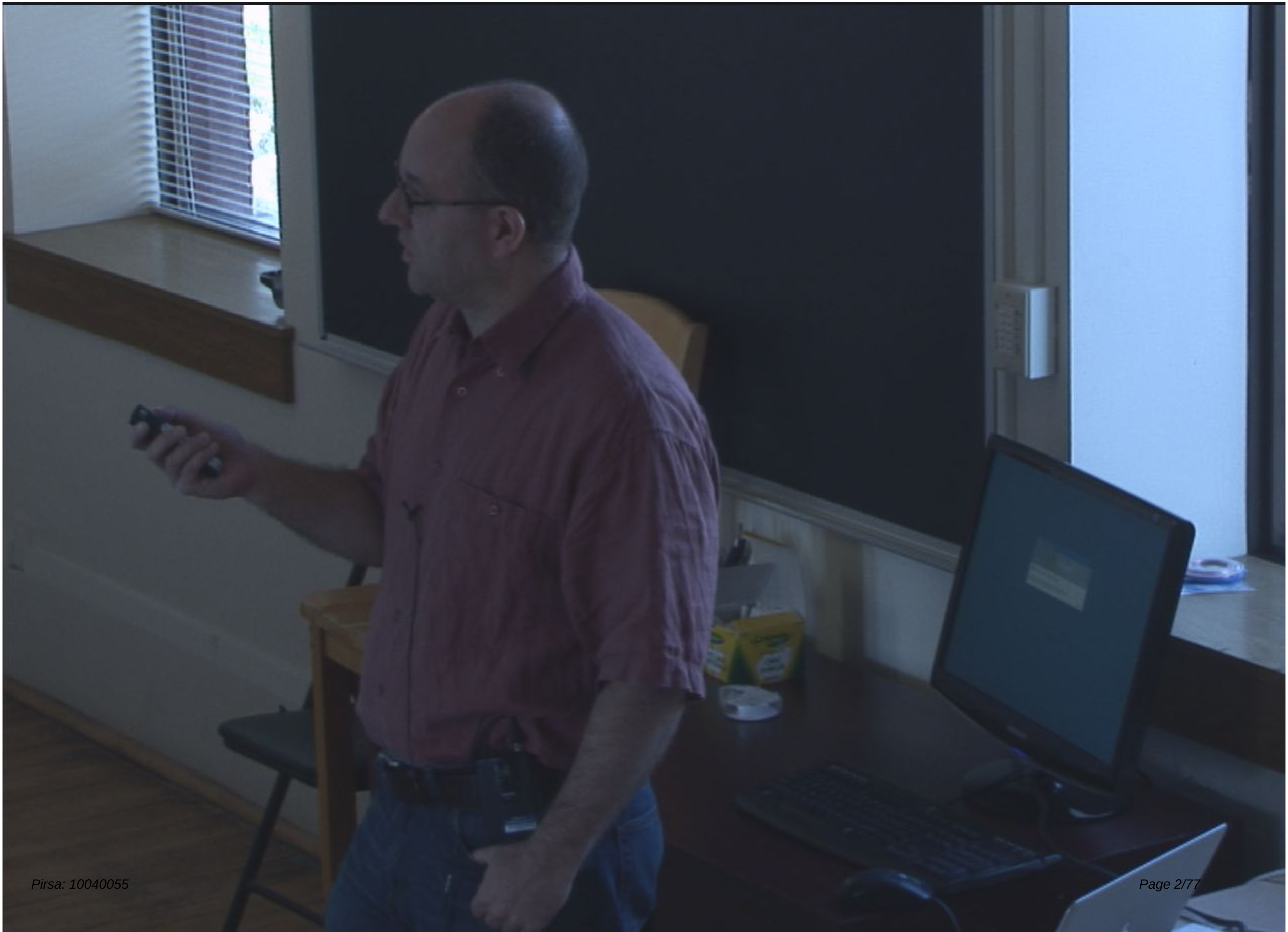
Title: Quantum Spin Simulations (PHYS 7380) - Lecture 13

Date: Apr 21, 2010 11:00 AM

URL: http://pirsa.org/10040055

Abstract:

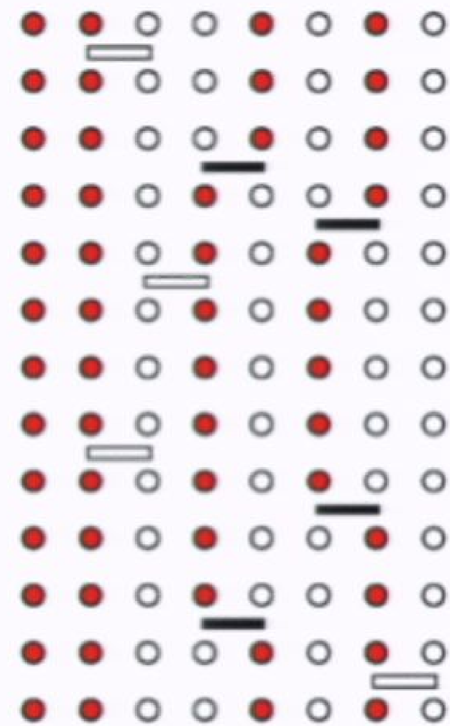


Monte Carlo sampling scheme

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

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

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

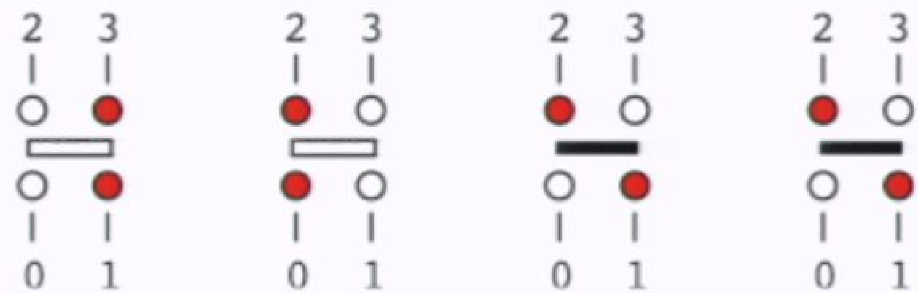


Diagonal update: $[0,0]_p \leftrightarrow [1,b]_p$

Attempt at p=0,...,L-1. Need to know |α(p)>
- generate by flipping spins when off-diagonal operator

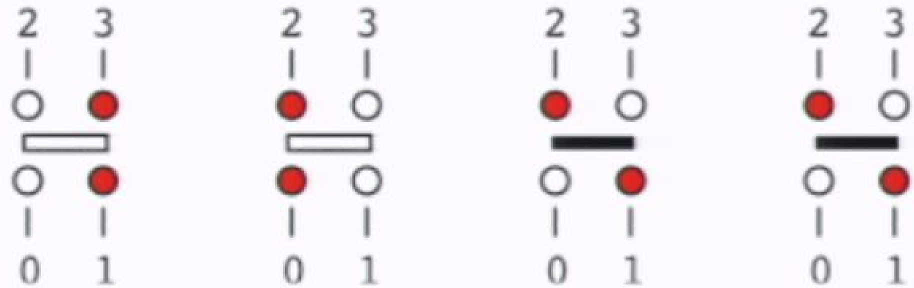
Linked vertex storage

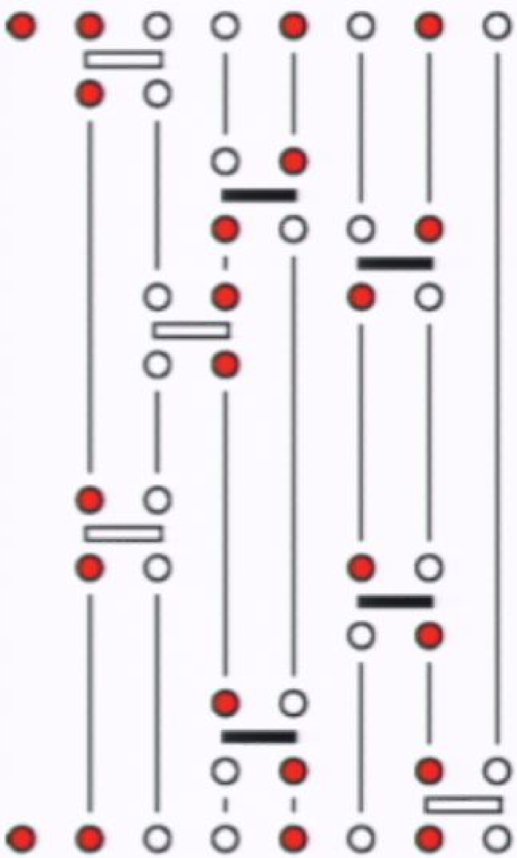
The "legs" of a vertex represents the spin states before (below) and after (above) an operator has acted



Linked vertex storage

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



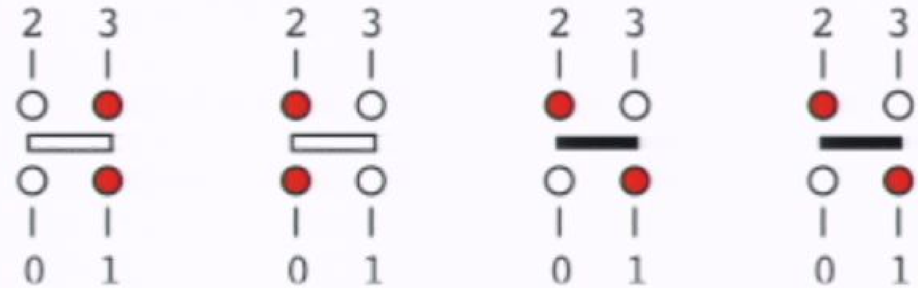


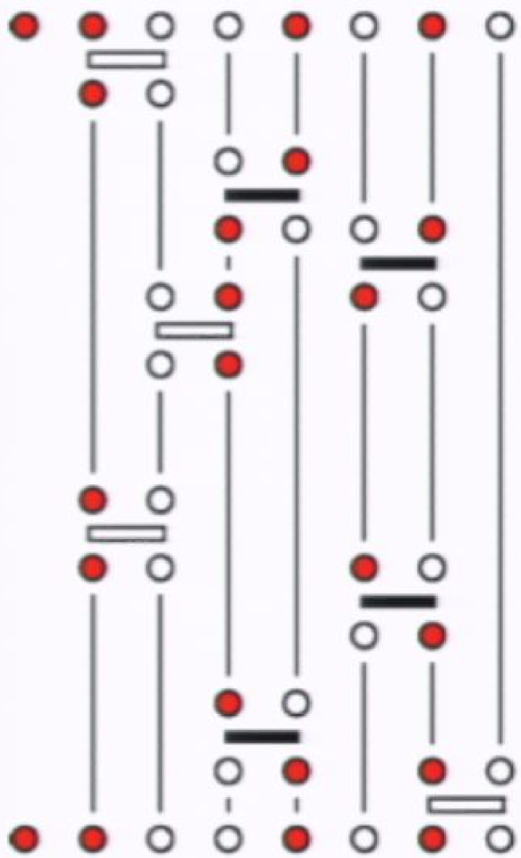
p	v	X(v)	v	X(v)	v	X(v)	v	X(v)
11	44	18	45	30	46	16	47	17
10	40	-	41	-	42	-	43	-
9	36	31	37	7	38	4	39	5
8	32	14	31	15	34	12	35	0
7	28	19	29	6	30	45	31	36
6	24	-	25	-	26	-	27	-
5	20	-	21	-	22	-	23	-
4	16	46	17	47	18	44	19	28
3	12	34	13	2	14	32	15	33
2	8	-	9	-	10	-	11	-
1	4	38	5	39	6	29	7	37
0	0	35	1	3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

Spin states between operations are redundant; represented by links

Linked vertex storage

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





p	v (l=0)	X(v) (l=0)	v (l=1)	X(v) (l=1)	v (l=2)	X(v) (l=2)	v (l=3)	X(v) (l=3)
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	31	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

X( ) = vertex list

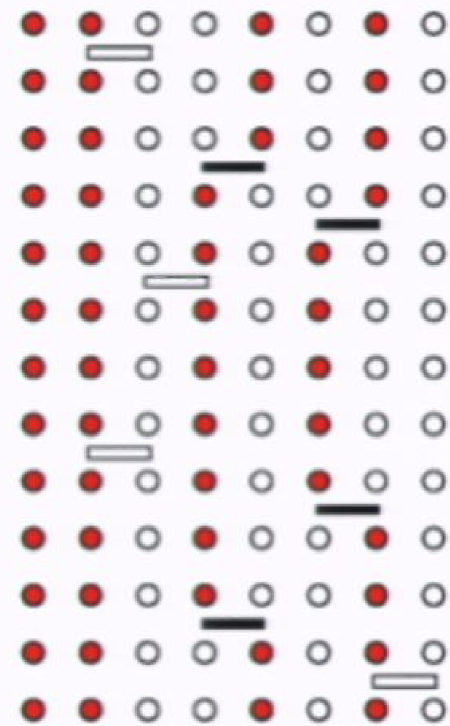
- operator at p→X(v)
  v=4p+l, l=0,1,2,3
- links to next and previous leg

Spin states between operations are redundant; represented by links

Monte Carlo sampling scheme

Change the configuration; $(\alpha, S_L) \to (\alpha', S'_L)$ $\qquad W(\alpha, S_L) = \left(\frac{\beta}{2}\right)^n \frac{(L-n)!}{L!}$

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



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

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

Diagonal update; pseudocode implementation

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

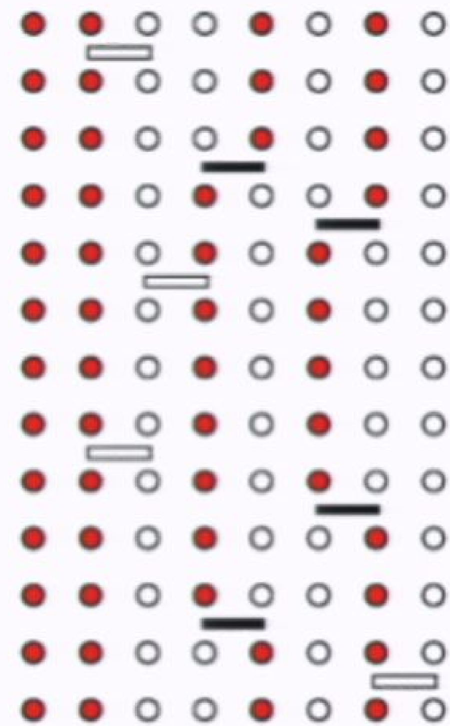
i(b),j(b) sites on bond b

Monte Carlo sampling scheme

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

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

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

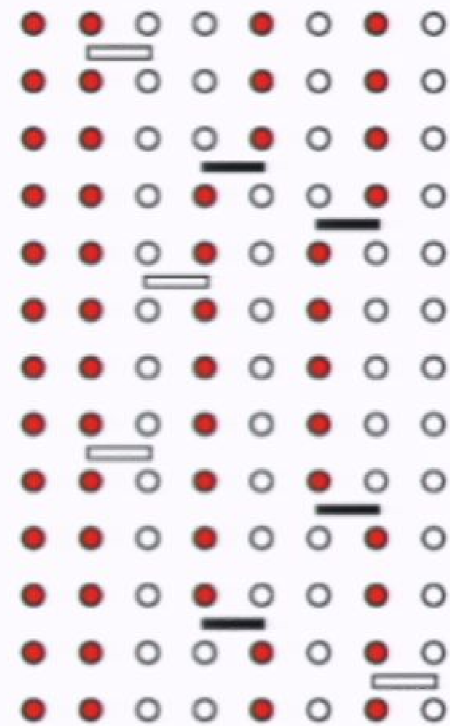


Monte Carlo sampling scheme

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

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

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



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

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

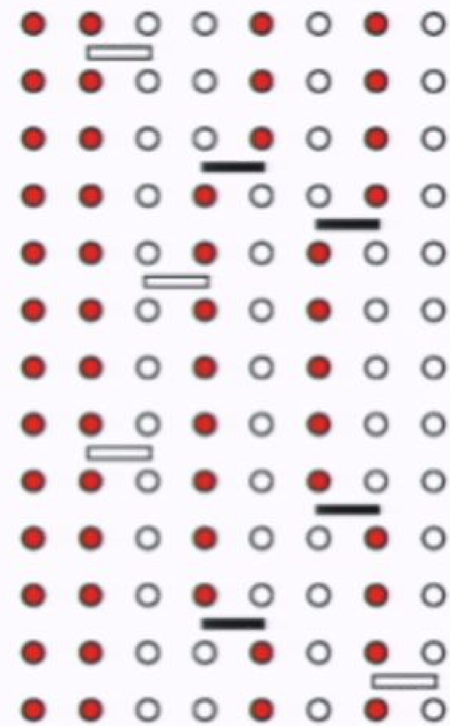
- generate by flipping spins when off-diagonal operator

Monte Carlo sampling scheme

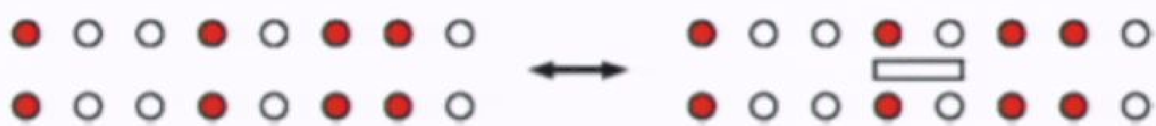
$$W(\alpha, S_L) = \left(\frac{\beta}{2}\right)^n \frac{(L-n)!}{L!}$$

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

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



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

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

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

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

n is the current power

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

Diagonal update; pseudocode implementation

```
do p = 0 to L − 1
    if (s(p) = 0) then
        b = random[1, ..., N_b]; if σ(i(b)) = σ(j(b)) cycle
        if (random[0 − 1] < P_insert(n)) then s(p) = 2b; n = n + 1 endif
    elseif (mod[s(p), 2] = 0) then
        if (random[0 − 1] < P_remove(n)) then s(p) = 0; n = n − 1 endif
    else
        b = s(p)/2; σ(i(b)) = −σ(i(b)); σ(j(b)) = −σ(j(b))
    endif
enddo
```

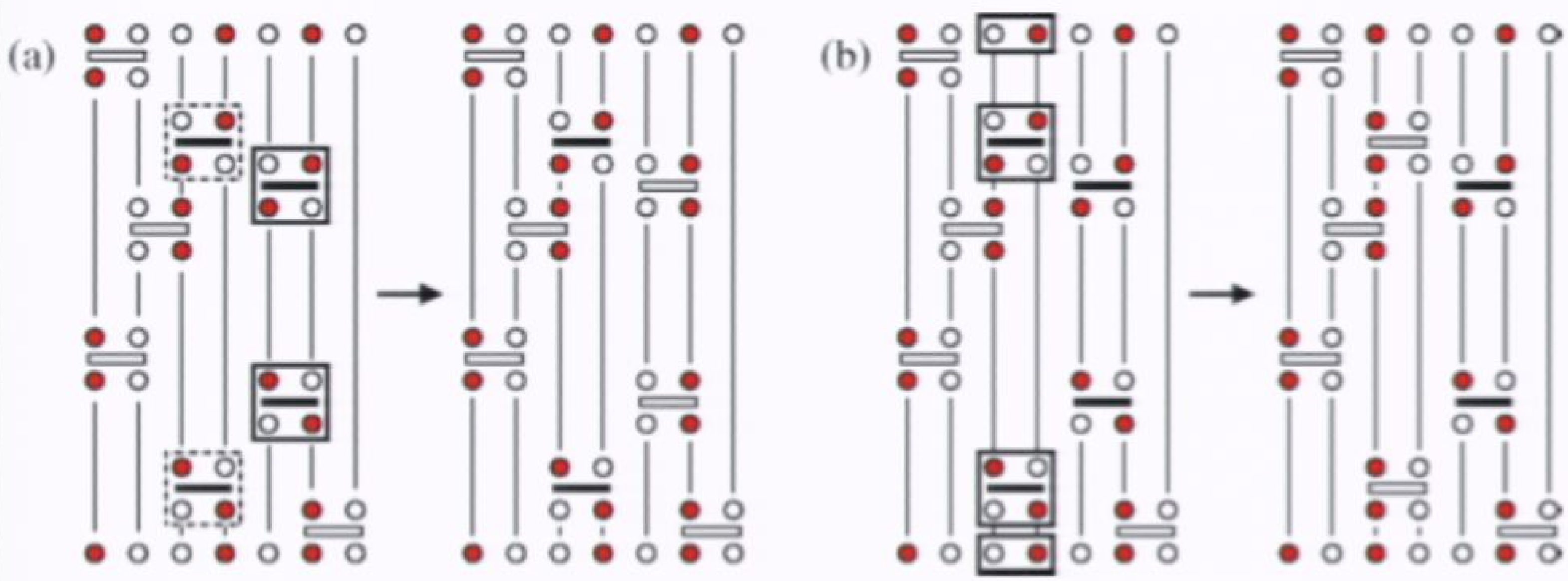
i(b),j(b) sites on bond b

Diagonal update; pseudocode implementation

```
do p = 0 to L − 1
    if (s(p) = 0) then
        b = random[1, . . . , N_b]; if  σ(i(b)) = σ(j(b)) cycle
        if (random[0 − 1] < P_insert(n)) then s(p) = 2b; n = n + 1 endif
    elseif (mod[s(p), 2] = 0) then
        if (random[0 − 1] < P_remove(n)) then s(p) = 0; n = n − 1 endif
    else
        b = s(p)/2; σ(i(b)) = −σ(i(b)); σ(j(b)) = −σ(j(b))
    endif
enddo
```

i(b),j(b) sites on bond b

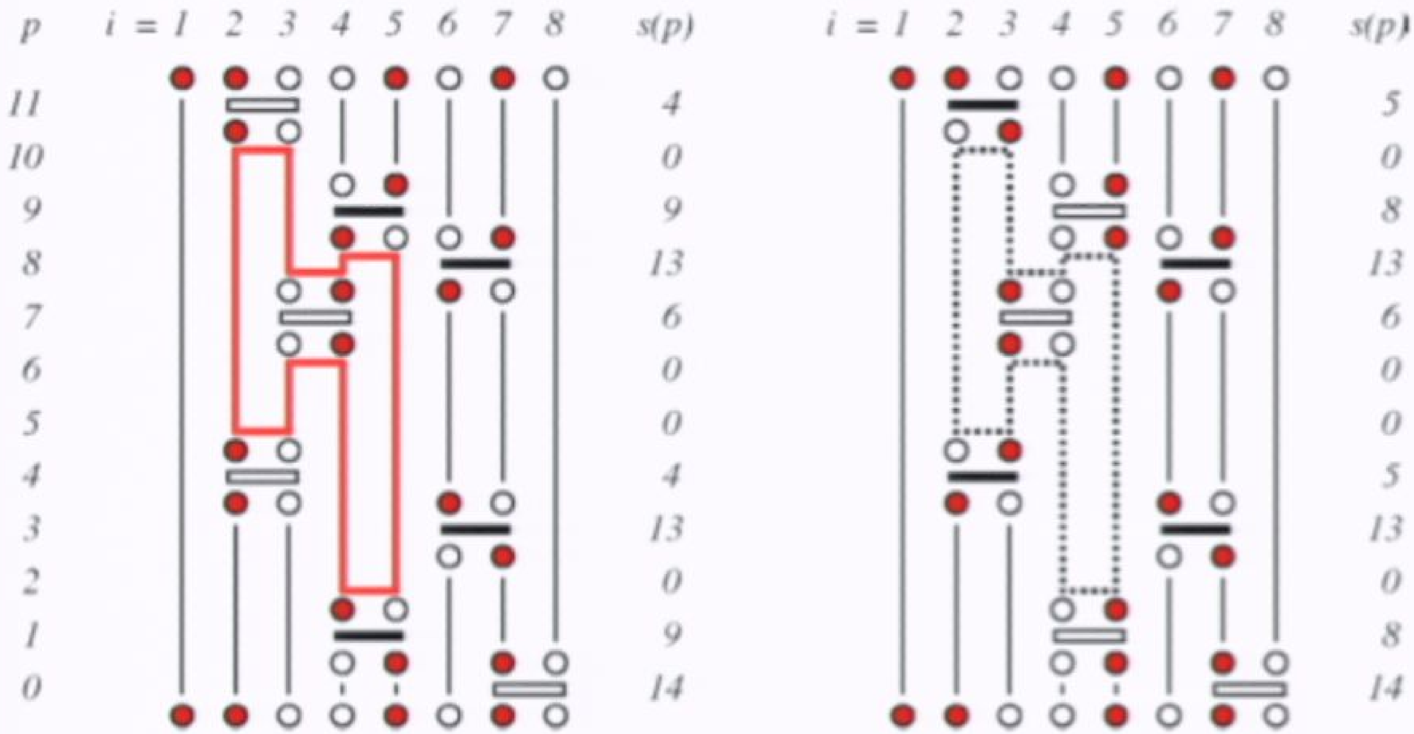
Local off-diagonal update



Switch the type (a=1 ↔ a=2) of two operators on the same spins

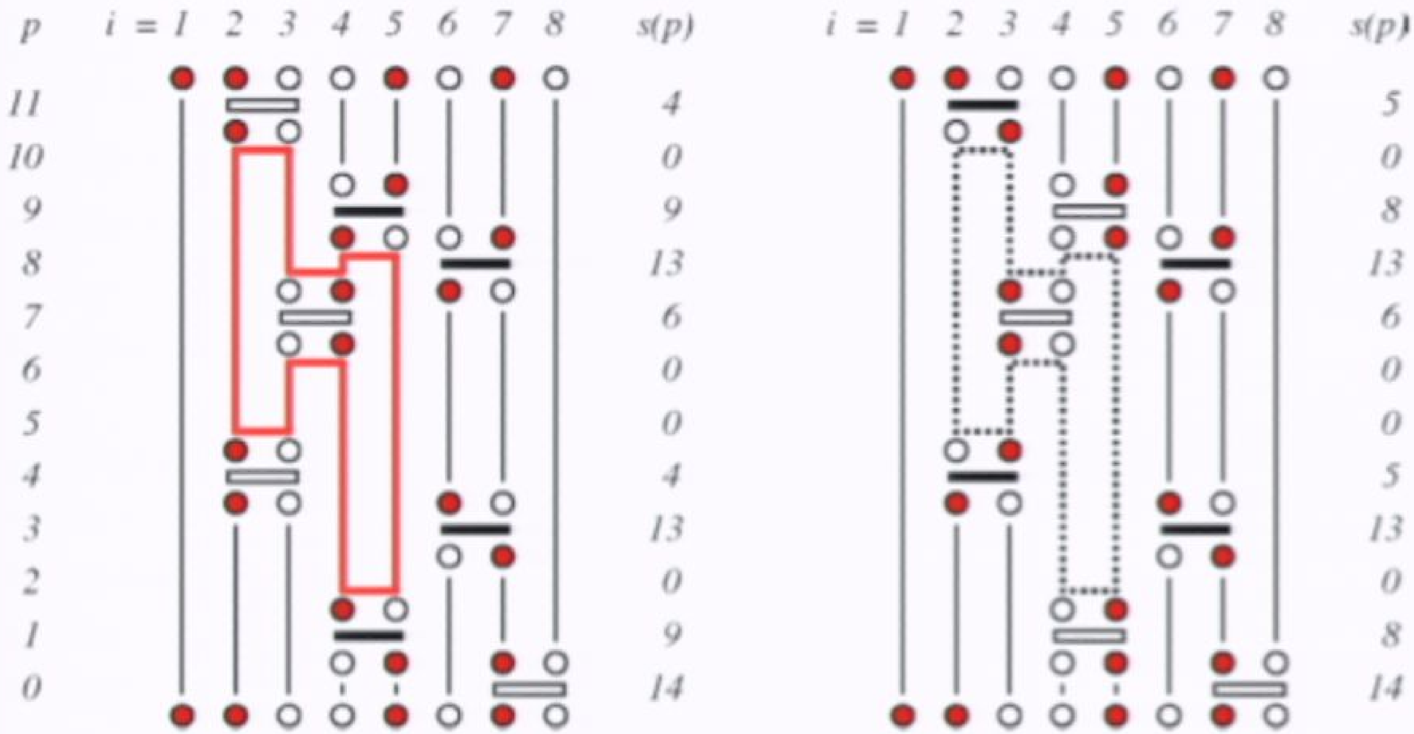
Operator-loop update

Many spins and operators can be changed simultaneously



Operator-loop update

Many spins and operators can be changed simultaneously



Pseudocode

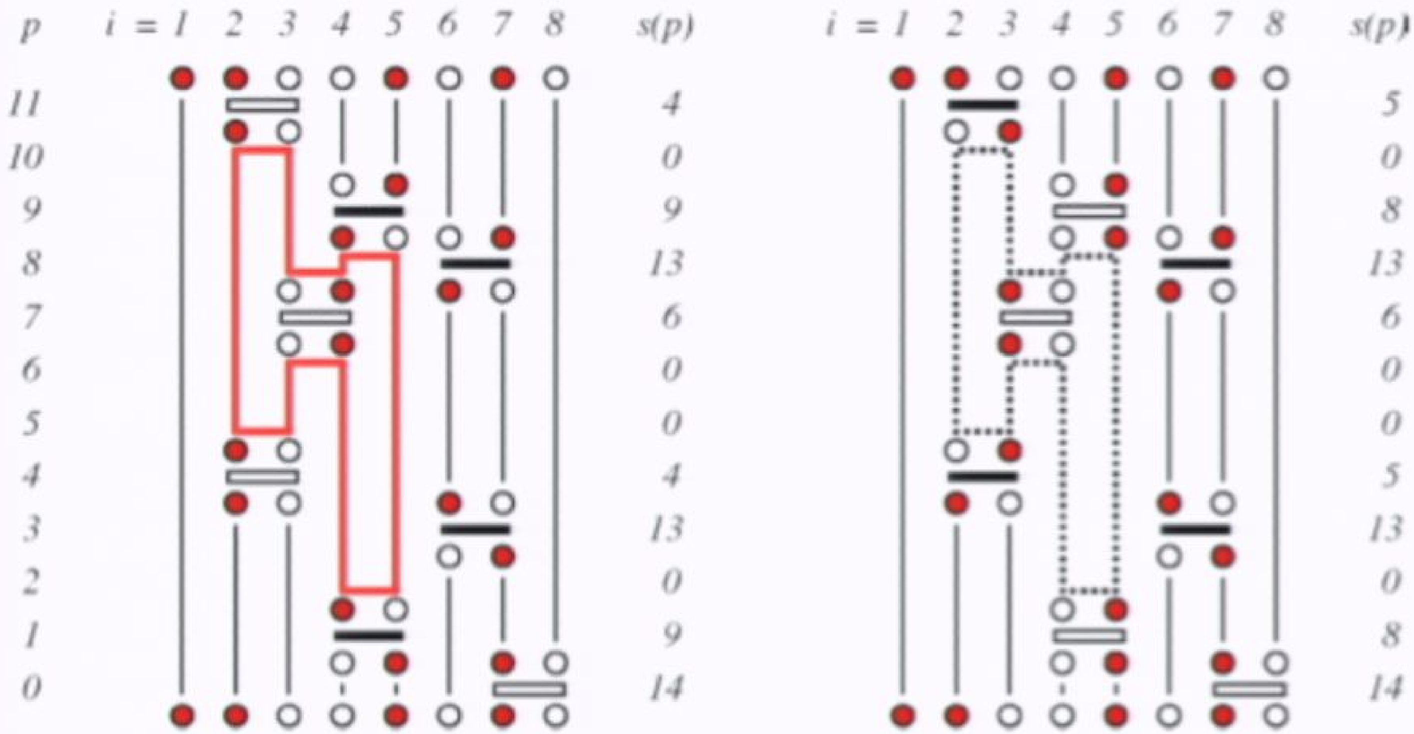
moving horizontally in the list corresponds to changing v even↔odd

- **flipbit**(*v*,*0*) flips bit 0 of v

- a given loop is only constructed once
- **vertices can be erased**
- *X(v)* < *0* = erased
- *X(v)* = -*1* not flipped loop
- *X(v)* = -2 flipped loop

Operator-loop update

Many spins and operators can be changed simultaneously



Pseudocode

moving horizontally in the list corresponds to changing v even↔odd

- **flipbit**(*v*,*0*) flips bit 0 of v

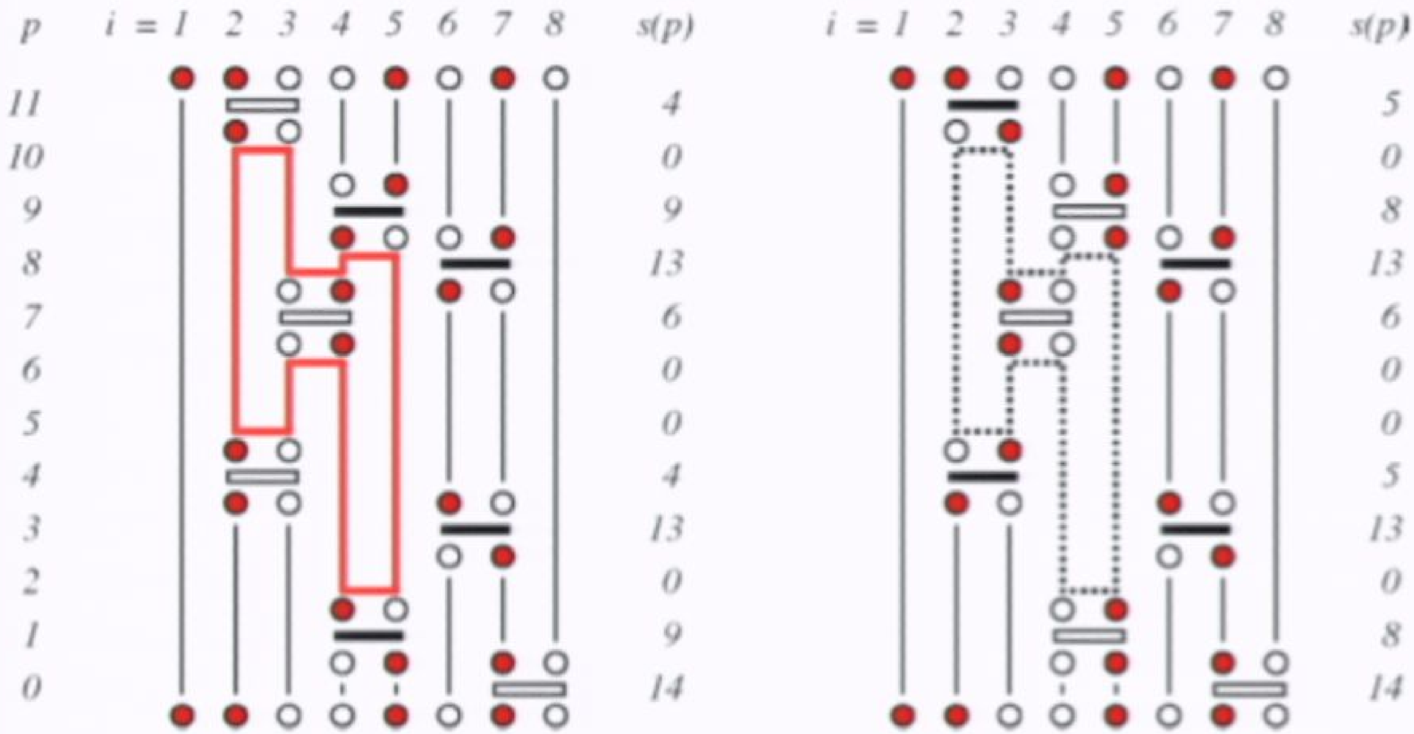
- a given loop is only constructed once
- **vertices can be erased**
- *X(v)* < *0* = erased
- *X(v)* = *-1* not flipped loop
- *X(v)* = -2 flipped loop

onstructing all loops, flip probability 1/2

```
lo 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
```

)perator-loop update

Many spins and operators can be changed simultaneously



Pseudocode

moving horizontally in the list corresponds to changing v even↔odd

- **flipbit**(*v*,*0*) flips bit 0 of v

- a given loop is only constructed once
- **vertices can be erased**
- *X*(*v*) < *0* = erased
- *X*(*v*) = -*1* not flipped loop
- *X*(*v*) = -2 flipped loop

onstructing all loops, flip probability 1/2

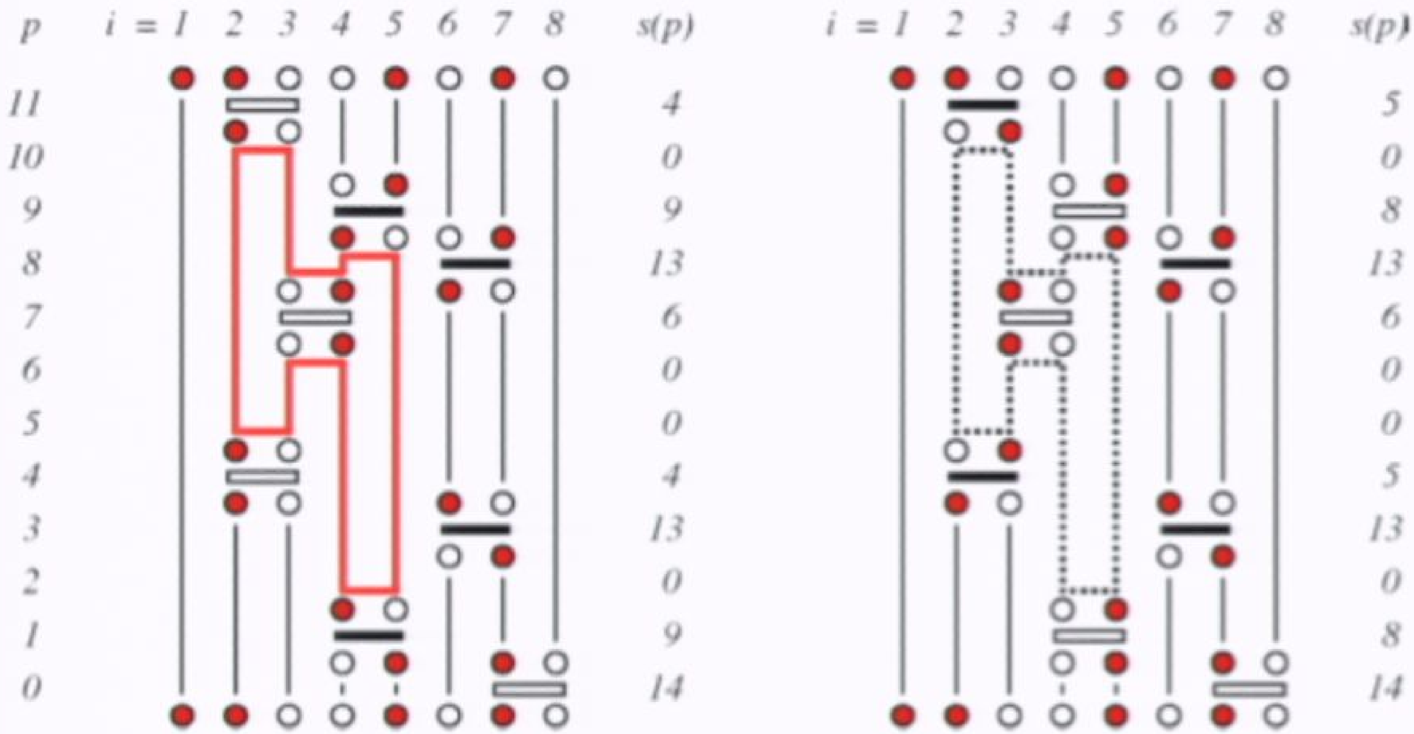
lo $v_0 = 0$ **to** $4L-1$ **step** 2
  if $(X(v_0) < 0)$ **cycle**
  $v = v_0$
  if (**random**$[0-1] < \frac{1}{2}$) **then**
    traverse the loop; for all v in loop, set $X(v) = -1$
  else
    traverse the loop; for all v in loop, set $X(v) = -2$
    flip the operators in the loop

construct and flip a loop

$v = v_0$
do
  $X(v) = -2$
  $p = v/4$; $s(p) =$ **flipbit**$(s(p),($
  $v' =$ **flipbit**$(v, 0)$
  $v = X(v')$; $X(v') = -2$

Operator-loop update

Many spins and operators can be changed simultaneously

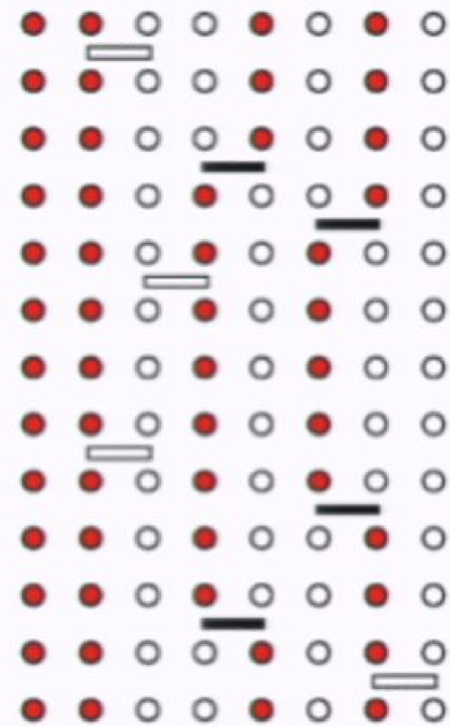


Monte Carlo sampling scheme

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

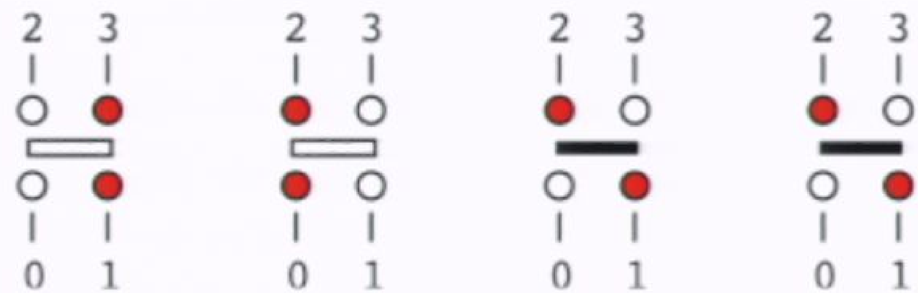
$$W(\alpha, S_L) = \left(\frac{\beta}{2}\right)^n \frac{(L-n)!}{L!}$$

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



Linked vertex storage

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

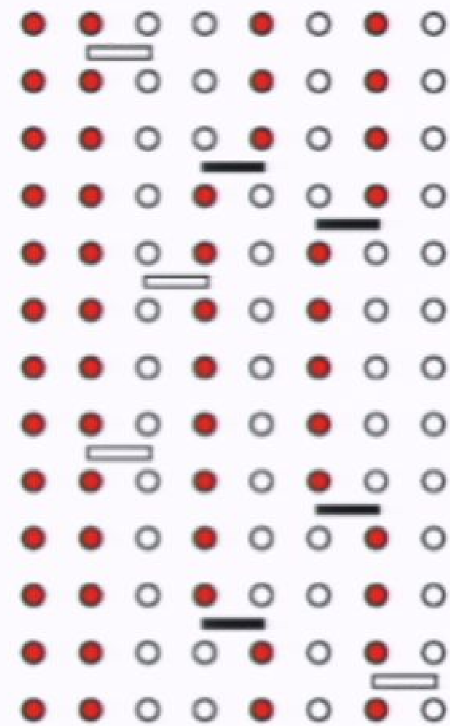


Monte Carlo sampling scheme

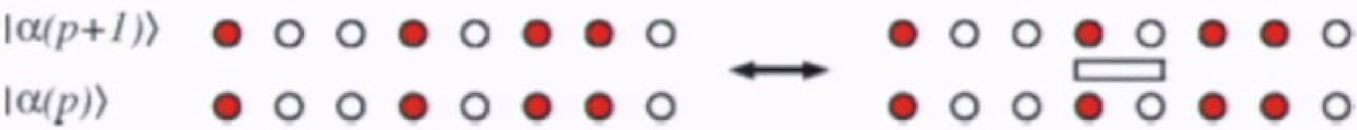
Change the configuration; $(\alpha, S_L) \rightarrow (\alpha', S'_L)$

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

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



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

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

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

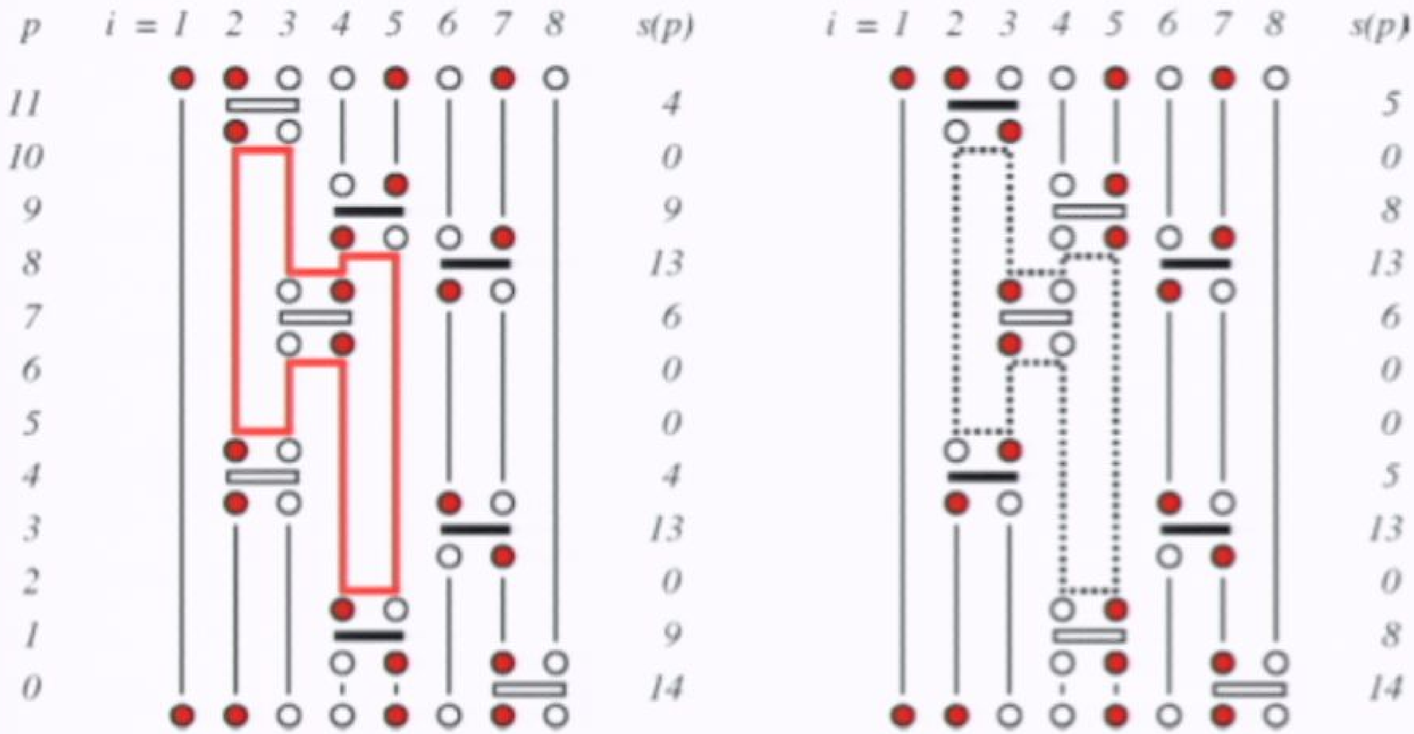
$$\frac{W(a=1)}{W(a=0)} = \frac{\beta/2}{L-n} \qquad \frac{W(a=0)}{W(a=1)} = \frac{L-n+1}{\beta/2}$$

n is the current power

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

Operator-loop update

Many spins and operators can be changed simultaneously



Pseudocode

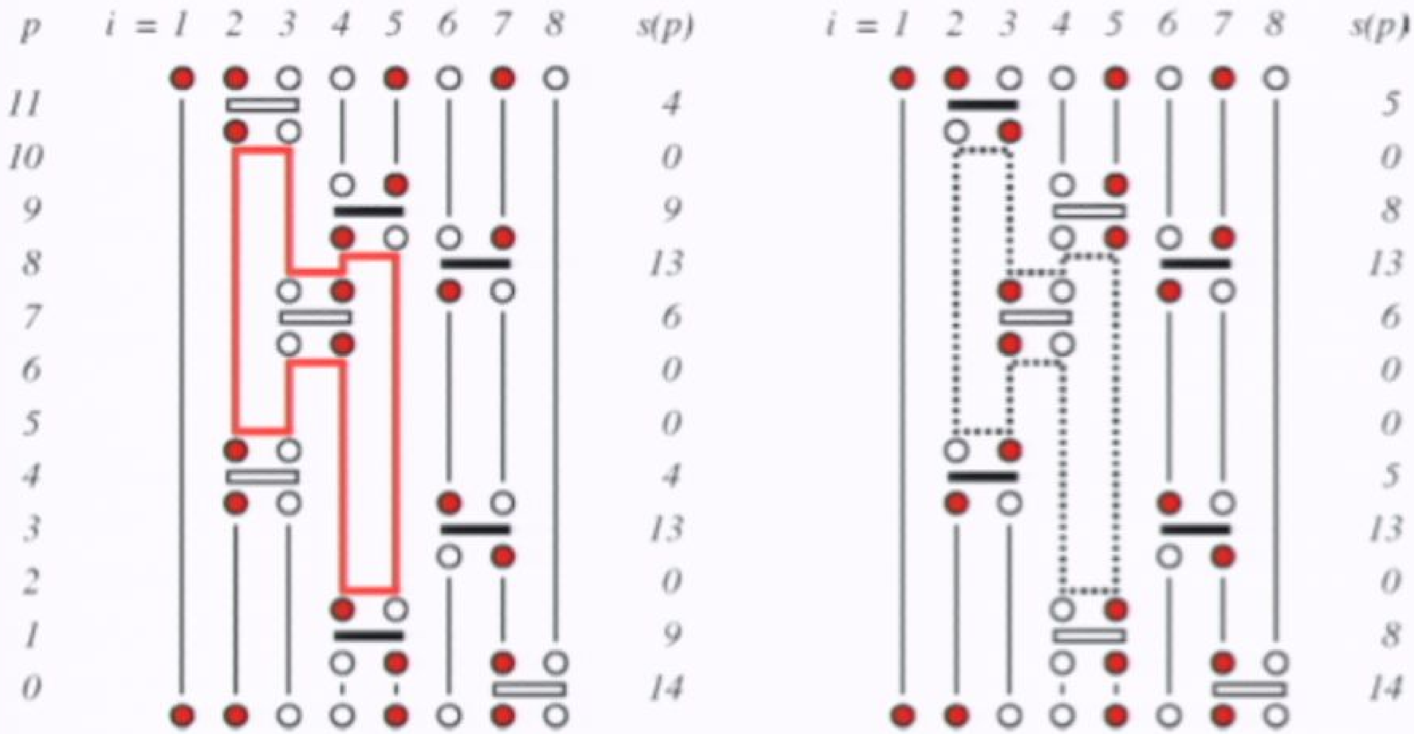
moving horizontally in the list corresponds to changing v even↔odd

- **flipbit**(*v*,*0*) flips bit 0 of v

- a given loop is only constructed once
- **vertices can be erased**
- *X(v)* < *0* = erased
- *X(v)* = -*1* not flipped loop
- *X(v)* = -2 flipped loop

)perator-loop update

Many spins and operators can be changed simultaneously



Pseudocode

moving horizontally in the list corresponds to changing v even↔odd

- **flipbit**(*v*,*0*) flips bit 0 of v

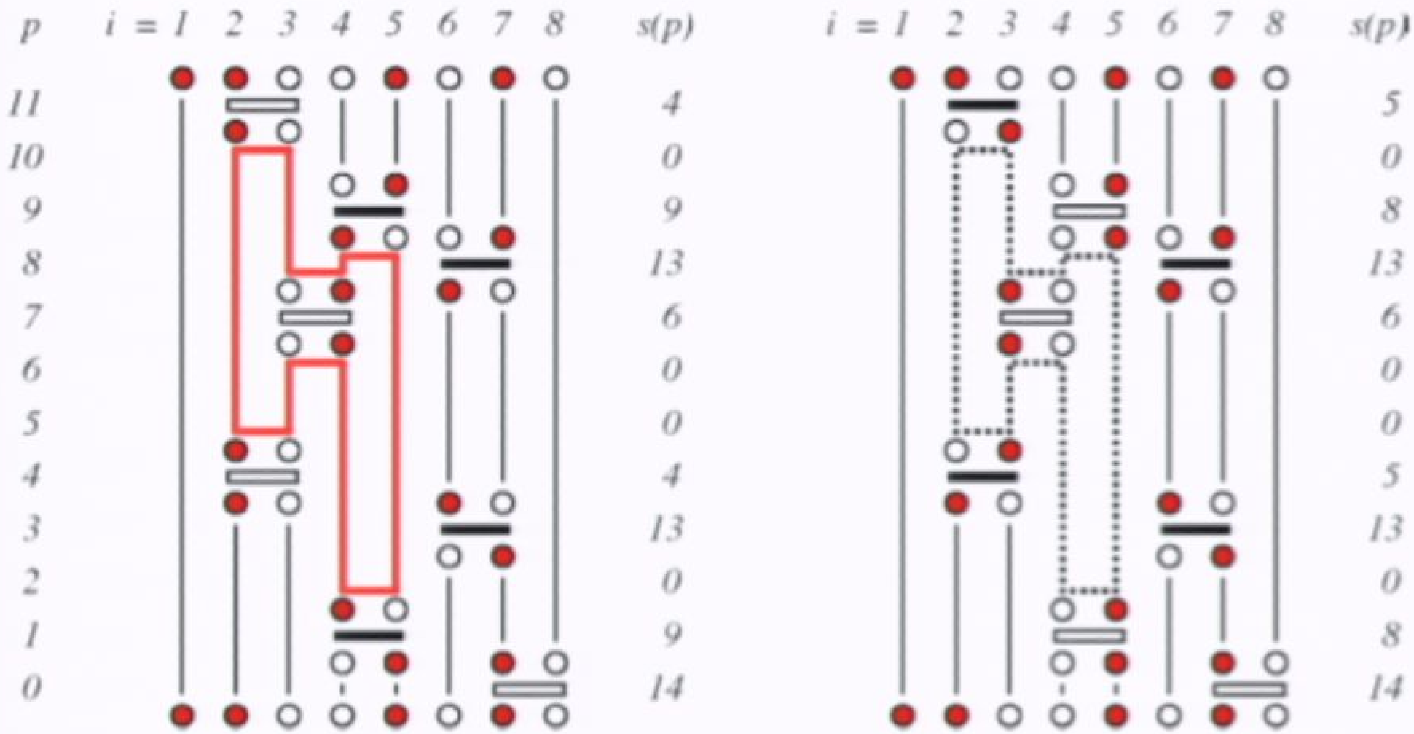
- a given loop is only constructed once
- **vertices can be erased**
- *X(v)* < *0* = erased
- *X(v)* = -*1* not flipped loop
- *X(v)* = -2 flipped loop

onstructing all loops, flip probability 1/2

lo $v_0 = 0$ **to** $4L - 1$ **step** 2
  if $(X(v_0) < 0)$ **cycle**
  $v = v_0$
  if (**random**$[0-1] < \frac{1}{2}$) **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

)perator-loop update

Many spins and operators can be changed simultaneously



Pseudocode

moving horizontally in the list corresponds to changing v even↔odd

- **flipbit**(*v*,*0*) flips bit 0 of v

- a given loop is only constructed once
- **vertices can be erased**
- *X(v)* < *0* = erased
- *X(v)* = *-1* not flipped loop
- *X(v)* = -2 flipped loop

onstructing all loops, flip probability 1/2

```
lo 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
```

construct and flip a loop

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

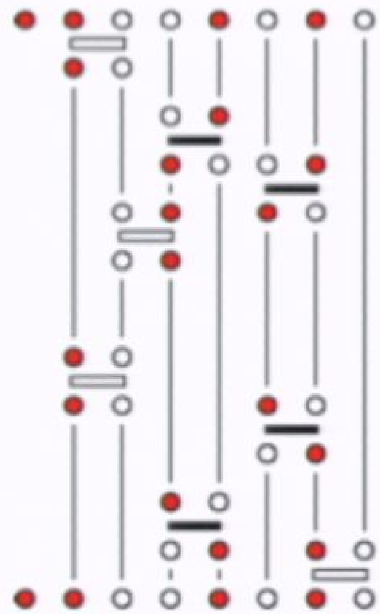
Constructing the linked vertex list

Traverse operator list $s(p)$, $p=0,\ldots,L-1$

- 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 (l=0)	X(v) (l=0)	v (l=1)	X(v) (l=1)	v (l=2)	X(v) (l=2)	v (l=3)	X(v) (l=3)
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	31	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

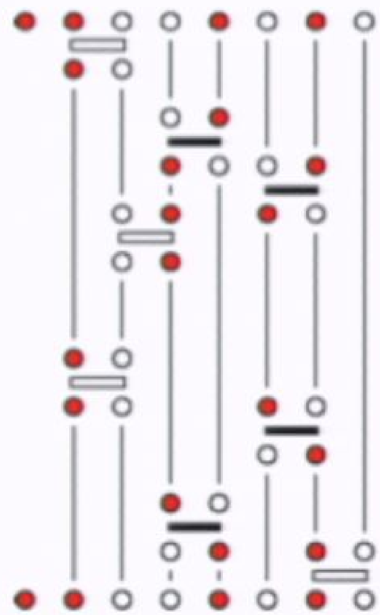
Constructing the linked vertex list

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 (l=0)	X(v) (l=0)	v (l=1)	X(v) (l=1)	v (l=2)	X(v) (l=2)	v (l=3)	X(v) (l=3)
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	31	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

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

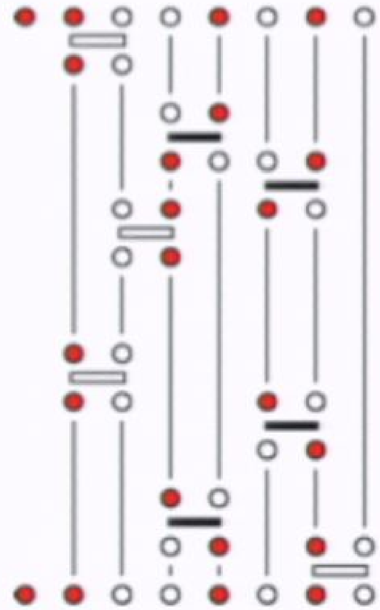
Constructing the linked vertex list

Traverse operator list $s(p)$, $p=0,\dots,L-1$

- 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

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



p	l=0: v	l=0: X(v)	l=1: v	l=1: X(v)	l=2: v	l=2: X(v)	l=3: v	l=3: 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	31	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

$V_{\rm first}(:) = -1$; $V_{\rm last}(:) = -1$
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
  $f = V_{\rm first}(i)$

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

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

```
do i = 1 to N
    v = V_first(i)
    if (v = -1) then
        if (random[0-1] < 1/2) σ(i) = -σ(i)
    else
        if (X(v) = -2) σ(i) = -σ(i)
    endif
enddo
```

v is the location of the first vertex leg on spin i

- flip it if X(v)=−2
- (do not flip it if X(v)=−1)
- no operation on i if $v_{first}(i)=-1$

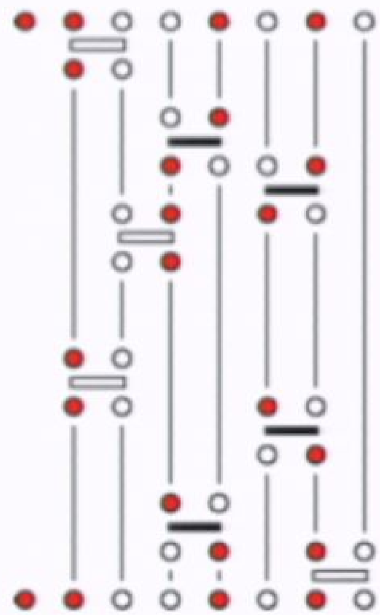
Constructing the linked vertex list

Traverse operator list $s(p)$, $p=0,\ldots,L-1$

- 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

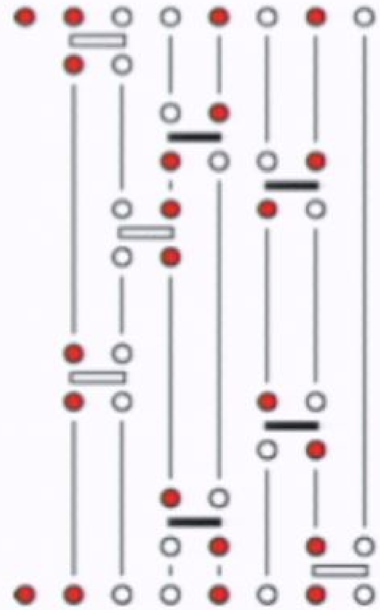


p	v (l=0)	X(v) (l=0)	v (l=1)	X(v) (l=1)	v (l=2)	X(v) (l=2)	v (l=3)	X(v) (l=3)
11	44	18	45	30	46	16	47	17
10	40	-	41	-	42	-	43	-
9	36	31	37	7	38	4	39	5
8	32	14	31	15	34	12	35	0
7	28	19	29	6	30	45	31	36
6	24	-	25	-	26	-	27	-
5	20	-	21	-	22	-	23	-
4	16	46	17	47	18	44	19	28
3	12	34	13	2	14	32	15	33
2	8	-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

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 (l=0)	X(v) (l=0)	v (l=1)	X(v) (l=1)	v (l=2)	X(v) (l=2)	v (l=3)	X(v) (l=3)
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	31	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

$V_{\rm first}(:) = -1;\ V_{\rm last}(:) = -1$
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

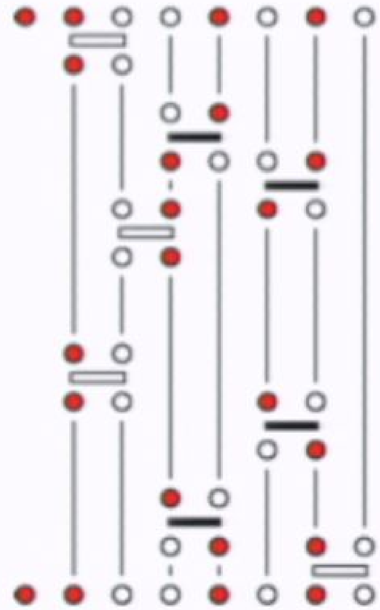
Constructing the linked vertex list

Traverse operator list $s(p)$, $p=0,\ldots,L-1$

- 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

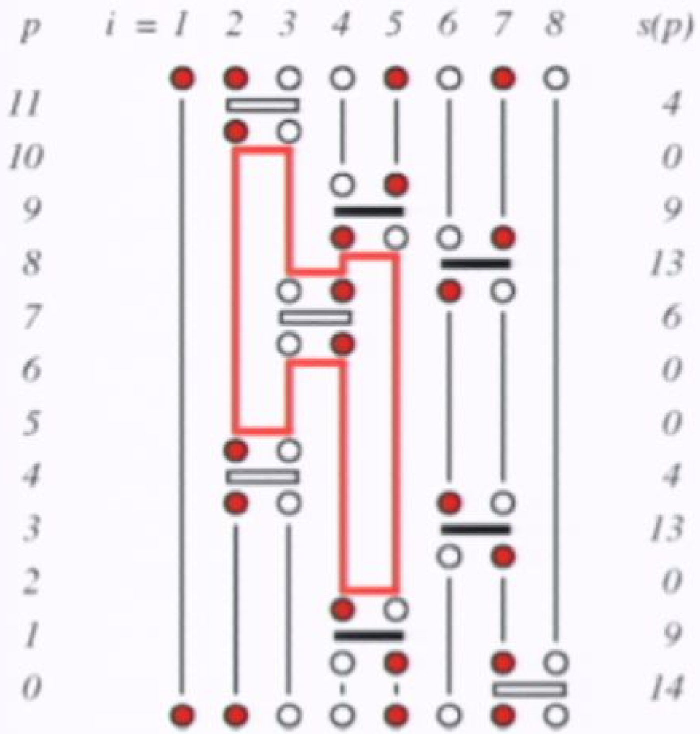
- $V_{first}(i)$ = location v of first leg on site i
- $V_{last}(i)$ = location v of last (currently) leg
- these are used to create the links
- initialize all elements to −1

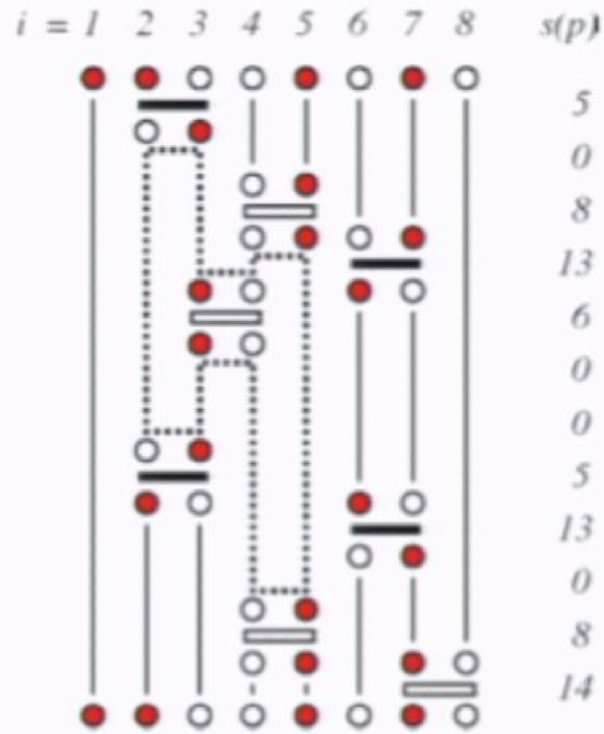


p	v (l=0)	X(v) (l=0)	v (l=1)	X(v) (l=1)	v (l=2)	X(v) (l=2)	v (l=3)	X(v) (l=3)
11	44	18	45	30	46	16	47	17
10	40	-	41	-	42	-	43	-
9	36	31	37	7	38	4	39	5
8	32	14	31	15	34	12	35	0
7	28	19	29	6	30	45	31	36
6	24	-	25	-	26	-	27	-
5	20	-	21	-	22	-	23	-
4	16	46	17	47	18	44	19	28
3	12	34	13	2	14	32	15	33
2	8	-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

Operator-loop update

Many spins and operators can be changed simultaneously





Pseudocode

moving horizontally in the list corresponds to changing v even↔odd

- **flipbit**(*v*,*0*) flips bit 0 of v

- a given loop is only constructed once
- **vertices can be erased**
- *X*(*v*) < *0* = erased
- *X*(*v*) = -*1* not flipped loop
- *X*(*v*) = -2 flipped loop

onstructing all loops, flip probability 1/2

lo $v_0 = 0$ **to** $4L-1$ **step** 2
  if $(X(v_0) < 0)$ **cycle**
  $v = v_0$
  if (**random**$[0-1] < \frac{1}{2}$) **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

construct and flip a loop

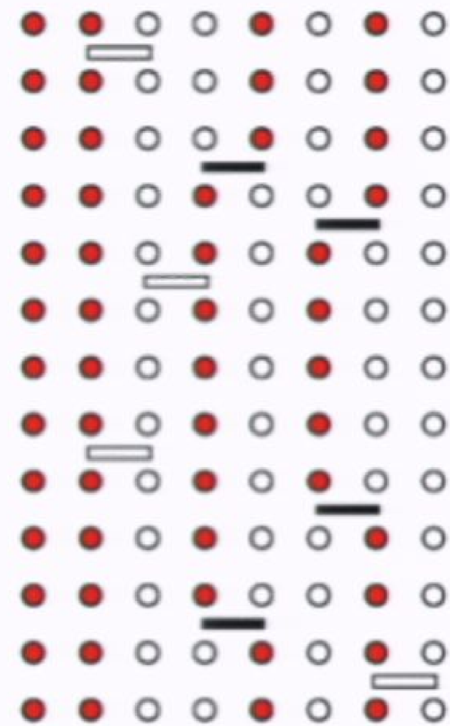
$v = v_0$
do
  $X(v) = -2$
  $p = v/4$; $s(p) =$ **flipbit**$(s(p),($
  $v' =$ **flipbit**$(v, 0)$
  $v = X(v')$; $X(v') = -2$

Monte Carlo sampling scheme

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

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

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

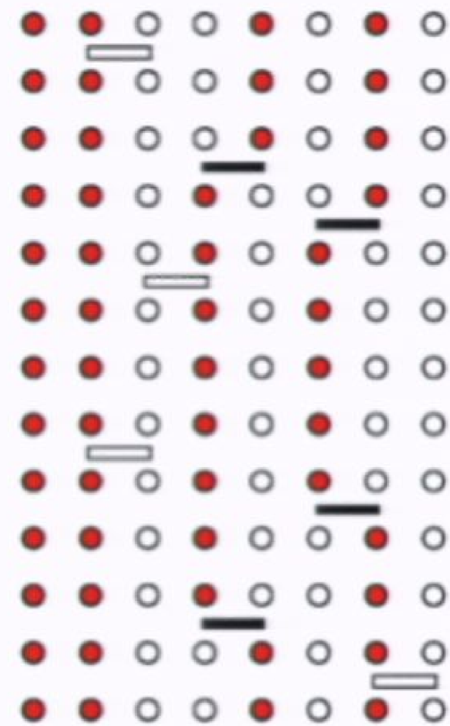


Monte Carlo sampling scheme

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

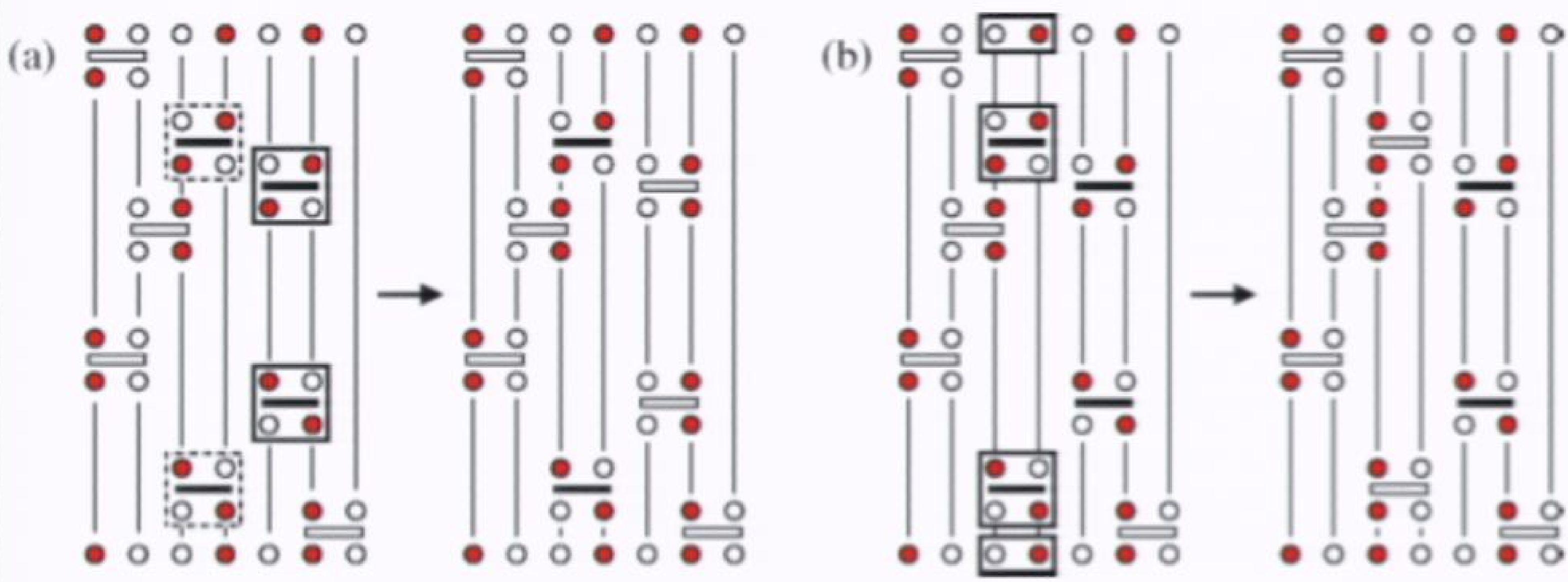
- n → n+1 (a=0 → a=1)
- n → n-1 (a=1 → a=0)

Diagonal update; pseudocode implementation

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

i(b),j(b) sites on bond b

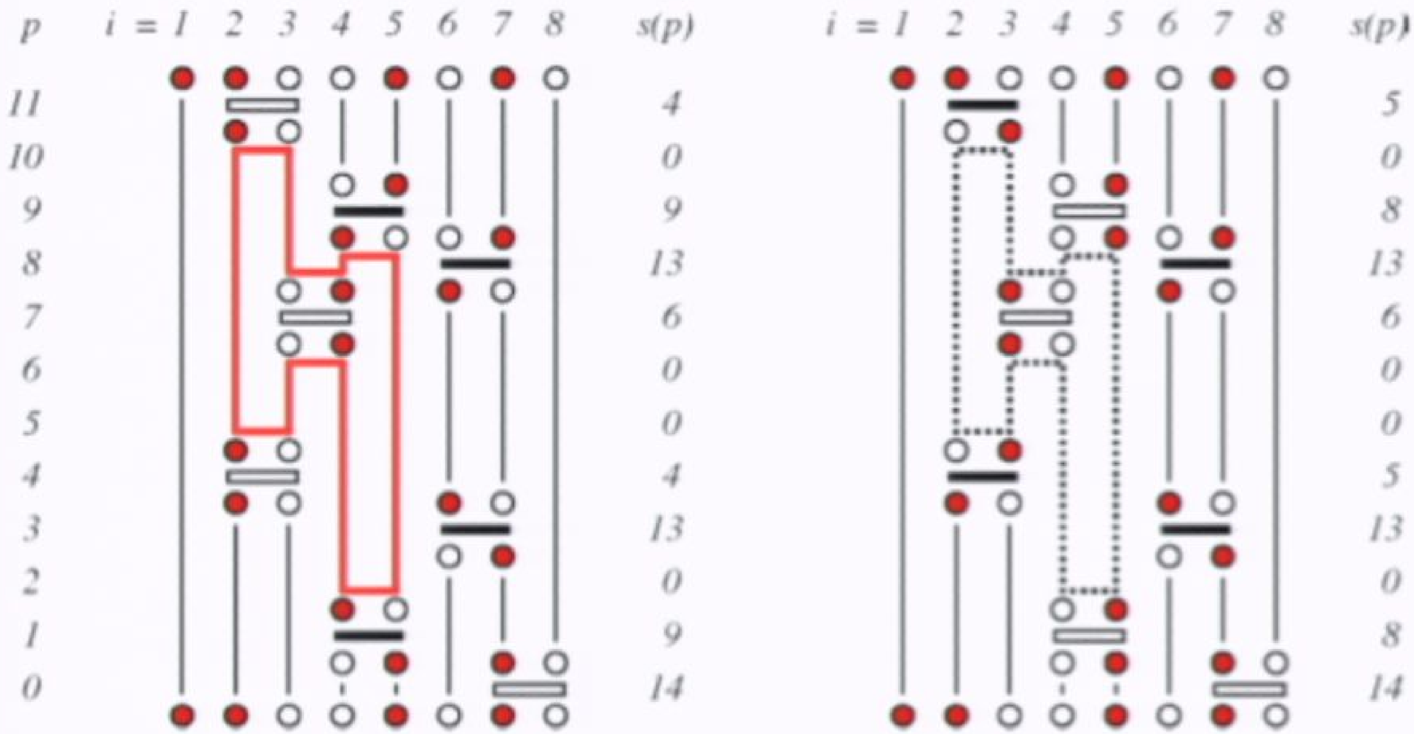
Local off-diagonal update



Switch the type (a=1 ↔ a=2) of two operators on the same spins

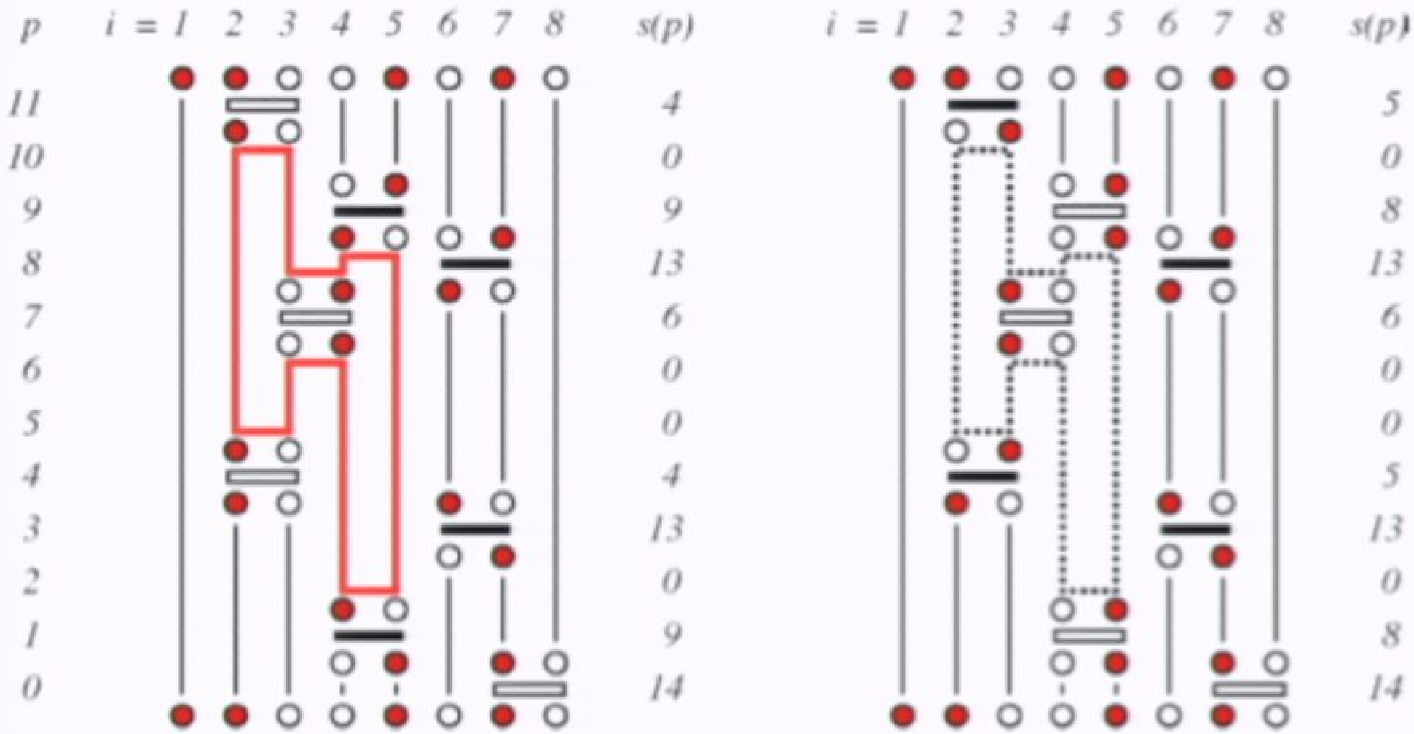
Operator-loop update

Many spins and operators can be changed simultaneously



Operator-loop update

Many spins and operators can be changed simultaneously



Pseudocode

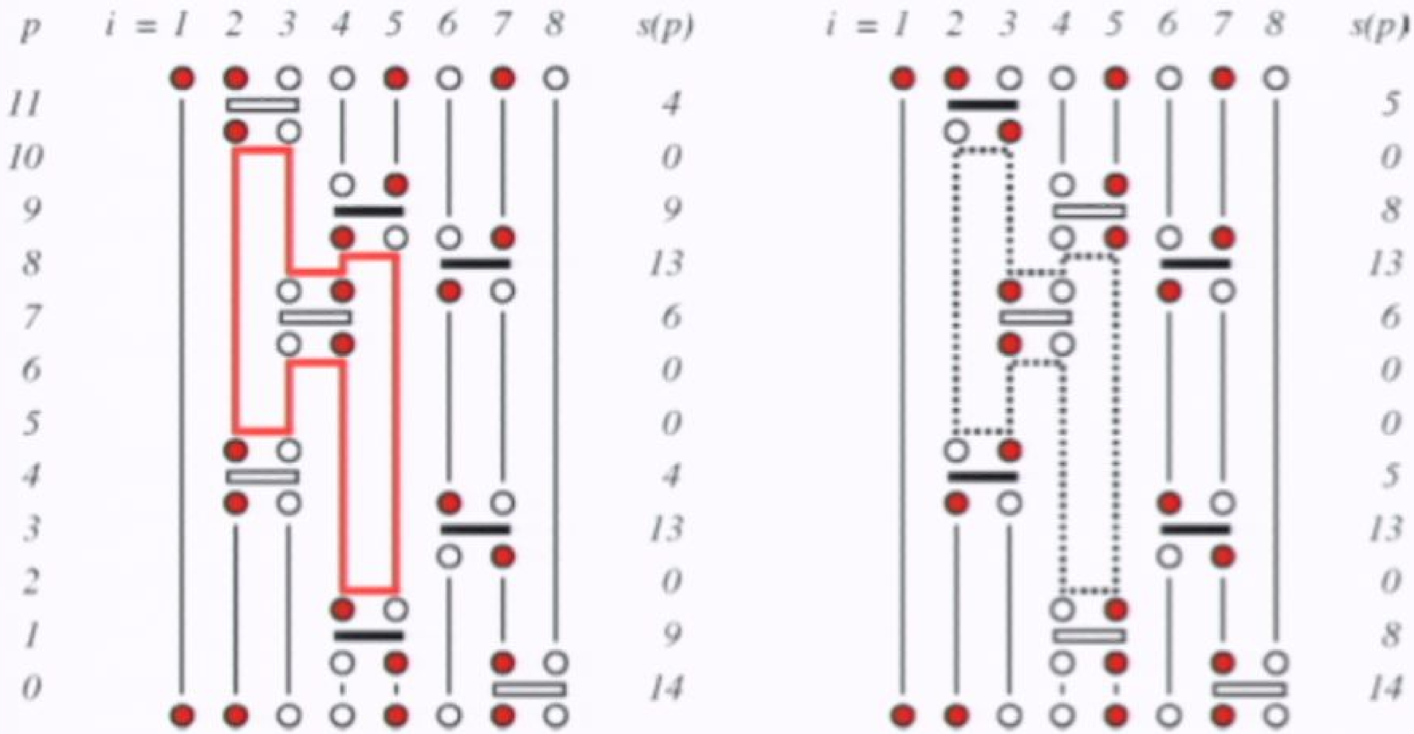
moving horizontally in the list corresponds to changing v even↔odd

- **flipbit**(*v*,*0*) flips bit 0 of v

- a given loop is only constructed once
- **vertices can be erased**
- *X(v) < 0* = erased
- *X(v)* = *-1* not flipped loop
- *X(v)* = -2 flipped loop

Operator-loop update

Many spins and operators can be changed simultaneously



Pseudocode

moving horizontally in the list corresponds to changing v even↔odd

- **flipbit**(*v*,*0*) flips bit 0 of v

- a given loop is only constructed once
- **vertices can be erased**
- *X(v)* < *0* = erased
- *X(v)* = -*1* not flipped loop
- *X(v)* = -2 flipped loop

onstructing all loops, flip probability 1/2

```
lo 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
```

construct and flip a loop

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

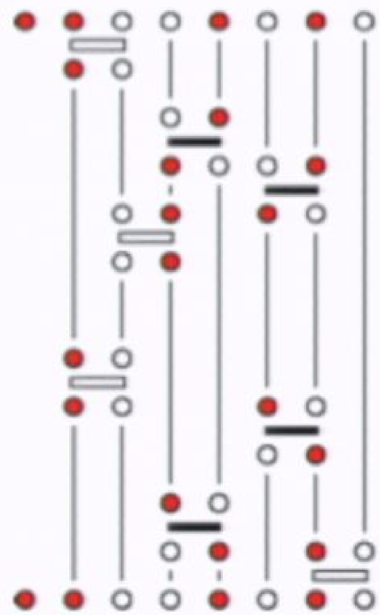
Constructing the linked vertex list

Traverse operator list $s(p)$, $p=0,\ldots,L-1$

- 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 (l=0)	X(v) (l=0)	v (l=1)	X(v) (l=1)	v (l=2)	X(v) (l=2)	v (l=3)	X(v) (l=3)
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	31	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

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

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

```
do i = 1 to N
    v = V_first(i)
    if (v = -1) then
        if (random[0-1] < 1/2) σ(i) = -σ(i)
    else
        if (X(v) = -2) σ(i) = -σ(i)
    endif
enddo
```

v is the location of the first vertex leg on spin i

- flip it if $X(v)=-2$
- (do not flip it if $X(v)=-1$)
- no operation on i if $v_{first}(i)=-1$

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

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

```
do i = 1 to N
    v = V_first(i)
    if (v = -1) then
        if (random[0-1] < 1/2) σ(i) = -σ(i)
    else
        if (X(v) = -2) σ(i) = -σ(i)
    endif
enddo
```

v is the location of the first vertex leg on spin i

- flip it if X(v)=–2
- (do not flip it if X(v)=–1)
- no operation on i if $v_{first}(i)=-1$

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***

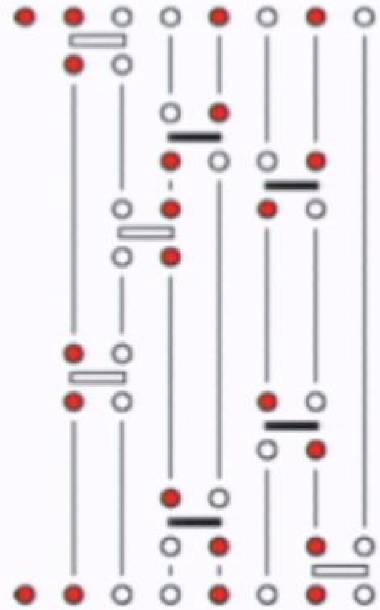
Constructing the linked vertex list

Traverse operator list $s(p)$, $p=0,\ldots,L-1$

- 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



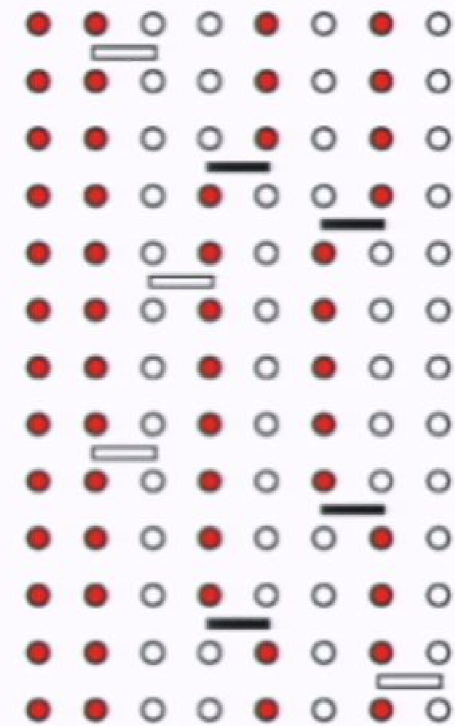
p	v ($l=0$)	X(v) ($l=0$)	v ($l=1$)	X(v) ($l=1$)	v ($l=2$)	X(v) ($l=2$)	v ($l=3$)	X(v) ($l=3$)
11	44	18	45	30	46	16	47	17
10	40	-	41	-	42	-	43	-
9	36	31	37	7	38	4	39	5
8	32	14	31	15	34	12	35	0
7	28	19	29	6	30	45	31	36
6	24	-	25	-	26	-	27	-
5	20	-	21	-	22	-	23	-
4	16	46	17	47	18	44	19	28
3	12	34	13	2	14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

Monte Carlo sampling scheme

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

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

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

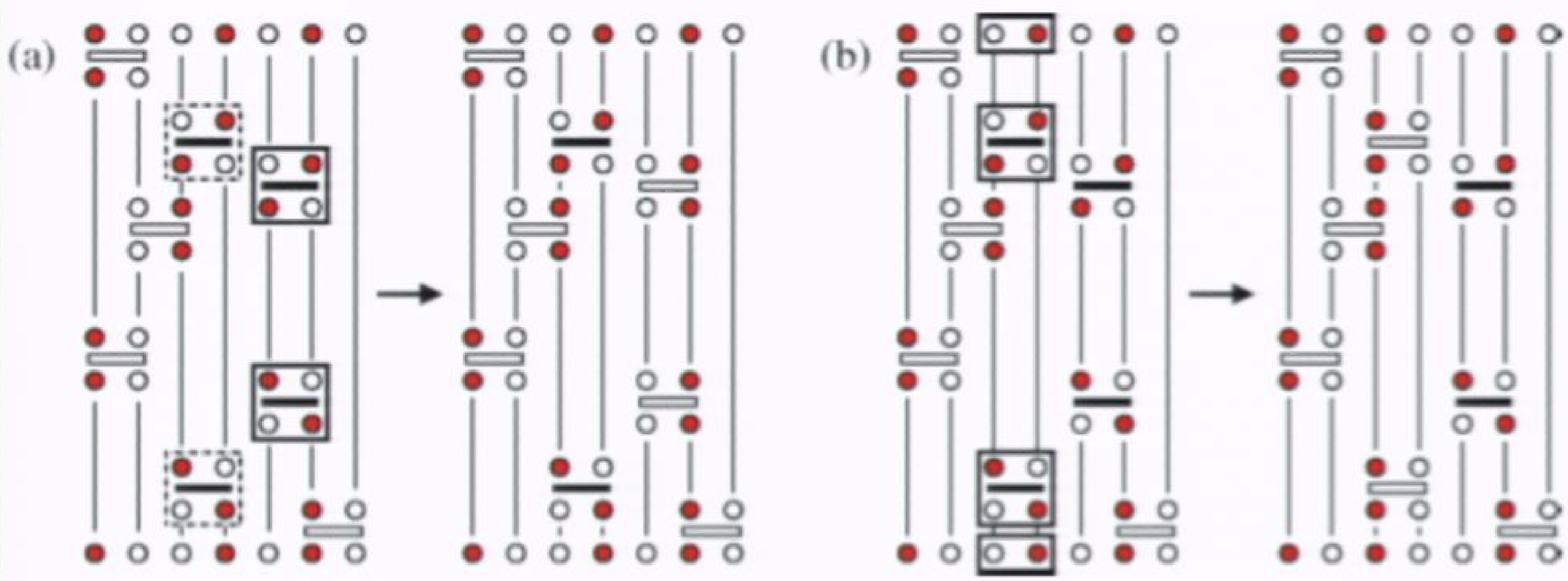


Diagonal update; pseudocode implementation

```
do p = 0 to L − 1
    if (s(p) = 0) then
        b = random[1, . . . , N_b]; if  σ(i(b)) = σ(j(b)) cycle
        if (random[0 − 1] < P_insert(n)) then s(p) = 2b; n = n + 1 endif
    elseif (mod[s(p), 2] = 0) then
        if (random[0 − 1] < P_remove(n)) then s(p) = 0; n = n − 1 endif
    else
        b = s(p)/2; σ(i(b)) = −σ(i(b)); σ(j(b)) = −σ(j(b))
    endif
enddo
```

i(b),j(b) sites on bond b

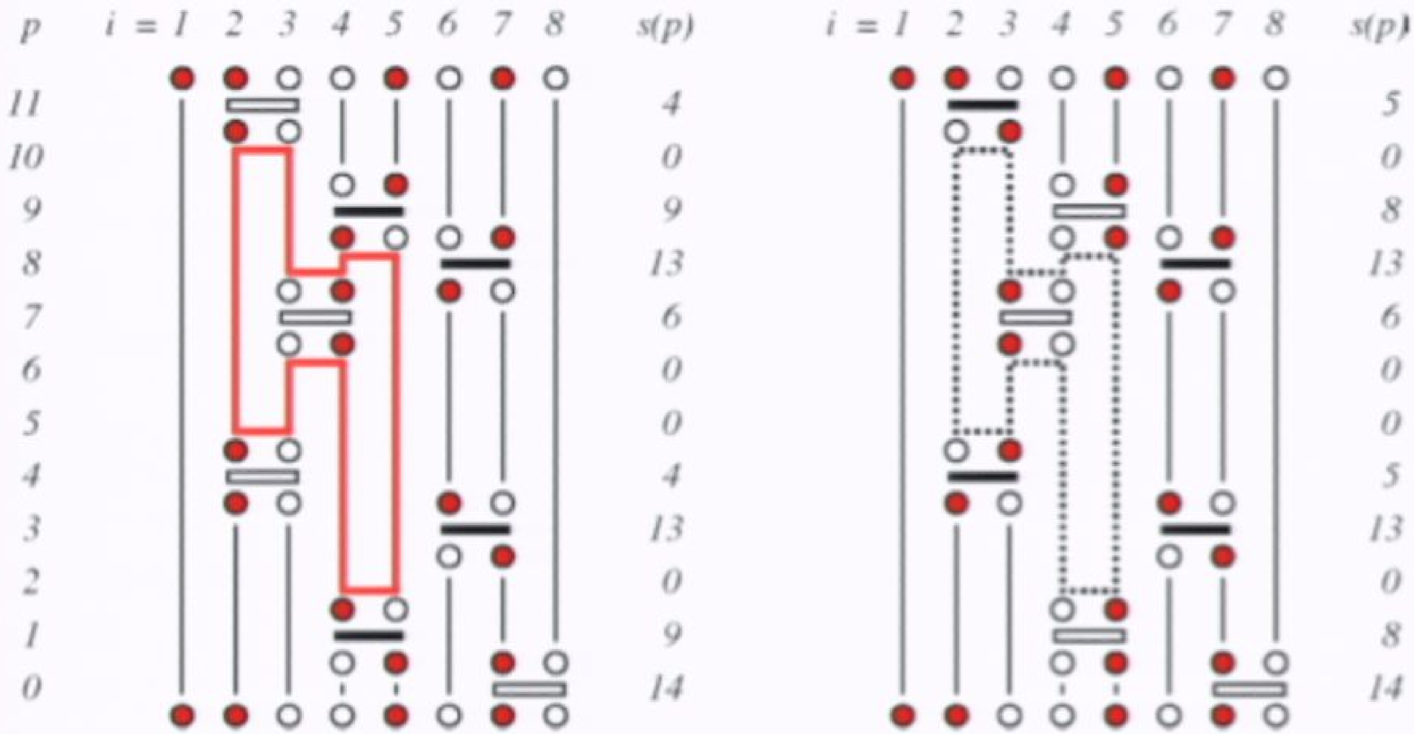
Local off-diagonal update



Switch the type (a=1 ↔ a=2) of two operators on the same spins

Operator-loop update

Many spins and operators can be changed simultaneously



Pseudocode

moving horizontally in the list corresponds to changing v even↔odd

- **flipbit**(*v*,*0*) flips bit 0 of v

- a given loop is only constructed once
- **vertices can be erased**
- *X(v) < 0* = erased
- *X(v) = -1* not flipped loop
- *X(v) = -2* flipped loop

onstructing 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 $(\textbf{random}[0-1] < \frac{1}{2})$ **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

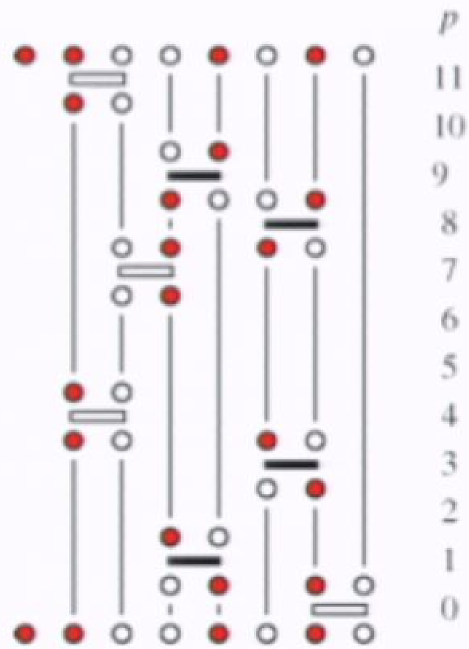
Constructing the linked vertex list

Traverse operator list $s(p)$, $p=0,\ldots,L-1$

- 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 (l=0)	X(v) (l=0)	v (l=1)	X(v) (l=1)	v (l=2)	X(v) (l=2)	v (l=3)	X(v) (l=3)
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	31	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

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

creating the last links across the "time" boundary

do $i = 1$ **to** N
  $f = V_{first}(i)$

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

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

```
do i = 1 to N
    v = V_first(i)
    if (v = -1) then
        if (random[0-1] < 1/2) σ(i) = -σ(i)
    else
        if (X(v) = -2) σ(i) = -σ(i)
    endif
enddo
```

v is the location of the first vertex leg on spin i

- flip it if X(v)=–2
- (do not flip it if X(v)=–1)
- no operation on i if $v_{first}(i)=-1$

Determination of the cut-off L

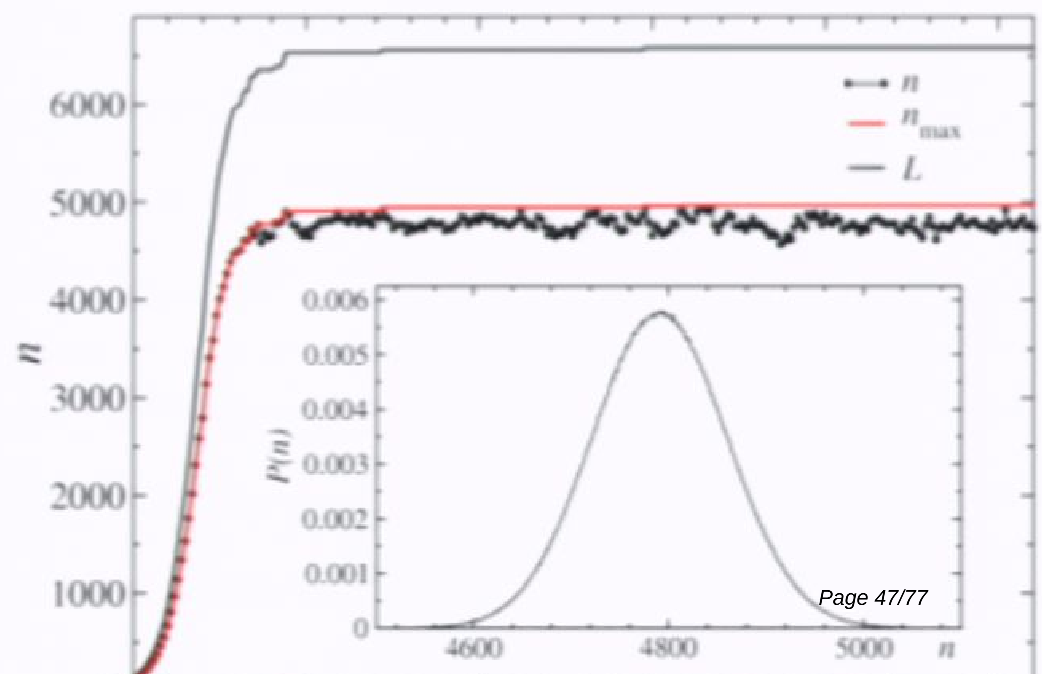
- adjust during equilibration
- start with arbitrary (small) n

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



Properties of the Heisenberg chain; large-scale SSE results

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

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

```
do i = 1 to N
    v = V_first(i)
    if (v = -1) then
        if (random[0-1] < 1/2) σ(i) = -σ(i)
    else
        if (X(v) = -2) σ(i) = -σ(i)
    endif
enddo
```

v is the location of the first vertex leg on spin i

- flip it if X(v)=–2
- (do not flip it if X(v)=–1)
- no operation on i if $v_{first}(i)=-1$

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

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

```
do i = 1 to N
    v = V_first(i)
    if (v = -1) then
        if (random[0-1] < 1/2) σ(i) = -σ(i)
    else
        if (X(v) = -2) σ(i) = -σ(i)
    endif
enddo
```

v is the location of the first vertex leg on spin i

- flip it if X(v)=−2
- (do not flip it if X(v)=−1)
- no operation on i if $v_{first}(i)=-1$

Determination of the cut-off L

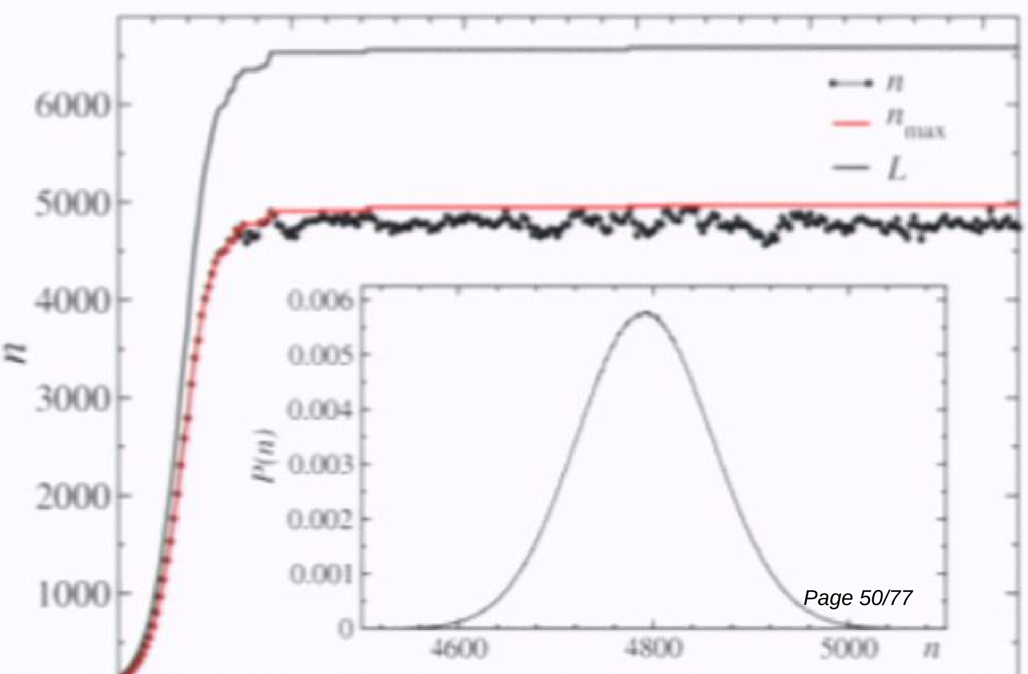
- adjust during equilibration
- start with arbitrary (small) n

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



Does it work?

Compare with exact results

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

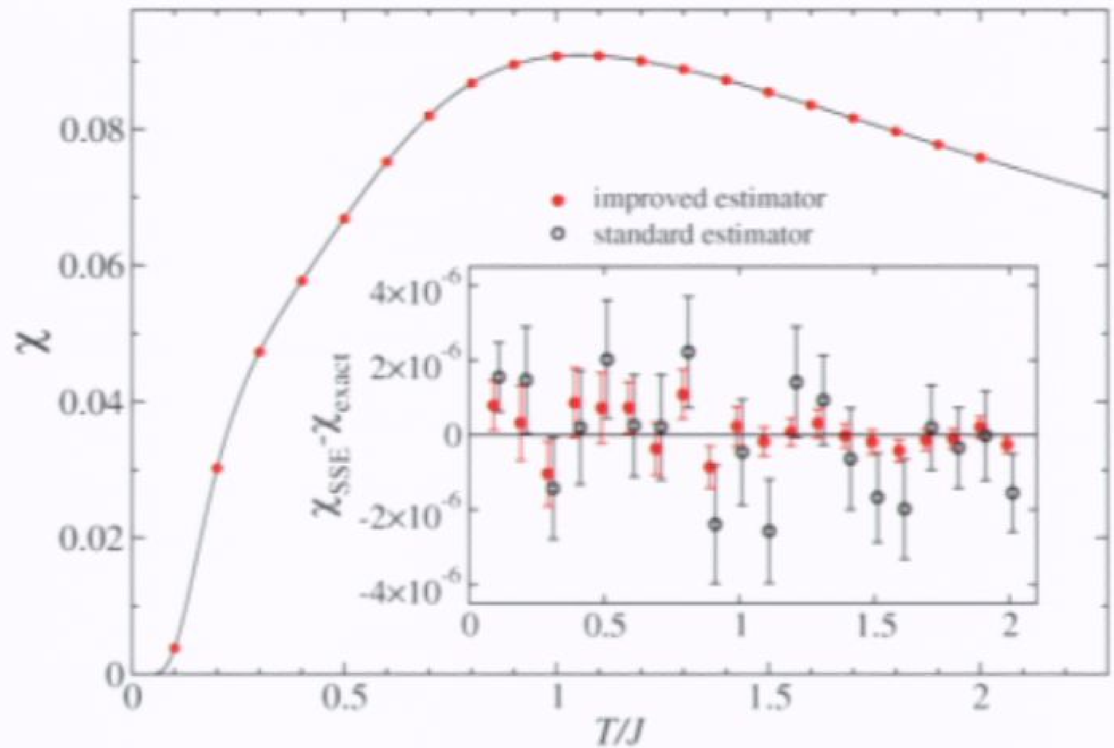
Does it work?

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)



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

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

```
do i = 1 to N
    v = V_first(i)
    if (v = -1) then
        if (random[0-1] < 1/2) σ(i) = -σ(i)
    else
        if (X(v) = -2) σ(i) = -σ(i)
    endif
enddo
```

v is the location of the first vertex leg on spin i

- flip it if $X(v)=-2$
- (do not flip it if $X(v)=-1$)
- no operation on i if $v_{first}(i)=-1$

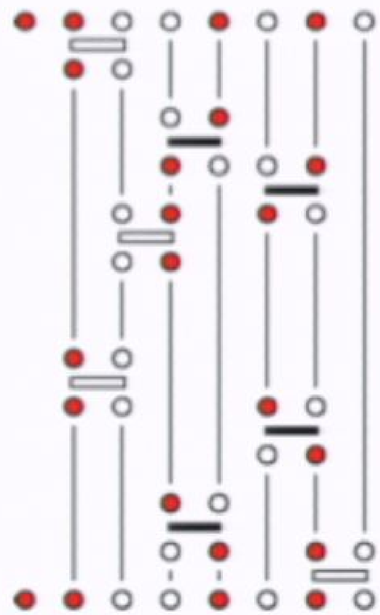
Constructing the linked vertex list

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

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

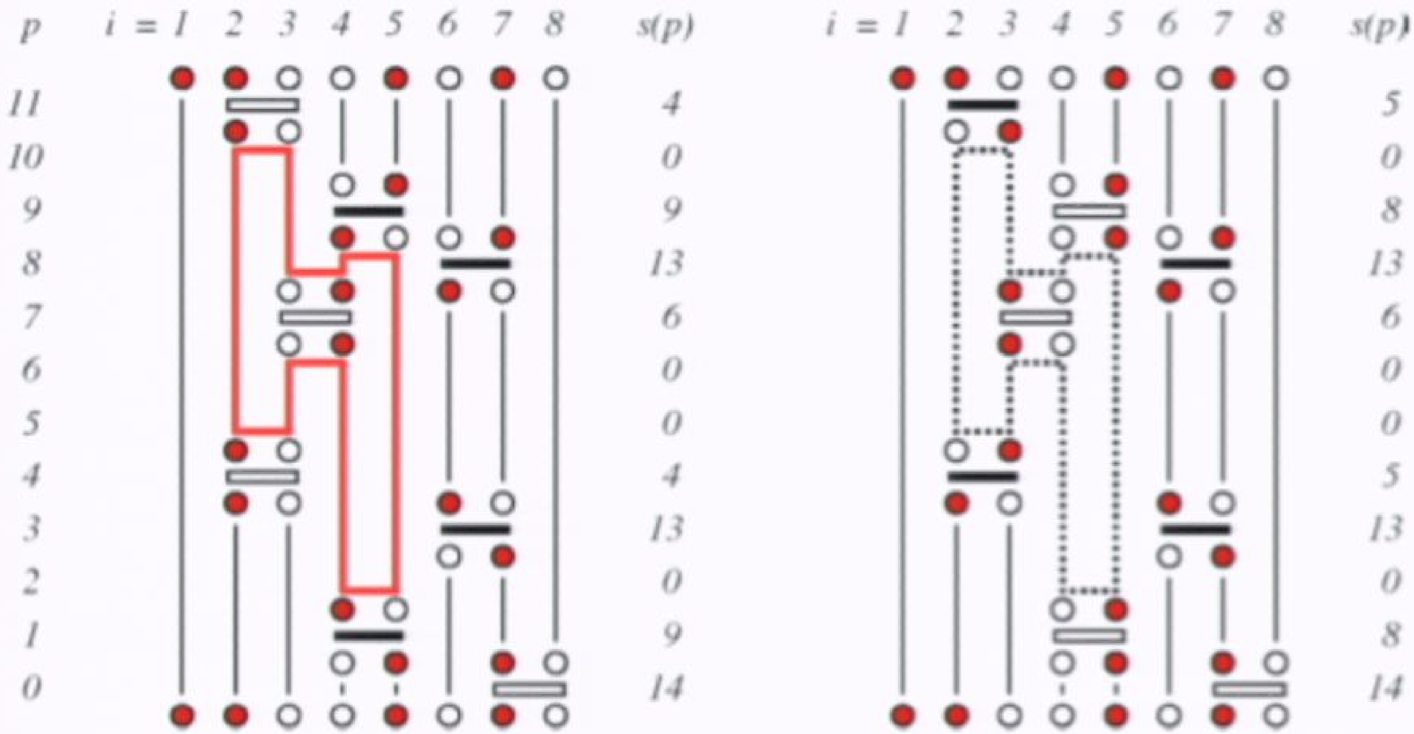


p	v ($l=0$)	$X(v)$ ($l=0$)	v ($l=1$)	$X(v)$ ($l=1$)	v ($l=2$)	$X(v)$ ($l=2$)	v ($l=3$)	$X(v)$ ($l=3$)
11	44	18	45	30	46	16	47	17
10	40	-	41	-	42	-	43	-
9	36	31	37	7	38	4	39	5
8	32	14	33	15	34	12	35	0
7	28	19	29	6	30	45	31	36
6	24	-	25	-	26	-	27	-
5	20	-	21	-	22	-	23	-
4	16	46	17	47	18	44	19	28
3	12	34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

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

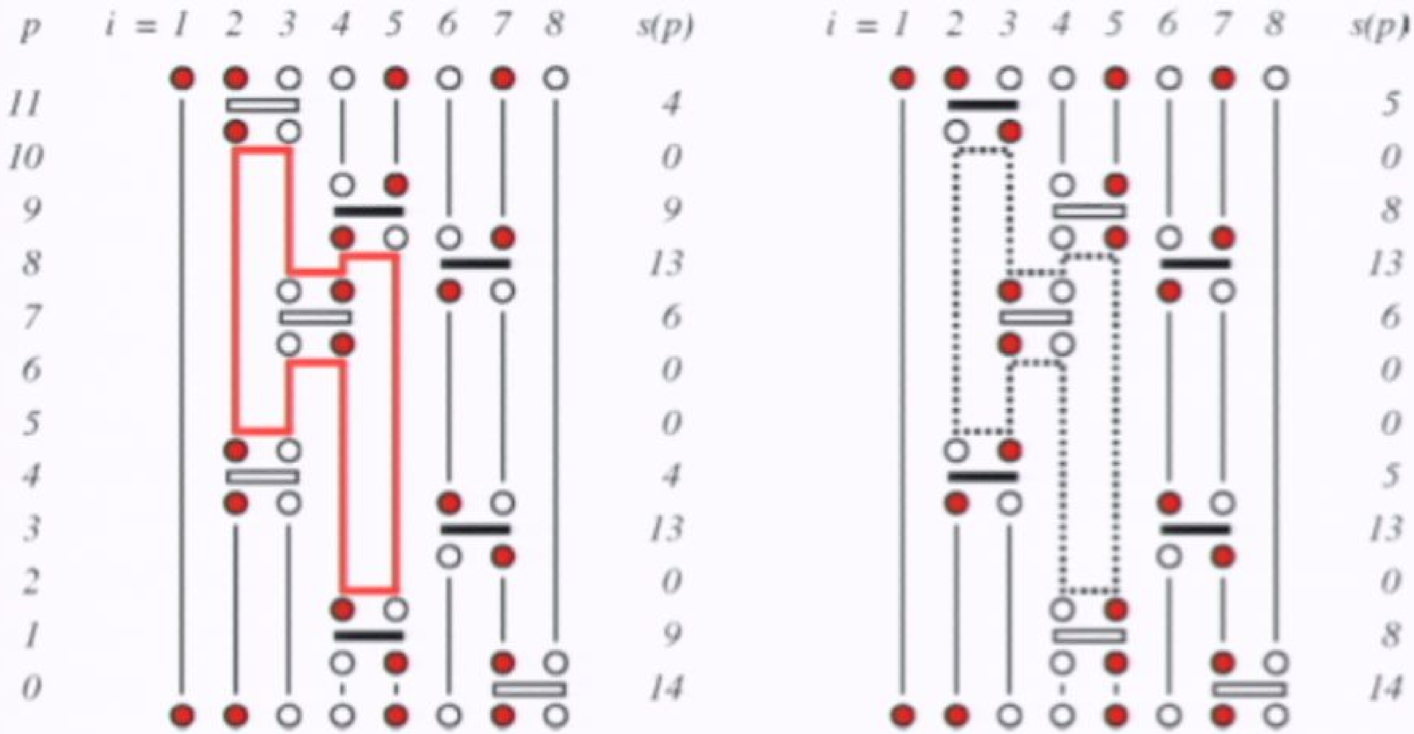
Operator-loop update

Many spins and operators can be changed simultaneously



)perator-loop update

Many spins and operators can be changed simultaneously



Pseudocode

moving horizontally in the list corresponds to changing v even↔odd

- **flipbit**(*v*,*0*) flips bit 0 of v

- a given loop is only constructed once
- **vertices can be erased**
- *X(v) < 0* = erased
- *X(v) = -1* not flipped loop
- *X(v) = -2* flipped loop

onstructing all loops, flip probability 1/2

lo $v_0 = 0$ **to** $4L-1$ **step** 2
  if $(X(v_0) < 0)$ **cycle**
  $v = v_0$
  if (**random**$[0-1] < \frac{1}{2}$) **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

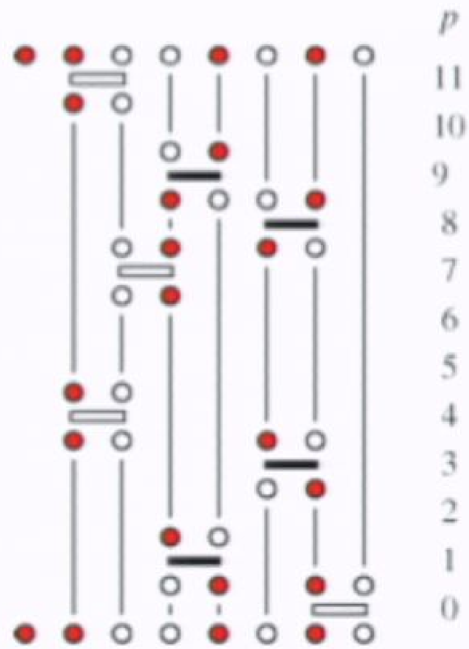
Constructing the linked vertex list

Traverse operator list $s(p)$, $p=0,\dots,L-1$

- 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 (l=0)	X(v) (l=0)	v (l=1)	X(v) (l=1)	v (l=2)	X(v) (l=2)	v (l=3)	X(v) (l=3)
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

Does it work?

Compare with exact results

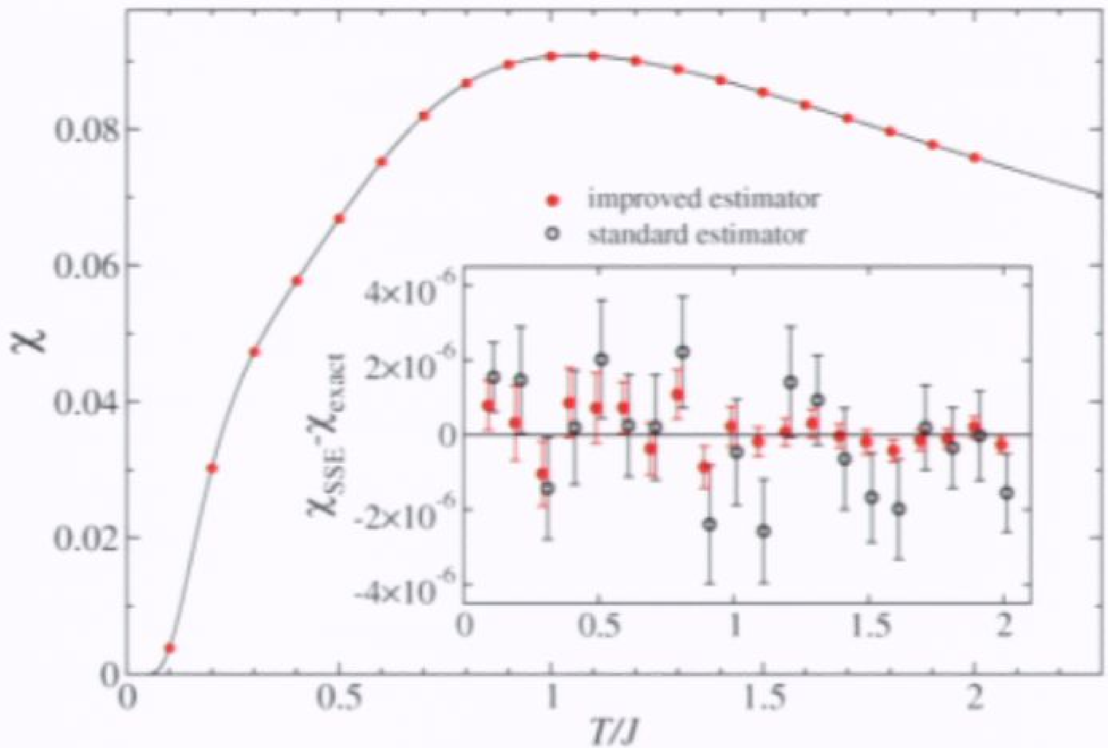
- 4×4 exact diagonalization
- Bethe Ansatz; long chains

Does it work?

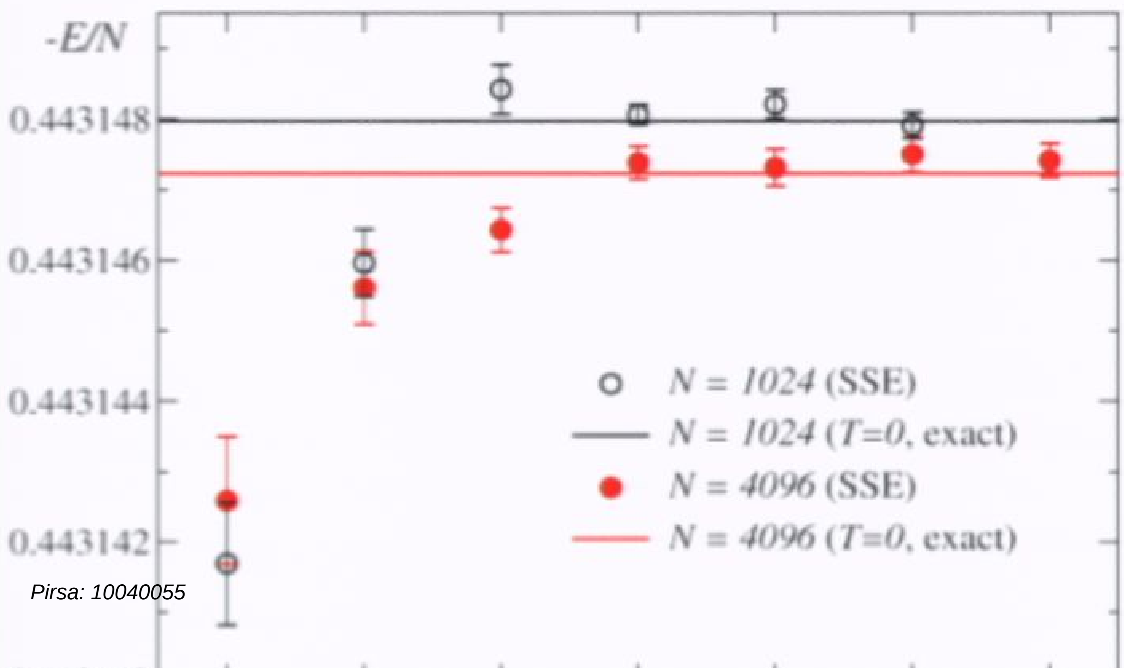
Compare with exact results

- 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)

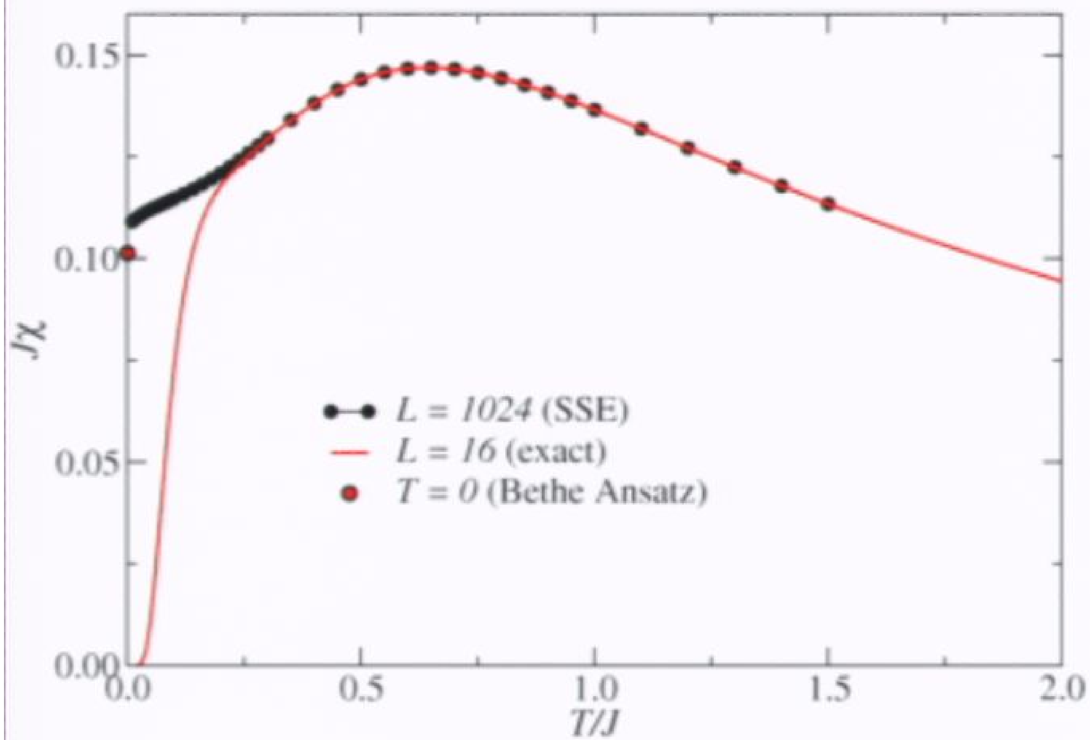


⇐ Energy for long 1D chains

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

Properties of the Heisenberg chain; large-scale SSE results

Properties of the Heisenberg chain; large-scale SSE results

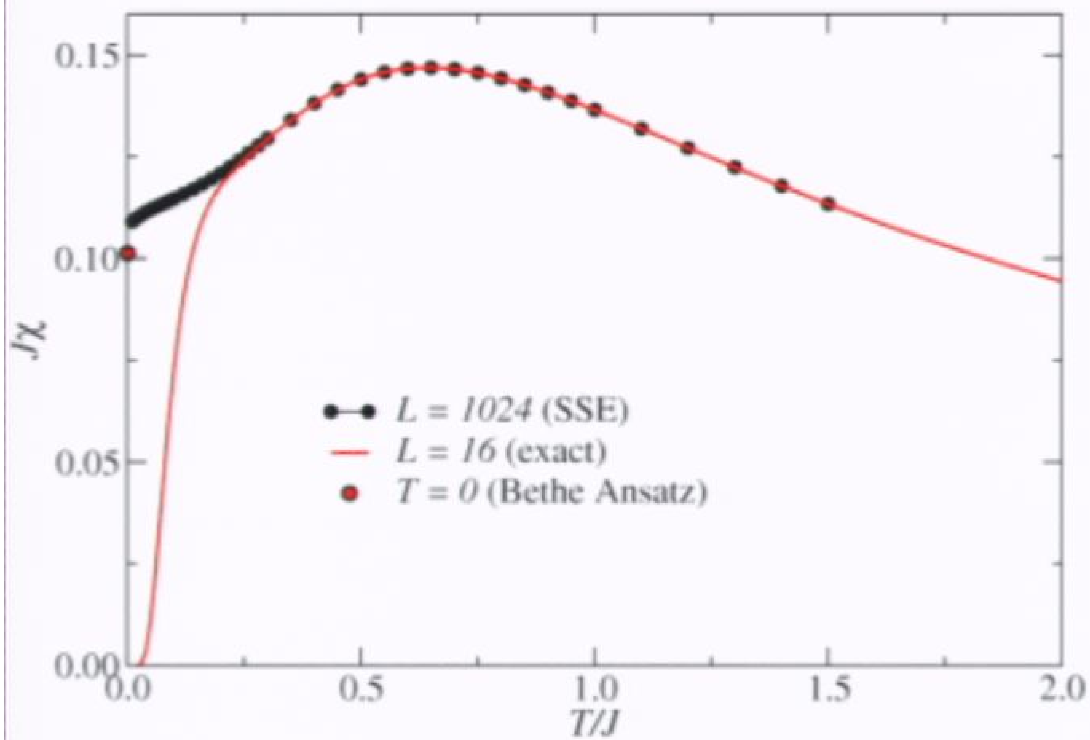


Magnetic susceptibility

anomalous behavior as T→0

- low-T results seem to disagree with known T=0 value obtained using the Bethe Ansatz method

Properties of the Heisenberg chain; large-scale SSE results



Magnetic susceptibility

anomalous behavior as T→0

- low-T results seem to disagree with known T=0 value obtained using the Bethe Ansatz method
- Reason: logarithmic correction at low T>0

Eggert, Affleck, Takahashi, PRL 73, 332 (1994)

$$\chi(T) = \frac{1}{2\pi c} + \frac{1}{4\pi c \ln(T_0/T)}$$

- Low-T form expected based on low-energy field theory

T=0 spin correlations

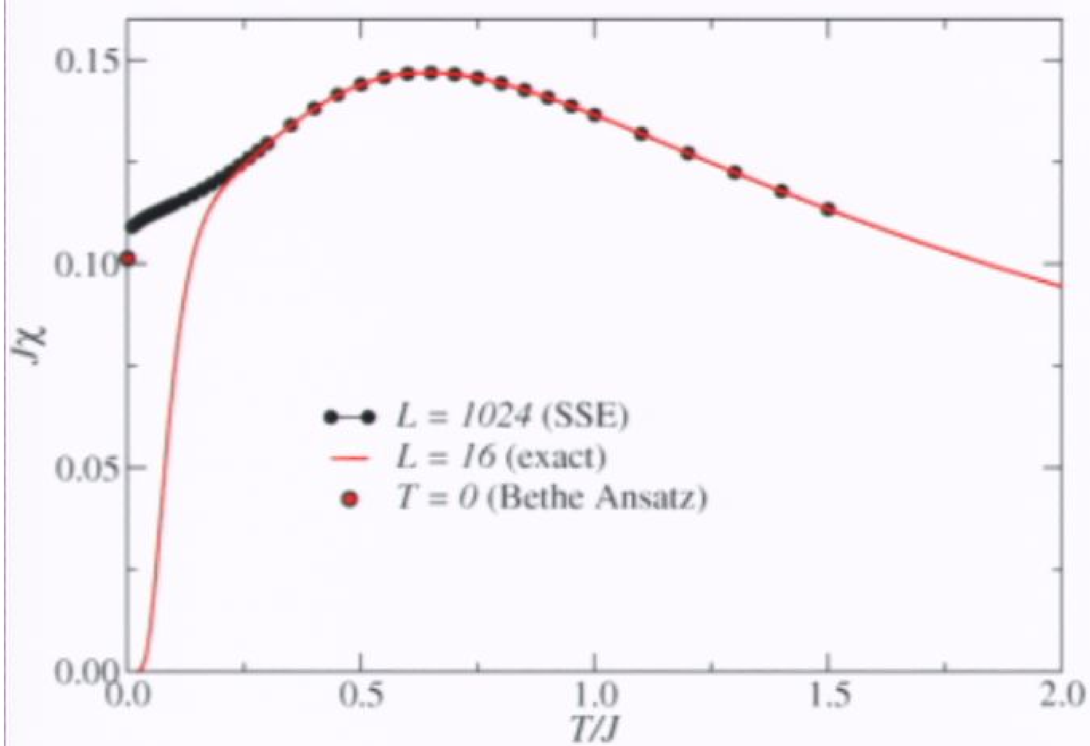
Low-energy field theory prediction

$$C(r) = A\frac{(-1)^r}{r}\ln\left(\frac{r}{r_0}\right)^{1/2}$$

SSE: converge to T=0 limit

- β dependence of *C(N/2)*, *N* = 4096 ⇒
- C(r) vs r and and r=N/2 ⇓

Properties of the Heisenberg chain; large-scale SSE results

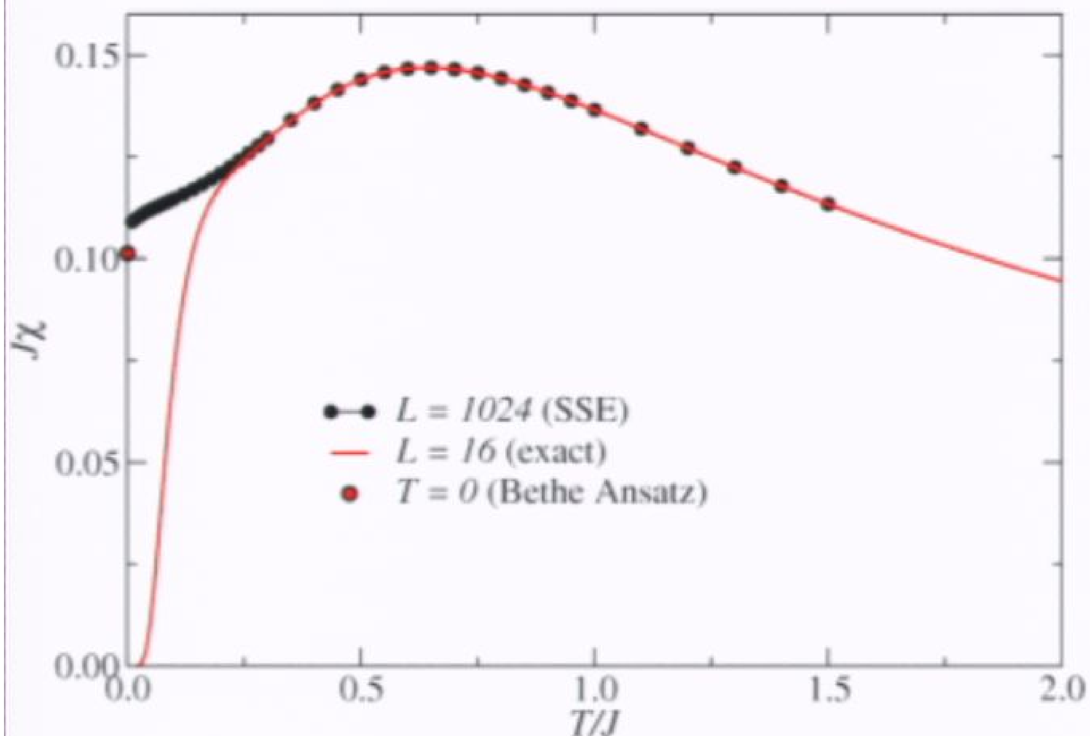


Magnetic susceptibility

anomalous behavior as T→0

- low-T results seem to disagree with known T=0 value obtained using the Bethe Ansatz method

Properties of the Heisenberg chain; large-scale SSE results



Magnetic susceptibility

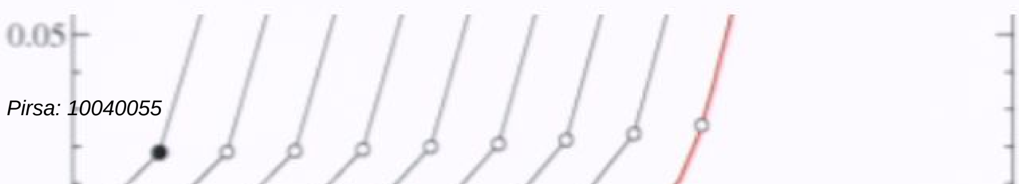
anomalous behavior as T→0

- low-T results seem to disagree with known T=0 value obtained using the Bethe Ansatz method
- Reason: logarithmic correction at low T>0

Eggert, Affleck, Takahashi, PRL 73, 332 (1994)

$$\chi(T) = \frac{1}{2\pi c} + \frac{1}{4\pi c \ln(T_0/T)}$$

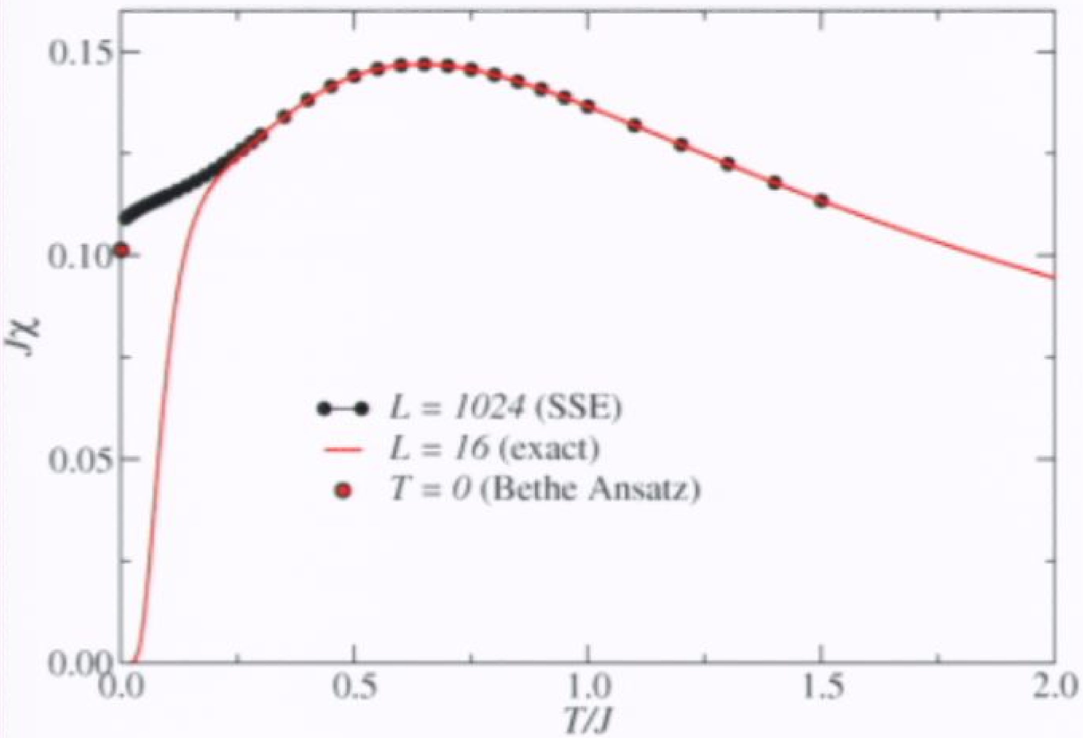
- Low-T form expected based on low-energy field theory

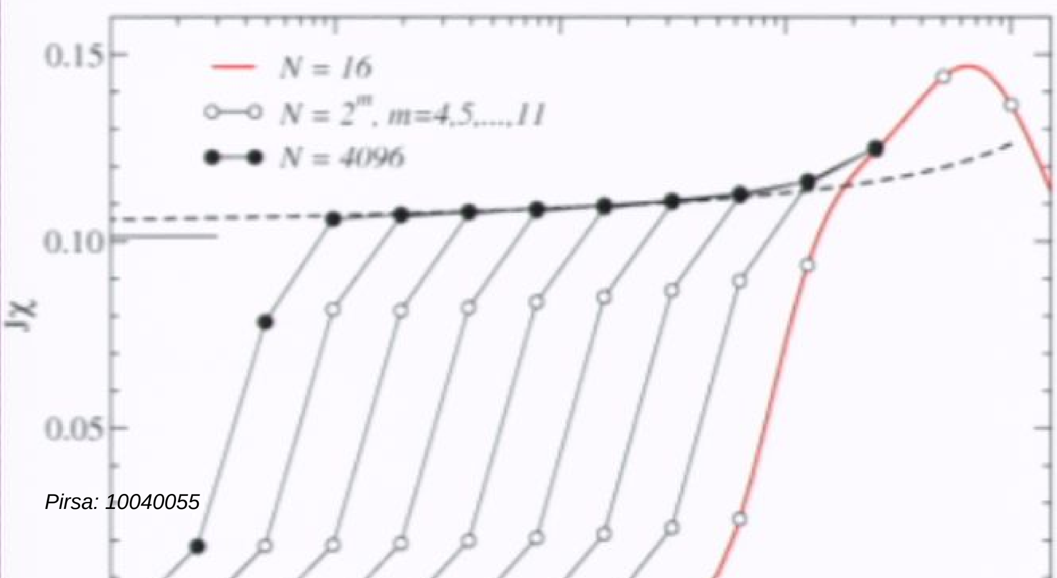


form, different parameters

Long chains needed for studying

Properties of the Heisenberg chain; large-scale SSE results





Magnetic susceptibility

anomalous behavior as T→0

- low-T results seem to disagree with known T=0 value obtained using the Bethe Ansatz method
- Reason: logarithmic correction at low T>0

Eggert, Affleck, Takahashi, PRL 73, 332 (1994)

$$\chi(T) = \frac{1}{2\pi c} + \frac{1}{4\pi c \ln(T_0/T)}$$

- Low-T form expected based on low-energy field theory
- For the standard chain

$$c = \pi J/2, \ T_0 \approx 7.7$$

- Other interactions → same form, different parameters

Long chains needed for studying

T=0 spin correlations

Low-energy field theory prediction

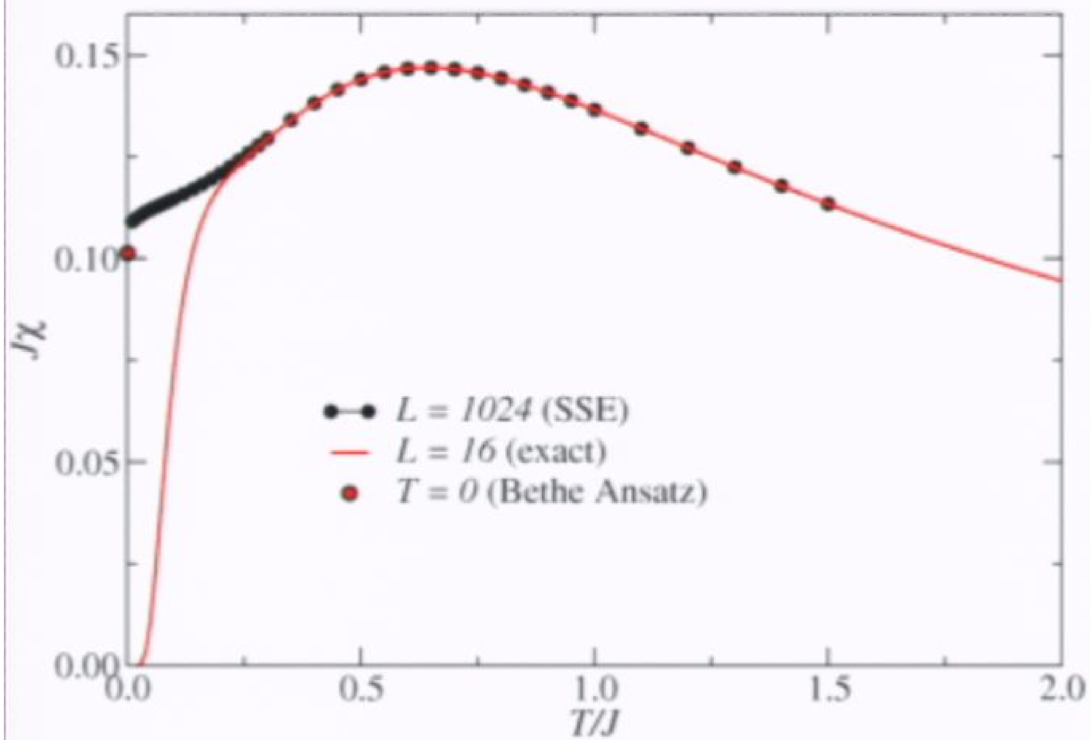
$$C(r) = A\frac{(-1)^r}{r}\ln\left(\frac{r}{r_0}\right)^{1/2}$$

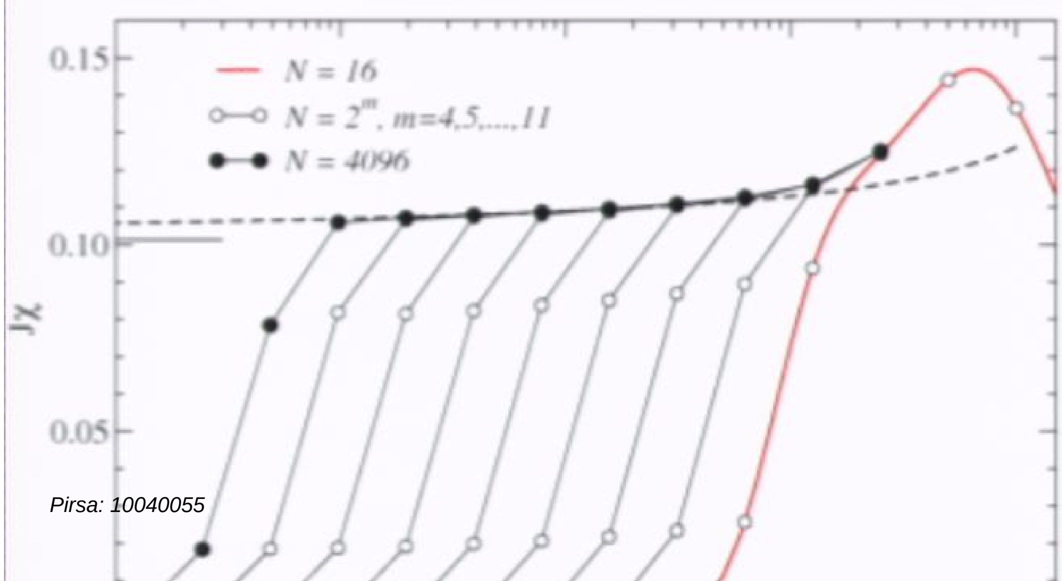
SSE: converge to T=0 limit

- β dependence of *C(N/2)*, *N* = 4096 ⇒
- C(r) vs r and and r=N/2 ⇓

Properties of the Heisenberg chain; large-scale SSE results

Properties of the Heisenberg chain; large-scale SSE results





Magnetic susceptibility

anomalous behavior as T→0

- low-T results seem to disagree with known T=0 value obtained using the Bethe Ansatz method
- Reason: logarithmic correction at low T>0

Eggert, Affleck, Takahashi, PRL 73, 332 (1994)

$$\chi(T) = \frac{1}{2\pi c} + \frac{1}{4\pi c \ln(T_0/T)}$$

- Low-T form expected based on low-energy field theory
- For the standard chain

$$c = \pi J/2, \;\; T_0 \approx 7.7$$

- Other interactions → same form, different parameters

Long chains needed for studying

T=0 spin correlations

Low-energy field theory prediction

$$C(r) = A\frac{(-1)^r}{r}\ln\left(\frac{r}{r_0}\right)^{1/2}$$

SSE: converge to T=0 limit

- β dependence of *C(N/2)*, *N* = 4096 ⇒
- C(r) vs r and and r=N/2 ⇓

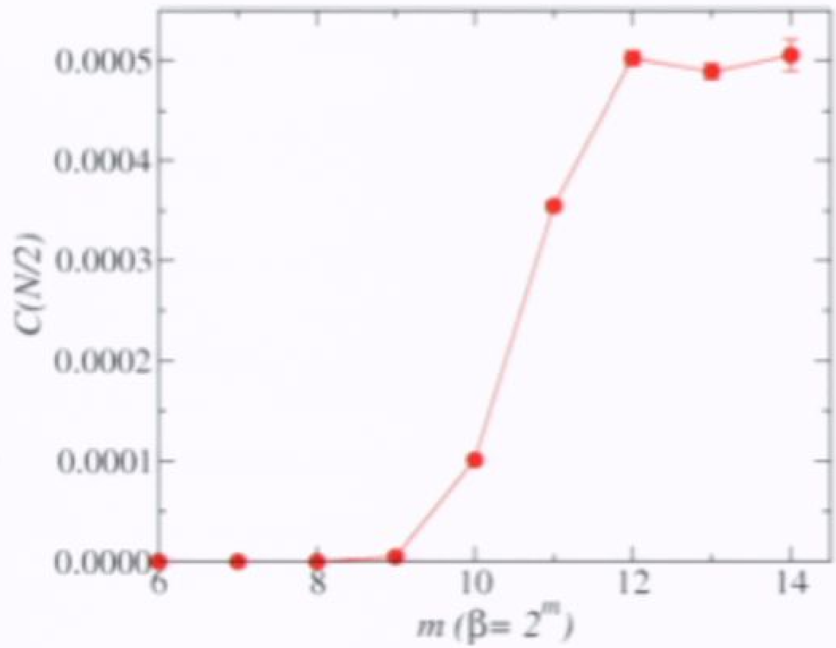
T=0 spin correlations

Low-energy field theory prediction

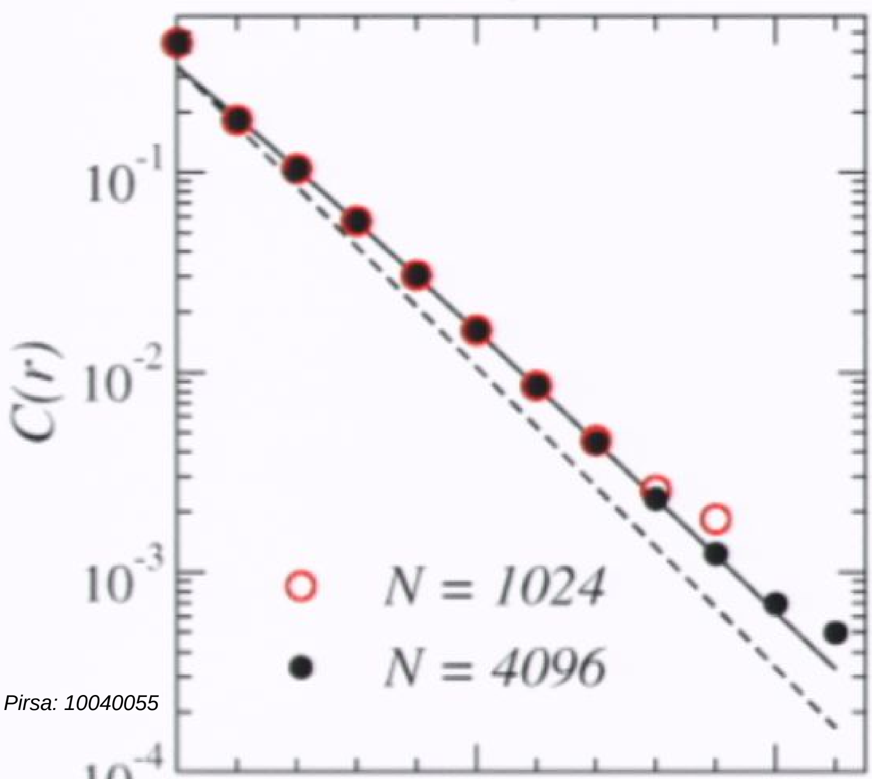
$$C(r) = A\frac{(-1)^r}{r}\ln\left(\frac{r}{r_0}\right)^{1/2}$$

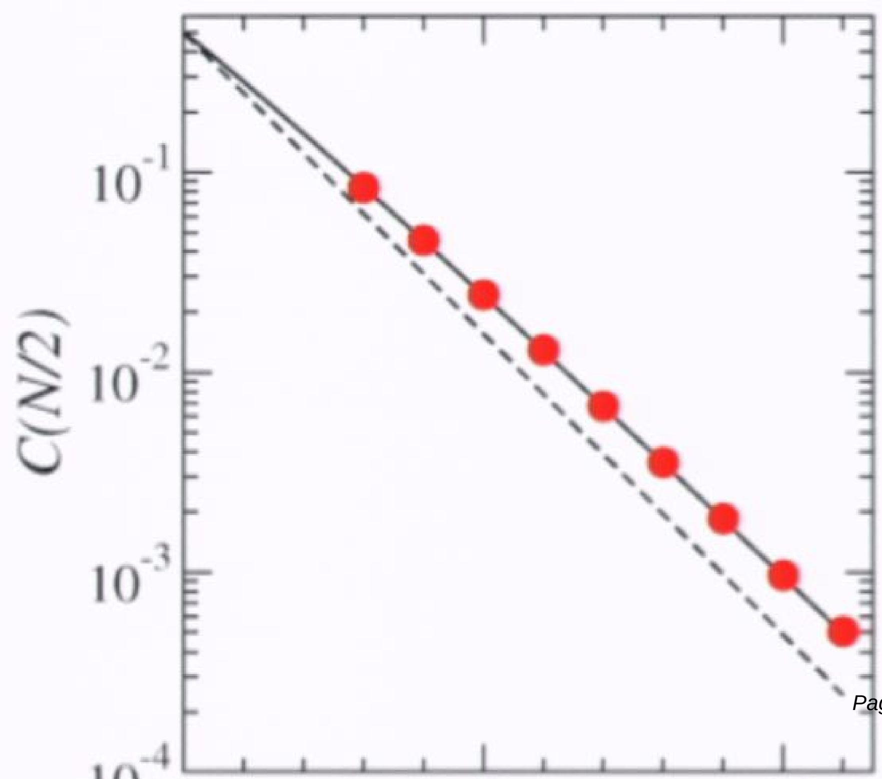
SSE: converge to T=0 limit

- β dependence of *C(N/2)*, *N* = 4096 ⇒
- C(r) vs r and and r=N/2 ⇓



A=0.21, r_0=0.08





Ladder systems

E. Dagotto and T. M. Rice, Science 271, 618 (1996)

Coupled Heisenberg chains; $L_x \times L_y$ spins, $L_y \to \infty$, L_x finite

- systems with even and odd L_y have qualitatively different properties
  - spin gap $\Delta > 0$ for L_y even, $\Delta \to 0$ when $L_x \to \infty$
  - critical state, similar to single chain, for odd L_y
  - the 2D limit is approached in different ways

Properties of Heisenberg ladders; large-scale SSE results

Magnetic susceptibility Low-T theoretical forms:

Odd L_y: from nonlinear -sigma model
Eggert, Affleck, Takahashi, PRL 73, 332 (1994)

$$\chi(T) = \frac{1}{2\pi c} + \frac{1}{4\pi c \ln(T_0/T)}$$

Even L_y: from large J_y/J_x expansion
Troyer, Tsunetsugu, Wurz, PRB 50, 13515 (1994)

$$\chi(T) = \frac{a}{\sqrt{T}} e^{-\Delta/T}$$

Ladder systems

E. Dagotto and T. M. Rice, Science 271, 618 (1996)

Coupled Heisenberg chains; $L_x \times L_y$ spins, $L_y \to \infty$, L_x finite

- systems with even and odd L_y have qualitatively different properties
  - spin gap $\Delta>0$ for L_y even, $\Delta \to 0$ when $L_x \to \infty$
  - critical state, similar to single chain, for odd L_y
  - the 2D limit is approached in different ways

Ladder systems

E. Dagotto and T. M. Rice, Science 271, 618 (1996)

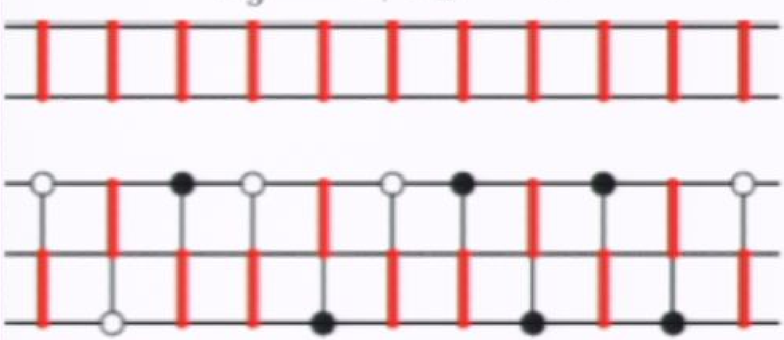
Coupled Heisenberg chains; $L_x \times L_y$ spins, $L_y \to \infty$, L_x finite

- systems with even and odd L_y have qualitatively different properties
  - spin gap $\Delta > 0$ for L_y even, $\Delta \to 0$ when $L_x \to \infty$
  - critical state, similar to single chain, for odd L_y
  - the 2D limit is approached in different ways

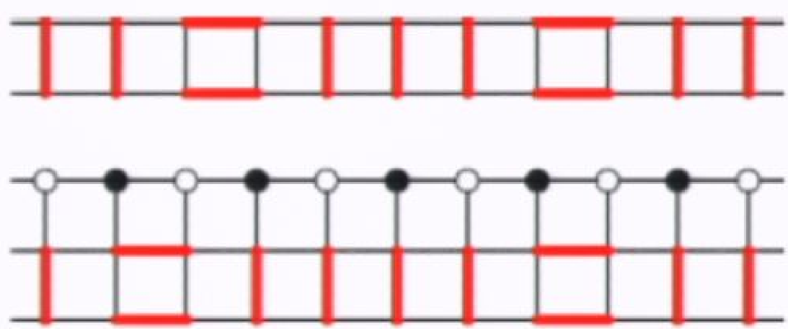
Consider anisotropic couplings; J_x and J_y

- the correct physics for all J_y/J_x can be understood based on large J_y/J_x
- short-range valence bond states

$J_y = 1, J_x = 0$



$0 < J_x/J_y \ll 1$



$(\uparrow\downarrow - \downarrow\uparrow)/\sqrt{2}$ $\uparrow$ $\downarrow$

$L_y = 2,4,\ldots$: $\Delta = J_y$ for $J_x = 0$

- gap persists for $J_x > 0$

$L_y = 3,5,\ldots$: $\Delta = 0$ for $J_x = 0$

Ladder systems

E. Dagotto and T. M. Rice, Science 271, 618 (1996)

Coupled Heisenberg chains; $L_x \times L_y$ spins, $L_y \to \infty$, L_x finite

- systems with even and odd L_y have qualitatively different properties
  - spin gap $\Delta > 0$ for L_y even, $\Delta \to 0$ when $L_x \to \infty$
  - critical state, similar to single chain, for odd L_y
  - the 2D limit is approached in different ways

Ladder systems

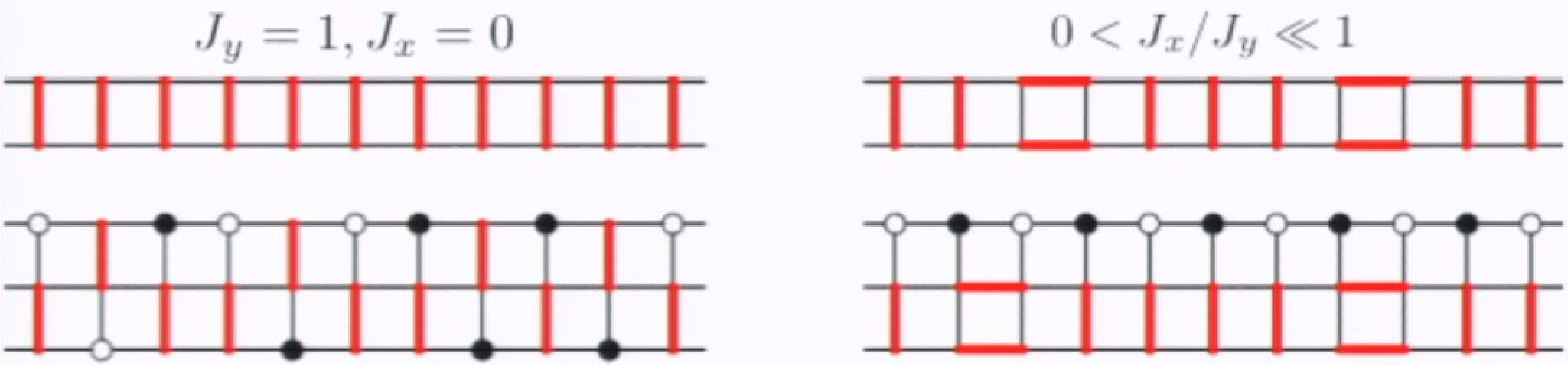
E. Dagotto and T. M. Rice, Science 271, 618 (1996)

Coupled Heisenberg chains; $L_x \times L_y$ spins, $L_y \to \infty$, L_x finite

- systems with even and odd L_y have qualitatively different properties
  - spin gap $\Delta > 0$ for L_y even, $\Delta \to 0$ when $L_x \to \infty$
  - critical state, similar to single chain, for odd L_y
  - the 2D limit is approached in different ways

Consider anisotropic couplings; J_x and J_y

- the correct physics for all J_y/J_x can be understood based on large J_y/J_x
- short-range valence bond states



$L_y = 2,4,\ldots$: $\Delta = J_y$ for $J_x = 0$

- gap persists for $J_x > 0$

$L_y = 3,5,\ldots$: $\Delta = 0$ for $J_x = 0$