1 Worm algorithm for the Ising model

$$H = \sum_{i}^{N} \sum_{\mu=1}^{d} K\sigma_i\sigma_{i+\mu}, \quad K = \beta J, \quad \mu = 1, \ldots, d, \quad \sigma_i = \pm 1. \tag{1}$$

$$Z = \sum_{\{\sigma_i\}} e^{-H/T} = \sum_{\{\sigma_i\}} \prod_{b=(i,\mu)}^{dN} e^{K\sigma_i\sigma_{i+\mu}} = \sum_{\{\sigma_i\}} \prod_{b=(i,\mu)}^{dN} \sum_{n_b=0}^{\infty} \frac{(K\sigma_i\sigma_{i+\mu})^{n_b}}{n_b!} \tag{2}$$

We'll rewrite the last expression in a more convenient form, which is better understood with an example. Consider $d = 1$ and $N = 3$, then

$$Z = \sum_{\sigma_1=\pm} \sum_{\sigma_2=\pm} \sum_{\sigma_3=\pm} \sum_{n_{(1,1)}} \sum_{n_{(2,1)}} \sum_{n_{(3,1)}} \frac{K^{n_{(1,1)}}}{n_{(1,1)}!} \frac{K^{n_{(2,1)}}}{n_{(2,1)}!} \frac{K^{n_{(3,1)}}}{n_{(3,1)}!} (\sigma_1\sigma_2)^{n_{(1,1)}} (\sigma_2\sigma_3)^{n_{(2,1)}} (\sigma_3\sigma_1)^{n_{(3,1)}} \tag{3}$$

$$= \sum_{\{n_b\}} \prod_{b=(i,\mu)} \left(\frac{K^{n_b}}{n_b!}\right) \sum_{\{\sigma_i\}} \sigma_1^{n_{(1,1)}+n_{(3,1)}} \sigma_2^{n_{(1,1)}+n_{(2,1)}} \sigma_3^{n_{(3,1)}+n_{(2,1)}} \tag{4}$$

$$= \sum_{\{n_b\}} \prod_{b=(i,\mu)} \left(\frac{K^{n_b}}{n_b!}\right) \sum_{\{\sigma_i\}} \sigma_1^{n_{(1,1)}+n_{(1,-1)}} \sigma_2^{n_{(2,1)}+n_{(2,-1)}} \sigma_3^{n_{(3,1)}+n_{(3,-1)}}$$

$$= \sum_{\{n_b\}} \prod_{b=(i,\mu)} \left(\frac{K^{n_b}}{n_b!}\right) \sum_{\{\sigma_i\}} \prod_{i} \sigma_i^{p_i}, \quad p_i = \sum_{\mu=1}^{d} (n_{(i,\mu)} + n_{(i,-\mu)}). \tag{5}$$

The last row is valid for any N and d, thus in general

$$Z = \sum_{\{n_b\}} \prod_{b=(i,\mu)} \left(\frac{K^{n_b}}{n_b!}\right) \sum_{\{\sigma_i\}} \prod_{i} \sigma_i^{p_i}, \quad p_i = \sum_{\mu=1}^{d} (n_{(i,\mu)} + n_{(i,-\mu)}). \tag{6}$$

Let us notice that

$$\sum_{\sigma_i=\pm} \sigma_i^{p_i} = \begin{cases} 2 & p_i \text{ even,} \\ 0 & p_i \text{ odd.} \end{cases} \tag{7}$$

Then

$$Z = 2^N \sum_{\{n_b\}_{\text{CP}}} \prod_{b=(i,\mu)} \left(\frac{K^{n_b}}{n_b!}\right), \tag{8}$$

where CP means *closed path* and stands for those configurations where all sites satisfy the restriction p_i even. We can understand the partition function configurations graphically if we consider that each term in the expansion of K is represented by a bond line. For instance, n_b lines correspond to $K^{n_b}/n_b!$. Graphically this means that an even number of lines must emanate from each lattice site, see figure 1

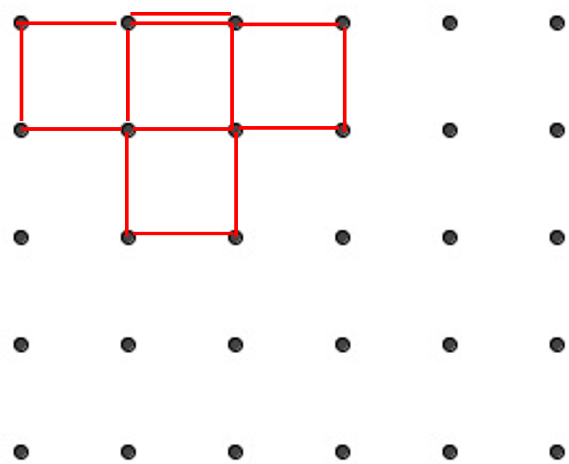


Figure 1: Typical configuration of the partition function. An even number of lines emanates from each lattice site.

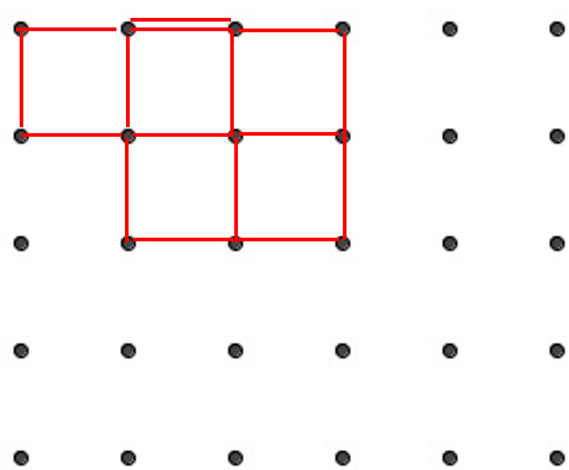


Figure 2: Even though the path made by the lines is closed, this configuration does not contribute to the partition function, since there is a site with an odd number of bonds.

Therefore we see that the spin configurations in the Ising model are equivalent to "closed" path configurations. The algorithm considers an extension of the configuration space to include "closed path" configurations with two open ends. To explain the idea let us consider the spin-spin correlation function

$$g(j_1 - j_2) = \frac{1}{Z} \sum_{\{\sigma_i\}} \sigma_{j_1} \sigma_{j_2} e^{-H/T} \equiv \frac{G(j_1 - j_2)}{Z}. \tag{9}$$

If we perform the same steps as before we obtain

$$G(j_1 - j_2) = \sum_{\{n_b\}} \left(\prod_{\{\sigma_i\}\, i \neq j_1, j_2} \sigma_i^{p_i} \right) \sum_{\sigma_{j_1} = \pm} \sigma_{j_1}^{p_{j_1}+1} \sum_{\sigma_{j_2} = \pm} \sigma_{j_2}^{p_{j_2}+1}. \tag{10}$$

The sums over σ_{j_1} and σ_{j_2} are equal to

$$\sum_{\sigma_j=\pm} \sigma_j^{p_j+1} = \begin{cases} 2 & p_j \ \text{odd}, \\ 0 & p_j \ \text{even}. \end{cases} \tag{11}$$

Then, the configurations that contribute to $G(j_1 - j_2)$ have an even number of bond lines emanating from the sites $i \neq j_1, j_2$ and an odd number of bond lines emanating from the sites j_1 and j_2. Note that if $j_1 = j_2$ we are back to the partition function.

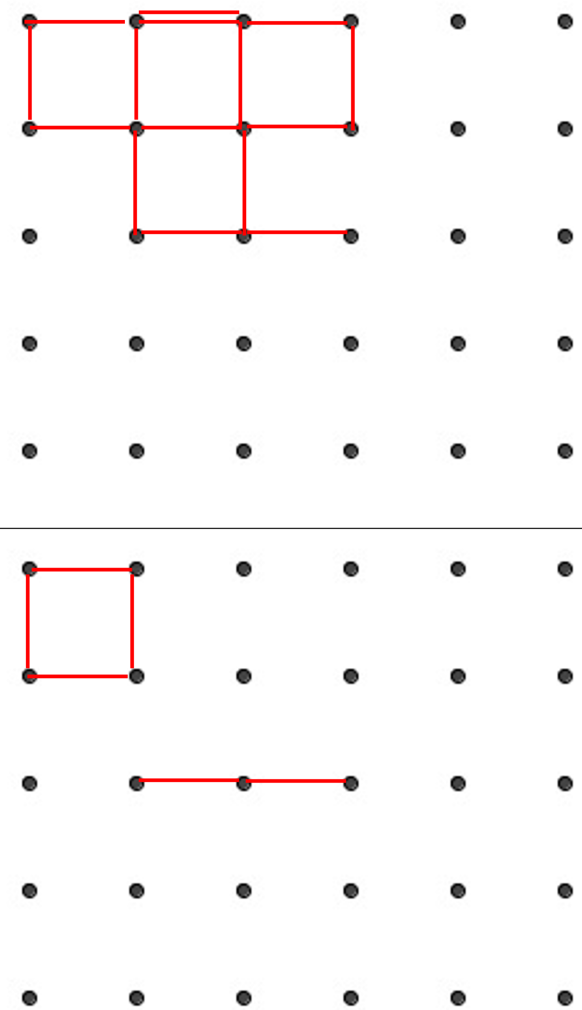


Figure 3: Typical configurations of $G(j_1 - j_2)$

The worm algorithm samples configurations in the G space. When $j_1 = j_2$ we return to the Z space. The steps of the algorithm are as follows:

1. In the beginning choose randomly j_1 and fix $j_1 = j_2$. Select randomly a direction to move j_2 to one of its neighbors, say j_3, and whether this shift will create or delete a bond.

2. Perform the shift from j_2 to j_3 with probability

$$R = \begin{cases} \frac{K}{n_b+1} & \text{when the bond number is increased,} \\ \frac{n_b}{K} & \text{when the bond number is decreased.} \end{cases} \tag{12}$$

3. Now j_3 is our new j_2 and we repeat the previous steps. If it comes out that $j_1 = j_2$ then we choose randomly a new site for j_1 and again fix $j_1 = j_2$.

If $i = j_1 - j_2$ we update $G(i) = G(i) + 1$. Only when $j_1 = j_2$ we update the partition function $Z = Z + 1$ and take measurements. The observables must be formulated in terms of the partition function. It can be proved that the magnetic susceptibility is equal to

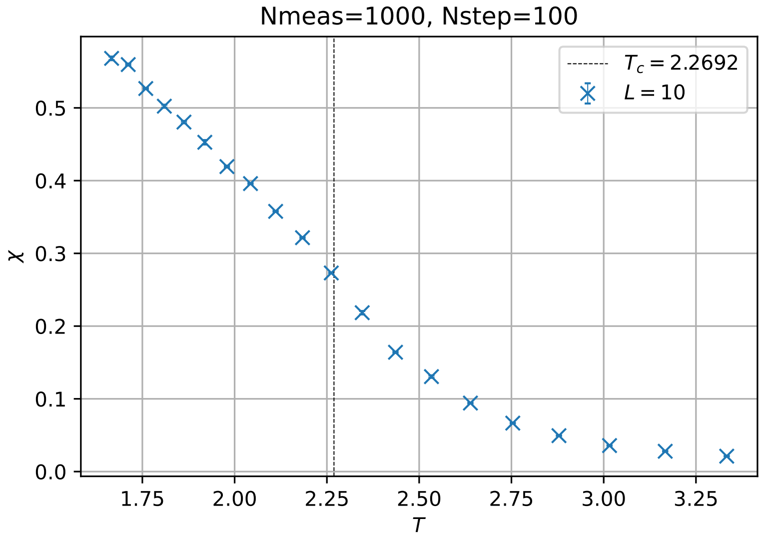
$$\chi = \beta \sum_i G(i)/Z, \tag{13}$$

while the heat capacity is

$$C_v = \langle N_b^2 - N_b \rangle - \langle N_b \rangle^2, \tag{14}$$

where N_b is the total number of bond lines on the lattice. Some useful references to better understand the details of the algorithm are [1, 2, 3] and a simple implementation of the algorithm can be found in [4]

2 Preliminary results



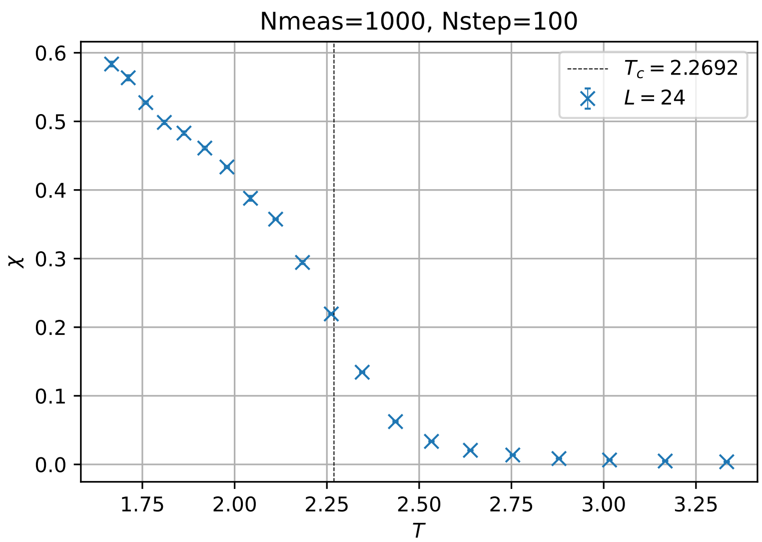
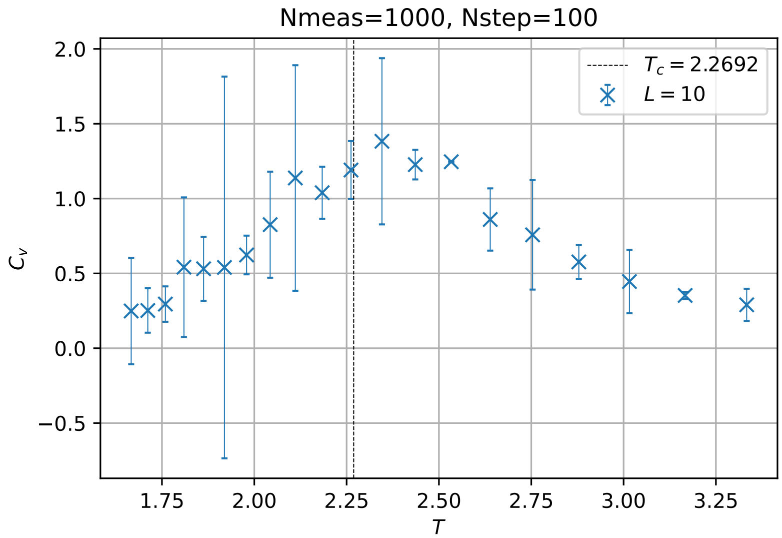
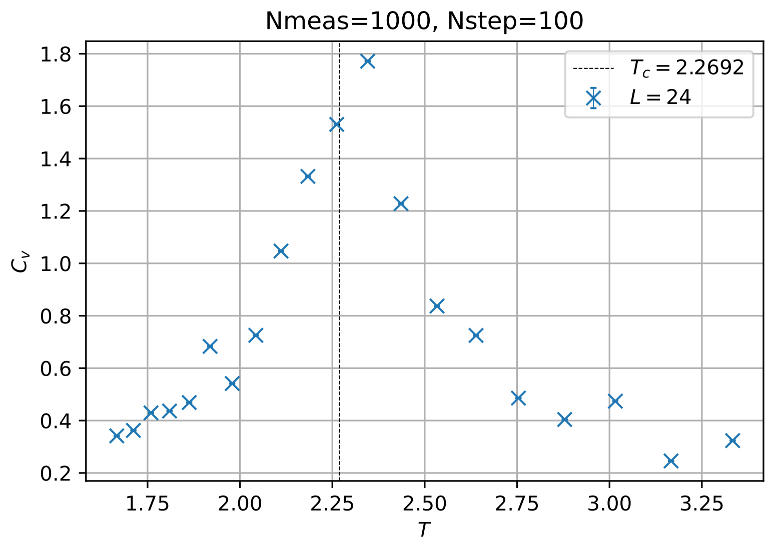


Figure 4: Magnetic susceptibility





References

[1] N. Prokof'ev and B. Svistunov, Worm Algorithms for Classical Statistical Models, Phys. Rev. Lett. **87** (2001)

[2] M. Szyniszewski, Simulating graphene impurities using the worm algorithm, Master Thesis, Lancaster University, 2013.

[3] Worm Algorithm and Diagrammatic Monte Carlo https://mcwa.csi.cuny.edu/umass/index.html

[4] https://github.com/Fabian2598/O-2-Model