Critical behaviour of the surface tension in the 3D Ising model

Federico Belliardo
Marco Costa

Dipartimento di Fisica
Università di Pisa

September 1, 2018

Summary

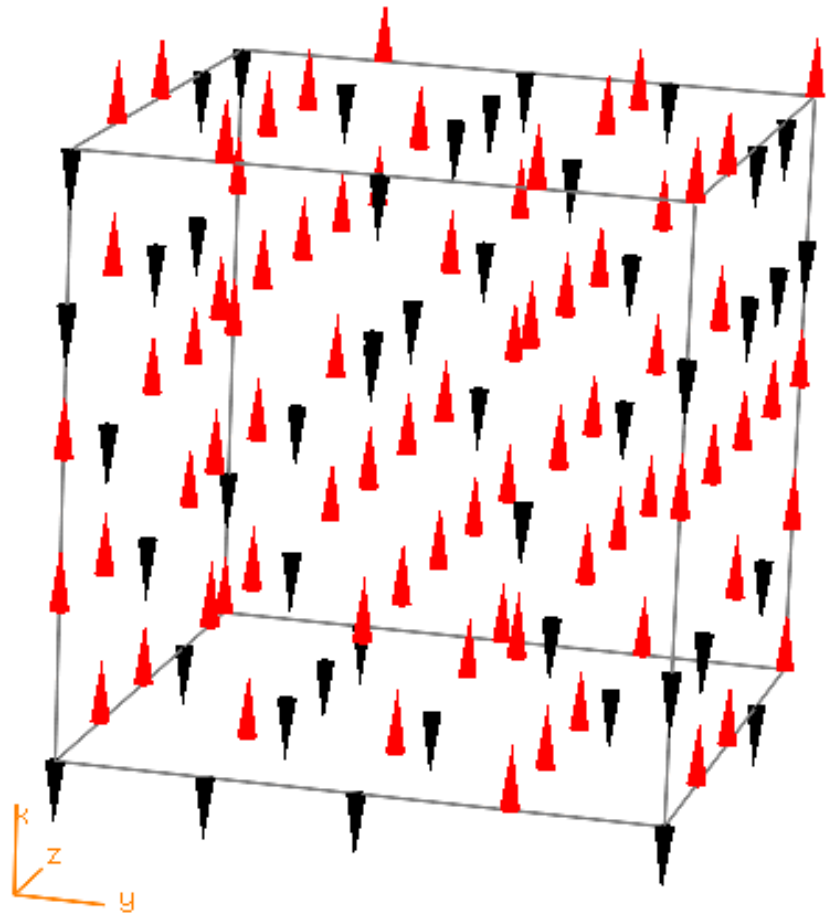
- 3D Ising models
- Definition of the surface tension
- Cluster algorithms and boundary flip
- (Notes on the implementation?)
- Estimation of the errors and autocorrelation
- Fit of the free energy
- Fit of the critical behaviour
- (Conclusion?)

3D Ising model

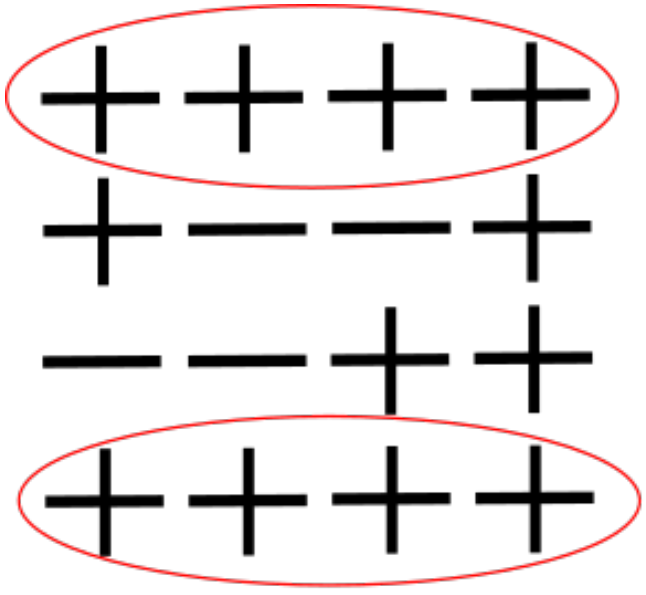
$$\mathcal{H} = -\sum_{\langle x,y \rangle} J_{\langle x,y \rangle} s_x s_y$$

$J_{\langle x,y \rangle} = 1$ ferromagnetic

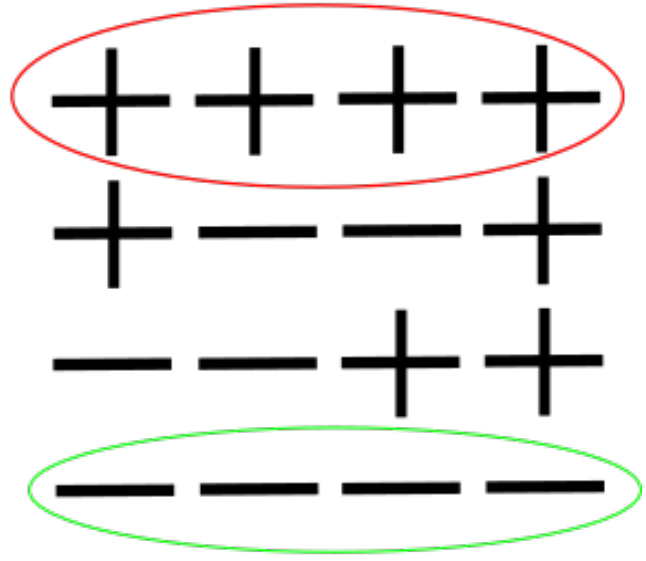
$J_{\langle x,y \rangle} = -1$ antiferromagnetic



Definition of the surface tension



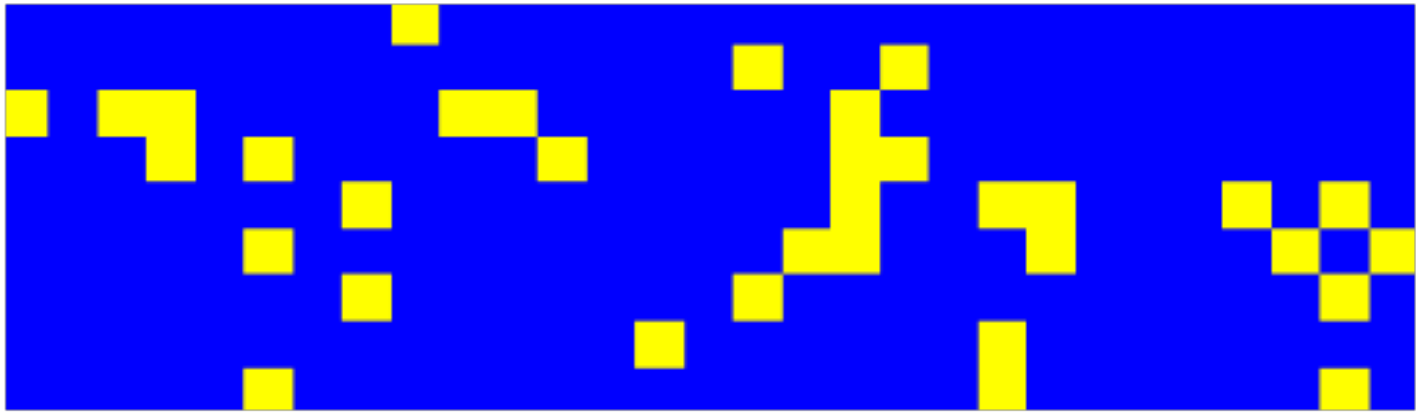
Z_{++}



Z_{+-}

$$\sigma = -\lim_{L\to\infty} \frac{1}{L^2} \log \frac{Z_{+-}}{Z_{++}} \qquad L \times L \times T,\ T = cL$$

Z_{+-}



Z_{++}

$$\sigma = -\lim_{L\to\infty} \frac{1}{L^2} \log \frac{Z_{+-}}{Z_{++}} = \lim_{L\to\infty} \frac{1}{L^2} (F_{+-} - F_{++}) = \lim_{L\to\infty} \frac{F_s}{L^2}$$

$\sigma =$ interface free energy per unit area

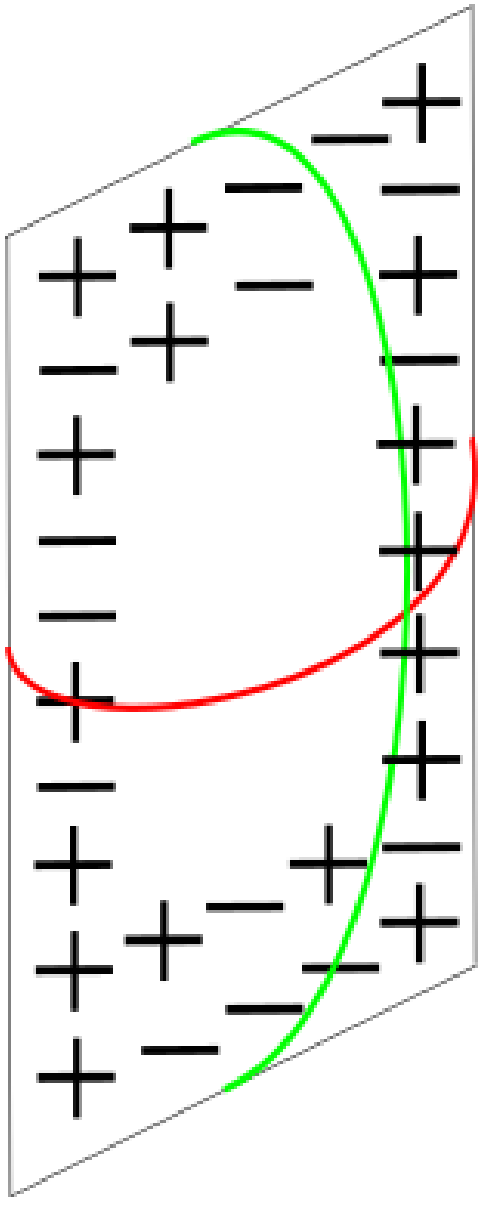
Redefinition of Z_{++} and Z_{+-}.

$Z_{++} \rightarrow$ ferromagnetic link between **top and bottom**.
$Z_{+-} \rightarrow$ **anti**ferromagnetic link between **top and bottom**.

Always ferromagnetic link in x and y.

Same definition for σ.

Periodic boundary conditions reduce the finite size effect.



Montecarlo simulations can't measure Z!

Solution: $J_{\langle x,y\rangle}$ between top and bottom becomes a **dinamical variable** that is summed over in Z.

$J_{\langle x,y\rangle} = 1$ (periodic b.c.) $\quad J_{\langle x,y\rangle} = -1$ (antiperiodic b.c.)

Other $J_{\langle x,y\rangle}$ remains ferromagnetic.

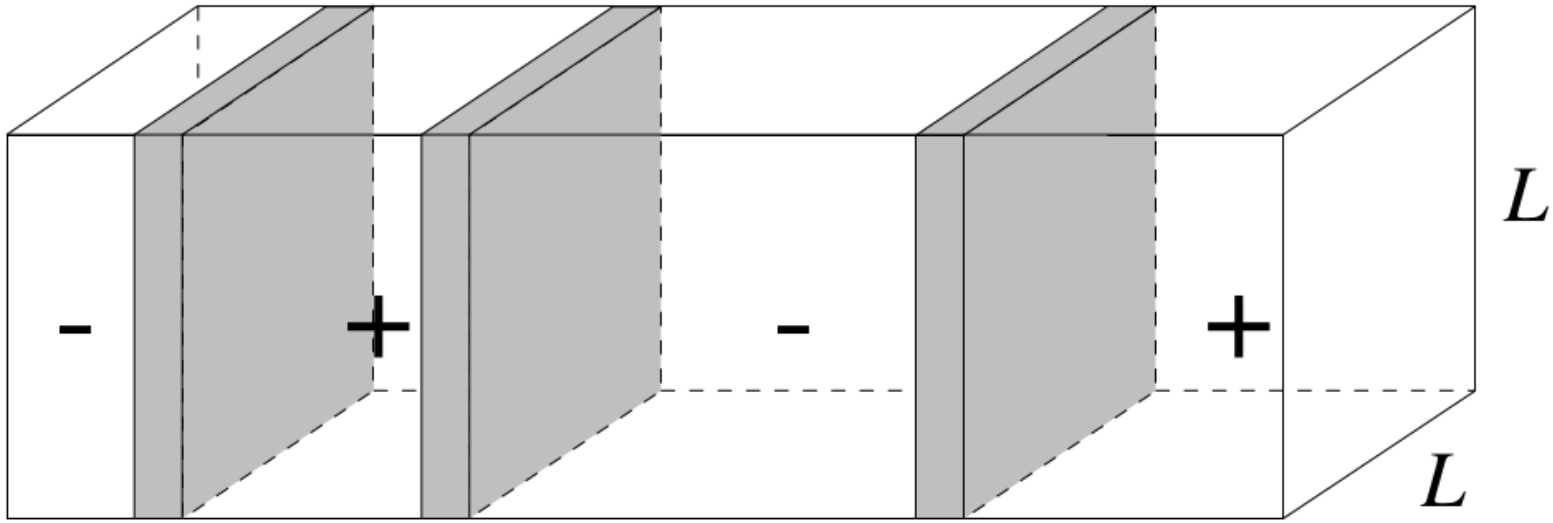
$$Z = \sum_{\{s\},J} \exp\left(\beta \sum_{\langle x,y\rangle} J_{\langle x,y\rangle} s_x s_y\right)$$

$$\frac{Z_{+-}}{Z_{++}} = \frac{\frac{Z_{+-}}{Z}}{\frac{Z_{++}}{Z}} = \frac{\langle \delta_{J=-1} \rangle}{\langle \delta_{J=+1} \rangle}$$

Ratio of measurable expectation values.

We redefine the free energy of the interface in order to improve the convergence proprieties of $\frac{F_s}{L^2}$ to σ when $L \to \infty$.
Thermodynamic limit $\longrightarrow$ only **one** interface

For finite L multiple interface can be present. An even number for Z_{++} and odd for Z_{+-}.



F_s is the free energy of a single surface. There are $\sim T$ different position for the interface.

$$Z_1 = T \exp(-F_s) = \exp(-F_S + \ln T)$$

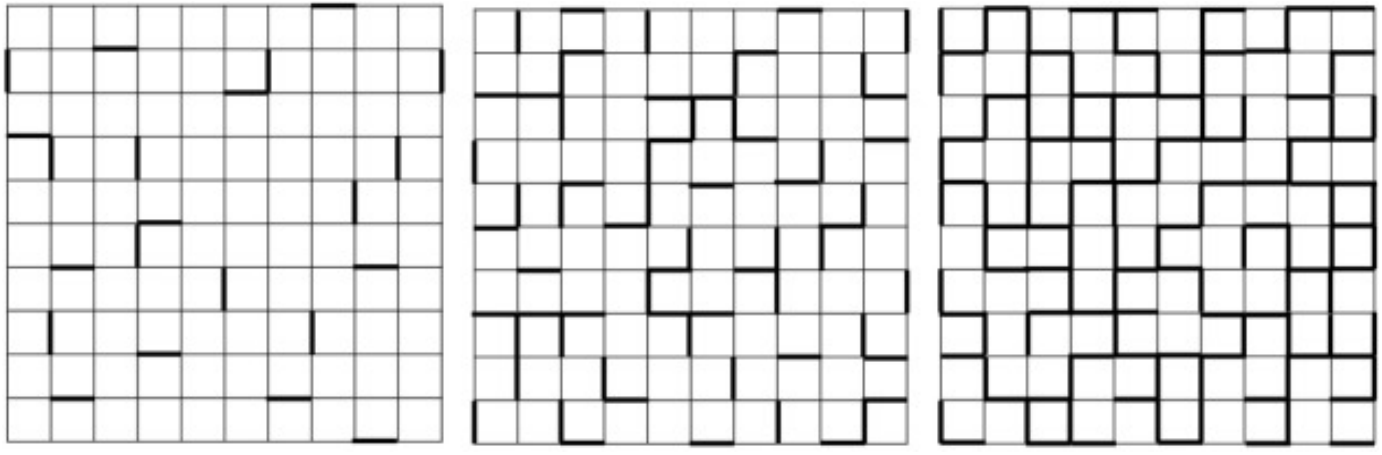
$$\frac{Z_{+-}}{Z_{++}} = \frac{Z_1 + \frac{Z_1^3}{3!} + \frac{Z_1^3}{5!} + \ldots}{1 + \frac{Z_1^2}{2!} + \frac{Z_1^4}{4!} + \ldots} = \tanh(\exp(-F_S + \ln T))$$

$$F_S = \ln(T) - \ln\left(\frac{1}{2}\ln\left(\frac{1 + \frac{Z_{+-}}{Z_{++}}}{1 - \frac{Z_{+-}}{Z_{++}}}\right)\right) \qquad \sigma = \lim_{L\to\infty} \frac{F_s}{L^2}$$

Cluster algorithms and boundary flip

Cluster algorithms allow for simultaneous updates of large parts of the lattice. Thus reducing the autocorrelation time and the critical slowling down. Swendsen and Wang (1987).

Introduce link variables $\sigma_{\langle x,y \rangle} = \{0, 1\}$ on the lattice:



$$Z = \sum_{\{s=\pm 1\}} \exp\left(\beta \sum_{\langle x,y\rangle} s_x s_y\right) = \sum_{\{s=\pm 1\}} \prod_{\langle x,y\rangle} e^{\beta s_x s_y} =$$

$$= e^{-dV\beta} \sum_{\{s=\pm 1\}} \prod_{\langle x,y\rangle} \left(1 + \delta_{s_x,s_y}\left(e^{2\beta} - 1\right)\right) =$$

$$= e^{-dV\beta} \sum_{\{s\}} \prod_{\langle x,y\rangle} \sum_{\{\sigma_{\langle x,y\rangle}=0,1\}} \left[\left(1 - \sigma_{\langle x,y\rangle}\right) + \sigma_{\langle x,y\rangle}\delta_{s_x,s_y}\left(e^{2\beta} - 1\right)\right]$$

Also valid for generic coupling $J_{\langle x,y\rangle}$:

$$Z = \sum_{\{s=\pm 1\}} \exp\left(\beta \sum_{\langle x,y\rangle} J_{\langle x,y\rangle} s_x s_y\right) = \sum_{\{s=\pm 1\}} \prod_{\langle x,y\rangle} e^{\beta J_{\langle x,y\rangle} s_x s_y} =$$

$$= e^{-dV\beta} \sum_{\{s=\pm 1\}} \prod_{\langle x,y\rangle} \left(1 + \delta_{J_{\langle x,y\rangle} s_x s_y, 1} \left(e^{2\beta} - 1\right)\right) =$$

$$= e^{-dV\beta} \sum_{\{s\}} \prod_{\langle x,y\rangle} \sum_{\{\sigma_{\langle x,y\rangle}=0,1\}} \left(1 - \sigma_{\langle x,y\rangle}\right) +$$

$$+ \sigma_{\langle x,y\rangle} \delta_{J_{\langle x,y\rangle} s_x s_y, 1} \left(e^{2\beta} - 1\right)$$

For a fixed spin configuration $\{s\}$ the links are indipendent.

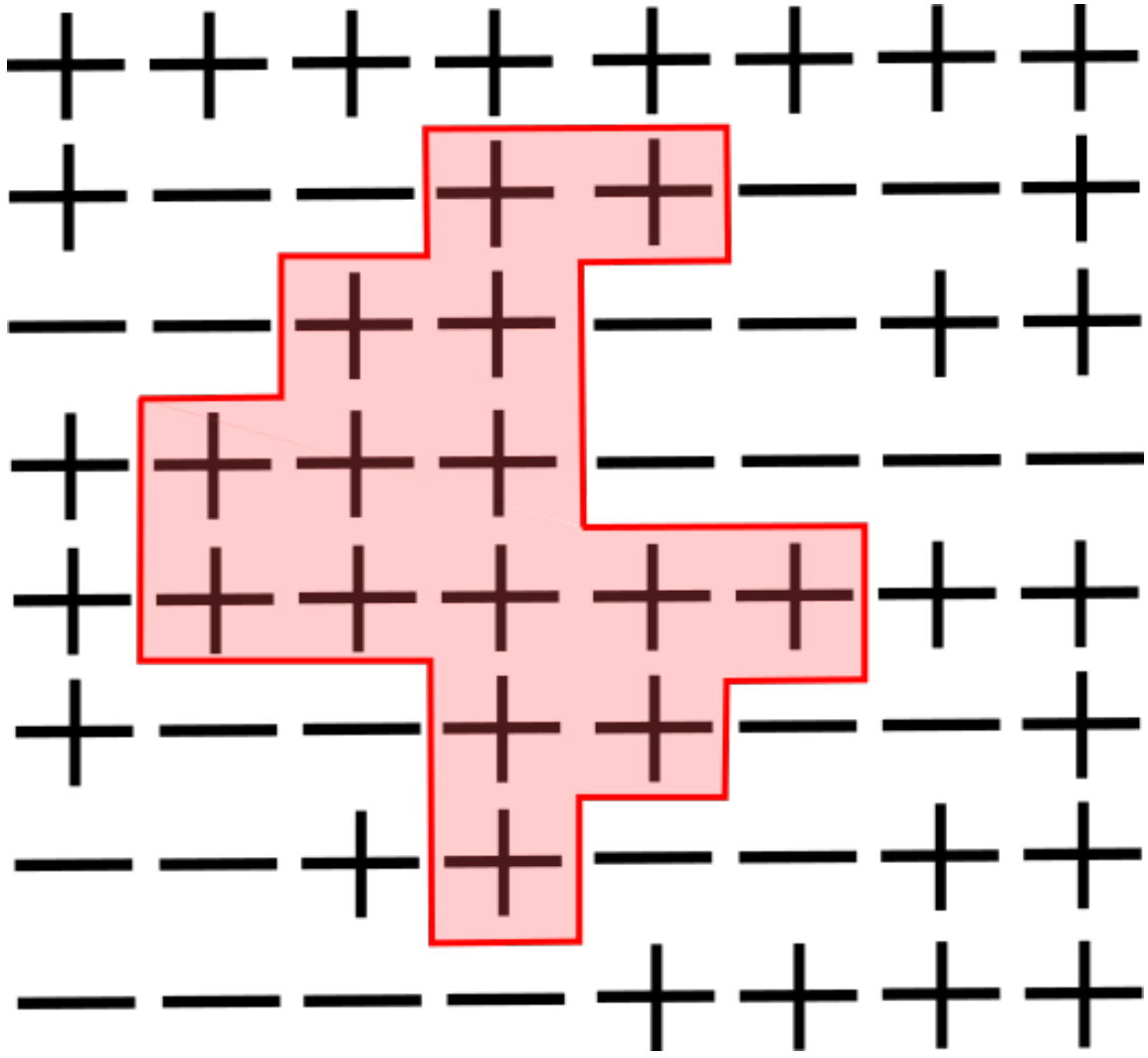
$$p_1 = p\left(\sigma_{\langle x,y\rangle} = 0\right) = \exp\left(-2\beta\delta_{J_{\langle x,y\rangle}s_x s_y,1}\right)$$

$$p_0 = p\left(\sigma_{\langle x,y\rangle} = 1\right) = 1 - p\left(\sigma_{\langle x,y\rangle} = 1\right)$$

(if $J_{\langle x,y\rangle}s_x s_y = 1$ the weights in Z are normalized to $e^{2\beta}$)

For simplicity let's put $J_{\langle x,y\rangle} = 1$. For fixed $\sigma_{\langle x,y\rangle}$ only configurations of spins that satisfy the constraint $s_x = s_y$ where $\sigma_{\langle x,y\rangle} = 1$ have a non zero probability. All configurations that satisfy the constraint have the same weight.

Definition: a **cluster** is a set of spin in the lattice path connected by links with $\sigma_{\langle x,y \rangle} = 1$. If $J_{\langle x,y \rangle} = 1$ in the ensamble all sites of a cluster are forced to have the same spin.



Swendsen and Wang algorithm:

- Generate a link configuration $\sigma_{\langle x,y \rangle}$ based on the current spin configuration by using probablities p_0 and p_1.
- For each cluster chosse a spin ($s = \pm 1$) with probability $\frac{1}{2}$. In general the updte step must be compatible with the constraint $J_{\langle x,y \rangle} s_x s_y = 1$ for all the spins of the cluster.
- The newly generted spin configuration is the next element of the Markov chain.

The SW is **ergodic**. We know prove that it satisfy the **detailed balance**.

In the cluster algorithm we update both the spin $\{s\}$ and the links $\{\sigma_{x,y}\}$. The ensamble contains both spins and link configurations: $\{s, \sigma\}$.

$$\frac{P\left(\{s_0, \sigma_0\} \to \{s_1, \sigma\}\right)}{P\left(\{s_1, \sigma_0\} \to \{s_0, \sigma\}\right)} = \frac{P\left(\{s_1\}|\{\sigma\}\right) P\left(\{\sigma\}|\{s_0\}\right)}{P\left(\{s_0\}|\{\sigma\}\right) P\left(\{\sigma\}|\{s_1\}\right)}$$

(this is not the detailer balance in the ensamble of $\{s, \sigma\}$ as the lonk configurations don't get exchnged!)

$P(\{s\}|\{\sigma\}) = \frac{1}{2^{\#cluster}}$ if $\{s\}$ is compatible with $\{\sigma\}$, null otherwise.

In general $P(\{s\}|\{\sigma\})$ must be a **constant** independent of $\{s\}$ on the allowed configurations.

Thus $P(\{s_1\}|\{\sigma\}) = P(\{s_2\}|\{\sigma\})$.

$$P\left(\{\sigma\}|\{s\}\right) = \prod_{\substack{\sigma_{\langle x,y\rangle}=0 \\ J_{\langle x,y\rangle}s_x s_y=1}} e^{-2\beta} \prod_{\sigma_{\langle x,y\rangle}=1} \left(1-e^{-2\beta}\right)$$

for $\{s\}$ compatible with $\{\sigma\}$.

The first factor arise from the unconnected links for which $J_{\langle x,y\rangle}s_x s_y = 1$ each being in this state with probability $e^{-2\beta}$. The second one is from the connected links (for which $J_{\langle x,y\rangle}s_x s_y = 1$ necessarily).

The second factor is **indipendent of** $\{s\}$ and will be neglettend in the sequent.

We obtain:

$$P\left(\{\sigma\}|\{s\}\right) = \prod_{\substack{\sigma_{\langle x,y\rangle}=0 \\ J_{\langle x,y\rangle}s_x s_y=1}} e^{-2\beta}$$

Now we compute (reminding $\{s_0\}$ and $\{s_1\}$ share the same $\{\sigma\}$:

$$\frac{e^{-\beta\mathcal{H}(\{s_1\})}}{e^{-\beta\mathcal{H}(\{s_0\})}} = \frac{\prod_{\langle x,y\rangle} e^{J_{\langle x,y\rangle}s_x^1 s_y^1}}{\prod_{\langle x,y\rangle} e^{J_{\langle x,y\rangle}s_x^0 s_y^0}} =$$

$$\frac{\prod_{\substack{\sigma=0 \\ Js_x^1 s_y^1=+1}} e^{\beta} \prod_{\substack{\sigma=0 \\ Js_x^1 s_y^1=-1}} e^{-\beta} \prod_{\substack{\sigma=1 \\ Js_x^1 s_y^1=+1}} e^{\beta}}{\prod_{\substack{\sigma=0 \\ Js_x^0 s_y^0=+1}} e^{\beta} \prod_{\substack{\sigma=0 \\ Js_x^0 s_y^0=-1}} e^{-\beta} \prod_{\substack{\sigma=1 \\ Js_x^0 s_y^0=+1}} e^{\beta}}$$

The last part having $\sigma = 1$ obviously depends only on $\{\sigma\}$ for all $Js_x s_y$ beeing forced to 1 if $\sigma = 1$.

We observe that:

$$\prod_{\substack{\sigma=0 \\ Js_x^1 s_y^1=-1}} e^{-\beta} \times \prod_{\substack{\sigma=0 \\ Js_x^1 s_y^1=1}} e^{-\beta} = \prod_{\sigma=0} e^{-\beta} = k$$

k depends only on the link configuration $\{\sigma\}$

$$\prod_{\substack{\sigma=0 \\ Js_x^1 s_y^1=-1}} e^{-\beta} = k \prod_{\substack{\sigma=0 \\ Js_x^1 s_y^1=1}} e^{\beta}$$

$$\frac{e^{-\beta\mathcal{H}(\{s_1\})}}{e^{-\beta\mathcal{H}(\{s_0\})}} = \frac{\prod_{\substack{\sigma=0 \\ Js_x^1 s_y^1=+1}} e^{2\beta}}{\prod_{\substack{\sigma=0 \\ Js_x^0 s_y^0=+1}} e^{2\beta}} = \frac{\prod_{\substack{\sigma=0 \\ Js_x^0 s_y^0=+1}} e^{-2\beta}}{\prod_{\substack{\sigma=0 \\ Js^1 s_x s_y^0=+1}} e^{+2\beta}}$$

We arrived at:

$$\frac{P\left(\{s_0,\sigma_0\}\to\{s_1,\sigma\}\right)}{P\left(\{s_1,\sigma_0\}\to\{s_0,\sigma\}\right)}=\frac{e^{-\beta\mathcal{H}(\{s_1\})}}{e^{-\beta\mathcal{H}(\{s_0\})}}$$

$$P\left(\{s_0\}\to\{s_1\}\right)=\sum_{\{\sigma\},\{\sigma_0\}}P\left(\{s_0,\sigma_0\}\to\{s_1,\sigma\}\right)$$

We thus obtain the detailed balance:

$$\frac{P\left(\{s_0\}\to\{s_1\}\right)}{P\left(\{s_1\}\to\{s_0\}\right)}=\frac{e^{-\beta\mathcal{H}(\{s_1\})}}{e^{-\beta\mathcal{H}(\{s_0\})}}$$

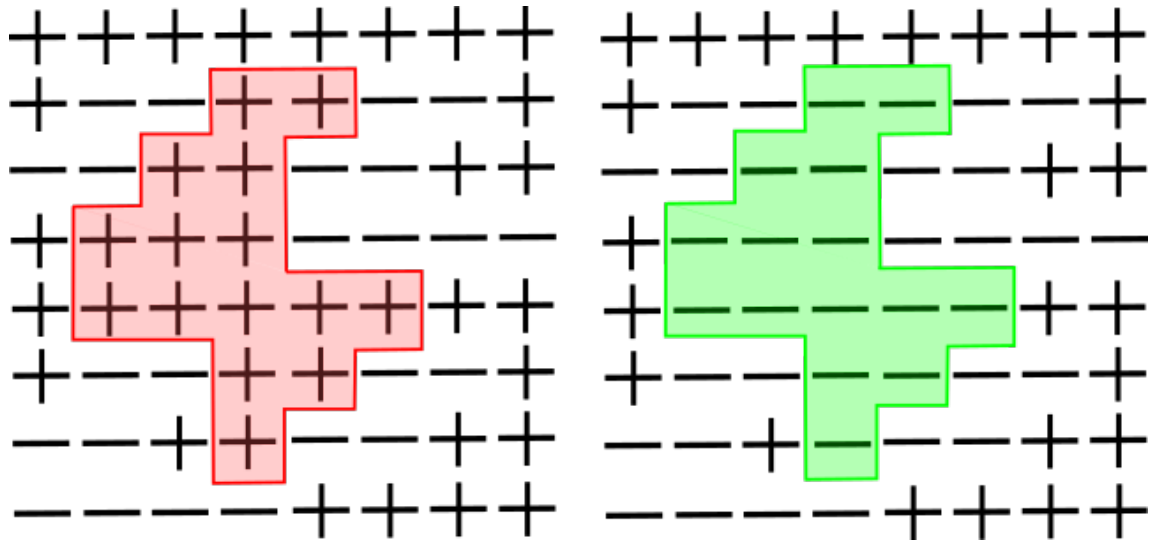
- The only requirement on the update step is that all the possible $\{s\}$ given $\{s\}$ which we choose to have a non null probability of happearing are indeed equiprobable.
- We can also update the coupling constants $J_{\langle x,y\rangle}$ as long as $J_{\langle x,y\rangle}$ where $\sigma = 1$.

This two observations give rise to two key modification of the SW algorithm: the Wolff algorithm and the boundary flip.

Single cluster update:

In the Wolff algorithm we chose at random one site of the lattice and flipping the cluster it belongs to with certanty.

It is equivalent to chossing at random one site of the lattice and consider it the seed for and expanding cluster and then flipping it.

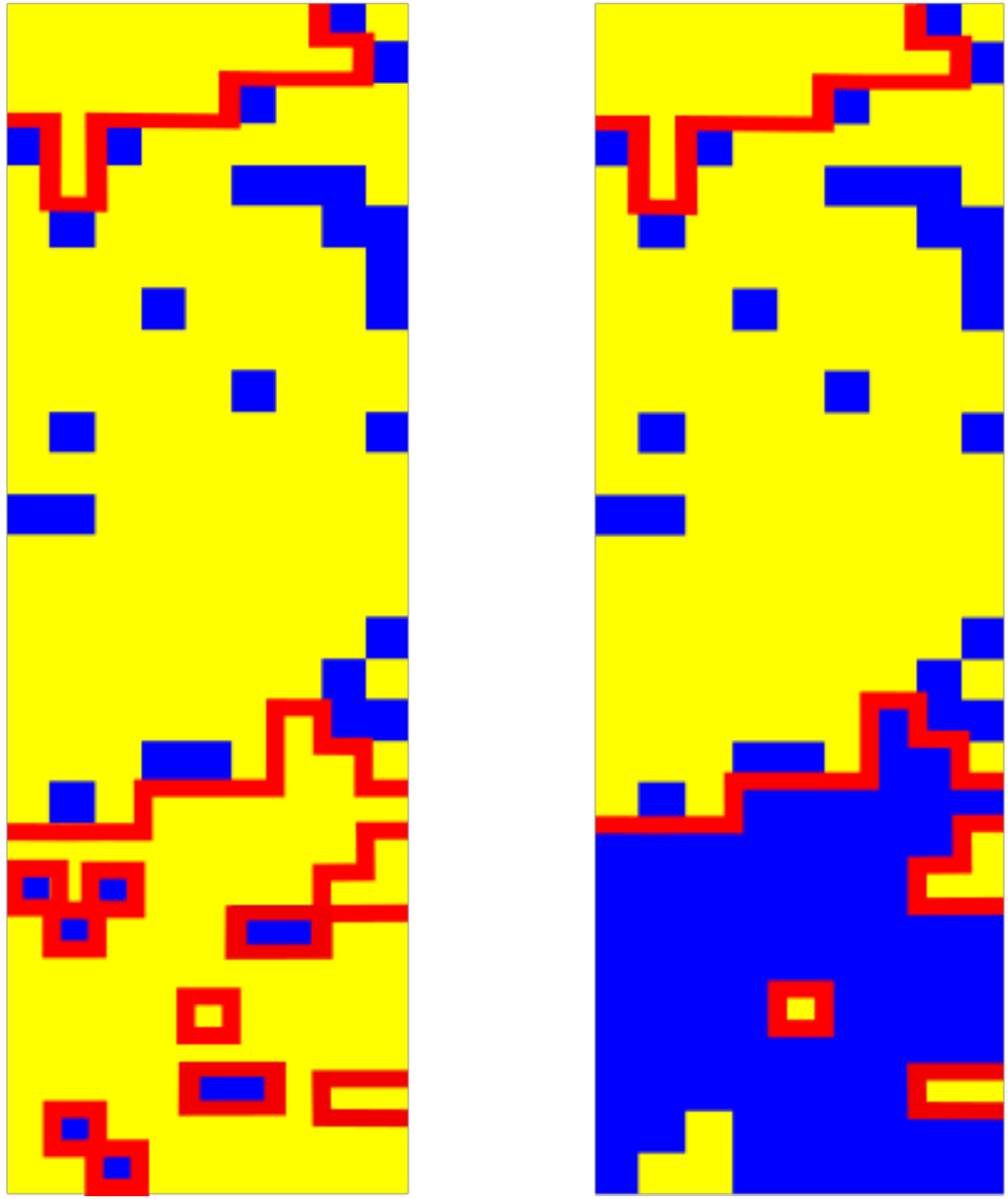


Boundary flip algorithm:

$J = 1$ in the bulk, but the coupling between the top and the bottom is now a dynamical variable to simulate:
$\{s, \sigma, J_{0,T-1} = \pm 1\}$.

A link $\sigma_{\langle 0,T-1\rangle} = 1$ demands $J_{0,T-1} s_0 s_{T-1} = 1$. Thus we can flip $J_{0,T-1}$, s_0 and all the spins connected to s_0 via some chain of links in the bulk obtaing a configuration compatible with $\{\sigma_{\langle x,y\rangle}\}$. This for all the spins on the bottom boundary.

Boundary condition update. Section of a $7 \times 7 \times 21$ lattice at temperature $\beta = 0.250$

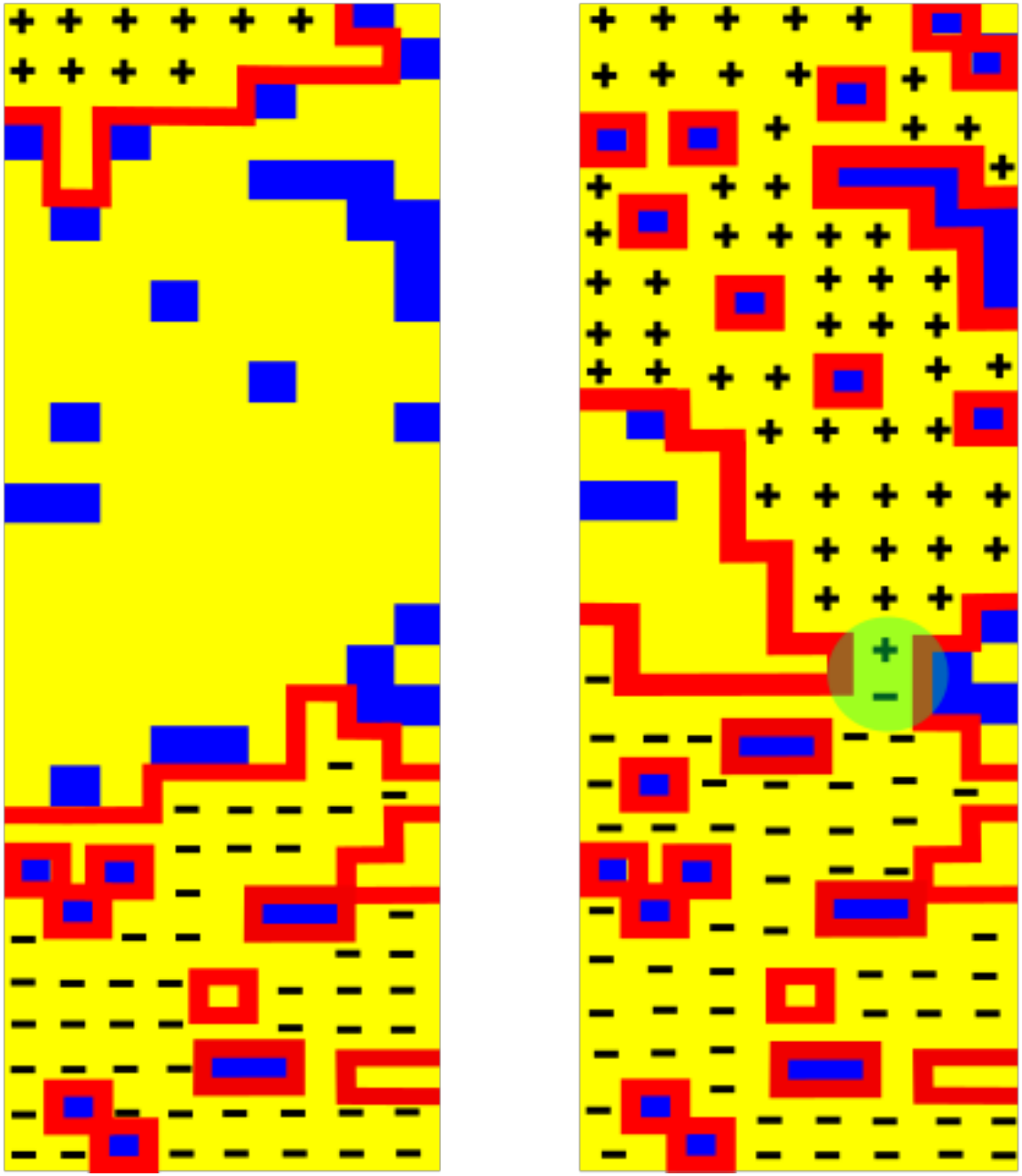


The boundary flip generates an interface between phases!

We must check this update can be done without violating the constraints imposed by the bulk links ($\sigma_{\langle x,y \rangle} = 1$ implies $s_x = s_y$).

Check all the clusters that contain sites of the lower surface and flip the boundary condition only if it can be done consistently on all the lattice.

Introduce an extra variable $c_x = \pm 1$ that gets propagated in the bulk during the costruction of the cluster but changes sign when crossing the boundary.



We do N total steps alternating the Wolff algorithm with the boundary flip.

After each update we count if the current configuration has ferromagnetic or antiferromagnetic coupling between top and bottom.

$$\langle \delta_{J=-1} \rangle = \frac{\#\text{Antiferromagnetic}}{N}$$

$$\langle \delta_{J=+1} \rangle = \frac{\#\text{Ferromagnetic}}{N}$$

Notes on the implementation

C++ for the Montecarlo and Jackknife algorithms, Python for data analysis, fits and plots.

The hot function of the simulation generates a cluster starting in a given position and exploring the neighbouring links. If a link is chosen to be $\sigma_{\langle x,y \rangle} = 1$ then the adjacent site is included in the cluster and the procedure is repeated. The extra variable c_x is also propagated.

```
stack<site> stack
stack.push(seed)
while(!stack.empty())
   site current = stack.top()
   value = cluster[current]
   if(cluster[current] is incostintent) flag = 1;
   else if(cluster[current] == 0)
      cluster[current] = cluster[old];
      for(d = 0; d < 3; d++)
         for(a = -1; a < 2; a = a + 2)
         next = current + a
         Check if we are on the boundary
         if(cluster[next] == 0)
            if(p > 0 and random < p) stack.push(next)
return flag
```

- The algorithm doesn't actually performs the spin and boundary flip it just counts for how many configuration it is possible.
- The simulation where executed with $N = 10^5, 10^6$ steps of the Markov chain.
- The first 5% of the Markov chain is ignored to avoid non-termalized configurations.
- The Jackknife is executed for blocks with size starting from 10000 to 10900 with step of 30 events.

Estimation of the errors and autocorrelation

Fit of the free energy - theory

Fit of the free energy - results

Fit of the critical behaviour

(Conclusions?)