Monte Carlo simulation of the Ising model – v4

Project for Computational Physics II, WS19/20 *February 4, 2020*

Calendar

Date	Activity
Jan 23	tutorial/lecture
Jan 28	planned tutorial/lecture
Jan 29	planned tutorial/lecture
Jan 30	tutorial/lecture
Feb 4	optional tutorial/lecture
Feb 5	optional tutorial/lecture
Feb 6	tutorial/lecture
Feb 11	planned tutorial/lecture
Feb 12	planned tutorial/lecture
Feb 13	tutorial/lecture
Feb 16	**deadline** at 23.59

- **This document will be updated. Please check the Moodle page regularly!**
- The final evaluation will be based on three compulsory projects and not four, as originally planned. Therefore the weight of the projects to the final mark is modified as follows

Weight	Project
25%	Analysis of sound signal
40%	Time-dependent Schrödinger equation
35%	Monte Carlo simulation of the Ising model

- The grade of this project will contribute to the 35% of the final grade. The grade g will be given on a scale from 0 to 100, and it can be converted to the German academic grading system by using the formula

$$f(g) = 1 + \frac{3}{50}(100 - g) \tag{1}$$

and by rounding to the closest German grade.

- The list of assessed competences and a description of their indicators can be found in the Moodle page of the course. The weight of each competence in the grading of this project is given in the following table.

Weight	Assessed competence
0%	Physical Transcription
20%	Planning
30%	Implementation
20%	Testing
20%	Running + Numerical Analysis
10%	Visualization + Physical Analysis

- Recommended group size: 3 or 4 students. Please send me an email per group with the list of group members.
- You can ask for feedback from me while you are working on the project (**and you should do it at least once per project**, for instance during the optional exercise session or in my office. We can discuss all aspects of the project: planning choices, code bugs and problems, theoretical and algorithmic aspects. However when you ask for my feedback you must come with clear questions, you must show to me that you have thought about problems.

Contents

1 Overview of goals

- Read the attached lectude notes.
- Design and write a code to generate a Markov chain which simulates the Ising model, using the Metropolis algorithm (explained in the lecture notes). You should consider a lattice with D dimensions, N points in each direction and periodic boundary conditions. You may recycle the geometry code developed for the time-dependent Schrödinger equation.

  Notice: It is intrisically hard to test the Metropolis algorithm, because of its stochastic nature. Only partial tests of the code are possible, e.g.

  - the usual geometry tests;
  - check the value of the Hamiltonian against analytic calculation on simple configurations, such as all spin up, all spin down, half space up and half space down;
  - isolate the function that calculates $b^s(z)$ and check that $H(s) = -\sum_z b^s(z)s(z)$ (this is a consistency check).
- As explained in the notes, the Metropolis algorithm produces a sequence of configurations. For each configuration, you should calculate energy and magnetization density. For a given of observable (e.g. the energy density), the sequence of values calculated on each configurations is called the *history* of that observable. Generate 50000 configurations and plot the history of observables for the following set of parameters

  - $N = 100$, $D = 2$, $\beta = 0.1$, $B = 0$;
  - $N = 100$, $D = 2$, $\beta = 0.3$, $B = 0$;
  - $N = 100$, $D = 2$, $\beta = 0.5$, $B = 0$;
  - $N = 100$, $D = 2$, $\beta = 0.7$, $B = 0$;
  - $N = 100$, $D = 2$, $\beta = 0.1$, $B = 0.01$;
  - $N = 100$, $D = 2$, $\beta = 0.3$, $B = 0.01$;
  - $N = 100$, $D = 2$, $\beta = 0.5$, $B = 0.01$;
  - $N = 100$, $D = 2$, $\beta = 0.7$, $B = 0.01$.
  - $N = 100$, $D = 2$, $\beta = 0.1$, $B = 0.03$;
  - $N = 100$, $D = 2$, $\beta = 0.3$, $B = 0.03$;
  - $N = 100$, $D = 2$, $\beta = 0.5$, $B = 0.03$;
  - $N = 100$, $D = 2$, $\beta = 0.7$, $B = 0.03$.

  Estimate the thermalization (by eye), calculate the autocorrelation function, the average and the statistical error of the two observables. Plot the average magnetization as a function of β, for the several values of B. Provide an interpretation of the results. For extra points, you can explore more values of β and B in the interesting region.
- **New:** As part of the testing process, compare the result of average energy and magnetization for the set of parameters $N = 100$, $D = 2$, $\beta = 0.5$, $B = 0.01$. Notice that the average energy and magnetization are only stochastically determined and they come with an error. How do you make sure that the comparison is meaningful?
- Extra: Estimate the value of β which corresponds to the ferromagnetic phase transition (at $B = 0$).

2 Monte Carlo simulation for the Ising model

In this section we will discuss Monte Carlo simulation of statistical systems, using the Ising model as a particular example. Monte Carlo methods are very important method in statistical physics, solid state physics, and in a branch of theoretical elementary particle physics, i.e. lattice field theory. It plays a central role in the research programmes of the Lattice Field Theory group at HU (Patella, Wolff) and in our collaboration partners at DESY-Zeuthen (Sommer, Schaefer, Jansen).

2.1 The Ising model on a cubic lattice

The Ising model is a simplified model of a ferromagnet. Its simplicity makes it a natural choice to illustrate how one can use numerical simulations to extract information about the physics. The first step is to define the model.

We imagine a regular cubic crystal (**lattice**), in a generic number D of dimensions ($D = 3$ corresponds to the real world). Lattice sites are identified by D integer coordinates:

$$\text{lattice site} = x = (x_0, x_1, .., x_{D-1}), \quad x_\mu \in \mathbb{Z} \ . \tag{2}$$

In order to be able to simulate the system on a computer, we need to consider a finite lattice (i.e. finite volume). Therefore we will require that the coordinates satisfy

$$0 \leq x_\mu < N \ . \tag{3}$$

The volume of the lattice is defined as

$$V = N^D \ . \tag{4}$$

We will be interested in thermodynamic properties of the system. A necessary condition for a physical system to obey the laws of thermodynamics is that its volume is very large, therefore we will need to take the $N \to \infty$ limit. Notice that the lattice spacing is taken to be $a = 1$. In this context, the lattice spacing is kept constant and is equal to the minimal distance between distinct atoms in the crystal (i.e. we are not interested in taking the continuum limit).

Ferromagnetic effects in solids are caused by electron spins. In more realistic models, the total spin of each atom (located in the lattice sites) obeys the laws of quantum mechanics and can be oriented in any possible direction. In the Ising model, the spin is thought as a classical vector with fixed magnitude which can be oriented only in two possible ways, which are conventionally referred to as *up* and *down*. Therefore the spin is described by a variable $s(x) \in \{-1, 1\}$ in each lattice site.

A **configuration** s of the system is specified when the value of the spin variable $s(x)$ is assigned in all lattice sites. There are in total 2^V distinct configurations. The set of all configurations is often referred to as phase space.

Statistical physics tells us that all thermodynamical properties (e.g. relation between energy density, temperature, entropy in thermal equilibrium) are completely determined once we know the energy H of the system for each configuration. In the Ising model this is given by

$$H(s) = -\sum_{\langle xy \rangle} s(x)s(y) - B\sum_x s(x) \ . \tag{5}$$

The first sum runs over all pairs $\langle xy \rangle$ of nearest neighbouring sites, i.e. pairs of site x and y whose distance is exactly 1. The pair $\langle xy \rangle$ contributes to the total energy with the term $-s(x)s(y)$. This contribution is positive and equal to 1 if the two spins are antiparallel, and it is negative and equal to -1 if the two spins are parallel. It is said that the system prefers to have all spins oriented in the same direction, and one needs to provide energy in order to make some spins flip. The second term in eq. (5) describes the interaction with an external magnetic field B which can be oriented only in two possible ways: up (if $B > 0$) or down

(if $B < 0$). Each site contributes to the energy with the term $-Bs(x)$. This contribution is positive and equal to $|B|$ if the spin has the opposite sign to B (i.e. it is antiparallel to the magentic field), and it is negative and equal to $|B|$ if the spin has the same sign as B (i.e. it is parallel to the magentic field). It is said that the system prefers to have all spins oriented in the same direction as the magnetic field, and one needs to provide energy in order to make some spins flip.

We will consider the lattice with periodic boundary conditions (but this is not the only possible choice). This means for instance that (with $D = 3$) the two lattice sites $(0, 0, 0)$ and $(0, N-1, 0)$ have distance equal to 1 and are considered as nearest neighbours. We want to write the sum over nearest neighbouring sites a bit more explicitly. We introduce the unit vectors

$$e_\mu = \Big(0 \quad \cdots \quad 0 \quad \underbrace{1}_{(\mu+1)\text{-th component}} \quad 0 \quad \cdots \quad 0 \Big) , \qquad \mu = 0, \ldots, D-1 . \tag{6}$$

Given an integer p we define $\mathtt{mod}(p, N)$ to be the reminder of the integer division of p by N. By definition, $0 \le \mathtt{mod}(p, N) < N$. Periodic boundary conditions are implemented by interpreting the sum of vector coordinates always modulo N, i.e.

$$v = \big(v_0, v_1, \ldots, v_{D-1}\big) \ , \tag{7}$$

$$w = \big(w_0, w_1, \ldots, w_{D-1}\big) \ , \tag{8}$$

$$v \oplus w \stackrel{\text{def.}}{=} \big(\mathtt{mod}(v_0 + w_0, N) \ , \mathtt{mod}(v_1 + w_1, N) \ , \ldots , \mathtt{mod}(v_{D-1} + w_{D-1}, N)\big) \ . \tag{9}$$

The nearest neighbouring interaction term of the energy can be written as

$$\sum_{\langle xy \rangle} s(x)s(y) = \sum_{x,\mu} s(x)s(x \oplus e_\mu) \ . \tag{10}$$

The sum in the r.h.s. runs over all sites x of the lattice, and over $\mu = 0, 1, \ldots, D-1$.

Programming tips

A program which simulates the Ising model needs to store the configuration into some array. One can choose to work with a D-dimensionsa array. However this requires that D is fixed once and for all. If we want to keep D generic, it is more convenient to work with one-dimensional arrays. This can be done by assigning an index $n(x)$ to each lattice site x, in such a way that the map between indices and lattice sites is one-to-one. The way indices are chosen is largely arbitrary. For instance one can choose

$$n(x) = x_0 + Nx_1 + N^2 x_2 + \cdots + N^{D-1} x_{D-1} \ . \tag{11}$$

When x varies through the whole lattice, $n(x)$ takes all possible values from 0 to $V - 1$. One can show that $n = n(x)$ if and only if

$$x_k = \mathtt{mod}(\mathtt{div}(n, N^k), N) \ , \tag{12}$$

where $\mathtt{div}(a, b)$ and $\mathtt{mod}(a, b)$ are respectively the quotient and the reminder of the integer division of a by b.

Therefore the configuration s can be stored into a one-dimensional array $\mathtt{s}$, such that

$$\mathtt{s}(n(x)) = s(x) \ . \tag{13}$$

The magnetic-field term of the energy can be written as

$$-B \sum_x s(x) = -B \sum_{n=0}^{V-1} \mathtt{s}(n) \ . \tag{14}$$

In order to write the nearest neighbour interaction, one needs to calculate the index associated to $x \otimes e_\mu$. It may be convenient to store this index in a new two-dimensional array `nn`, defined as

$$\texttt{nn}(n(x), \mu) = n(x \otimes e_\mu) \ . \tag{15}$$

Once this array is calculated, the nearest neighbour interaction can be written simply as

$$-\sum_{\langle xy \rangle} s(x)s(y) = -\sum_{n=0}^{V-1} \texttt{s}(n) \left[\sum_{\mu=0}^{D-1} \texttt{s}(\texttt{nn}(n, \mu)) \right] \ . \tag{16}$$

The pseudocode 1 shows a possible way to structure the program that calculates the map $x \mapsto n(x)$ and its inverse, calculates the array `nn`, and calculates the energy of a given configuration.

2.2 Observable in thermodynamic equilibrium

A microscopics state for the Ising model is competely determined by the configuration s. In a system with many degrees of freedom (e.g. of order of the Avogadro number), the complete characterization of the microscopic state is impossible in practice. In many applications, one has only a probabilistic knowledge of the microscopics state of the system, i.e. one knows that the system is in the state s with probability $P(s)$. One also says that the system is not described by a microscopic state, but by a **statistic ensemble** defined by the **probability distribution** P.

Give some fairly general conditions (which we will not discuss here), a system with energy function H in thermodynamic equilibrium with temperature $T = (k_\mathrm{B}\beta)^{-1}$ is described by the canonical ensemble,[1] i.e. the statistic ensemble defined by the probability distribution

$$P(s) = \frac{1}{Z} e^{-\beta H(s)} \ , \qquad Z = \sum_s e^{-\beta H(s)} \ . \tag{17}$$

The normalization factor Z is referred to as the **partition function**. We remind that the sum runs over all possible 2^V configurations.

A physical observable is a general function $A(s)$ of the configuration. The **thermal average** $\langle A \rangle$ of the observable A is given by the **expectation value** with the probability of the canonical ensemble, i.e.

$$\langle A \rangle = \sum_s A(s)P(s) = \frac{1}{Z} \sum_s A(s) e^{-\beta H(s)} \ . \tag{18}$$

Interesting observables are e.g. the **energy**

$$\langle H \rangle = \frac{1}{Z} \sum_s H(s) e^{-\beta H(s)} = -\frac{\partial \ln Z}{\partial \beta} \ , \tag{19}$$

or the **magnetization**

$$M(s) = \sum_x s(x) \ , \tag{20}$$

$$\langle M \rangle = \frac{1}{Z} \sum_s M(s) e^{-\beta H(s)} = \frac{1}{\beta} \frac{\partial \ln Z}{\partial B} \ . \tag{21}$$

Both the energy and the magnetization are **extensive quantities**, i.e. they are proportional to the volume. In the thermodynamic limit $N \to \infty$ it is more interesting to consider the **energy and magnetization densites**, i.e.

$$\epsilon = \frac{1}{V} \langle H \rangle \ , \qquad m = \frac{1}{V} \langle M \rangle \ . \tag{22}$$

[1] In natural units, which we will assume from now on, the Boltzmann constant is given by $k_\mathrm{B} = 1$.

Algorithm 1 Ising model: geometry and energy

```
1: global variables: int D, int N
2:                   int V = N^D
3:                   int nn[V]
4:                   double B

5: function Coord2Index(int x[D])                ▷ Returns the index n(x) of the site x
6:     local variables: int n
7:     n ← 0
8:     for all k = 0, 1, ..., D − 1 do
9:         n ← n + N^k mod(x_k, N)
10:    end for
11:    return n
12: end function

13: function Index2Coord(int n)                  ▷ Returns the site x with index n = n(x)
14:    local variables: int x[D]
15:    for all k = 0, 1, ..., D − 1 do
16:        x_k ← mod(div(n, N^k), N)
17:    end for
18:    return x
19: end function

20: procedure SetGeometry                        ▷ Defines the nn array
21:    local variables: int x[D]
22:    for all n = 0, 1, ..., V − 1 do
23:        x ←Index2Coord(n)
24:        for all μ = 0, 1, ..., D − 1 do
25:            x_μ ← x_μ + 1
26:            nn(n, μ) ←Coord2Index(x)
27:            x_μ ← x_μ − 1
28:        end for
29:    end for
30: end procedure

31: function Energy(int s[V])                    ▷ Returns the energy of the configuration s
32:    local variables: double H, d
33:    H ← 0
34:    for all n = 0, 1, ..., V − 1 do
35:        d ← 0
36:        for all μ = 0, 1, ..., D − 1 do
37:            d ← d + s(nn(n, μ))
38:        end for
39:        H ← H − (d + B) s(n)
40:    end for
41:    return H
42: end function
```

Other intersting quantities are the n-point correlation functions, i.e.

$$G_n(x^1, x^2, \ldots, x^n) = \langle s(x^1)s(x^2)\cdots s(x^n)\rangle \ . \tag{23}$$

In particular one can look at the 2-point correlation function (or simply correlation function)

$$G(x, y) = \langle s(x)s(y)\rangle \ . \tag{24}$$

It describes the correlation between two possibly distant spins. For example, if it is positive, there is a tendency for these spins to be parallel. If this is the case for spins at infinite distance, even at $B = 0$, then spontaneous magnetization is present, which is the actual phenomenon of ferromagnetism. This happens in the Ising model for $D > 2$ and sufficiently small temperature (large β).

2.3 Monte Carlo simulation using the Ising model as an example

In this section we discuss how to calculate the expectation value of observables numerically. We may think to calculate the sum

$$\sum_s e^{-\beta H(s)} \tag{25}$$

exactly. Consider a two dimensional system with $N = 10$ sites in each direction. Then the sum runs over 2^{100} configurations. Assume that we can calculate the function $e^{-\beta H(s)}$ for each configuration in 10^{-9}s (which corresponds to a frequency of 1GHz), then the whole sum will be calculated in

$$2^{100} \times 10^{-9} \text{ s} = 1.27 \times 10^{21} \text{ s} = 4 \times 10^{13} \text{ years} \simeq 3000 \times \text{age-of-the-universe} \ . \tag{26}$$

We need to resort to other techniques!

Assume $B > 0$, then the absolute minimum of the energy is obtained for the constant configuration $\bar{s}(x) = 1$, and

$$E_{\min} = \min_s H(s) = H(\bar{s}) = -(D + B)V \ . \tag{27}$$

We rewrite the expecation value of a generic observable as

$$\langle A\rangle = \frac{\sum_s A(s)e^{-\beta H(s)}}{\sum_s e^{-\beta H(s)}} = \frac{\sum_s A(s)e^{-\beta[H(s)-E_{\min}]}}{\sum_s e^{-\beta[H(s)-E_{\min}]}} \ . \tag{28}$$

The exponential now satisfies

$$0 < e^{-\beta[H(s)-E_{\min}]} \leq 1 \ . \tag{29}$$

Configurations with an energy very close to $E_{\min}$ will give an important contribution to the sum, while configurations with energy much larger than $E_{\min}$ give a negligible contribution. For instance, if $D = 2$, $N = 10$, $\beta = B = 1$, the configuration $s = 0$ contributes with

$$e^{-\beta[H(0)-E_{\min}]} = e^{\beta E_{\min}} = e^{-\beta(D+B)V} = 5 \times 10^{-131} \ . \tag{30}$$

Monte Carlo algorithms with important sampling allow to construct only configurations that contribute significantly to the sum.

The main idea is to construct a sequence $s_1, s_2, s_3, \ldots, s_N$ of configurations with the property that

$$\langle A\rangle = \lim_{N\to\infty} \frac{1}{N} \sum_{i=1}^{N} A(s_i) \ , \tag{31}$$

i.e. the average over the sequence (s_i) is equal to the expectation value with respect to the probability $P(s)$, in the limit of infinitely long sequence.

Ideally one would like to require that the configurations s_i are independent, and individually distributed with probability $P(s)$. This strategy is called *independent sampling*. For general systems (and in particular for the Ising model), algorithms for independent sampling are not known. Instead, we will require that configuration s_{i+1} depends only on the configuration s_i, and not on the previously generated ones. A sequence of this type is called **Markov chain**. We will construct our algorithm by iterating always the same elementary step (**Monte Carlo step or iteration**). We start from some arbitrarily-chosen **initial configuration** s_1. We apply the MC step to s_1, and we produce s_2. We apply the MC step to s_2, and we produce s_3, and so on. Schematically

$$s_1 \xrightarrow{MC} s_2 \xrightarrow{MC} \dots \xrightarrow{MC} s_{i-1} \xrightarrow{MC} s_i \xrightarrow{MC} s_{i+1} \xrightarrow{MC} \dots \xrightarrow{MC} s_N \ . \tag{32}$$

The algorithm is completely specified once the MC step is defined. The MC step needs to be chosen in such a way that eq. (31) is satisfied. We will present here a particular choice for the MC step.

We introduce the *spin flip* function $\mathcal{F}^z$. For any configuration s, we define $\mathcal{F}^z[s]$ as the configuration which is obtained from s by flipping the sign of $s(z)$, i.e.

$$\mathcal{F}^z[s](x) = \begin{cases} s(x) & \text{if } x \neq z \\ -s(x) & \text{if } x = z \end{cases} \ . \tag{33}$$

The MC step is defined by the following algorithm.

1. **Input:** s
2. **for all** z **in the lattice, do**
3. **replace** $s \leftarrow \mathcal{F}^z[s]$ **with probability** $\min\{1, e^{-\beta[H(\mathcal{F}^z[s])-H(s)]}\}$
4. **end for**
5. **Output:** s

We want to rewite this algorithm a bit more explicitly. First we notice that the exponent appearing in the probability in step 3 can be rewritten as

$$H(\mathcal{F}^z[s]) - H(s) = 2 \sum_{x \text{ neighbour of } z} s(x)s(z) + 2Bs(z) \stackrel{\text{def.}}{=} 2b^s(z)s(z) \ . \tag{34}$$

Notice that most of the terms cancel in the difference, so one should not calculate them. Step 3 can be reformulated as

3. **replace** $s(z) \leftarrow -s(z)$ **with probability** $\min\{1, e^{-2\beta b^s(z)s(z)}\}$

The behaviour of the algorithm is different in the two cases: $e^{-2\beta b^s(z)s(z)} < 1$ and $e^{-2\beta b^s(z)s(z)} \geq 1$. If $e^{-2\beta b^s(z)s(z)} \geq 1$, then the sign of $s(x)$ must be flipped with probability 1, i.e. with certainty. If $e^{-2\beta b^s(z)s(z)} < 1$, then the sign of $s(x)$ must be flipped with probability $0 < e^{-2\beta b^s(z)s(z)} < 1$. This can be done with the help of a pseudorandom generator. Step 3 can be reformulated as

3a. **set** $p \leftarrow e^{-2\beta b^s(z)s(z)}$
3b. **if** $p \geq 1$ **then**
3c. **replace** $s(z) \leftarrow -s(z)$
3d. **else**
3e. **let** r **be a pseudorandom number with uniform distribution in** $[0, 1]$
3f. **if** $r < p$ **then**
3g. **replace** $s(z) \leftarrow -s(z)$
3h. **end if**
3i. **end if**

This concludes the description of the algorithm which performs a MC step.

2.4 Observable measurement and analysis

We are interested in expectation values of observables. For defineteness, we will focus on the energy, but the same analysis can be repeated for any other observable (e.g. the magnetization). We have already noticed that the simulation algorithm produces a sequence $s_1, s_2, s_3, \ldots, s_N$ of configurations with the property that

$$\langle H \rangle = \lim_{N\to\infty} \frac{1}{N} \sum_{i=1}^{N} H(s_i) \ . \tag{35}$$

However in practice we do not have an infinitely-long sequence, but only a finite one. Therefore we are not able to calculate the expectation value of the energy, but only an estimate given by the *mean* or *average* over the finite sequence. Since this is not exact, we also want to find a way to estimate the error. The process of extracting the average and error from the data goes under the name of *statistical analysis*. We will present here a simple way to perform the statistical analysis (in more complicated simulations, this usually needs to be refined).

1. The index n which identifies the n-th configuration s_n is usually referred to as *Monte Carlo time* (this is just terminology, and there is no relation to real time). The first step is always to plot the *history* of the observable, i.e. the sequence $H(s_1), H(s_2), H(s_3), \ldots$ as a function of the Monte Carlo time. You will notice that, typically, the value of the observables starts from a number $H(s_1)$ which is determined by the initial condition, it moves away from this value and starts fluctuating around some typical value (sometimes it may jump between different values). The transient phase is called *thermalizaton*. One needs to estimate, usually by eye, the length of the thermalization. Configurations in the theralization phase are thrown away and never considered in the analysis. In the following, $n = 1$ will correspond to the first thermalized configuration, and N will be the number of thermalized configurations.

2. Calculate now the average and standard deviation on the thermalized configurations, i.e.

$$\bar{H} = \frac{1}{N} \sum_{i=1}^{N} H(s_i) \ , \tag{36}$$

$$\sigma(H) = \sqrt{\frac{1}{N} \sum_{i=1}^{N} [H(s_i) - \bar{H}]^2} \ . \tag{37}$$

   The average is an estimate for the expectation value of the energy. The standard deviation is an estimate for the size of the thermal fluctuations of the energy. We want to provide an estimate for the error of the average. The *central limit theorem* tells you that, if the configurations are statistically independent, then the error ΔH on the average is related to the standard deviation by the simple formula $\Delta H = \frac{\sigma(H)}{\sqrt{N}}$ (notice that the error goes to zero for $N \to \infty$). *However a Markov chain does not produce independent configurations: every configuration depends on the previous one.* The correct way to deal with this issue is to look at autocorrelations.

3. The autocorrelation function (of the particular observable we are considering) is defined for $k \geq 0$ as

$$C(k) = \frac{1}{N-k} \sum_{i=1}^{N-k} [H(s_{i+k}) - \bar{H}][H(s_i) - \bar{H}] \tag{38}$$

   where again the thermalization is not considered in the sum. One can prove that the behaviour for large k of the autocorrelation function is given by

$$C(k) \overset{N,k\to\infty}{\simeq} A e^{-\frac{k}{\tau}} \tag{39}$$

   where τ is referred at the *autocorrelation time* of the considered observable. There are several ways to estimate τ. One possibility is to fit a function $Ae^{-\frac{k}{\tau}}$ to the autocorrelation function $C(k)$. Another possibility is to use the so-called *integrated autocorrelation time*, defined as

$$\tau^{-1} = \frac{1}{C(0)} \sum_{k=0}^{\infty} C(k) \ . \tag{40}$$

You should plot the autocorrelation function, estimate the autocorrelation time using both methods, and compare the two methods.

4. If the estimated autocorrelation time is given by τ, this means that two configurations separated by a few units of τ (e.g. 2τ) are almost independent. For instance, in a set of N configurations, one can roughly say that there are effectively

$$N_{\text{eff}} = \frac{N}{2\tau} \tag{41}$$

independent configurations. Then the error on the average of the energy can be estimated to be

$$\Delta H = \frac{\sigma(H)}{\sqrt{N_{\text{eff}}}} \ . \tag{42}$$

2.5 Extra: Transition probability and detailed balance

The MC step is not a deterministic procedure. Given the input configuration s, the output configuration s' is constructed by means of a stochastic process. It make sense to define the probability that the output configuration be s' *given that* the input configuration is s. This probability is called *transition probability* of the Markov chain and is denoted by

$$W(s' \leftarrow s) \ . \tag{43}$$

The transition probability must satisfy the axioms of probability, i.e.

$$W(s' \leftarrow s) \geq 0 \ , \tag{44}$$

$$\sum_{s'} W(s' \leftarrow s) = 1 \ . \tag{45}$$

We want to write an expression for W. Notice that the MC step is obtained by iterating the elementary step 3 for each point of the lattice. If we fix the lattice point z, we can define the transition probability for the elementary step 3, which we will denote by $w^z(s' \leftarrow s)$. If s is the input configuration for step 3, the output configuration is $\mathcal{F}^z[s]$ with probability

$$p^z(s) = \min\left\{1, \frac{e^{-\beta H(\mathcal{F}^z[s])}}{e^{-\beta H(s)}}\right\} \ , \tag{46}$$

and s itself with probability $1 - p^z(s)$. Therefore the transition probability

$$w^z(s' \leftarrow s) = \delta_{s',\mathcal{F}^z[s]} p^z(s) + \delta_{s',s}(1 - p^z(s)) \ . \tag{47}$$

Let us say not that we apply step 3 twice in a row. We start from the configuration s, we apply step 3 on the site z^i, which produces the configuration s'. Then we apply step 3 on the site z^i to s', which produces the configuration s''. The probability to obtain s'' after these two steps is given by

$$(w^{z^{i+1}} \circ w^{z^i})(s'' \leftarrow s) = \sum_{s'} w^{z^{i+1}}(s'' \leftarrow s') w^{z^i}(s' \leftarrow s) \ . \tag{48}$$

The algorithm runs over all possible lattice points $(z^1, z^2, \ldots, z^V\}$ in a given order. The transition probability for the complete MC step is given by

$$W = w^{z^V} \circ w^{z^{V-1}} \circ \cdots \circ w^{z^2} \circ w^{z^1} \ . \tag{49}$$

At this point we have described the algorithm, but we have not proven that it works. Recall that our goal is to have an algorithm which produces a sequence of configurations, such that eq. (31) is satisfied. There is a theorem for Markov chains which can be used to show that our algorithm works. The theorem says that, if the following two conditions are satisfied:

1. (*stability*) the target probability $P(s)$ is a stationary probability for W, i.e. if the initial configuration is randomly chosen with probability $P(s)$, after applying one MC step, the output configuration s' is still distributed with probability $P(s)$, i.e.

$$\sum_s W(s' \leftarrow s)P(s) = P(s') \ , \tag{50}$$

2. (*ergodicity*) the Markov chain is ergodic, i.e. given two configurations s and s', an integer n exists such that the probability to produce s' starting from s after n MC steps is larger than zero, i.e.

$$W^n(s' \leftarrow s) = (\underbrace{W \circ W \circ \cdots \circ W}_{n \text{ times}})(s' \leftarrow s) > 0 \ . \tag{51}$$

These properties can be proven to hold for our algorithm. *Ergodicity* is quite difficult to prove. *Stability* follows from *detailed balance*: given two configurations s and s'

$$W(s' \leftarrow s)P(s) = W(s \leftarrow s')P(s') \ , \tag{52}$$

which is easy to prove for our algorithm.