# 2-d classical Ising Monte Carlo simulator

Ising model is the go-to toy model of a system undergoing a phase transition to become ferromagnetic. The lattice consists of spins with values either -1 or 1. They have nearest neighbour (i.e. directly up, down, left or right in 2d) interactions and they can also be coupled to an external magnetic field. The Hamiltonian is -J $\Sigma_{<i,j>} \sigma_i \sigma_j$-h $\Sigma_i \sigma_i$, where $\sigma_i$ is the spin at site i, <i,j> indicates a sum over the nearest neighbour pairs, J is a coupling constant (usually J = 1) and h is the value of the external magnetic field. Low-energy states tend to have their spins aligned while in high-energy states the spin changes sign from site to site.

The typical thing to do would be to compute the canonical partition function, Z[h] = $\Sigma_{configurations}$ Exp[-$\beta$ H[h]] and to study many properties of the system including magnetization. However, this only doable for a few cases and we have to resort to numerical magic. The 2d model is integrable when h=0 but we also wish to consider the effect of the magnetic field. In two and higher dimensions, the system undergoes a 2nd order phase transition from $\langle\sigma_i\rangle = 0$ (paramagnetic) phase to $\langle\sigma_i\rangle \neq 0$ (ferromagnetic) phase at a special value of $\beta = \beta_c$ and at h=0.

We wish to study this system and its phase transition on a two-dimensional periodic square lattice. We do this using Monte Carlo simulation. We construct both a local and a cluster update method. The idea of Monte Carlo simulations is to generate typical configurations of the systems by using incremental updates to the configuration that are accepted or rejected with some $\beta$ dependent probability. It can be shown that the statistical properties of these generated states approach the thermodynamic values.

---

## Setup

### Generic functions:

0. For good random numbers, initialize Mersenne Twister

1. Construct a square lattice (array) of size L^2. Give it some initial state, i.e. fill it with -1's and 1's. Stick to L above 16 but below 200.

2. Make the lattice periodic, i.e. construct functions s.t. $\sigma$(L+1,y) = $\sigma$(1,y), $\sigma$(x,1+L)=$\sigma$(x,1), $\sigma$(1-1,y)=$\sigma$(0,y) etc.

3. Construct a function that computes the Hamiltonian on the lattice.

### 4. Construct a function that gives you a random initial site.

## Local update

Local updates are done by making a pre-emptive flip of a single spin. If the energy of the system goes down, the spin flip is accepted. If the energy goes up, it is accepted with a probability Exp[- $\beta$ $\delta$H], where $\delta$H is the change of energy and reverted with probability 1-Exp[- $\beta$ $\delta$H]. Note that change occurs in a small area affecting only 4 “bonds” (due to 4 neighbours) and the energy shift due to the magnetic field so you can stick to computing a local part of the Hamiltonian to determine the probability of flipping or not. This is the so-called Metropolis method.

These spin-by-spin flips are applied to the whole lattice and they are flipped separately. The order is not very important. You can do them row by row, or doing even and odd sites separately (checker board) or in a completely random order.

### 5. Construct a function to determine, whether a single spin should be flipped or not.

### 6. Construct a function for a lattice-wide update.

### 7. Plot the lattice using MatrixPlot with Dynamic updating and run a few thousand rounds (for small lattices, for bigger lattices you need much less) of your update method. Track the total energy of the system and the magnetization: M= $\frac{1}{L\text{^}2}\Sigma_i \sigma_i$. Keeping h=0 and J = 1, when does the system exhibit spontaneous magnetization? (It might be better to track |M|). The analytic value is $1/2\,\text{Log}\left[1+\sqrt{2}\right]$=0.44... (Note: Beyond the critical value $\beta_c$, the magnetization would not go to zero even when considering larger lattices)

## Cluster update

You might have noticed (probably not...) that your local update is slow to forget about the previous configuration when $\beta$ is close to its critical value or higher. There is a way around this problem by considering larger updates. Consider any lattice configuration and pick site i. Site i is the first site included in your cluster. You increase the cluster as follows: For all the sites in the cluster, consider the nearest neighbours to it. If they have the same spin, connect the two sites with probability 1-Exp[-2 $\beta$ J]. If they don’t have the same spin, ignore the site. Note that a site can be joined to the cluster from more than one neighbour so you can try to connect it many times. Once there are no more sites to add to the cluster, you flip the spin in all the sites of the cluster. And you start all over again. (Shortly: clusters are connected regions whose sites all have their spins aligned and new sites are connected with probability 1-exp(-2 $\beta$ J))

These are known as Fortuin-Kasteleyn clusters and the update method is known as the Wolff algorithm. These are much more efficient in sampling new new independent lattice configurations than local updates when $\beta$ increases. In our implementation, h=0 implicitly. There are ways to add h to the cluster

method but it is out of the scope of this project (see e.g. ghost spins).

## 8. Construct a second lattice (array) for keeping track of the sites already visited by the cluster expansion algorithm.

## 9. Construct an algorithm for growing a FK cluster and then flip the spins

## 10. Same as problem 7 but use your new cluster method. This time, h=0 in our implementation.

```
magn = {}; energy = {};
J = 1.0; h = 0; β = 0.44;
Do[constructcluster[β, J, randomsite[]];
 magn = Append[magn, Total[lattice0, 2] / Times @@ Dimensions[lattice0] // N];
 energy = Append[energy, Ham[J, h] // N];, {500}]
```

```
$Aborted
```

```
Dynamic[{MatrixPlot[lattice0], ListPlot[magn], ListPlot[energy]}]
Dynamic[Mean[Abs[magn]]]
```

```
{MatrixPlot[lattice0], ListPlot[magn], ListPlot[energy]}
```

```
Mean[Abs[magn]]
```