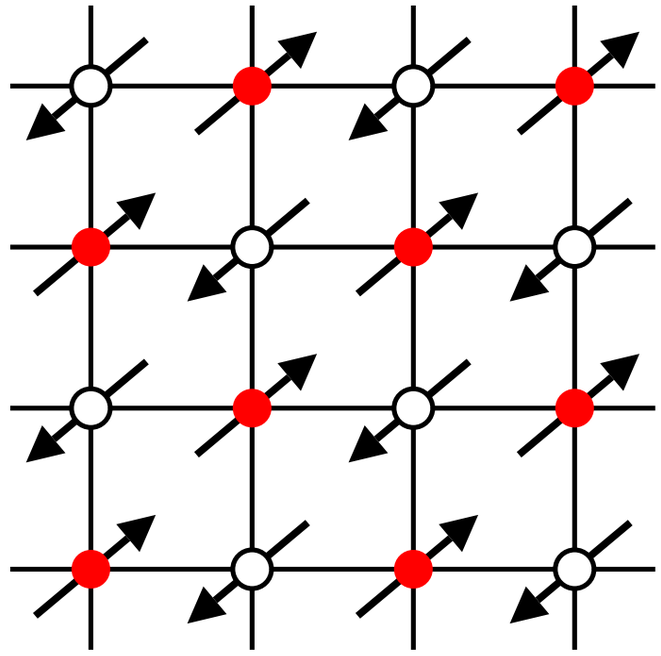
- Look at program if we have time

# Quantum antiferromagnets

Starting model for spin-isotropically interacting S=1/2 spins:

The **Heisenberg model** with nearest-neighbor <i,j> interactions

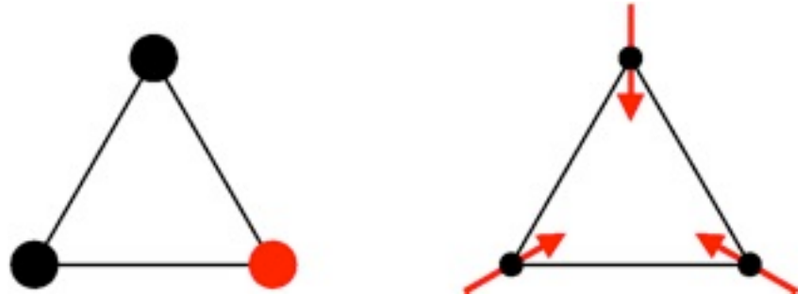
$$H = J \sum_{\langle i,j \rangle} \vec{S}_i \cdot \vec{S}_j, \qquad J > 0$$



Lattices can be classified as:

**Bipartite**

- nearest-neighbors i,j always on different sublattices
- compatible with Neel order
- but other states possible

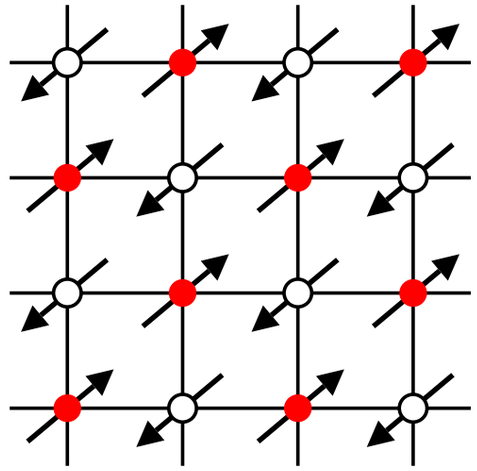


**Non-bipartite**

- no bipartition is possible
- frustrated antiferromagnetic interactions
- different kinds of order or no long-range order (spin liquid)

## Order Parameter: Sublattice magnetization

$$\vec{m}_s = \frac{1}{N}\sum_{i=1}^{N} \phi_i \vec{S}_i, \qquad \phi_i = (-1)^{x_i + y_i} \quad \text{(2D square lattice)}$$



- In a classical bipartite system at T=0: $|\langle \vec{m}_s \rangle| = S$

- In finite system, symmetry not broken, use: $\langle m_s^2 \rangle = S^2$

- At T > 0 thermal fluctuations reduce order: $\langle m_s^2 \rangle < S^2$

## Quantum system: Fully ordered Neel state not an eigenstate

of H even on a bipartite lattice

- $|\mathbf{m_s}| < \mathbf{S}$ at T=0 (can have $|\mathbf{m_s}| = \mathbf{0}$)

## Mermin-Wagner theorem

(on breaking a continuous symmetry):

- No Neel order in 1D quantum Heisenberg model
- Neel order possible only at T=0 in 2D system
- Order possible also at T>0 in 3D

# Quantum Monte Carlo

Rewrite the quantum-mechanical expectation value into a classical form

$$\langle A \rangle = \frac{\mathrm{Tr}\{Ae^{-\beta H}\}}{\mathrm{Tr}\{e^{-\beta H}\}} \rightarrow \frac{\sum_c A_c W_c}{\sum W_c}$$

Monte Carlo sampling in the space **{c} with weights $W_c$** (if positive-definite...)

**Different ways of doing it**

("sign problem" if not the case)

- World-line methods for spins and bosons
- Stochastic series expansion for spins and bosons
- Fermion determinant methods

For ground state calculations we can also do projection from a "trial state"

$$|\Psi_m\rangle \sim H^m |\Psi_0\rangle \qquad |\Psi_m\rangle \rightarrow |0\rangle \quad \text{when} \quad m \rightarrow \infty$$

$$|\Psi_\beta\rangle \sim \mathrm{e}^{-\beta H} |\Psi_0\rangle \qquad |\Psi_\beta\rangle \rightarrow |0\rangle \quad \text{when} \quad \beta \rightarrow \infty$$

Particularly simple and efficient schemes exist for S=1/2 models

$$H = -J \sum_{b=1}^{N_b} \left( \frac{1}{4} - \mathbf{S}_{i(b)} \cdot \mathbf{S}_{j(b)} \right)$$

(+ certain multi-spin terms)

No sign problem on **bipartite lattices**

# Path integrals on the lattice, imaginary time

We want to compute a thermal expectation value

$$\langle A \rangle = \frac{1}{Z} \mathrm{Tr}\{ A \mathrm{e}^{-\beta H} \}$$

where β=1/T (and possibly T→0). How to deal with the exponential operator?

"Time slicing" of the partition function

$$Z = \mathrm{Tr}\{\mathrm{e}^{-\beta H}\} = \mathrm{Tr}\left\{ \prod_{l=1}^{L} \mathrm{e}^{-\Delta_\tau H} \right\} \qquad \Delta_\tau = \beta / L$$

Choose a basis and insert complete sets of states;

$$Z = \sum_{\alpha_0} \sum_{\alpha_1} \cdots \sum_{\alpha_L - 1} \langle \alpha_0 | \mathrm{e}^{-\Delta_\tau H} | \alpha_{L-1} \rangle \cdots \langle \alpha_2 | \mathrm{e}^{-\Delta_\tau H} | \alpha_1 \rangle \langle \alpha_1 | \mathrm{e}^{-\Delta_\tau H} | \alpha_0 \rangle$$

Use approximation for imaginary time evolution operator. Simplest way

$$Z \approx \sum_{\{\alpha\}} \langle \alpha_0 | 1 - \Delta_\tau H | \alpha_{L-1} \rangle \cdots \langle \alpha_2 | 1 - \Delta_\tau H | \alpha_1 \rangle \langle \alpha_1 | 1 - \Delta_\tau H | \alpha_0 \rangle$$

Leads to error $\propto \Delta_\tau$. Limit $\Delta_\tau \to 0$ can be taken

# Example: hard-core bosons

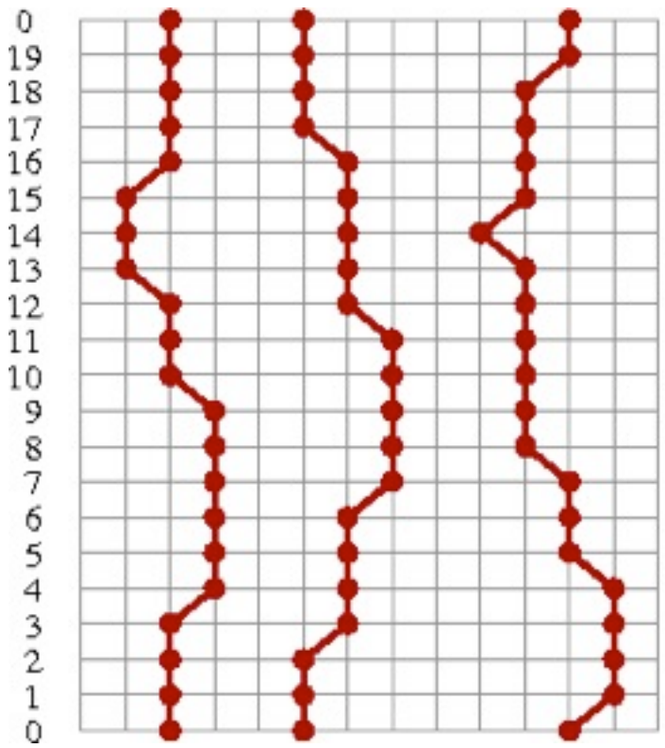
$$H = K = -\sum_{\langle i,j \rangle} K_{ij} = -\sum_{\langle i,j \rangle} (a_j^\dagger a_i + a_i^\dagger a_j) \qquad n_i = a_i^\dagger a_i \in \{0, 1\}$$

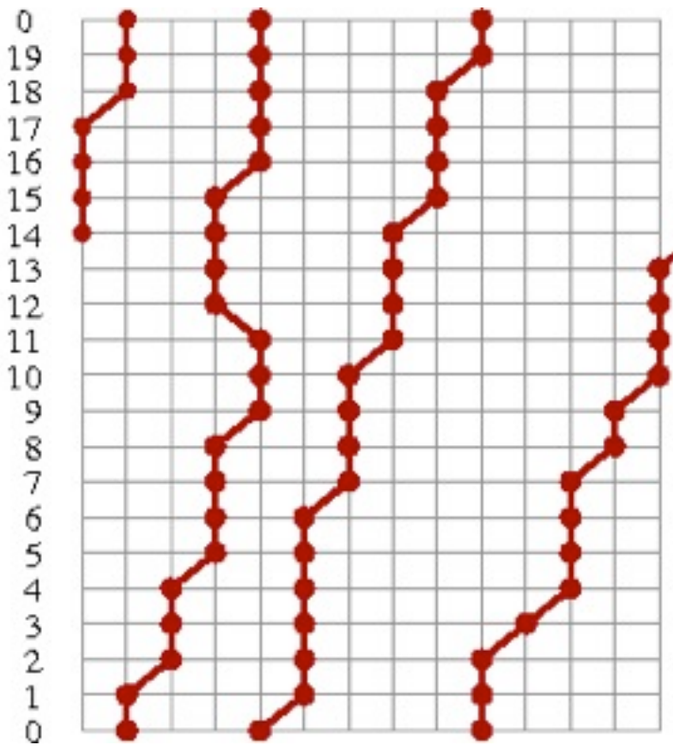
Equivalent to S=1/2 XY model

$$H = -2\sum_{\langle i,j \rangle} (S_i^x S_j^x + S_i^y S_j^y) = -\sum_{\langle i,j \rangle} (S_i^+ S_j^- + S_i^- S_j^+), \quad S^z = \pm\frac{1}{2} \sim n_i = 0, 1$$

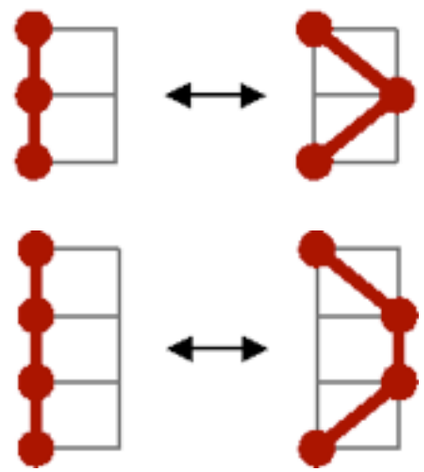
"World line" representation of

$$Z \approx \sum_{\{\alpha\}} \langle \alpha_0 | 1 - \Delta_\tau H | \alpha_{L-1} \rangle \cdots \langle \alpha_2 | 1 - \Delta_\tau H | \alpha_1 \rangle \langle \alpha_1 | 1 - \Delta_\tau H | \alpha_0 \rangle$$




world line moves for Monte Carlo sampling



$$Z = \sum_{\{\alpha\}} W(\{\alpha\}), \quad W(\{\alpha\}) = \Delta_\tau^{n_K}$$

$n_K$ = number of "jumps"

# Expectation values

$$\langle A\rangle = \frac{1}{Z}\sum_{\{\alpha\}}\langle\alpha_0|\mathrm{e}^{-\Delta_\tau}|\alpha_{L-1}\rangle\cdots\langle\alpha_2|\mathrm{e}^{-\Delta_\tau H}|\alpha_1\rangle\langle\alpha_1|\mathrm{e}^{-\Delta_\tau H}A|\alpha_0\rangle$$

We want to write this in a form suitable for MC importance sampling

$$\langle A\rangle = \frac{\sum_{\{\alpha\}} A(\{\alpha\})W(\{\alpha\})}{\sum_{\{\alpha\}} W(\{\alpha\})} \quad\longrightarrow\quad \langle A\rangle = \langle A(\{\alpha\})\rangle_W$$

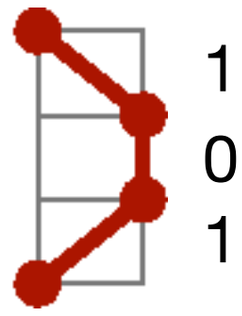
$$W(\{\alpha\}) = \text{weight}$$
$$A(\{\alpha\}) = \text{estimator}$$

For any quantity diagonal in the occupation numbers (spin z):

$$A(\{\alpha\}) = A(\alpha_n) \quad \text{or} \quad A(\{\alpha\}) = \frac{1}{L}\sum_{l=0}^{L-1} A(\alpha_l)$$

Kinetic energy (here full energy). Use

$$K\mathrm{e}^{-\Delta_\tau K} \approx K \qquad K_{ij}(\{\alpha\}) = \frac{\langle\alpha_1|K_{ij}|\alpha_0\rangle}{\langle\alpha_1|1-\Delta_\tau K|\alpha_0\rangle} \in \{0, \frac{1}{\Delta_\tau}\}$$




Average over all slices → count number of kinetic jumps

$$\langle K_{ij}\rangle = \frac{\langle n_{ij}\rangle}{\beta}, \quad \langle K\rangle = -\frac{\langle n_K\rangle}{\beta} \qquad \langle K\rangle \propto N \rightarrow \langle n_K\rangle \propto \beta N$$

**There should be of the order βN "jumps"** (regardless of approximation used)

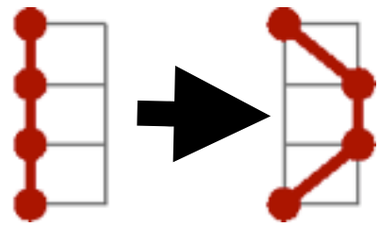
# Including interactions

For any diagonal interaction V (Trotter, or split-operator, approximation)

$$\mathrm{e}^{-\Delta_\tau H} = \mathrm{e}^{-\Delta_\tau K}\mathrm{e}^{-\Delta_\tau V} + \mathcal{O}(\Delta_\tau^2) \rightarrow \langle\alpha_{l+1}|\mathrm{e}^{-\Delta_\tau H}|\alpha_l\rangle \approx \mathrm{e}^{-\Delta_\tau V_l}\langle\alpha_{l+1}|\mathrm{e}^{-\Delta_\tau K}|\alpha_l\rangle$$

Product over all times slices →

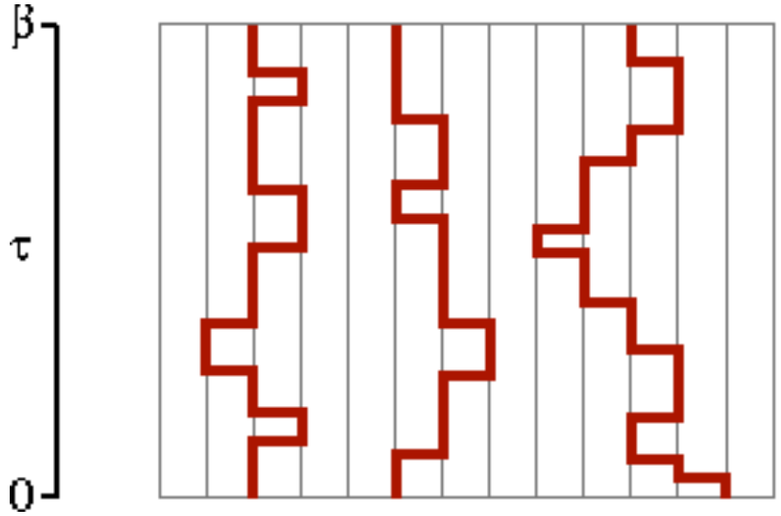
$$W(\{\alpha\}) = \Delta_\tau^{n_K}\exp\left(-\Delta_\tau\sum_{l=0}^{L-1}V_l\right)$$



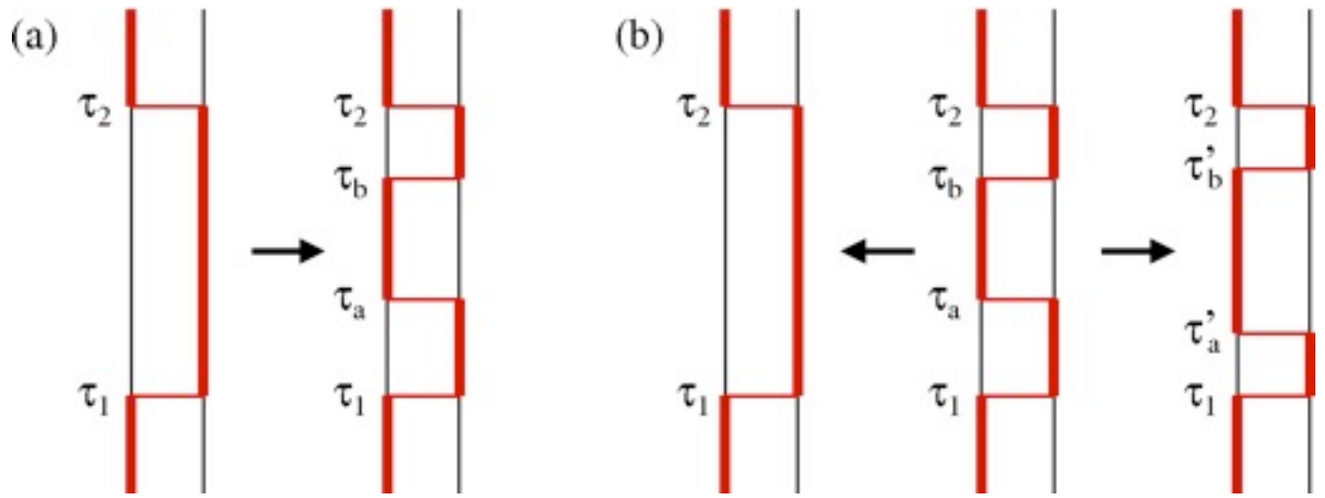
$$P_{\mathrm{acc}} = \min\left[\Delta_\tau^2\exp\left(-\frac{V_{\mathrm{new}}}{V_{\mathrm{old}}}\right),1\right]$$

# The continuous time limit

Limit $\Delta_\tau \rightarrow 0$: number of kinetic jumps remains finite, store events only




Special methods (**loop and worm updates**) developed for efficient sampling of the paths in the continuum




**local updates** (problem when $\Delta_\tau \rightarrow 0$?)

- consider probability of inserting/removing events within a time window

⇐ Evertz, Lana, Marcu (1993), Prokofev et al (1996)
Beard & Wiese (1996)

# Series expansion representation

Start from the Taylor expansion (no approximation)

$$Z = \mathrm{Tr}\{\mathrm{e}^{-\beta H}\} = \sum_{n=0}^{\infty} \frac{(-\beta)^n}{n!} \sum_{\alpha_0} \langle \alpha_0 | H^n | \alpha_0 \rangle$$

Define index sequences (string) referring to terms of H

$$H = \sum_{i=1}^{m} H_i \qquad S_n = (a_1, a_2, \ldots, a_n), \quad a_i \in \{1, \ldots, m\}$$

Break up $H^n$ into strings:

$$Z = \sum_{n=0}^{\infty} \frac{(-\beta)^n}{n!} \sum_{\alpha_0} \sum_{S_n} \langle \alpha_0 | H_{a_n} \cdots H_{a_2} H_{a_1} | \alpha_0 \rangle$$

We should have (always possible): $H_i |\alpha_j\rangle \propto |\alpha_k\rangle$

- no branching during propagation with operator string
- some strings not allowed (illegal operations)

For hard-core bosons the (allowed) path weight is: $W(S_n, \alpha_0) = \dfrac{\beta^n}{n!}$

We can make this look more similar to a path integral by introducing partially propagated states: $|\alpha_p\rangle = H_{a_p} \cdots H_{a_2} H_{a_1} |\alpha_0\rangle$

$$Z = \sum_{n=0}^{\infty} \frac{(-\beta)^n}{n!} \sum_{\alpha_0} \sum_{S_n} \langle \alpha_0 | H_{a_n} | \alpha_{n-1} \rangle \langle \alpha_{n-2} | \cdots | \alpha_2 \rangle \langle \alpha_1 | H_{a_1} | \alpha_0 \rangle$$

$$|\alpha_n\rangle = |\alpha_0\rangle$$

Same-looking paths, different-looking weights
- but become equivalent with time continuum in path integral

**Energy:** $$\langle H \rangle = \frac{1}{Z} \sum_{n=0}^{\infty} \frac{(-\beta)^n}{n!} \sum_{\alpha_0} \langle \alpha_0 | H^n H | \alpha_0 \rangle$$

Relabel terms of n-sum: replace n+1 by n

$$\langle H \rangle = -\frac{1}{Z} \sum_{n=1}^{\infty} \frac{(-\beta)^n}{n!} \frac{n}{\beta} \sum_{\alpha_0} \langle \alpha_0 | H^n | \alpha_0 \rangle$$

Therefore the energy is: $E = -\langle n \rangle / \beta$

Can also derive specific heat: $C = \langle n^2 \rangle - \langle n \rangle^2 - \langle n \rangle$

Follows: $\langle n \rangle \propto \beta N, \quad \sigma_n \propto \sqrt{\beta N}$

## Fixed string-length scheme

- n fluctuating → varying size of the sampled configurations
- the expansion can be truncated at some $n_{max}$=L
  (exponentially small error if large enough)
- cutt-off at n=L, fill in operator string with unit operators **$H_0$=I**

n=10 | $H_4$ | $H_7$ | $H_1$ | $H_6$ | $H_2$ | $H_1$ | $H_8$ | $H_3$ | $H_3$ | $H_5$ | $\Longrightarrow$

M=14 | $H_4$ | I | $H_7$ | I | $H_1$ | $H_6$ | I | $H_2$ | $H_1$ | $H_8$ | $H_3$ | $H_3$ | I | $H_5$ |

- conisider all possible locations in the sequence
- overcounting of original strings, correct by $\binom{L}{n}^{-1} = \frac{n!(L-n)!}{L!}$

$$Z = \sum_{\alpha_0} \sum_{S_L} \frac{(-\beta)^n (L-n)!}{L!} \langle \alpha_0 | H_{a_m} \cdots H_{a_2} H_{a_1} | \alpha_0 \rangle$$

Here n is the number of $H_i$, i>0 instances in the sequence of L ops
- the summation over n is now implicit

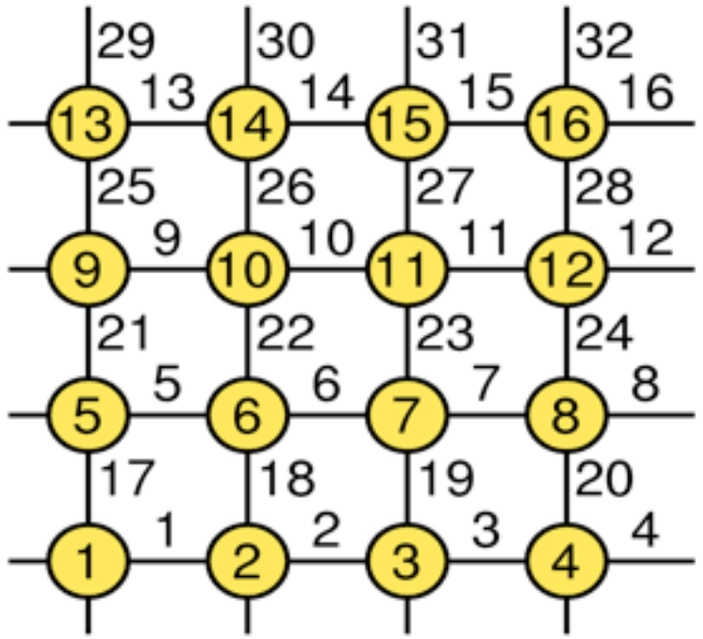
**L can be chosen automatically by the simulation**

# Stochastic Series expansion (SSE): S=1/2 Heisenberg model

Write H as a bond sum for arbitrary lattice

$$H = J \sum_{b=1}^{N_b} \mathbf{S}_{i(b)} \cdot \mathbf{S}_{j(b)},$$

2D square lattice
bond and site labels




Diagonal (1) and off-diagonal (2) bond operators

$$H_{1,b} = \frac{1}{4} - S^z_{i(b)} S^z_{j(b)},$$

$$H_{2,b} = \frac{1}{2}(S^+_{i(b)} S^-_{j(b)} + S^-_{i(b)} S^+_{j(b)}).$$

$$H = -J \sum_{b=1}^{N_b} (H_{1,b} - H_{2,b}) + \frac{J N_b}{4}$$

Four non-zero matrix elements

$$\langle \uparrow_{i(b)} \downarrow_{j(b)} | H_{1,b} | \uparrow_{i(b)} \downarrow_{j(b)} \rangle = \frac{1}{2} \qquad \langle \downarrow_{i(b)} \uparrow_{j(b)} | H_{2,b} | \uparrow_{i(b)} \downarrow_{j(b)} \rangle = \frac{1}{2}$$

$$\langle \downarrow_{i(b)} \uparrow_{j(b)} | H_{1,b} | \downarrow_{i(b)} \uparrow_{j(b)} \rangle = \frac{1}{2} \qquad \langle \uparrow_{i(b)} \downarrow_{j(b)} | H_{2,b} | \downarrow_{i(b)} \uparrow_{j(b)} \rangle = \frac{1}{2}$$

Partition function

$$Z = \sum_{\alpha} \sum_{n=0}^{\infty} (-1)^{n_2} \frac{\beta^n}{n!} \sum_{S_n} \left\langle \alpha \left| \prod_{p=0}^{n-1} H_{a(p),b(p)} \right| \alpha \right\rangle$$

$n_2$ = number of a(i)=2
(off-diagonal operators)
in the sequence

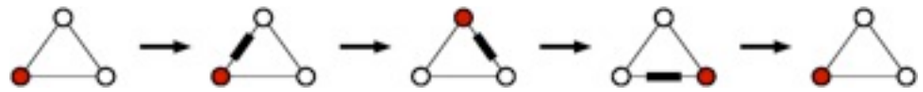
Index sequence: $S_n = [a(0), b(0)], [a(1), b(1)], \ldots, [a(n-1), b(n-1)]$

For fixed-length scheme

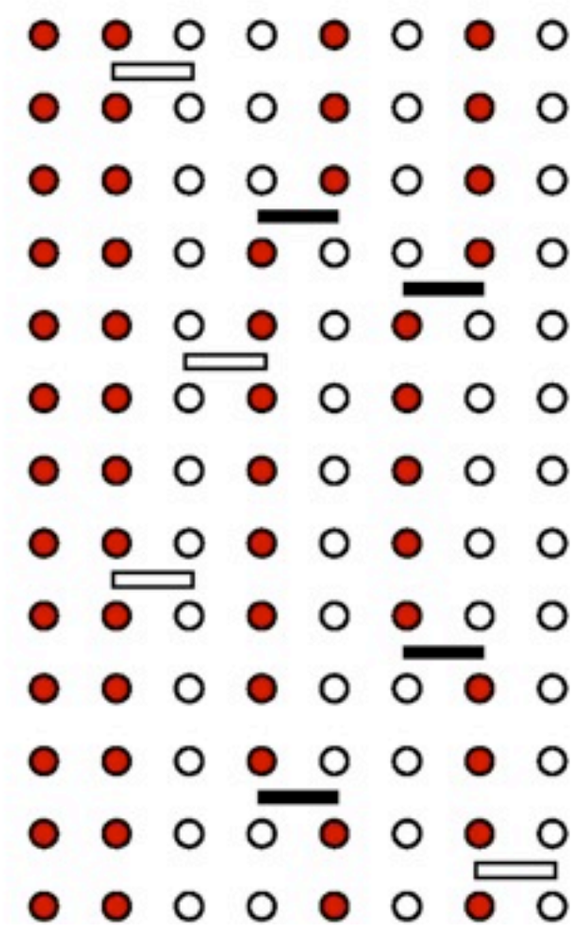
$$Z = \sum_{\alpha} \sum_{S_L} (-1)^{n_2} \frac{\beta^n (L-n)!}{L!} \left\langle \alpha \left| \prod_{p=0}^{L-1} H_{a(p),b(p)} \right| \alpha \right\rangle \qquad W(\alpha, S_L) = \left(\frac{\beta}{2}\right)^n \frac{(L-n)!}{L!}$$

Propagated states: $|\alpha(p)\rangle \propto \prod_{i=0}^{p-1} H_{a(i),b(i)} |\alpha\rangle$

W>0 ($n_2$ even) for bipartite lattice
Frustration leads to **sign problem**



| $i$ = | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| $\sigma(i)$ = | -1 | +1 | -1 | -1 | +1 | -1 | +1 | +1 |



| $p$ | $a(p)$ | $b(p)$ | $s(p)$ |
|---|---|---|---|
| 11 | 1 | 2 | 4 |
| 10 | 0 | 0 | 0 |
| 9 | 2 | 4 | 9 |
| 8 | 2 | 6 | 13 |
| 7 | 1 | 3 | 6 |
| 6 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 |
| 4 | 1 | 2 | 4 |
| 3 | 2 | 6 | 13 |
| 2 | 0 | 0 | 0 |
| 1 | 2 | 4 | 9 |
| 0 | 1 | 7 | 14 |

**In a program:**

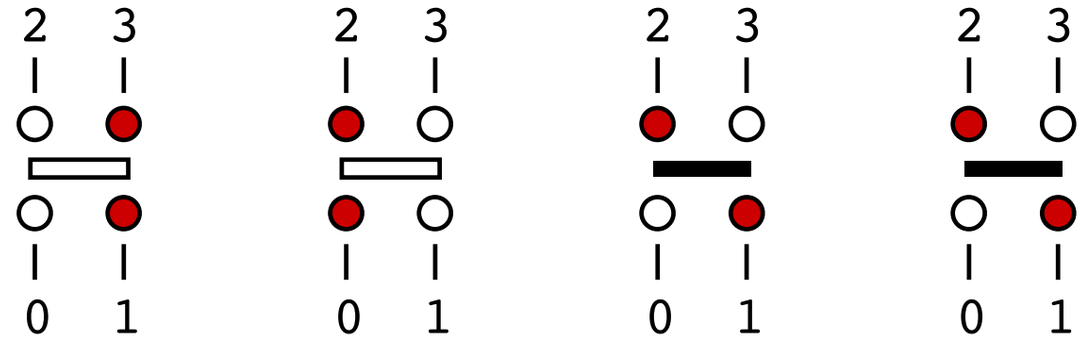
s(p) = operator-index string
- **s(p) = 2*b(p) + a(p)-1**
- diagonal; s(p) = even
- off-diagonal; s(p) = off

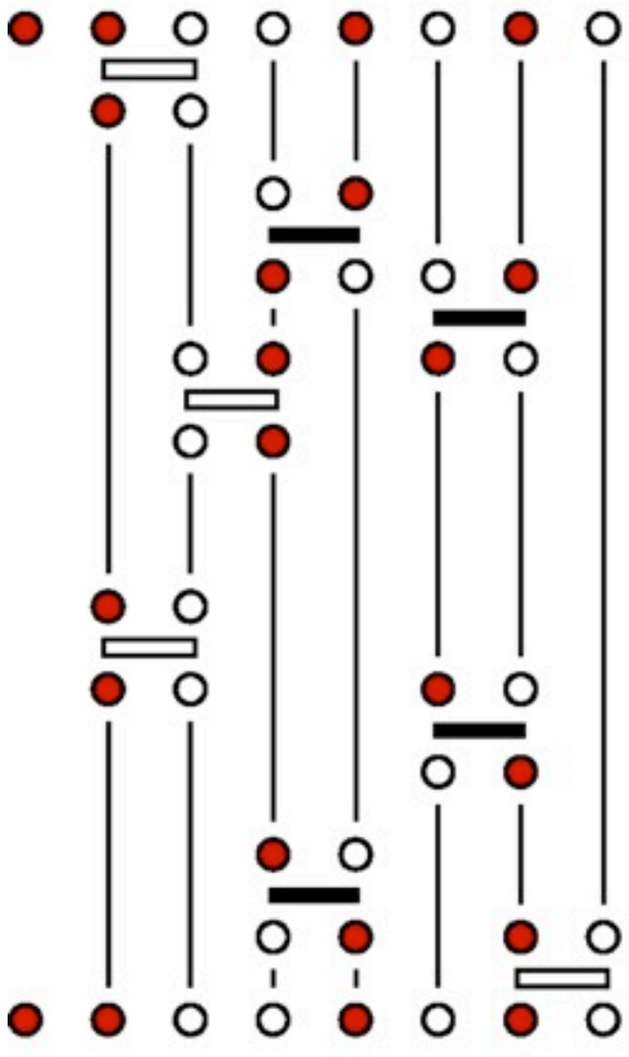
σ(i) = spin state, i=1,...,N
- only one has to be stored

**SSE effectively provides a discrete representation of the time continuum**
- computational advantage; only integer operations in sampling

# Linked vertex storage

The “legs” of a vertex represents the spin states before (below) and after (above) an operator has acted



| p | v X(v) | | v X(v) | | v X(v) | | v X(v) | |
|---|---|---|---|---|---|---|---|---|
| 11 | 44 | 18 | 45 | 30 | 46 | 16 | 47 | 17 |
| 10 | 40 | - | 41 | - | 42 | - | 43 | - |
| 9 | 36 | 31 | 37 | 7 | 38 | 4 | 39 | 5 |
| 8 | 32 | 14 | 33 | 15 | 34 | 12 | 35 | 0 |
| 7 | 28 | 19 | 29 | 6 | 30 | 45 | 31 | 36 |
| 6 | 24 | - | 25 | - | 26 | - | 27 | - |
| 5 | 20 | - | 21 | - | 22 | - | 23 | - |
| 4 | 16 | 46 | 17 | 47 | 18 | 44 | 19 | 28 |
| 3 | 12 | 34 | 13 | 2 | 14 | 32 | 15 | 33 |
| 2 | 8 | - | 9 | - | 10 | - | 11 | - |
| 1 | 4 | 38 | 5 | 39 | 6 | 29 | 7 | 37 |
| 0 | 0 | 35 | 1 | 3 | 2 | 13 | 3 | 1 |
| | l=0 | | l=1 | | l=2 | | l=3 | |

X( ) = vertex list

- operator at p→X(v)
  **v=4p+l, l=0,1,2,3**
- links to next and previous leg

Spin states between operations are redundant; represented by links

- network of linked vertices will be used for loop updates of vertices/operators

## Monte Carlo sampling scheme

Change the configuration; $(\alpha, S_L) \to (\alpha', S'_L)$

$$W(\alpha, S_L) = \left(\frac{\beta}{2}\right)^n \frac{(L-n)!}{L!}$$

$$P_{\text{accept}} = \min\left[\frac{W(\alpha', S_L)}{W(\alpha, S_L)} \frac{P_{\text{select}}(\alpha', S'_L \to \alpha, S_L)}{P_{\text{select}}(\alpha, S_L \to \alpha', S'_L)}, 1\right]$$

**Diagonal update:** $[0,0]_p \leftrightarrow [1,b]_p$

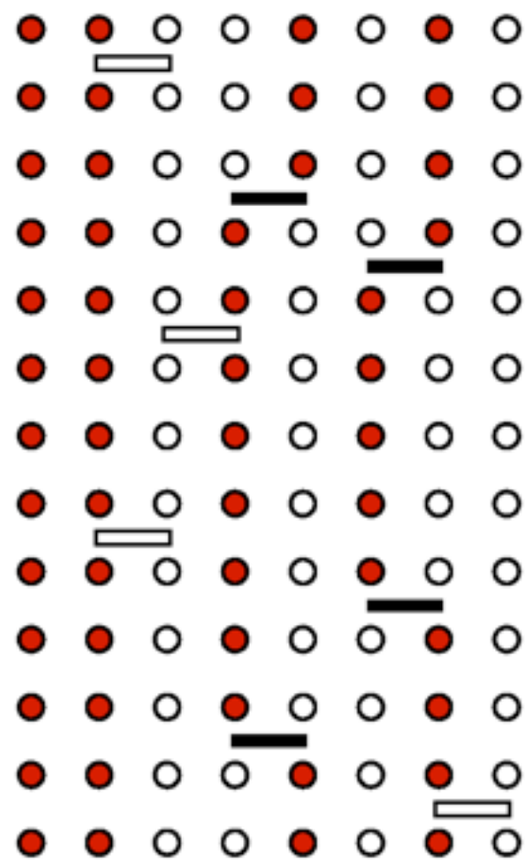

Attempt at p=0,...,L-1. Need to know |α(p)>

- generate by flipping spins when off-diagonal operator

$$P_{\text{select}}(a = 0 \to a = 1) = 1/N_b, \quad (b \in \{1, \ldots, N_b\})$$

$$P_{\text{select}}(a = 1 \to a = 0) = 1$$

$$\frac{W(a=1)}{W(a=0)} = \frac{\beta/2}{L-n} \qquad \frac{W(a=0)}{W(a=1)} = \frac{L-n+1}{\beta/2}$$



n is the current power

- n → n+1 (a=0 → a=1)
- n → n-1 (a=1 → a=0)

## Acceptance probabilities

$$P_{\text{accept}}([0,0] \to [1,b]) = \min\left[\frac{\beta N_b}{2(L-n)}, 1\right]$$

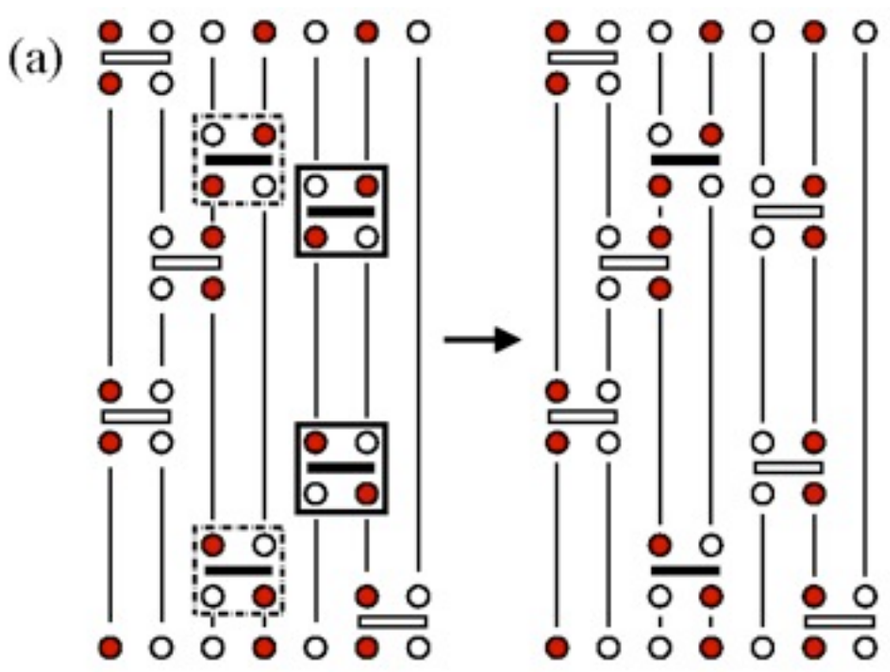
$$P_{\text{accept}}([1,b] \to [0,0]) = \min\left[\frac{2(L-n+1)}{\beta N_b}, 1\right]$$

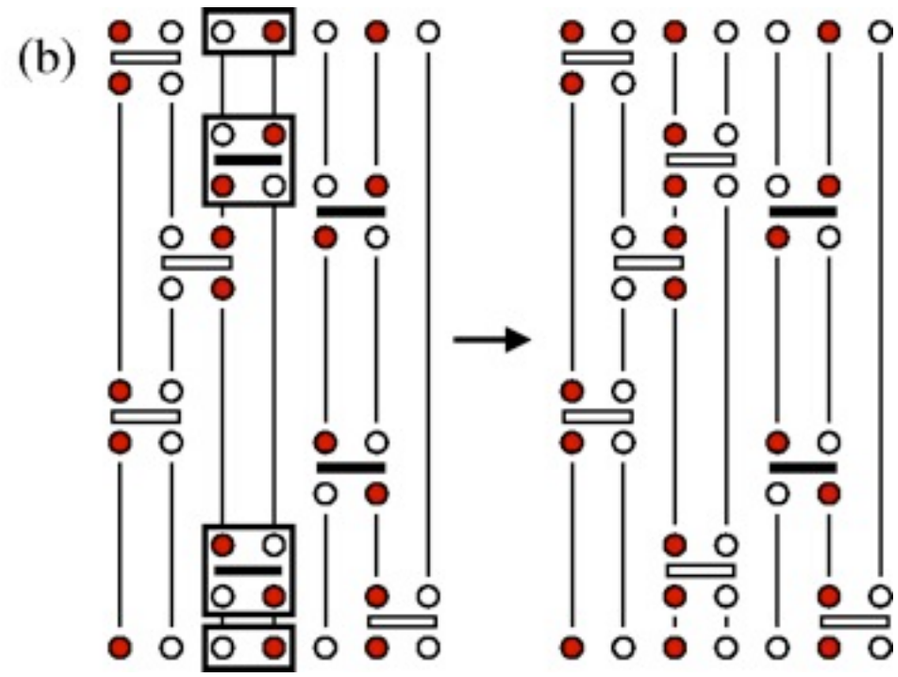
## Pseudocode: Sweep of diagonal updates

**do** $p = 0$ **to** $L - 1$
  **if** $(s(p) = 0)$ **then**
    $b =$ **random**$[1, \ldots, N_b]$
    **if** $\sigma(i(b)) = \sigma(j(b))$ **cycle**
    **if** (**random**$[0 - 1] < P_{\text{insert}}(n)$) **then** $s(p) = 2b;\ n = n + 1$ **endif**
  **elseif** (**mod**$[s(p), 2] = 0$) **then**
    **if** (**random**$[0 - 1] < P_{\text{remove}}(n)$) **then** $s(p) = 0;\ n = n - 1$ **endif**
  **else**
    $b = s(p)/2;\ \sigma(i(b)) = -\sigma(i(b));\ \sigma(j(b)) = -\sigma(j(b))$
  **endif**
**enddo**

- To insert operator, bond b generated at random among 1,...,$N_b$
  - can be done only if connected spins i(b),j(b) are anti-parallel
  - if so, do it with probability $P_{insert}(n)$
- Existing diagonal operator can always be removed
  - do it with probability $P_{remove}(n)$
- If off-diagonal operator, advance the state
  - extract bond b, flip spins at i(b),j(b)

# Off-diagonal updates

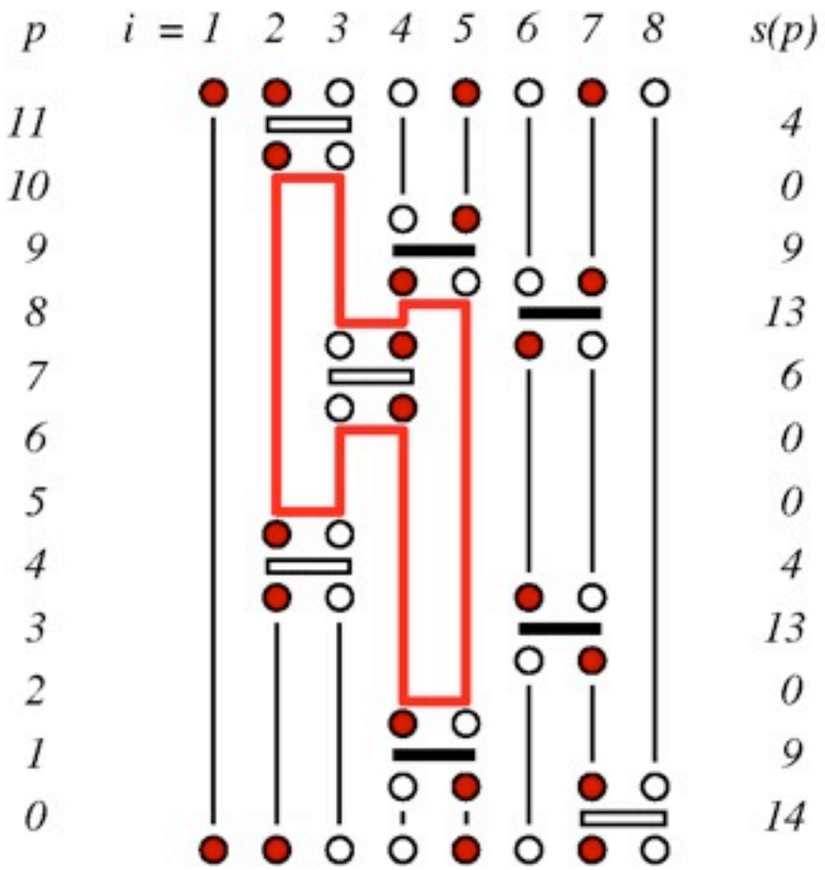


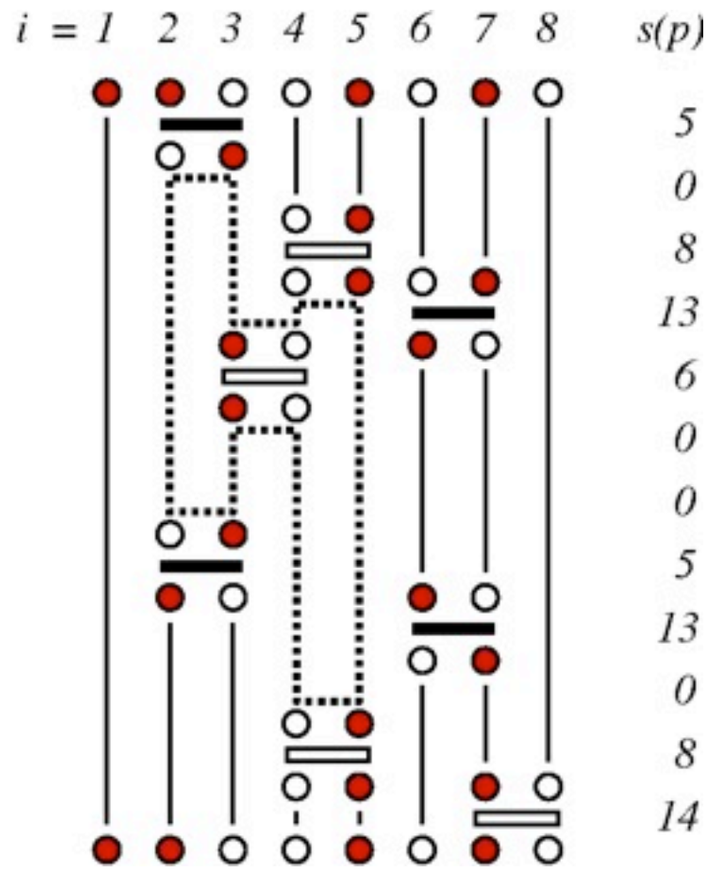




**Local update**

Change the type of two operators

- constraints
- inefficient
- cannot change winding numbers




**Operator-loop update**

- Many spins and operators can be changed simultaneously
- can change winding numbers

# Pseudocode: Sweep of loop updates

constructing all loops, flip probability 1/2

```
do v_0 = 0 to 4L − 1 step 2
    if (X(v_0) < 0) cycle
    v = v_0
    if (random[0 − 1] < 1/2) then
        traverse the loop; for all v in loop, set X(v) = −1
    else
        traverse the loop; for all v in loop, set X(v) = −2
        flip the operators in the loop
    endif
enddo
```

→

construct and flip a loop

```
v = v_0
do
    X(v) = −2
    p = v/4; s(p) = flipbit(s(p), 0)
    v' = flipbit(v, 0)
    v = X(v'); X(v') = −2
    if (v = v_0) exit
enddo
```

- by flipping bit 0 of s(p), the operator changes from diagonal to off-diagonal, or vise versa
- moving on the vertex to the adjacent spin is also done with a bit flip

## We also have to modify the stored spin state after the loop update

- we can use the information in $V_{\text{first}}()$ and X() to determine spins to be flipped
- spins with no operators, $V_{\text{first}}(i)=-1$, flipped with probability 1/2

**do** $i = 1$ **to** $N$
  $v = V_{\text{first}}(i)$
  **if** $(v = -1)$ **then**
    **if** (**random**[0-1]$< 1/2$) $\sigma(i) = -\sigma(i)$
  **else**
    **if** $(X(v) = -2)$ $\sigma(i) = -\sigma(i)$
  **endif**
**enddo**

**$v=V_{\text{first}}(i)$** is the location of the first vertex leg on site i

- flip the spin if X(v)=–2
- (do not flip it if X(v)=–1)
- no operation on i if $v_{\text{first}}(i)=-1$; then it is flipped with probability 1/2

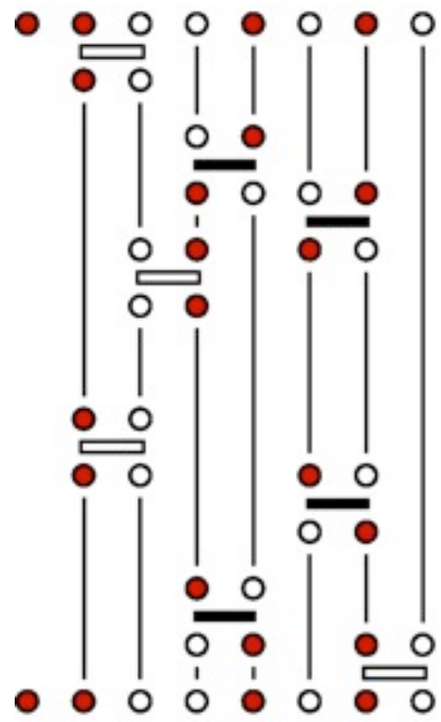
## Constructing the linked vertex list

Traverse operator list $s(p)$, $p$=0,...,$L$–1

- vertex legs **v=4p,4p+1,4p+2,4p+3**

Use arrays to keep track of the first and last (previous) vertex leg on a given spin

- $V_{first}(i)$ = location v of first leg on site i
- $V_{last}(i)$ = location v of last (currently) leg
- these are used to create the links
- initialize all elements to –1



| p | v | X(v) | v | X(v) | v | X(v) | v | X(v) |
|---|---|---|---|---|---|---|---|---|
| 11 | 44 | 18 | 45 | 30 | 46 | 16 | 47 | 17 |
| 10 | 40 | - | 41 | - | 42 | - | 43 | - |
| 9 | 36 | 31 | 37 | 7 | 38 | 4 | 39 | 5 |
| 8 | 32 | 14 | 33 | 15 | 34 | 12 | 35 | 0 |
| 7 | 28 | 19 | 29 | 6 | 30 | 45 | 31 | 36 |
| 6 | 24 | - | 25 | - | 26 | - | 27 | - |
| 5 | 20 | - | 21 | - | 22 | - | 23 | - |
| 4 | 16 | 46 | 17 | 47 | 18 | 44 | 19 | 28 |
| 3 | 12 | 34 | 13 | 2 | 14 | 32 | 15 | 33 |
| 2 | 8 | - | 9 | - | 10 | - | 11 | - |
| 1 | 4 | 38 | 5 | 39 | 6 | 29 | 7 | 37 |
| 0 | 0 | 35 | 1 | 3 | 2 | 13 | 3 | 1 |
| | l=0 | | l=1 | | l=2 | | l=3 | |

$V_{\rm first}(:) = -1;\ V_{\rm last}(:) = -1$
**do** $p = 0$ **to** $L-1$
  **if** $(s(p) = 0)$ **cycle**
  $v_0 = 4p;\ b = s(p)/2;\ s_1 = i(b);\ s_2 = j(b)$
  $v_1 = V_{\rm last}(s_1);\ v_2 = V_{\rm last}(s_2)$
  **if** $(v_1 \neq -1)$ **then** $X(v_1) = v_0;\ X(v_0) = v_1$ **else** $V_{\rm first}(s_1) = v_0$ **endif**
  **if** $(v_2 \neq -1)$ **then** $X(v_2) = v_0;\ X(v_0) = v_2$ **else** $V_{\rm first}(s_2) = v_0 + 1$ **endif**
  $V_{\rm last}(s_1) = v_0 + 2;\ V_{\rm last}(s_2) = v_0 + 3$
**enddo**

creating the last links across the “time” boundary

**do** $i = 1$ **to** $N$
  $f = V_{\rm first}(i)$
  **if** $(f \neq -1)$ **then** $l = V_{\rm last}(i);\ X(f) = l;\ X(l) = f$ **endif**
**enddo**

## Determination of the cut-off L

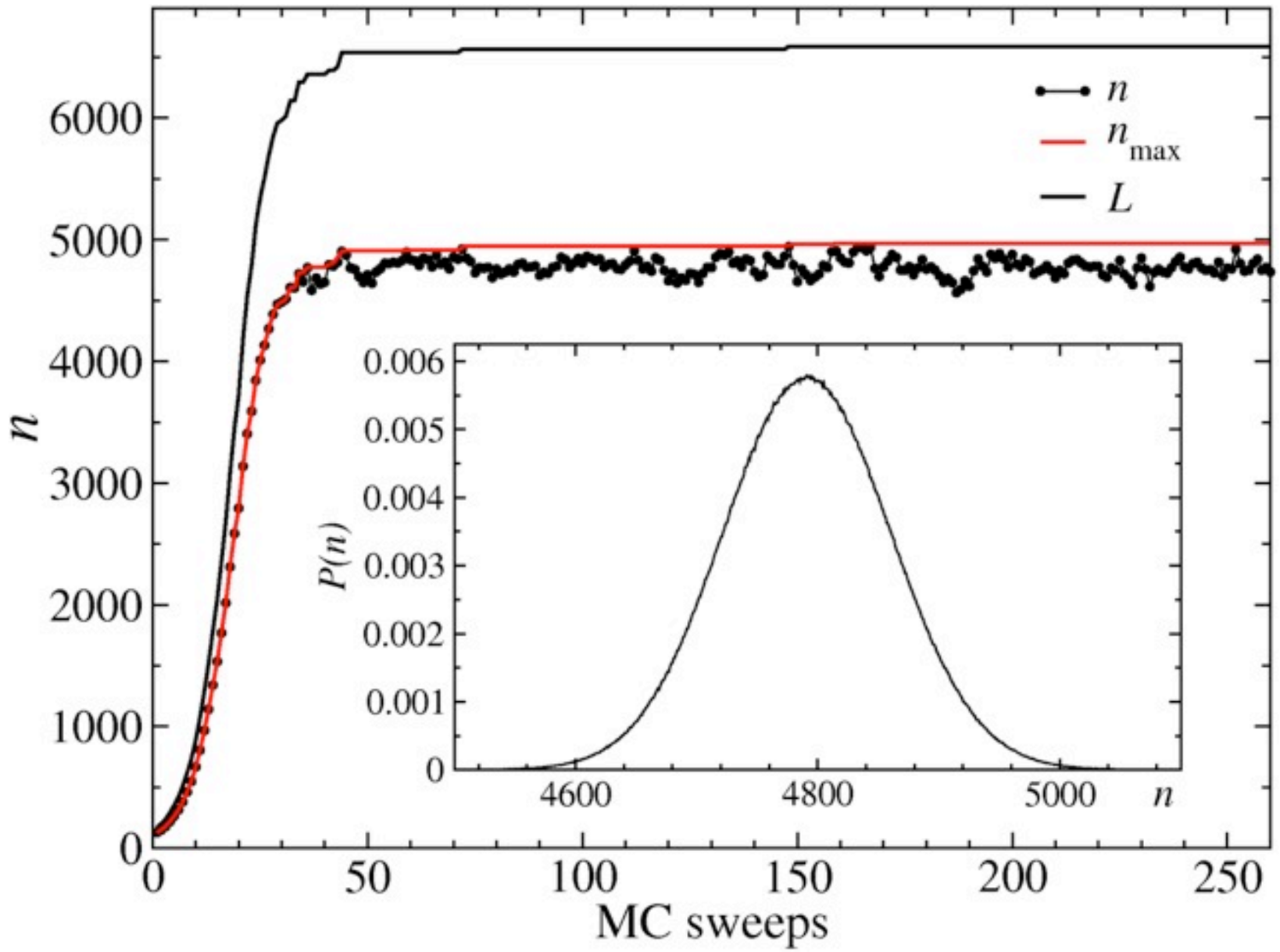
- adjust during equilibration
- start with arbitrary (small) n

Keep track of number of operators n

- increase L if n is close to current L
- e.g., ***L=n+n/3***

**Example**

- 16×16 system, β=16 ⇒
- evolution of L
- n distribution after equilibration
- truncation is no approximation

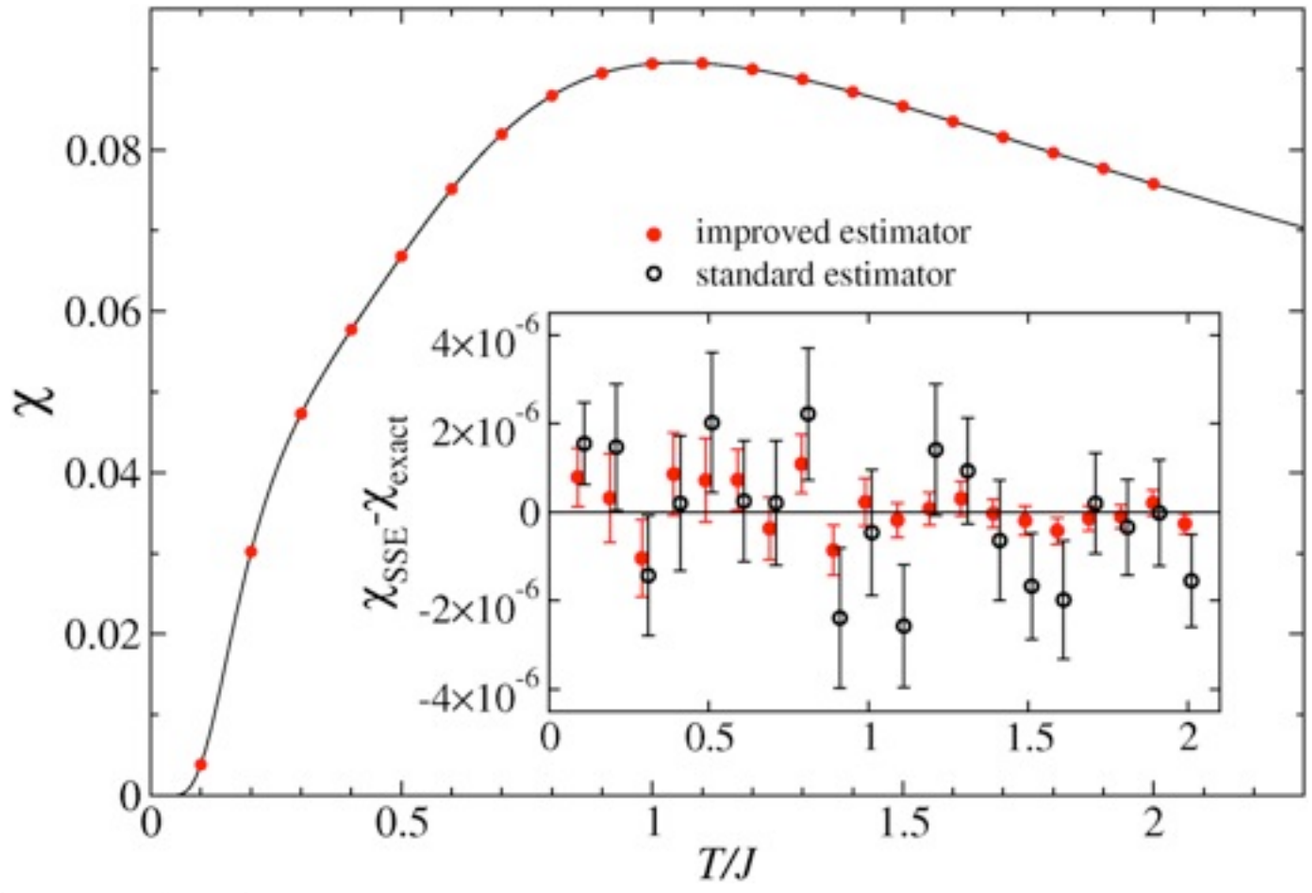


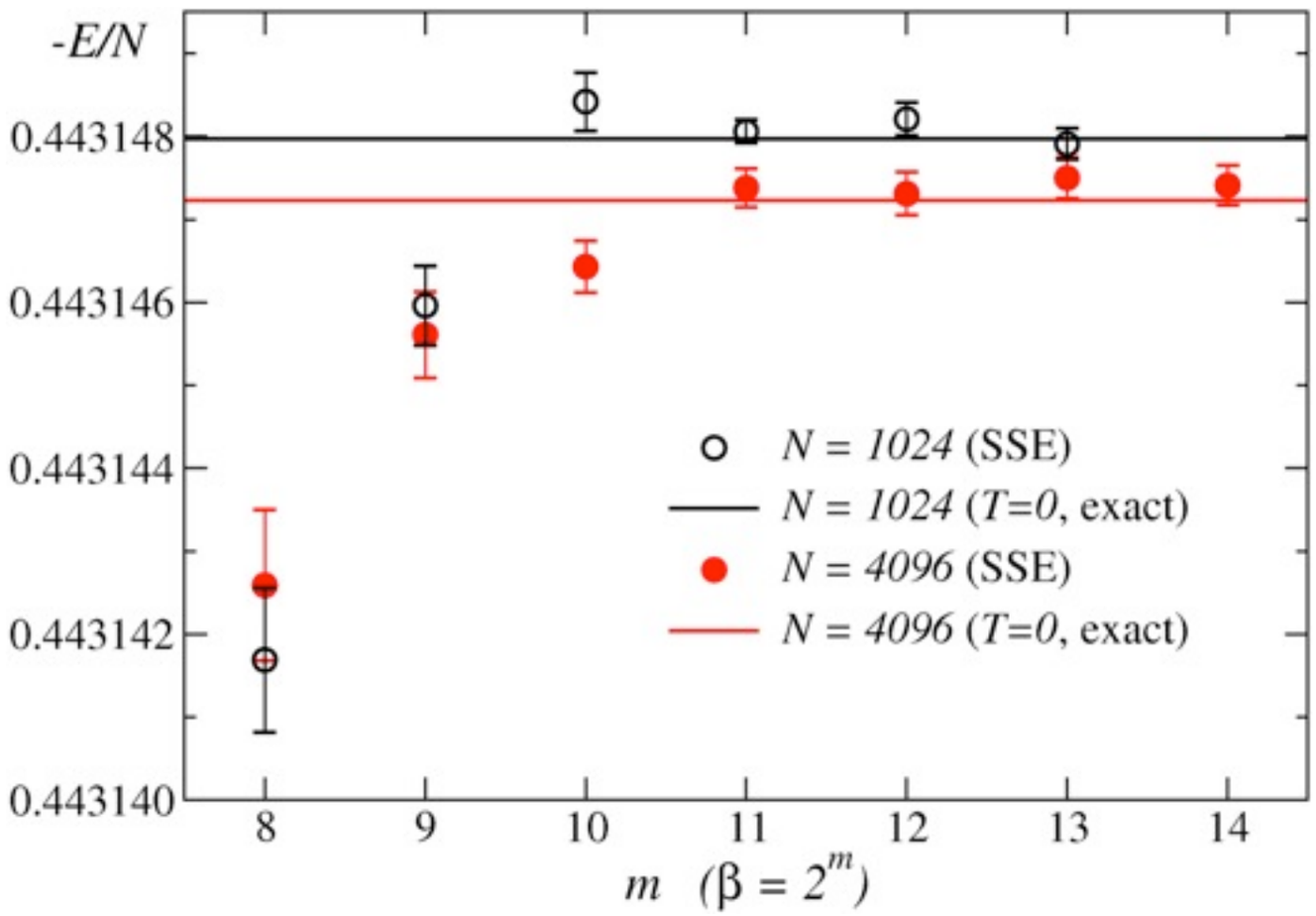
**Does it work?**
**Compare with exact results**

- 4×4 exact diagonalization
- Bethe Ansatz; long chains

**Susceptibility of the 4×4 lattice ⇒**

- SSE results from $10^{10}$ sweeps
- improved estimator gives smaller error bars at high T (where the number of loops is larger)




**⇐ Energy for long 1D chains**

- SSE results for $10^6$ sweeps
- Bethe Ansatz ground state E/N
- SSE can achieve the ground state limit (T→0)

**Basic SSE code (Fortran90) available on-line:**

**https://physics.bu.edu/~sandvik/trieste15/**

Simulation of the 2D Heisenberg model with N=Lx*Ly spins
- periodic boundary conditions
- Ly=1 for a chain (only x-periodic)
- Ly=2 for a 2-leg ladder (only x-periodic)

Calculates:
- energy and specific heat per site
- uniform magnetic susceptibility
- squared sublattice magnetization