

# Wang Landau algorithm for 4D Ising gauge model



# Chapter 1

## Theory

The Ising model is based on a set of discrete variables that take value  $\pm 1$ . The most common Ising model is the Ising spin model, used to describe the magnetism in matter, where the variables are the spin values associated to the sites of a lattice. The Ising gauge model is a particular Ising model where the variables are defined on the links between two sites and the action is usually written as follow:

$$\begin{aligned} S[U] &= \beta \sum_x \sum_{\{U_\rho\}=\pm 1} \sum_{0 \leq \mu < \nu \leq 3} U_\mu(\vec{x}) U_\nu(\vec{x} + a\hat{\mu}) U_{-\mu}(\vec{x} + a\hat{\mu} + a\hat{\nu}) U_{-\nu}(\vec{x} + a\hat{\nu}) = \\ &= \beta \sum_x \sum_{\{U_\rho\}=\pm 1} \sum_{0 \leq \mu < \nu \leq 3} \sigma_{\mu\nu}(\vec{x}) = S[\sigma]. \end{aligned} \quad (1.1)$$

Where:

- $\vec{x}$  are the coordinates of the sites of the lattice
- $\mu, \nu$  are the directions
- $a$  is the distance between the sites. In the codes this distance is fixed to 1
- $U_\mu(\vec{x})$  is the link from the site of coordinate  $\vec{x}$  to the site of coordinate  $\vec{x} + a\hat{\mu}$
- $\sigma_{\mu\nu}(\vec{x})$  is the so called "elementary plaquette", defined as the smallest loop on the lattice. In our case, is a square with side-length 1
- $\beta$  is a parameter, which represent the inverse of the coupling constant of the theory.

Let's consider the coordinate  $\mu = 3$  the time-like direction. In our model, we introduce an anisotropy as follow:

$$S[\sigma] = \beta_s \sum_x \sum_{\{U_\rho\}=\pm 1} \sum_{0 \leq \mu < \nu \leq 2} \sigma_{\mu\nu}(\vec{x}) + \beta_t \sum_x \sum_{\{U_\rho\}=\pm 1} \sum_{0 \leq \mu \leq 2} \sigma_{3\mu}(\vec{x}) \quad (1.2)$$

Thus, the coupling constant for the space-like elementary plaquettes is, in principle, different from that of the time-like elementary plaquettes.

## Chapter 2

# Metropolis algorithm

The Wang-Landau algorithm is based on the idea that the partition function can be written as

$$Z = \sum_{\{\text{configurations}\}} e^{E/(K_b T)} = \sum_E g(E) e^{E/(K_b T)} \quad (2.1)$$

The function  $g(E)$  is the density of states, and does not depend from the temperature  $T$ . The aim of the Wang-Landau algorithm is calculate  $g(E)$ . This algorithm perform a random walk in the energy space by changing the states of the links randomly. During the random walk, we accumulate the histogram  $H(E)$ , which is a quantity that keeps track of the number of visits at each energy level  $E$  ( each time an energy  $E$  is visited, the corresponding entry in  $H(E)$  is incremented by one); thus, the algorithm modifies the estimate of the density of states by a multiplicative factor  $f$ .

Thus, the Wang-Landau algorithm is as follows:

- Initialize the lattice in an arbitrary configuration with energy  $E_1$ .
- Initialize  $g(E)=1$  for all possible energies  $E$ .
- Choose an initial modifier factor  $f$  ( usually  $\exp(0)$  ).
- Set  $H(E) = 0$  for all  $E$ .
- Pick one link and change its state, thus the new energy of the system is  $E_2$ .
- The new state is accepted with probability  $\min\left(\frac{g(E_1)}{g(E_2)}, 1\right)$ .
- If the new states is accepted,  $g(E_2) \rightarrow g(E_2) \times f$ , otherwise  $g(E_2) \rightarrow g(E_1) \times f$ .
- If the new states is accepted,  $H(E_2) \rightarrow H(E_2) + 1$ , otherwise  $H(E_1) \rightarrow H(E_1) + 1$ .
- When  $H(E)$  is "flat",  $f \rightarrow \sqrt{f}$ .

- Set  $H(E) = 0$  for all  $E$  and repeat the algorithm.
- We stopped the algorithm when  $f$  is smaller than a predefined value.

It is impossible to obtain a perfectly flat histogram and the phrase "flat histogram" means that the histogram  $H(E)$  for all possible  $E$  is not less than  $x\%$  of the average histogram  $\langle H(E) \rangle$ , where  $x\%$  is chosen according to the size and complexity of the system and the desired accuracy of the density of states.

## Chapter 3

# Structure

Due to the complexity of the model, the program perform four simulations, and every 200000000 check the flatness condition of  $H(E) = H1(E) + H2(E) + H3(E) + H4(E)$ , where  $H1(E), H2(E), \dots$  are the histograms of each simulation. When the total  $H(E)$  is flat,  $f \rightarrow \sqrt{f}$  and  $H_i(E) = 0$  for all possible  $E$  a for  $i=\{1,2,3,4\}$ .

When  $f < \text{Threshold}$ , the final  $g(E)$  is the average of the  $g(E)$  of each simulation, so  $g(E) = (g1(E) + g2(E) + g3(E) + g4(E)) / 4$ .

The lattice has volume  $N_s^2 \times N_z \times N_t$ , where  $N_s$  is the lenght of the firsts two spatial-like sides of the lattice,  $N_z$  is the lenght of the third spatial-like side and  $N_t$  is the lenght of the time-like side.

When we will talk about of site or link  $i$ , the letter  $i$  is a compact coordinate; for instance, a site of the lattice with coordinate  $(x,y,z,t)$  has compact coordinate  $i = x * N_s \times N_z \times N_t + y * N_z \times N_t + z * N_t + t$ .

The element `lattice[i][j]` corresponds to the link from site  $i$  along direction  $j$ .

The vector `Plaquette[]` stores all the elementary plaquette, thus it is not necessary to calculate them at every Monte Carlo step, but the condition is that the lenght of every side of the lattice have to be  $\geq 1$ .

The vector `link_to_plaquette[i][j][ ]` stores the positions in the vector `Plaquette` of the elementary plaquettes attached to the link  $i$  along the direction  $j$ .

The element `map_nn[i][j]` is the first neighbour along the direction  $j$  of the site  $i$ .

- `latticeRandom_creation` creates a lattice in a random configuration.
- `lattice_creation_frozen` creates a lattice in the frozen configuration (all the links are +1).
- `lattice_creation_minus_frozen` creates a lattice in the configuration with all the links -1.
- `plaquette_creation` creates the vector `Plaquettes`.
- `Wang_Landau` is the core of the code, is the Wang Landau algorithm.

- Energy compute the average energy using the distribution obtained with Wang Landau.
- EnergyUp2 compute the average energy<sup>2</sup> using the distribution obtained with Wang Landau.

```
g++-15 -std=c++17 -O Gauge Wang_Landau_Ising_4D.cpp
```

To run the program:

```
./Gauge
```