Cluster algorithm for the Ising model The loop update is an example of a cluster algorithm. Detour:

Define bond index corresponding to pair of interacting spins

bond $b = 1, 2, ..., N_b$, interacting spins $\sigma_{i(b)}, \sigma_{i(b)}$

Write the energy of the Ising ferromagnet as Number of bonds $N_b = dN$ for a d-dimensional cubic lattice

$$
E = -|J| \sum_{b=1}^{N_b} [\sigma_{i(b)} \sigma_{j(b)} + 1] = - \sum_{b=1}^{N_b} E_b
$$

Write the partition function as

$$
Z = \sum_{\sigma} e^{-E(\sigma)/T} = \sum_{\sigma} \prod_{b=1}^{N_b} e^{E_b/T} = \sum_{\sigma} \prod_{b=1}^{N_b} [1 + (e^{E_b/T} - 1)]
$$

Define bond functions with arguments 0,1 (bond variable):

$$
F_b(0) = 1
$$

\n
$$
F_b(1) = e^{E_b/T} - 1
$$

\n
$$
Z = \sum_{\sigma} \prod_{b=1}^{N_b} [F_b(0) + F_b(1)]
$$

Introduce bond variables

 \overline{N}

$$
\tau_b = 0, 1, \tau = \{\tau_1, \tau_2, \ldots, \tau_{N_b}\}
$$

Partition function can be written as sum over spins and bonds

 \overline{N}

$$
Z = \sum_{\sigma} \prod_{b=1}^{N_b} [F_b(0) + F_b(1)] = \sum_{\sigma} \sum_{\tau} \prod_{b=1}^{N_b} F_b(\tau_b)
$$

The functions F_b depend on the spins:

$$
F_b(0) = 1
$$

\n
$$
F_b(1) = e^{E_b/T} - 1 = \begin{cases} e^{2|J|/T} - 1, & \text{if } \sigma_{i(b)} = \sigma_{j(b)} \\ 0, & \text{if } \sigma_{i(b)} \neq \sigma_{j(b)} \end{cases}
$$

 $\tau_b = 1$ allowed only between parallel spins Probabilities: For everything else fixed, probability for a given b

$$
P(\tau_b) = \frac{F(\tau_b)}{F(0) + F(1)} = \frac{F(\tau_b)}{e^{2|J|/T}}
$$

If parallel spins on bond b, probabilities for the bond variable

$$
P(\tau_b = 0) = e^{-2|J|/T}
$$
, $P(\tau_b = 1) = 1 - e^{-2|J|/T}$

If anti-parallel spins on bond b

$$
P(\tau_b = 0) = 1, \quad P(\tau_b = 1) = 0
$$

For a fixed bond configuration, spins forming clusters (spins connected by "filled" bonds) can be flipped and then give a configuration (term) with the same weight in Z $(F_b=1)$ for all bonds between clusters, F_b unchanged inside cluster).

 $N(\tau_b = 1) =$ No. of filled bonds $W = (e^{2|J|/T} - 1)^{N(\tau_b=1)}$ (unchanged after flip)

Spins not connected to any filled bonds are single-spin clusters

Swendsen-Wang algorithm

- Start from spin configuration
- Generate bond configuration
- Identify clusters of spins connected by bonds
- Flip each cluster with probability $1/2$
- Generate new bonds with the current spins, etc

SSE: Linked vertex storage for loop update

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

 $v X(v)$

47 17

5

 θ

 $|43|$

 39

 $|35|$

27

 23

 $I\bar{I}$

 $\overline{7}$

 \mathfrak{Z}

 $l = 3$

 $31|36$

 $19|28$

 $15|33$

37

 \overline{O}

2 3

$$
X() = \text{vertex list}
$$
\n• operator at p→X(v)\n v=4p+1, 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

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\textbf{if } (\textbf{random}[0-1] < \frac{1}{2}) \textbf{ then}traverse the loop; for all v in loop, set X(v) = -1else
         traverse the loop; for all v in loop, set X(v) = -2flip the operators in the loop
    endif
enddo
```
• visited vertices are no longer needed and we set them to a negative value -1 or -2 , to indicate that the loop has been visited (-1) or visited and flipped (-2)

construct and flip a loop

 $v = v_0$ do $X(v) = -2$ $p = v/4$; $s(p) =$ **flipbit**($s(p)$, 0) $v' = \text{flipbit}(v, 0)$ $v = X(v')$; $X(v') = -2$ if $(v = v_0)$ exit enddo

- p is the location of the operator in the original length-L list of operatotors
- **•** by flipping bit 0 of s(p), the operator changes from diagonal to offdiagonal, or vice 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_{first}() and X() to determine spins to be flipped • spins with no operators, V_{first}(i)=−1, flipped with probability 1/2

$$
\begin{aligned}\n\mathbf{do} \ i &= 1 \ \mathbf{to} \ N \\
v &= V_{\text{first}}(i) \\
\mathbf{if} \ (v = -1) \ \mathbf{then} \\
& \mathbf{if} \ (\mathbf{random}[0-1] < 1/2) \ \sigma(i) = -\sigma(i) \\
\mathbf{else} \\
& \mathbf{if} \ (X(v) = -2) \ \sigma(i) = -\sigma(i) \\
\mathbf{endif} \\
\mathbf{endo}\n\end{aligned}
$$

v=V_{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_{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

$$
V_{\text{first}}(:) = -1; V_{\text{last}}(:) = -1
$$
\ndo p = 0 to L - 1
\nif (s(p) = 0) cycle
\nv₀ = 4p; b = s(p)/2; s₁ = i(b); s₂ = j(b)
\nv₁ = V_{last}(s₁); v₂ = V_{last}(s₂)
\nif (v₁ \neq -1) then X(v₁) = v₀; X(v₀) = v₁ else V_{first}(s₁) = v₀ endif
\nif (v₂ \neq -1) then X(v₂) = v₀; X(v₀) = v₂ else V_{first}(s₂) = v₀ + 1 endif
\nV_{last}(s₁) = v₀ + 2; V_{last}(s₂) = v₀ + 3
\nenddo

creating the last links across the "time" boundary

do $i = 1$ to N $f = V_{\text{first}}(i)$ if $(f \neq -1)$ then $l = V_{\text{last}}(i)$; $X(f) = l$; $X(l) = f$ 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 $\Rightarrow \times$

- SSE results from 10¹⁰ 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 10⁶ sweeps
- **•** Bethe Ansatz ground state E/N
- **•** SSE can achieve the ground state limit (T→0)

Improved Estimators

SSE with loop updates is an example of a cluster algorithms - we can utilize improved estimators for many observables For a fixed bond configuration, spins forming clusters SE WIth loop updates is an example of a cluster algorithms we can utilize improved estimators for many observables

Classical example: Swendsen-Wang Ising cluster algorithm

 $N(\tau_b = 1) =$ No. of filled bonds $W = (e^{2|J|/T} - 1)^{N(\tau_b=1)}$ (unchanged after flip)

Write magnetization as sum over clusters of size n_C , sign s $_C$:</sub>

$$
M = \sum_{i=1}^{N} \sigma_i = \sum_{C=1}^{N_{\text{clus}}} \sum_{i \in C} \sigma_i = \sum_{C=1}^{N_{\text{clus}}} s_{C} n_{C} \qquad \langle M^2 \rangle = \sum_{C=1}^{N_{\text{clus}}} \sum_{C'=1}^{N_{\text{clus}}} \langle n_{C} n_{C'} s_{C} s_{C'} \rangle
$$

All cluster orientations (signs) have same weight - average over all 2^{Nclus} orientations →

$$
\langle M^2 \rangle = \sum_{C=1}^{N_{\rm clus}} \langle n_C^2 \rangle
$$

 $\frac{N_{\text{clus}}}{N}$ $\frac{N_{\text{clus}}}{N^2}$ This is the improved estimator of <M²> $\langle M \rangle = \sum_{C=1} \langle n_C \rangle$ - only depends on cluster structure

Improved estimators in SSE the staggered structure factor is simply given by

terms ⟨*ms*(*i*)*ms*(*j*)⟩ = 0 (for *i* ̸= *j*). One is then left with just the *i* = *j* contributions, and

Consider a given slice (propagated state) of an SSE configuration .
configi ∑ *n*2 *j* " e (propagated state)

 \cdot label the sites according to the loops passing through → clusters **nightary** passing the passing the phase of the phases of the phases

<u>In given loop</u> care takes the stage of the stage of the structure with the structure wit All spins on given $\qquad \qquad \bigoplus \bigoplus \Phi \qquad \bigoplus$ sub-lattice A or B are same, different on A, B \bigcirc **3 3 6** \bigcirc the fact that the true structure factor must be real-valued for any **q**. $\text{SUD-}\text{Iattice A or B are}$

in the SSE method $\tilde{1}$ 90]. The improvements in performance relative to local updates to local updates to local updates in performance relative to local updates in performance relative to local updates to local updates

 \circ

FIGURE 63. Asserve of the size of the size of the cluster - changes sign when loop flipped **FIGURE 61.** A linked-vertex SSE configuration with one loop traced out and shown in both of its in both of its Staggered magnetization on a cluster is 1/2 of the size of the cluster \sim 10 for the improved estimator than in code ${\overline{3}}$ for the simple simple ${\overline{3}}$ estaggered magnetization on a cluster is 1/2 of

for classical Monte Carlo simulations of the Ising model \mathcal{I}_1 as performed \mathcal{I}_2

 $\frac{1}{2}$ - similar to magnetization in SW algorithm - Sirinar to magnetization in ow-algontum

$$
\langle M_{z,{\rm stagg}}^2\rangle = \frac{1}{4}\sum_{C=1}^{N_{\rm clus}}\langle n_C^2\rangle
$$

 $S_{\rm eff}$ at other wave-vectors involve the full space-time loop structure, not just μ

5.2.5. Improved estimators The uniform magnetization requires the staggered phases tive world line system for the *S* = 1/2 Heisenberg model constructed using the discrete

$$
\chi = \frac{\beta}{4N} \left\langle \sum_{j=1}^{C} \left(\sum_{i=1}^{n_j} \phi_i \right)^2 \right\rangle \qquad \phi_i = \left\{ \begin{array}{c} +1 & i \text{ on A site} \\ -1 & i \text{ on B site} \end{array} \right.
$$

Transverse-field Ising model <u>**ISVEISE</u>**</u> " *j* <u>iield isiriy mode</u> **high for the random or the random or uniform and of any sign. The dimensionality is an adjusting to be exponentially** \cdot \circ , **Transverse-field Ising model** [PRE 68, 056701]

$$
H = \sum_{i,j} J_{ij} \sigma_i^z \sigma_j^z - h \sum_i \sigma_i^x
$$

l!1

strength of the interaction between spins *i* and *j*, which can

transverse field of strength *h*,

transverse field of strength *h*,

Elementary operators for the SSE strings: Arbitrary interactions (incl. random, long-range,…) strength of the interaction between spins *i* and *j*, which can **be Arbitrary interactions (incl. random, long-range,...) Elementary operators for the SSE strings:** written as a power-series expansion, with the trace expansion, $\frac{1}{2}$, $\frac{1}{2}$, $\frac{1}{2}$ ators for the SSE strings: Arbitrary interactions (incl. random, long-range,...) **Flementary** may act only on states with "*ⁱ* !" *^j z ^z* if *Ji j*&0 #ferromagnetic\$, **Elementary operators for the SSE strings:**

strength of the interaction between spins *i* and *j*, which can be random or uniform and of any sign. The dimensionality is arbitrary. Define the operators *H*0,0!1, #2a\$ *H*0,0!1, #2a\$ *Hi*,0!*h*#"*ⁱ* \$\$"*ⁱ* "\$, *i*%0, #2b\$ *Hi*,*i*!*h*, *i*%0, #2c\$ *Hi*, *^j*!!*Ji j*!"*Ji j*"*ⁱ z* " *j ^z* , *i*, *j*%0, *i*% *j*. #2d\$ *^Z*! ¹ *^L*! ! !)*n Hi*,*i*!*h*, *i*%0, #2c\$ *Hi*, *^j*!!*Ji j*!"*Ji j*"*ⁱ* " *j* , *i*, *j*%0, *i*% *j*. #2d\$ Up to a constant, the Hamiltonian can be written as has a fixed length. basis. Using Eq. #3\$ then gives *^Z*!! ! *ⁿ*! -+!. where *Sn* denotes a sequence of *n* operator-index pairs #here-In order to construct an efficient sampling scheme, it is useful to cut expansion #4\$ at some power *n*!*L*, sufficiently high for the remaining truncation error to be exponentially small and completely negligible &*L* clearly has to be 0)*NIN*(*J*)]. One can then obtain an expansion for which the length of the operator sequence is constant, by considering random insertions of *L*"*n* unit operators *H*0,0 in the An SSE configuration is illustrated in Fig. 1. The vertical direction in this representation will be referred to as the SSE *propagation direction*. It can be related to the imaginary-time direction in standard path-integral representations &26'. Note that this full configuration, including all the states !+(*p*)/ explicitly, does not have to be stored in the simulation. A

Hi,0!*h*#"*ⁱ* \$\$"*ⁱ* "\$, *i*%0, #2b\$ *Hi*,*i*!*h*, *i*%0, #2c\$ erators!. All possible vertices for the transverse Ising model are shown in Fig. 3. Note that only those Ising vertices that are compatible with the sign of the interaction between a FIG. 3. All the possible four-leg and two-leg vertices. "a! Ferromagnetic Ising vertices, "b! antiferromagnetic Ising vertices, "c! constant vertices, and "d! spin-flip vertices. Up to a constant, the Hamiltonian can be written as *^H*!"! *N* ! *N Hi*, *^j* . #3\$ The constants *Hi*,*ⁱ* are introduced for purposes that will be-*^H*!"! *i*!1 *N* ! *j*!0 *N Hi*, *^j* . #3\$ The constants *Hi*,*ⁱ* are introduced for purposes that will become clear below. Note that *H*0,0 is not included as a term in in *SL* : *Sn*!&*i*#1\$, *j*#1\$',...,&*i*#*n*\$, *j*#*n*\$', #5\$ with *i*(*l*)!(1, . . . ,*N** and *j*(*l*)!(0, . . . ,*N**. The standard basis (!+/*!(!"¹ ,...," *^N ^z* /* is used. Because of the constants added to *Hi*, *^j* in Eq. #2d\$, the product in Eq. #4\$. Adjusting for the (*ⁿ Z*! 1 *^L*! ! + ! *SL*)*n* #*L*"*n*\$!-+!. *l*!1 *L Hi*(*l*), *^j*(*l*)!+/, #7\$ single state and the operator sequence suffice for reproducing all the states, and such a representation is used in some stages of the simulation. For some updates it is convenient to generate other representations, as will be discussed below. **B. Local updates** The sampling of Eq. #7\$ can be carried out using simple

Possible local u " *j z* , *i*, *j*%0, *i*% *j*. #2d\$ come clear below. Note that *H*0,0 is not included as a term in Possible local updates:

^H!"!

as a sum over diagonal matrix elements in a suitably chosen

basis. Using Eq. #3\$ then gives

has a fixed length. The fixed length $\mathcal{L}_{\mathcal{A}}$

$$
[0,0]_p \leftrightarrow [i,j]_p, \quad i,j \neq 0,
$$
 (diagonal)

$$
[i,i]_{p_1}[i,i]_{p_2} \leftrightarrow [i,0]_{p_1}[i,0]_{p_2}, \quad i \neq 0
$$
 (off-diagonal)

or "*ⁱ z*

Hence, the size of the size of the size of the operator sequence to be stored in com-

!"" *^j*

!

where the subscript *p* indicates the position (*p*!1, . . . ,*L*) of

ing random insertions of *L*"*n* unit operators *H*0,0 in the

In order to construct an efficient sampling scheme, it is

system. Here the truncation *L*#49, and the expansion order of the

Cluster update for Transverse-field Ising models legs on the stack. The reason that the cluster can always be moving a Hamiltonian operator, there are no constraints and !)
(1)(1)(2)(2)(2)(3)(2)(3)(3)(3) <u>p process to processe the contract of the course of</u>

trix element of the Ising bond operator is not affected when Start at arbitrary in-leg and apply weight-preserving operator changes Start at arbitrary in-leg and apply weight-preserving operator changes

- growing "to do" list (stack) from each out-leg

all \uparrow \uparrow \uparrow \rightarrow arowing "to do" list (stack) from each ou
- until stack is empty generated successively. For an $\frac{1}{2}$ $\frac{1}{2}$ update $\frac{1}{2}$ in $\frac{1}{2}$ in $\frac{1}{2}$ is empty. \blacksquare until stack is empty
- divides the entire system into clusters

updating cycle, the operator sequence *SL* and the state

- similar to classical S-W algorithm **-** Similar to classical 3-W algont **probability. In the similar to classical S-W algorithm** eration to eraboreal of the arguments

Long-range interactions <u>spin grows. This thin short horizontal</u> ong-range interactions

Goal: avoid summations over interactions J_{ij} indices *i*, *j* are left undetermined and it is assumed that any abai: avoid summations over interactions a_{ij}

 $F = F \cdot F$ - we can first assume that all diagonal operators are allowed, then

$$
P([0,0] \to [i,j]) = \frac{\beta \left(Nh + 2\sum_{ij} |J_{ij}|\right)}{L - n + \beta \left(Nh + 2\sum_{ij} |J_{ij}|\right)}
$$

$$
P([i,j] \to [0,0]) = \frac{L - n + 1}{L - n + 1 + \beta \left(Nh + 2\sum_{ij} |J_{ij}|\right)}
$$

where [i, j] in [0,0] > [i, i] is a still independent
periodic functions

where [i,j] in [0,0]->[i,j] is a still undetermined interaction normalized cumulative probabilities *Pc*(*k*!1, . . . ,*N*) are where $\left[\begin{matrix}1, \end{matrix}\right]$ in $\left[\begin{matrix}0, 0\end{matrix}\right]$ is a suit undetermined interaction achieving \mathbb{R} the spin-flip \mathbb{R} of two spin-flipping spin-flipping spin-flip wriere $\left[\!\left[1, \right]\!\right]$ in $\left[\!\left[0, 0\right]\!\right]$ is a suil driue *P* = [l,j] in [0,0]->[l,j] is a still ure.
Pactual li il is chosen in a se

- the actual [i,j] is chosen in a second step using cumulative probabilities: \mathbf{p} as \mathbf{p} as usual. The probability of \mathbf{p} 2010 Bullin a Second St ,

$$
P_c(k) = \frac{\sum_{i=1}^{k} P(i)}{\sum_{i=1}^{N} P(i)}
$$
 N is the total num
Choose op #k usii
- time scales as ln

inserted, a random number 0&*R*"1 is generated. The table

spin state. This is the reason for the advantage of this simu-

lowed if !and only if" no Ising operators acting on site *i* are

Crioose op # search the cumulative probability table"s# in the case of longrhoer of diagonal ops. **Partitions** dating all of them according to Eq. "9b# requires on the order Choose op #k using bisection in the table Pc() (*p*1), . . . ,!%*ⁱ* (*p*² prefactors in Eq. !7" when *n*→*n*%1: of *L* operations. Hence, the number of operations for a full \mathbf{u} $\sum_{i=1}^{k} P(i)$ and in the total number of diagonal ops - time scales as ln(N) of the SSE configuration. However, the off-diagonal update of k *%¹ #*L*!!*n*%1"\$! านm $P_{\mu}(k) = \frac{i-1}{N}$ Choose op #k using $\frac{1}{2}$ $\frac{1}{2}$ $\frac{1}{2}$ as image as in

 $\frac{1}{2}$ step $\frac{1}{2}$ interaction. Cluster update remains the same for any kind of interaction

- total time for updating sweep scales as β NIn(N) with long-range interactions *interaction, and that the update requires no knowledge of the* **Example 3** *is* 8 *NIn*(N) with long-range interactions ements of all #*i*, *j*\$ operators. Staying with the assumption $\frac{1}{2}$ and $\frac{1}{2}$ are $\frac{1}{2}$ *ng* sweep scales as βN

generated successively. For an #*i*, *j*\$→#0,0\$ update, i.e., re-

calculate the weight change when flipping a spin interacting

that any #*i*, *j*\$ is allowed in the update #0,0\$→#*i*, *j*\$, the rela-