

## Cluster

- Vicino alla temperatura critica gli spin formano grandi gruppi (cluster)
- all'interno di ogni gruppo è molto improbabile riuscire a flippare uno spin perché ci sarebbe una grande perdita di energia,
- l'algoritmo di Metropolis diventa molto lento
- è conveniente flippare un gruppo di spin uguali tutti allo stesso tempo

## Tempo di correlazione

- Nel calcolare il valore di  $E$  e  $M$  ho fatto la media su diversi stati alla stessa temperatura
- Questi stati, per dare una media significativa, devono essere statisticamente indipendenti
- Il tempo  $\tau$  che deve trascorrere perché abbia stati statisticamente indipendenti è il tempo di correlazione
- Vicino alla transizione di fase, gli spin si raggruppano in cluster dello stesso segno con dimensione tipiche di  $\xi$ , che viene detta lunghezza di correlazione
- Vicino alla transizione di fase

$$\xi \sim \left( \frac{T_c}{T - T_c} \right)^\nu \quad \text{e} \quad \tau = \xi^z$$

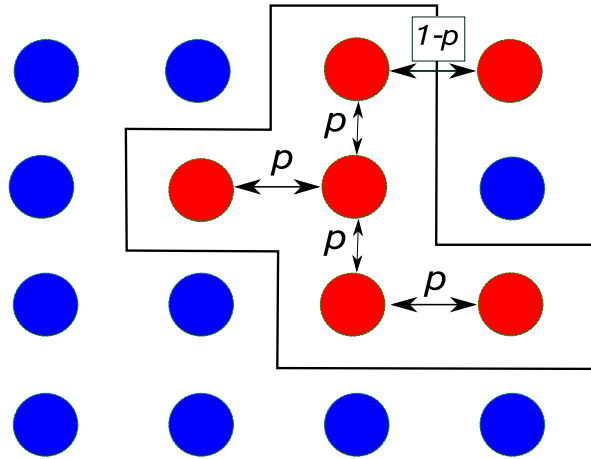
- Il numero  $z$  dipende dall'algoritmo usato, mentre  $\nu$  è universale. Per Metropolis  $z \approx 2.1$

## Algoritmi per cluster

- Considero un algoritmo che mi cambi il segno dello spin di un intero gruppo di spin in un colpo solo
- Scelgo il cluster in modo che non comprenda tutti gli spin dello stesso segno
- il tempo di correlazione sarà molto minore
- Devo soddisfare le condizioni di ergodicità e bilancio dettagliato
- La prima è soddisfatta se posso avere cluster di un solo spin, ricadendo quindi nel caso di Metropolis

## Metodo di Wolff

- Prendo un singolo spin di segno arbitrario
- Mi guardo intorno per vedere se ci sono altri spin ugualmente orientati
- Aggiungo ognuno di questi al cluster con probabilità  $P_{add}$
- Ripeto il procedimento ricorsivo per ciascuno dei siti aggiunti al cluster
- Se arrivo a un sito attraverso due link diversi, devo cercare di aggiungerlo al cluster due volte. In pratica è come se aggiungessi i link piuttosto che i siti



Quando la procedura sarà terminata il cluster avrà dei precisi confini, ed i legami con gli spin fuori dal cluster saranno  $m$  con spin uguali e  $n$  con spin opposti. La differenza di energia è quindi  $\Delta E = 2(m - n)$  La probabilità di selezionare questo cluster sarà quindi

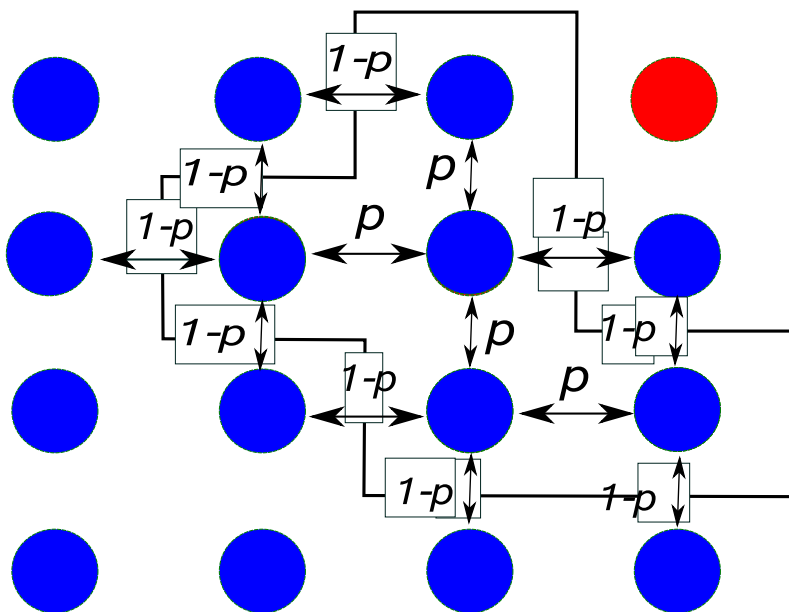
$$P_{add}^i \cdot (1 - P_{add})^m$$

dove il primo pezzo si riferisce ai legami interni al cluster, il secondo a quelli esterni

## Bilancio dettagliato per il metodo di Wolff

Suppongo ora di cambiare lo stato di tutti gli spin del cluster, e di voler calcolare la probabilità di selezionare lo stesso cluster di prima, ma con spin invertiti. In questo caso avrò  $n$  legami con spin uguali e  $m$  con spin opposto. La probabilità di ottenere lo stesso cluster sarà quindi

$$P_{add}^i \cdot (1 - P_{add})^n$$



A questo punto posso applicare il bilancio dettagliato

$$\frac{g(i \rightarrow j)A(i \rightarrow j)}{g(j \rightarrow i)A(j \rightarrow i)} = \frac{(1 - P_{add})^m A(i \rightarrow j)}{(1 - P_{add})^n A(j \rightarrow i)}$$

Scegliendo  $P_{add} = (1 - e^{-2\beta})$  trovo

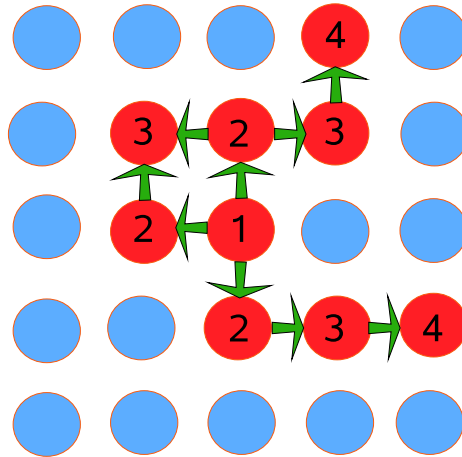
$$\frac{e^{-2m\beta} A(i \rightarrow j)}{e^{-2n\beta} A(j \rightarrow i)} = e^{-2\beta(m-n)}$$

che soddisfa il bilancio dettagliato nel modo migliore perché consente di scegliere le ampiezze di transizione  $A(i \rightarrow j)$  uguali ad uno in tutti i casi.

Per Wolff  $z \approx 0.25$

## Implementazione numerica

L'algoritmo di Wolff si presta ad un procedimento ricorsivo



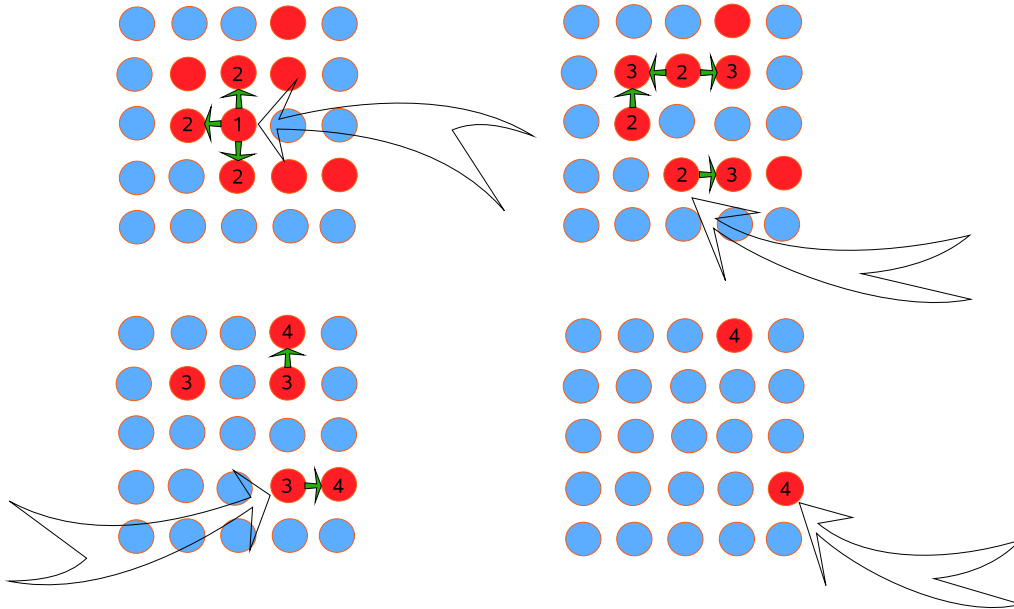
1. Parto da un sito a caso
2. guardo i primi vicini
  - se sono di spin diverso li trascuro
  - se hanno lo stesso spin li aggiungo con probabilità  $P_{add} = (1 - e^{-2\beta})$
3. ripeto per tutti i siti selezionati in questo modo



## Algoritmo con stack

- Sostituisce le chiamate ricorsive con l'uso di uno stack, cioè un vettore in memoria
- lo stack è di tipo LIFO (Last In First Out)
- ogni sito aggiunto al cluster viene messo nello stack
- c'è un puntatore all'ultimo elemento dello stack
- flippo l'ultimo elemento dello stack e cerco siti da aggiungere al cluster tra i suoi primi vicini
- se non ho aggiunto nuovi siti, torno indietro nello stack
- il cluster è terminato e completamente flip-pato quando lo stack è vuoto

Se immagino di selezionare i primi vicini in senso antiorario, indico qual è l'elemento in cima allo stack dopo ogni iterazione



La sequenza di elementi sullo stack è

$(2, 2) \rightarrow (1, 2), (2, 1), (3, 2) \rightarrow (1, 2), (2, 1), (3, 3) \rightarrow$   
 $\rightarrow (1, 2), (2, 1), (3, 4) \rightarrow (1, 2), (2, 1) \rightarrow (1, 2), (1, 1)$   
 $\rightarrow (1, 2) \rightarrow (0, 2) \rightarrow vuoto$

## Quale metodo è più veloce?

- Decido come misurare il tempo impiegato dal programma: una possibilità è contare le chiamate al generatore di numeri casuali
- due programmi hanno le stesse prestazioni se danno la stessa precisione nello stesso tempo
- misuro la precisione dalla varianza dell'energia per sito ( $N$  è il numero di stati su cui faccio la media)

$$\sigma_N^2 = \frac{\langle (E - \langle E \rangle)^2 \rangle}{N}$$