# Monte Carlo simulations of the Harmonic Oscillator and the $\phi^4$ Ising model

Federico Anastasi

February 15, 2021

## Contents

**Abstract**

This article presents two applications of the Monte Carlo method: a one-dimensional harmonic oscillator system and a three-dimensional scalar field theory, usually called $\phi^4$. These systems are interesting for different reasons. The first one is one of the simplest quantum mechanics systems, thus it has an analytic solution whereby it is possible to compare the result of the experiment. In particular, the interesting quantities are the energy gap of the first excited state, $\Delta E = E_1 - E_0$, and the first non-vanishing matrix element $|\langle E_0|x|E_1\rangle|$.

On the other hand, there is not an exact solution to the scalar field theory, so numerical simulations are crucial. In three dimensions, this theory models a spin system, which undergoes a phase transition at the critical point. Measuring the magnetization and the susceptibility, it is possible to estimate the critical exponents which guide the behavior of these quantities close to the critical point.

Both systems are discrete versions of the real ones. In the harmonic oscillator, the time is discrete, and to recover the analytic solutions it is required to interpolate the data to the continuum limit. Regarding the scalar theory, the field has a finite number of degrees of freedom which correspond to its volume $V$, so in this case, the infinite volume limit has been considered.

# Part I
# Harmonic Oscillator

## 1 Introduction

In the last fifty years, the numerical methods in physics became one of the most powerful strategy to solve complicated system that often have not analytic solutions. In particular, to investigate quantum field theories in the non-perturbative regime, the Monte Carlo methods turn out to be the only possibility. The first part of this paper discuss an implementation of these methods: the Metropolis algorithm [10], which can be considered the simplest way to build a Monte Carlo simulation.

The harmonic oscillator has nothing to do with quantum field theories, rather this system is one of simplest systems to solve in the Schrödinger picture of quantum mechanics. Thanks to its simplicity, the implementation of a Monte Carlo algorithm is straightforward. Moreover, this system gives the possibility to check the accuracy of this method, indeed the results can be compared with the analytic solutions. For these reasons, this simulation represents the ideal first approach to Monte Carlo methods.

The goal of this experiment is to recover the same results that the theory gives; however, to implement numerically any system it is required to discretize it. In this case, the harmonic oscillator is one-dimensional, so there is a scalar

position operator $\hat{x}(t)$, which is time-dependent, but the time is discrete, $t_i = ia$, in which $a$ is the lattice spacing and $i = 0, \ldots, N-1$. The system is always observed for a fixed time, $T = Na$, and the interesting quantities are the energy gap between the first excited state and the vacuum, $\Delta E = E_1 - E_0$, and the first non-vanishing matrix element, $|\langle E_1|\hat{x}|E_0\rangle|$. Clearly, the discrete system is different from the continuum one, thus to recover the analytic solution it needs to consider the continuum limit, namely $a \to 0$, $N \to \infty$ with $T = constant$. In order to do that, smaller and smaller lattice spacing will be considered and finally the results will be interpolate.

## 1.1 The system

The Hamiltonian of the harmonic oscillator is well known as its solution written below.

$$H = \frac{p^2}{2m} + V \qquad \text{with:} \quad V = \frac{1}{2}m\omega^2 x^2. \tag{1.1.1}$$

$$\Delta E = \omega \quad \langle E_0|x|E_1\rangle = \frac{1}{\sqrt{2m\omega}} \tag{1.1.2}$$

Actually, it is required to define two more Hamiltonians, given that a numerical implementation works with discrete quantities. The first one, $\tilde{H}$, is implicitly defined by the transfer matrix of the system $T$ as follows:

$$T = e^{-a\tilde{H}} = e^{-\frac{a}{2}V} e^{-a\frac{p^2}{2m}} e^{-\frac{a}{2}V}. \tag{1.1.3}$$

Then, to capture the discretization errors it is possible to define an auxiliary Hamiltonian $\bar{H}$:

$$\bar{H} = \frac{p^2}{2m} + \frac{1}{2}m\bar{\omega}^2 x^2 \quad \text{with} \quad \bar{\omega}^2 = \omega^2\Big(1 + \frac{a^2\omega^2}{4}\Big). \tag{1.1.4}$$

in which $\bar{\omega}$ tends to $\omega$ as $a \to 0$, namely in the continuum limit. Then, writing $\bar{H}$ in terms of annihilation and creation operators, it is possible to show that $[T, \bar{H}] = 0$. So, the transfer matrix and the auxiliary Hamiltonian have the same eigenvectors and this means that one can diagonalize both operators with respect to the same basis, namely $|\tilde{E}_n\rangle = |\bar{E}_n\rangle$.

With this notation, the energy gap between two consecutive state is defined as $\tilde{\omega} = \tilde{E}_{n+1} - \tilde{E}_n$. What the simulation computes as energy gap is precisely $\tilde{\omega}$ because the transfer matrix, thus $\tilde{H}$, defines the path integral. On the other hand, $\bar{\omega}$ is the quantity of interest of the experiment, given that:

$$\Delta\bar{E} = \bar{\omega} \xrightarrow{a\to 0} \Delta E = \omega \tag{1.1.5}$$

Since $T$ and $\bar{H}$ have the same eigenvectors, a relation between $\tilde{\omega}$ and $\bar{\omega}$ can be found.

$$a\tilde{\omega} = \log\Big(1 + a\bar{\omega} + \frac{a^2\omega^2}{2}\Big) \tag{1.1.6}$$

Finally, the transfer matrix leads to the Euclidean path integral with the following discrete Euclidean action:

$$S = \sum_{i=1}^{N} a\Big[\frac{1}{2}m\Big(\frac{x_{i+1}-x_i}{a}\Big)^2 + \frac{1}{4}m\omega^2(x_{i+1}^2+x_i^2)\Big] \tag{1.1.7}$$

for simplicity $m$ and $\omega$ are fixed to 1 and periodic boundary conditions are enforced, namely $x_0 = x_N$. Actually, in the simulation there is no explicit $a$ dependence in $S$, indeed it is possible to redefine the parameters as follows:

$$S = \sum_{i=1}^{N}\Big[\frac{1}{2}M(x_{i+1}-x_i)^2 + \frac{1}{4}M\Omega^2(x_{i+1}^2+x_i^2)\Big] \qquad \text{with: } M = \frac{1}{a} \quad \Omega = a \tag{1.1.8}$$

In order to compute the energy gap and the matrix element, the correlator $\langle x(l)x(k)\rangle$ has to be computed, as for large enough values of $N$ and $|l-k|$ [11], the leading term contains both of them:

$$\langle x(l)x(k)\rangle = 2|\,\langle \tilde{E}_1|\hat{x}|\tilde{E}_0\rangle\,|^2 e^{-\frac{N}{2}a(\tilde{E}_1-\tilde{E}_0)}\cosh\Big[\Big(\frac{N}{2}-|l-k|\Big)a(\tilde{E}_1-\tilde{E}_0)\Big] \tag{1.1.9}$$

For this reason, the algorithm has to compute as many correlators as many are the combinations of the pair $(l,k)$ for each configuration of the Markov chain. Fortunately, the correlator is a function of $|l-k| = t$ so there are $N$ different correlators to compute.

## 1.2 The simulation

The Euclidean path integral remembers a lot the statistical mechanics partition function, it weights each possible path with $e^{-S[\gamma]}$ in which $\gamma$ represents a generic path. In this case, a certain path, or configuration $\gamma_\alpha$, is a set of positions at each time, $\gamma_\alpha = \{x_1^\alpha, \ldots, x_N^\alpha\}$. In general, it can be defined as the set of values of the field which is integrated in the path integral.

Given $N_{sweep}$ configurations of the system, generated through the Metropolis algorithm, to compute numerically the expectation value of any operator $A$ the algorithm takes their average.

$$\langle A\rangle = \frac{\int Dx\, A\, e^{-S(x)}}{\int Dx\, e^{-S(x)}} \quad \Leftrightarrow \quad \overline{\langle A\rangle} = \frac{1}{N_{sweep}}\sum_{\gamma} A[\gamma] \tag{1.2.1}$$

The Metropolis generates a new path $\gamma_\alpha$ considering the value of $e^{-S[\gamma_\alpha]}$. The details of how it works will be discussed in section (2.2), but the main feature of this algorithm is that it produces a new path $\gamma_\beta$ from the old one and then it decides to move the system from $\gamma_\alpha$ to $\gamma_\beta$ whether the variation of the action is little. This process simulates on a discrete lattice what the path integral does in the continuum.

To catch the continuum behavior, computation of observables have been done tuning the parameter $a$. The time interval is related to the discretization errors of the numerical simulation, but then there are also statistical errors that depend on the number of configurations considered, $N_{sweep}$. As showed in the previous section, discretization errors are taken in account in the equation (1.1.6). This fact underlines why this system is so interesting: not only it has analytic solutions but these solutions care about the discretization. For example, considering a generic potential, to find the results in the continuum limit the only possibility is to interpolate for $a \to 0$.

The following table resumes the parameters of the action used in the simulations.

| $\frac{1}{a}$ | $N$ | $M$ | $\Omega$ |
|---|---|---|---|
| 0.5 | 32 | 0.5 | 2 |
| 1 | 64 | 1 | 1 |
| 2 | 128 | 2 | 0.5 |
| 4 | 256 | 4 | 0.25 |
| 8 | 512 | 8 | 0.125 |
| 16 | 1024 | 16 | 0.0625 |

**Table 1:** Parameters of the action used in the simulations. The system is observed for a fixed time $T = Na = 64$.

The structure of the algorithm is such that the computational time grows quadratically with $N$ and linearly with $N_{sweep}$, so $N$ bigger than 1024 was not considered to avoid too long simulations. Concerning $N_{sweep}$, it has been fixed considering the errors on observables starting from $10^6$ .

Actually, there are three more parameters to be fixed before to compute an observable: the thermalization time $ThermalT$, the dimension of the bin $D_{bin}$, and the Metropolis parameter $\lambda$. These parameters concern the algorithm not the action. Their role and values will be discussed in the next section in which all steps of the algorithm will be clear. The table (3) of the section (3), in which the final results will be presented, really contains all parameters of the algorithm chosen for each value of $N$.

# 2 Materials and methods

## 2.1 General structure of the algorithm

The algorithm is quite simple: the code which implement the Metropolis is discussed in section (2.2) and, given a configuration $\gamma_\alpha$, it is possible to compute the correlator as a product $x(l)x(k)$. Then, the average of it is taken over the $N_{sweep}$ configurations.

Actually, before to compute any correlator, it is required to fix the algorithm parameters. The set of paths on which the average is taken constitutes a *Markov*

*chain* [16] just after the simulation thermalized. Given a generic initial configuration, the number of Metropolis step required to thermalize is $ThermalT$ and it is fixed in section (2.3). Moreover, a Markov chain produces correlated data and to minimize statistical error it is crucial to work with independent data. The autocorrelation function, that will be discussed in section (2.4), quantifies how much data are correlated, and looking at it, it is possible to fix $D_{bin}$.

The algorithm reads the parameter values from the input file `infile`; this allows to launch the same program with different parameters, even in parallel. The position $x(t)$ is a global variable defined as a pointer in `gobal.h` since its length $N$ is a parameter of the algorithm. The first thing that the program does is to read these parameters from the input file and allocate dynamically the memory for $x(t)$ and for the other variables which length is a parameter in the input file.

In the folder `main` there is the main program called `mainEnergy.c` which computes directly $\tilde{\omega}$ and the matrix elements, while in `devel` there are several main programs which have been used to check the correctness of the code and to set the best parameters, such as the thermalization time $ThermalT$ or the length of the bin $D_{bin}$. Finally, the folder `modules` contains `functions.c` in which the functions called from the main programs are defined. In the following sections the results of the test programs will be discussed in order to motivate the choice of the free parameter's values.

## 2.2 Metropolis

Given the probability $\pi_\alpha$ to be in the configuration $\gamma_\alpha$ and $P_{\alpha\beta}$ to move in the new configuration $\gamma_\beta$, if

$$\pi_\alpha P_{\alpha\beta} = \pi_\beta P_{\beta\alpha} \tag{2.2.1}$$

the detailed balance is satisfied, therefore the Markov chain is ergodic. To satisfy this constraint, the algorithm proposes one coordinate at a time that can be accepted or rejected. If the system is in the configuration $\gamma^\alpha$ the Metropolis algorithm proposes the new coordinate with the rule

$$x_i^{\alpha+1} = x_i^\alpha + \lambda\big(r - \frac{1}{2}\big)$$

in which $r \in [0,1)$ is a random number and $\lambda$ a free parameter of the algorithm. Then, computing the variation of the action, $\Delta S_i = S(x_i^{\alpha+1}) - S(x_i^\alpha)$, the new coordinate can be accepted or rejected:

$$\begin{cases} \text{if} \quad \frac{\pi_{\alpha+1}}{\pi_\alpha} \geq 1 \quad \text{accept} \\ \text{if} \quad \frac{\pi_{\alpha+1}}{\pi_\alpha} < 1 \quad \text{accept with probability} \quad \frac{\pi_{\alpha+1}}{\pi_\alpha} = e^{-\Delta S_i} \end{cases}$$

thus the probability to go from $\gamma_\alpha$ to $\gamma_\beta$ is

$$P_{\alpha\beta} = \begin{cases} Q_{\alpha\beta} & \text{for} \quad \frac{\pi_\beta}{\pi_\alpha} \geq 1 \\ Q_{\alpha\beta}\frac{\pi_\beta}{\pi_\alpha} & \text{for} \quad \frac{\pi_\beta}{\pi_\alpha} < 1 \end{cases} \qquad \text{with} \quad Q_{\alpha\beta} = \begin{cases} 1 & \text{for} \quad -\frac{\lambda}{2} < x_i^\alpha - x_i^\beta < \frac{\lambda}{2} \\ 0 & \text{otherwise} \end{cases} \tag{2.2.2}$$

Given that $Q_{\alpha\beta} = Q_{\beta\alpha}$, the detailed balance is satisfied. Changing only one coordinate defines a new configuration, however it is almost equal to the previous one, so this process is repeated for all $N$ coordinates. At the end, a certain number of coordinate will be different and the system is really in a new configuration. Since the variation of the action $\Delta S_i$ is computed for each coordinate $x_i$, the Metropolis can be considered a local algorithm. This means than for each path the algorithm computes $N$ times $\Delta S_i$. It is not a good idea to change all coordinates together to reduce the computational time, because the new configuration would be rarely accepted. This is due to the fact that the algorithm proposes the new configuration randomly, so it doesn't know the physics of the system, every informations about the physics are contained in the accept/reject step of the algorithm.

The parameter $\lambda$ plays a role in the optimization of the whole algorithm, it has to be fixed considering the acceptance rate of the proposed paths which is related to the decorrelation time, $T_M$. Despite most of the coordinate changes, consecutive elements of a Markov chain are correlated. This means that they are not so different. In section (2.4) the decorrelation time will be discussed since it fixes $D_{bin}$. These type of optimization of the algorithm will be discussed further in the next sections.

The following code implement the Metropolis algorithm described above: at each call of this function the system moves in a new configuration.

```
void AggCatena(int N, double M, double W, double lambda)
{
  ranlxs(r,2*N);
  for(j=0; j<N; j++)
  {
    A=(double)lambda*(r[j+N]-0.5);
    DS=DeltaS(j,A,N,M,W);

    if(exp(-1*DS) > r[j])
    xx[j]=xx[j]+A;
  }
}
```

**Figure 1:** Screenshot of the function which update the Markov chain. The computational time is proportional to $N$.

## 2.3 Thermalization

The first parameter to set is the thermalization time and the file `mainAction.c` has this scope: after the initialization of the positions $x(t)$ the Metropolis step update the positions $ThermalT$ times, for each configuration the action value is printed.

Two different starting configurations of the system have been analysed to make sure that the results are independent of the starting configuration. The Cold configuration is that one with the initial position at every time fixed to zero and the Hot one corresponds to the initial position fixed at a random value, different at each time between 0 and 10.

The Cold configuration leads to an action which start from zero and grows up until the thermalization value $T_c$ has been reached, while the Hot configuration starts from a higher value and fall down to the thermalization value $T_h$. Then, when $T_c \simeq T_h$ it is possible to say that the system thermalized, namely the configurations reached the stationary distribution $e^{-S}$ and it forms a Markov chain.

Before any computation of observable quantities it is required to run the Monte Carlo algorithm as many times as needed to be sure that the system reaches the thermalization point. To find out what is the thermalization point an arbitrary big value can be chosen as $ThermaT = 500$.

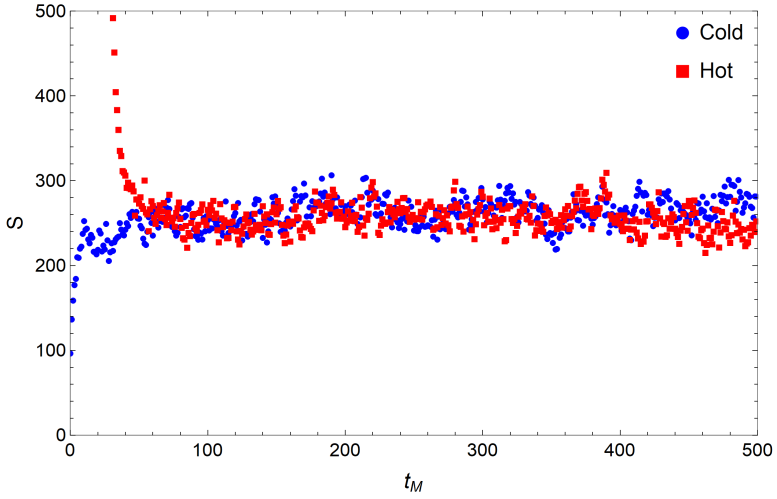



**Figure 2:** Action values for each different element of the Markov chain, $t_M$ is called Markov time. In red the data inizialized with the Hot configuration, in blue that one with the Cold one. ($N = 512$)

From the figure (9) it is possible to say that after 500 iterations of the Metropolis step the system thermalized. This computation allows to fix $ThermalT = 500$ and moves on. Further, the action values grow proportionally to $N$, as it should be, so this part of the code works.

## 2.4 Autocorrelation

It is known that consecutive configurations of a Markov chain are correlated and taking average over correlated data is not only a waste of time but also it will increase the errors. The normalized autocorrelation function, $\Gamma(t_M)$, which takes values in $[0, 1]$, quantifies the correlation between configurations. Given the correlator $C_\alpha$, evaluated for the configuration $\alpha$ of the system at the physical time $t = |l - k| = 1$, the autocorrelation function is defined as follows:

$$\Gamma(t_M) = \frac{1}{N_{sweep}} \sum_{\alpha=1}^{N_{sweep}} \frac{C_\alpha C_{\alpha+t_M} - \langle C \rangle^2}{\langle C^2 \rangle - \langle C \rangle^2} \tag{2.4.1}$$

This definition comes from that one of the variance,

$$\sigma_{\bar{C}}^2 = \frac{1}{N_{sweep}}(\langle C^2\rangle - \langle C\rangle^2)\Big(1+2\sum_{t_M=1}^{\infty}\Gamma(t_M)\Big) = \frac{\sigma^2}{N_{sweep}}\Big(1+2\sum_{t_M=1}^{\infty}\Gamma(t_M)\Big) \quad (2.4.2)$$

in which $\sigma^2$ is the usual variance for uncorrelated data thus: the faster $\Gamma$ goes to zero the smaller the errors will be. The file `mainCor.c` computes first the correlator $C(|l-k|) = \langle x(l)x(k)\rangle$ for each configuration, it saves the average values for each time and then computes the autocorrelation $\Gamma$. The following figure represents $\Gamma$ as a function of the Markov time $t_M$ for $N = 64$.

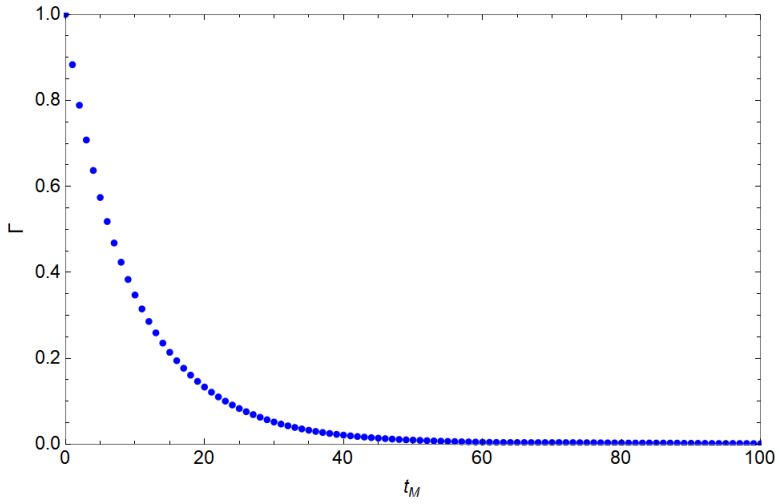



**Figure 3:** Autocorrelation $\Gamma$ as a function of the Markov time $t_M$. ($N = 64$, $N_{sweep} = 10^7$)

It has an exponential behavior, so it goes to zero quickly, for example $\Gamma(40) \simeq 0.021$. However, it is possible to make $\Gamma \simeq 0$ even before $t_M \simeq 40$, this can be done building packages of $D_{bin}$ data and considering only the average of them which represents all. Choosing $D_{bin}$ big enough, consecutive packages will be decorrelated. This means that they are really significant instead of almost equal. Then, the variance takes almost zero contribution from the sum over $\Gamma_B(t_M)$. For $N = 64$ and $D_{bin} = 10^3$, $\Gamma_B(1) \simeq -0.018$ which means that consecutive binned data are not significantly correlated. The following plot shows how the values of $\Gamma_B$ are distributed around zero.

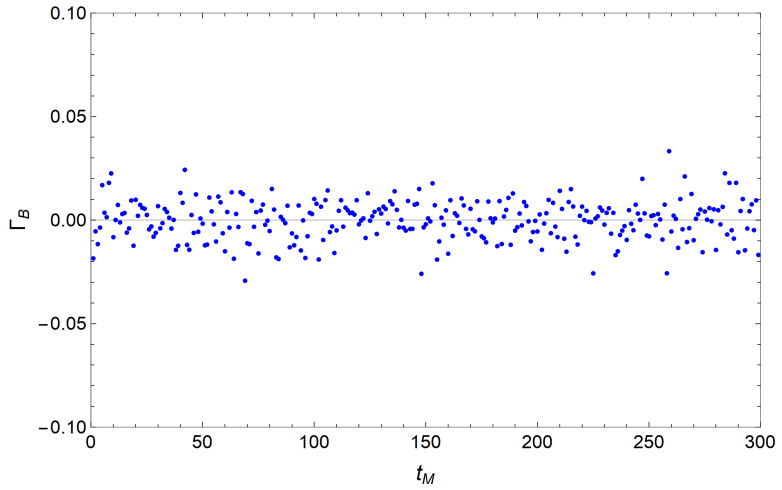



**Figure 4:** Autocorrelation $\Gamma_B$ computed with the binned data as a function of the Markov time $t_M$. ($N = 64$, $N_{sweep} = 10^7$, $D_{bin} = 10^3$)

For bigger values of $N$, $\Gamma$ goes to zero at different Markov time, in the case of $N = 512$ the autocorrelation goes to zero at $t_M$ much grater than 40.

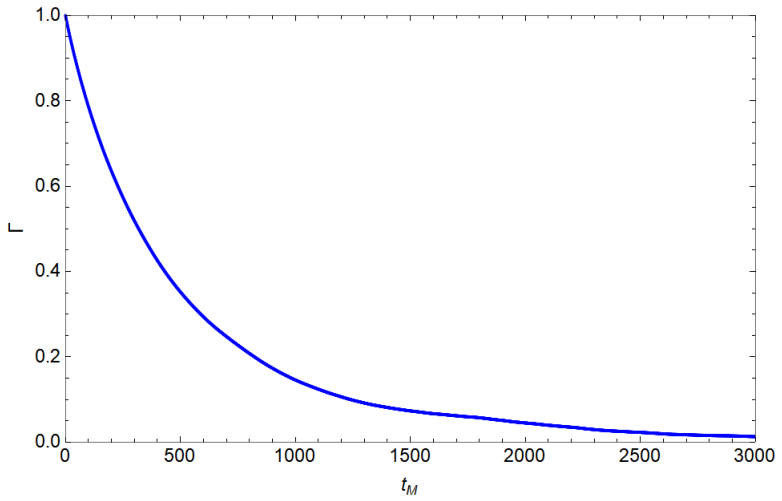



**Figure 5:** Autocorrelation $\Gamma$ as a function of the Markov time. ($N = 512$, $N_{sweep} = 10^7$)

The correlation of the data depends on how new configurations are proposed. As said before, the acceptance rate depends on the free parameter $\lambda$ in the Metropolis step. If $\lambda$ is small, the Metropolis will often accept the new configuration but it will be really similar to the old one. On the other hand, if $\lambda$ is big, the new configuration will rarely be accepted but, the accepted ones will be really different from the old one. So, both extremes lead to have correlated data at high values of $t_M$ which means that a bigger $D_{bin}$ is required. Choosing long packages of data to bin is not forbidden but it will increase the computational time at fixed error. For example, choosing $D_{bin} = 10^3$ with $N_{sweep} = 10^7$ the number of data on which the average is taken is $\frac{N_{sweep}}{D_{bin}} = 10^4$, thus for $D_{bin} = 10^4$ and fixed error the computational cost increase by a factor of 10.

Defining $T_M$ as $\Gamma(T_M) = 0.3$, it is possible to find the minimum of $T_M$ tuning $\lambda$ for different $N$. Actually, for $N = 32, 64, 128, 256$ the values of $T_M$ and the computational cost is low thus, the optimization is not strictly required. For $N = 512, 1024$ it is crucial to find the minimum.

### 2.4.1 Lambda tuning

The graph below shows the behaviour of $T_M$ as a function of $\lambda$, for small and big values of $\lambda$ $T_M$ grows so the minimum occurs around at $\lambda \simeq 1.3$.

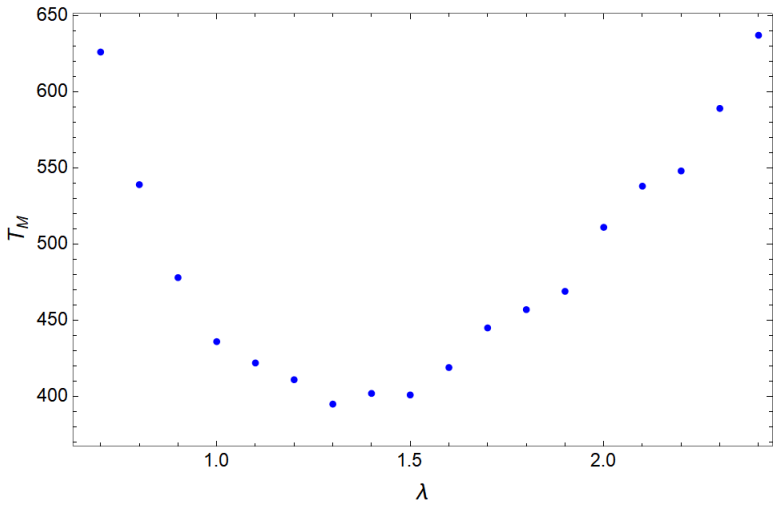



**Figure 6:** Values of $T_M$ as a function of $\lambda$. ($N = 512$, $N_{sweep} = 10^7$)

The minimum value is $T_M \simeq 400$, so $D_{bin} = 10^3$ could be enough to get $\Gamma(1) \simeq 0$. To make sure that it happens, the autocorrelation has been computed with the binned data. The results is $\Gamma_B(1) \simeq 0.25$ which is not negligible, but considering $D_{bin} = 5 \times 10^3$ and $N_{sweep} = 5 \times 10^3$, $\Gamma_B(1) \simeq 0.042$. In general, considering at least $D_{bin} = 10 \times T_M$, $\Gamma_B(1)$ should be negligible.

For what concern the case $N = 1024$, the minimum occurs around $\lambda \simeq 0.91$ with $T_M \simeq 1517$. Thus, for $N = 1024$, $D_{bin} = 15 \times 10^3$ has been chosen, the plot below represents the results, for $N_{sweep} = 10^7$.

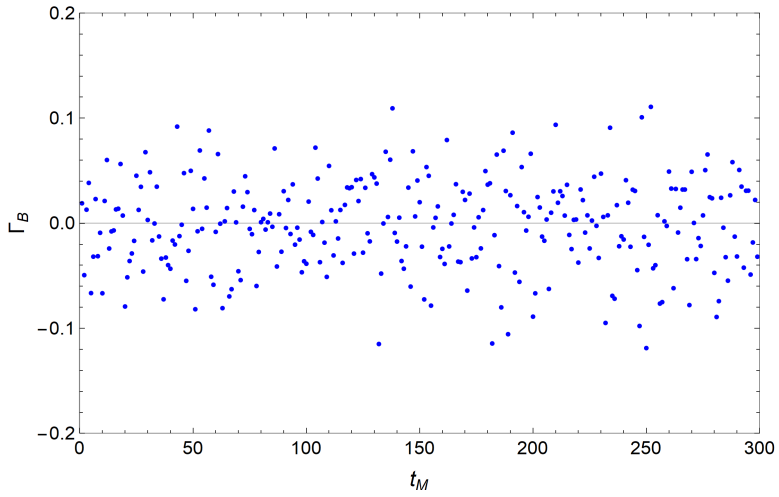



**Figure 7:** Autocorrelation $\Gamma_B$ computed with the binned data as a function of the Markov time. ($N = 1024$, $N_{sweep} = 10^7$, $D_{bin} = 15 \times 10^3$)

Comparing this result with that one obtained for $N = 64$, it is clear that the distributions of values around zero is larger. The reason is that the ratio between $N_{sweep}$ and $D_{bin}$ is smaller so the correlators have bigger error. To get a similar result one should consider $N_{sweep} = 15 \times 10^7$. Unfortunately, this simulation exceeds a reasonable computational time, thus it has not been performed. Nevertheless, $\Gamma_B(1) \simeq 0.019$ and the maximum value of $|\Gamma_B|$ is 0.12 which is not such big value. Before to throw away the data of the simulation with $N = 1024$, it has been decided to look at the final results. Since there are analytic solution, it is possible to say whether the errors are too small. It would mean that the contribution to $\sigma^2_{\bar{C}}$ of the autocorrelation is not negligible.

Finally, the values of $T_M$ as a function of $N$ has been plotted and fitted with a quadratic model.

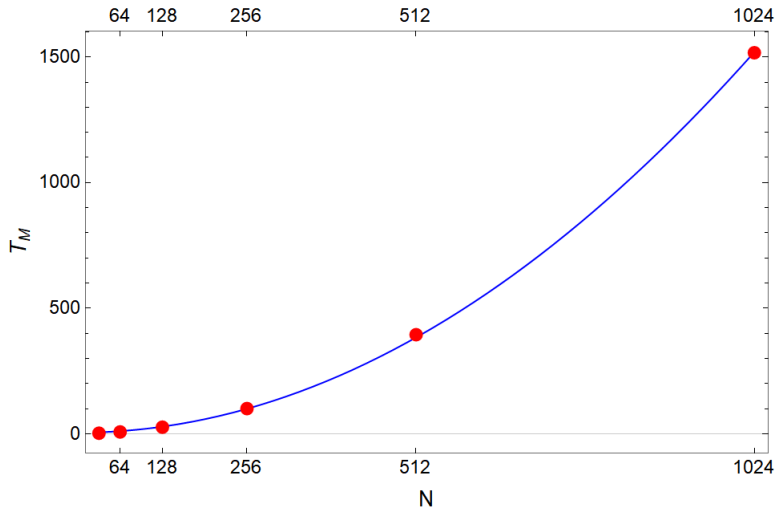



**Figure 8:** $T_M$ as a function of $N$. The fitted curve is: $a + bN^2$ with $R^2 \simeq 0.999$. ($N_{sweep} = 10^7$)

As an estimate of the goodness of the fit the Pearson's $R^2$ coefficient has been computed getting $R^2 \simeq 0.999$, so it's possible to conclude that the decorrelation $T_M$ time goes as $N^2$.

## 2.5 Correlator

From the analytic expression of the correlator $\langle x(l)x(k)\rangle$, it is possible to compute the energy gap between the ground state and the first excited level, and the matrix element $|\langle E_1|x|E_0\rangle|$ using the following relations with $t = |l-k|$:

$$\begin{aligned} \Delta E(t)a &= \operatorname{arccosh}\left(\frac{\langle x(t+1)x(0)\rangle + \langle x(t-1)x(0)\rangle}{2\langle x(t)x(0)\rangle}\right) \\ |\langle E_1|\hat{x}(t)|E_0\rangle| &= \sqrt{\frac{\langle x(t)x(0)\rangle\ e^{\frac{N}{2}\Delta E(t)a}}{2\cosh[(\frac{N}{2}-t)\Delta E(t)a]}} \end{aligned} \tag{2.5.1}$$

Then the error, which is given by $\langle x(l)^2 x(k)^2\rangle$, is independent on $|l-k|$ since the operator $\hat{x}^2$ is even and thus $\langle E_0|\hat{x}^2|E_0\rangle \neq 0$. The only way to minimize the error is to consider a big number of $N_{sweep}$, thus the estimated error is:

$$\sigma^2_{\overline{\langle x(l)x(k)\rangle}} = \frac{\sigma^2_{\langle x(l)x(k)\rangle}}{N_{sweep}} \quad \Rightarrow \quad \sigma_{\overline{\langle x(l)x(k)\rangle}} \propto \frac{1}{\sqrt{N_{sweep}}} \tag{2.5.2}$$

Given these relations, the algorithm compute first the correlator at each time $t$ and then it builds the binned correlators taking the average over packages of $D_{bin}$ consecutive data.

The following picture shows the average values of the correlator for $N = 512$. It behaves exactly as an hyperbolic cosine and the errors are tiny since $N_{sweep} = 10^7$.

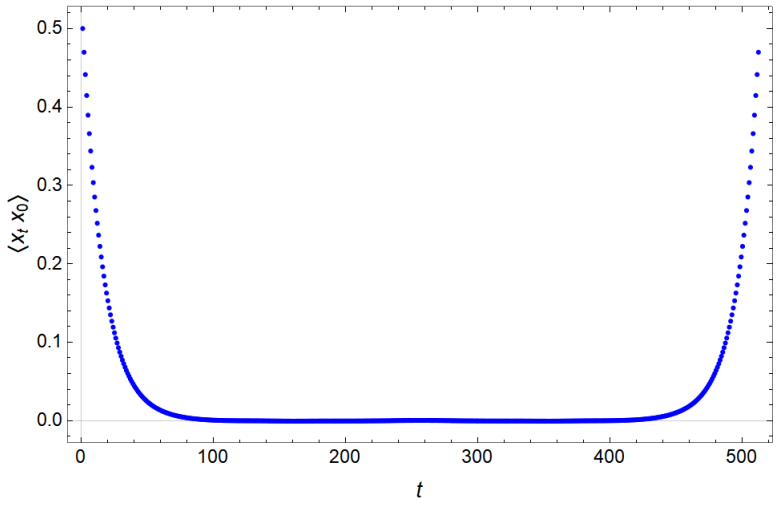



**Figure 9:** Average correlator as a function of the physical time $t$. ($N = 512$, $N_{sweep} = 5 \times 10^7$, $D_{bin} = 5 \times 10^3$)

The error bars are undetectable, given that the average error is $1.3(1) \times 10^{-4}$. Considering less sweep, to verify the behavior as an inverse square root of the errors, a computation with $N_{sweep} = 10^5$ has been performed. In this case the average error is $1.3(2) \times 10^{-3}$, so the ratio, which is about 10, corresponds to the square root of the ratio of $N_{sweep}$, thus it correctly behaves. Whether the data look like an hyperbolic cosine the algorithm works correctly, so looking at the average correlators is a good test. Given its theoretical behavior, a plot with a logarithmic scaling of the $y$ axis would be interesting. Unfortunately, in the central region the noise is dominant and this leads to deviation from the cosh behavior even with negative values that are not allowed as argument of the logarithm. Accordingly, this type of graph has been avoided, nevertheless the extreme regions follow the theoretical behavior and as a consequence these regions are mainly used to find the energy gap and the matrix element. The time required to compute the correlators $x(t)x(0)$ is $N^2$. Although there are $N$ of them, each $x(t)x(0)$ is the mean over $i$ of $x(i)x(i+t)$.
The results that will be showed in the next section have been computed with the Jackknife method, which is discussed in (A.1) and [9]. The reason is that both the energy gap and the matrix element are function of the primary observable which is the correlator, so to compute errors on the expectation values propagation of errors formula one should used. The Jackknife method avoids the use of these formula: it makes the algorithm faster and simpler.

# 3 Results

The main program, `mainEnergy.c`, computes both the energy gap and the matrix element, with their errors, starting from the computation of the binned correlator and their means for each time. In figure (10) the results for $N = 64$ are plotted.

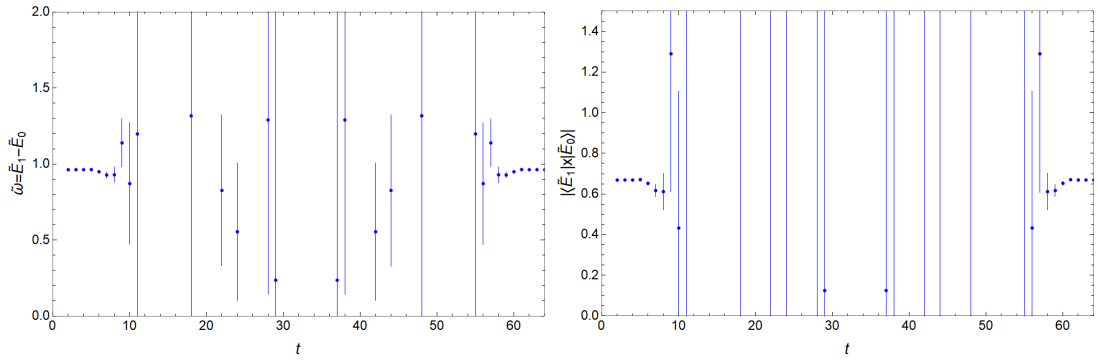



**Figure 10:** On the left the energy gap, on the right the matrix element. ($N = 64$, $N_{sweep} = 10^7$, $D_{bin} = 10^3$)

Comparing these graphs with that one of the correlator, it is possible to figure out the reason why there are such big error bars in the center: the errors are related with the average correlator which is really close to zero. Moreover, in the relations (2.5.1) there is a correlator or an hyperbolic cosine at the denominator.

When it approaches to zero, both the matrix element and the energy blow up and the computation gives "nan" so, many data miss in the center.

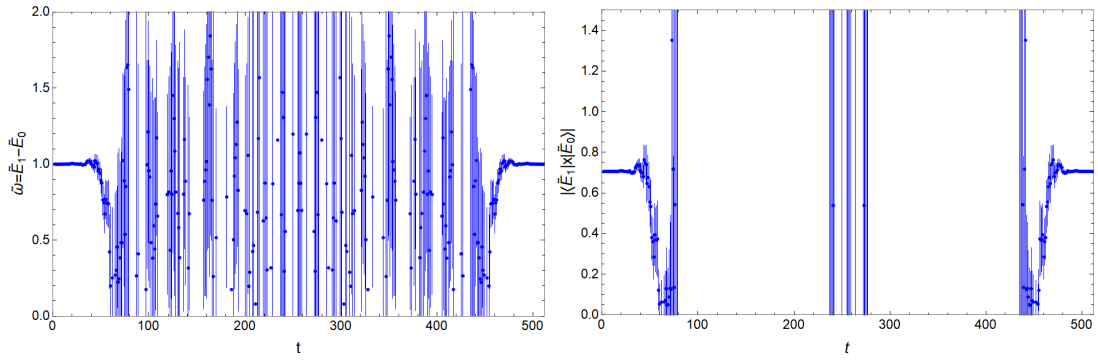



**Figure 11:** On the left the energy gap, on the right the matrix element. ($N = 512$, $N_{sweep} = 5 \times 10^7$, $D_{bin} = 5 \times 10^3$)

In figure (11) the results for $N = 512$ are plotted, clearly the same problems arise as the $N = 64$ case. However, it is possible, through a simple Mathematica code, to throw away the "nan" and to compute the weighted averages with their errors. In the table (2) theoretical results coming from expressions (1.1.6) are resumed, considering the discretization errors and the results of the simulations. The table (3) shows the parameter values chosen for the different simulations.

| $N$ | *theory* | | *simulation* | |
|---|---|---|---|---|
| | *energy* | *matrix* | *energy* | *matrix* |
| 32 | 0.881374 | 0.594604 | 0.88117(23) | 0.594618(55) |
| 64 | 0.962424 | 0.66874 | 0.96256(15) | 0.668785(50) |
| 128 | 0.989866 | 0.696471 | 0.989907(96) | 0.696491(31) |
| 256 | 0.997414 | 0.704371 | 0.99748(11) | 0.704363(38) |
| 512 | 0.99935 | 0.706418 | 0.999331(99) | 0.706393(34) |
| 1024 | 0.999837 | 0.706934 | 1.00103(28) | 0.706498(98) |

**Table 2:** Theoretical values given by the equation (1.1.6) on the left and results of the simulations on the right.

| $\frac{1}{a}$ | $N$ | $M$ | $\Omega$ | $N_{sweep}$ | $D_{bin}$ | $\lambda$ |
|---|---|---|---|---|---|---|
| 0.5 | 32 | 0.5 | 2 | $10^7$ | $10^3$ | 3.5 |
| 1 | 64 | 1 | 1 | $10^7$ | $10^3$ | 3.2 |
| 2 | 128 | 2 | 0.5 | $10^7$ | $10^3$ | 2.4 |
| 4 | 256 | 4 | 0.25 | $10^7$ | $10^3$ | 1.9 |
| 8 | 512 | 8 | 0.125 | $5 \times 10^7$ | $5 \times 10^3$ | 1.2 |
| 16 | 1024 | 16 | 0.0625 | $10^7$ | $15 \times 10^3$ | 0.91 |

**Table 3:** Parameter values used in the simulation, $ThermaT$ is always fixed to 500.

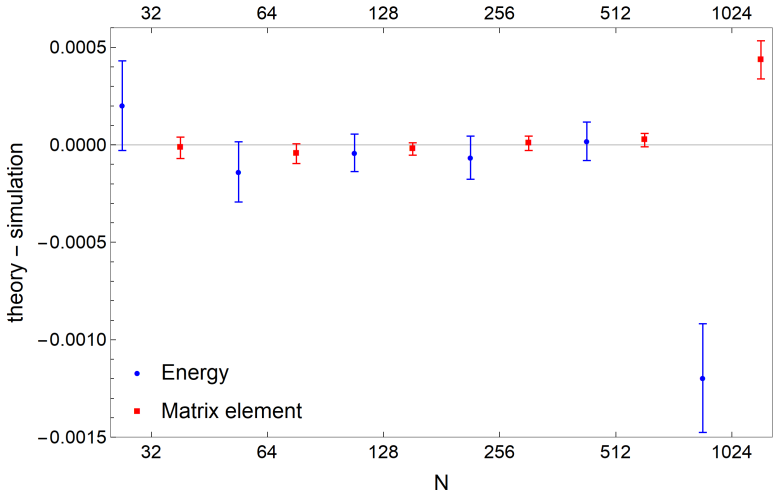



**Figure 12:** The figure shows the difference between the theoretical values and the results of the simulations. The data correspond to different values of $N$ from the smallest to the highest.

Up to $N = 512$, as figure (12) shows, the theoretical values are within the error bars of the data, while the values for $N = 1024$ are too distant from the theory, or the errors are too little. This means that the data are not sufficiently uncorrelated, and, as noticed in section (2.4.1), they have to be throw away. However, the other systems give the predicted results but this is possible because the discretization error is known. Otherwise, a fitted model can be used to find the continuum limit of both the interesting quantities.

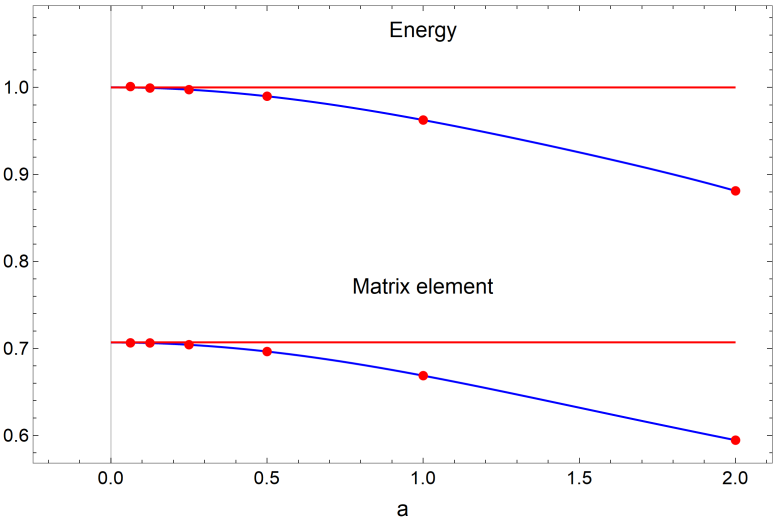



**Figure 13:** In red the theoretical values at $a = 0$, in blue the fitted curves which are a sixth degree polynomial with even power only.

The fitted models consider only systems up to $N = 512$, from which it is possible to compute the confidence interval at $a = 0$: $(0.997607, 1.00263)$ for the energy, while $(0.705438, 0.708354)$ for the matrix element, with a confidence level

of 0.99. Given that the theoretical values, $\omega = 1$ and $|\langle E_1|\hat{x}|E_0\rangle| \simeq 0.707107$, are within the confidence interval, so in conclusion the simulations give the correct estimates up to statistical errors.

# 4 Discussion and conclusion

In the first part of this paper, the harmonic oscillator system has been numerically investigated through the simplest implementation of the Monte-Carlo methods, the Metropolis algorithm. Its implementation is straightforward until the required computational time is reasonable short. Indeed, as discussed when the autocorrelation results have been presented, to consider the system with $N = 1024$, bigger $D_{bin}$ and $N_{sweep}$ are required. However, this is not an insurmountable problem because with more computational power, that simulations can be done as well. Clearly, more computational power allows to reduce the statistical errors on every systems increasing $N_{sweep}$, since it goes as $\frac{1}{\sqrt{N_{sweep}}}$. The CPU-time required to compute the interesting quantities is proportional to $N_{sweep} \times N^2$, which is due to the computation of $\langle x(l)x(k)\rangle$.

Since the Metropolis algorithm is a local algorithm, it lead to a computational time to update the chain proportional to $N$. From this point of view, a global algorithm that proposes paths considering the physics of the system could be smarter. Nevertheless, its implementation allows to recover the analytic results with a discrete accuracy: this was the experiment's goal.

The algorithm that has been presented can be used for more advanced one-dimension quantum mechanical systems. For this purpose it is just required to change the potential. In general, a different potential has not exact results, so the interpolation to the continuum limit, $a \to 0$, is crucial. However, in these cases the numerically algorithms are really useful.

# Part II
# Scalar theory

## 5 Introduction

A spin system is characterized by an infinite volume lattice at which each site has a spin degree of freedom. The Ising model considers the interaction between next-neighbor and interaction with a magnetic field. In $D = 3$ there is not an analytic solution, a simple approximation of the Ising model is called *mean field* approximation. However, it does not take into account fluctuations which turn out to be crucial in a phase transitions.

In 1937, Landau and Ginzburg [8] showed that the Ising model can be formulated in terms of a coarse-grained variable, the field $\phi$, with a power-law as Hamiltonian of the system. Moreover, without the magnetic field interaction, the energy of the system has a $\mathbb{Z}_2$ symmetry which allows to consider only even powers of the field. The Landau-Ginzburg theory considers fluctuations, thus it catches the phase transition which occurs in ferromagnetic phenomena.

The $\phi^4$ theory has two different vacuum configurations. If the coefficient of the quadratic term is positive, there is only a stable vacuum at $\phi = 0$. If it is negative, this vacuum is unstable and the true vacuum of the theory doubles at $\phi = \pm\phi_0$. These vacuum configurations constitute a model of *Spontaneous Symmetry Breaking* [2]. The stable vacuum at $\phi = 0$ is associated with the disordered phase of the spin system, namely with zero magnetization, while the double stable vacuum at $\phi = \pm\phi_0$ is associated with the ordered phase in which the magnetization can be $\pm m_0$. At the critical point, the system moves from the disordered phase to the ordered one and this breaks the $\mathbb{Z}_2$ in favor of one of the two equivalent magnetization values.

The focus of this experiment is on the phase transition of the spin system. Furthermore, the simulation of $\phi^4$ constitutes a first approach to lattice field theory: it is the simplest interacting one and for $D = 4$ it is a quantum field theory for a spin-zero particle.

As known from statistical mechanics [8], a phase transition is characterized by an infinite value of the susceptibility at the critical point and its behavior is guided by a critical exponent as well as the correlation length or the magnetization. To estimate these critical exponents, but also the location of the critical point, the magnetization and the susceptibility have been measured. However, to study numerically a field theory it is required to confine the field in a box of volume $V$, which means reducing the number of degrees of freedom from infinity to $V$. By the way the system properties change, indeed phase transitions do not occur in finite volume systems. Nevertheless, considering larger and larger volumes, it is possible to estimates the critical exponents and the critical point.

## 5.1 The model

The derivation of the discrete action starts from the standard $\phi^4$ QFT action. Precisely, moving from $D = 3 + 1$ Minkowski space-time to $D = 3$ Euclidean space, the action becomes the Lagrangian; there is no more time integration. Actually, there is no physical time dependence at all. The lattice $\Lambda$ is a set of points with periodic boundary conditions whose number of elements coincide with $V$.

The following action is the natural discretization of the continuum one:

$$S = a^D \sum_{x \in \Lambda} \Big[ \frac{1}{2} \sum_{\mu=0}^{D-1} (\partial\phi)^2 + \frac{1}{2} m_0^2 \phi^2 + \frac{1}{4!} g_0 \phi^4 \Big] \tag{5.1.1}$$

the integral over $\mathbb{R}^3$ is replaced by the sum over $\Lambda$ and the discrete derivative is defined as:

$$\partial_\mu \phi = \frac{\phi(\vec{x} + a\vec{\mu}) - \phi(\vec{x})}{a} \qquad\qquad |\vec{\mu}| = 1 \tag{5.1.2}$$

in which $a$ is the lattice spacing. It is useful to manipulate this action as follows:

$$\phi = \sqrt{2k} a^{\frac{2-D}{2}} \phi, \qquad \mathcal{L} \mapsto \mathcal{L} + \lambda, \qquad g_0 = \frac{6\lambda}{k^2} a^{D-4}, \qquad m_0^2 a^2 = \frac{1 - 2\lambda}{k} - 2D, \tag{5.1.3}$$

these redefinitions lead to an equivalent shape of the action:

$$S = \sum_{x \in \Lambda} \Big[ -2k \sum_{\mu=0}^{D-1} \phi(x)\phi(x + \mu) + \phi^2 + \lambda(\phi^2 - 1)^2 \Big] \tag{5.1.4}$$

This shape of the action has not explicit dependence on the lattice spacing. Actually, what matters is the ratio $V/a$ which define the number of point of the lattice, so the natural convention is $a = 1$. Moreover, from this action formulation, the next-neighbor interaction term is explicit and its coupling is $k$. In the limit $\lambda \to 0, \infty$ it is possible to show that it converges respectively to the gaussian model or the Ising model [17], in which the field value is constrained to be $\pm 1$.

The magnetization per unit of volume $M$, which is the derivative of the energy of the system with respect to the magnetic field $B$ is defined as follows:

$$M = \frac{\langle m \rangle}{V} \qquad m = \sum_{x \in \Lambda} \phi(x) \tag{5.1.5}$$

in which $\langle m \rangle$ is the average over all configurations of the system. The susceptibility $\chi$, which is the derivative of $M$ with respect to $B$ is defined as:

$$\chi = \frac{1}{V} (\langle m^2 \rangle) - \langle m \rangle^2) \tag{5.1.6}$$

Actually, to implement numerically the computation of $M$ and $\chi$ it is required to modify these definitions substituting $\langle m \rangle$ with $\langle |m| \rangle$. The reason is simple: to

compute $\langle m \rangle$ as a function of the coupling $k$, it needs to run many simulations at different values of $k$. In some of them, the system will be in the vacuum associated with negative values of $m$, while in some others the system will choose the positive vacuum. In the end, the result is a function that jumps between positive and negative values of $\langle m \rangle$. This has no physical meaning because in a real experiment the temperature is tuned and the system will definitely decide to stay in one of the two vacua. Fortunately, considering $\langle |m| \rangle$ instead of $\langle m \rangle$ does not affect the values of critical point or critical exponents.

## 5.2 The simulation

In the case of the harmonic oscillator, the Metropolis algorithm has been used to produce the configurations of the system. However, this algorithm is not enough efficient for $\phi^4$. The reason is that a global algorithm is required but, if the new field is randomly proposed, namely a random value for each lattice site $x$, the variation of the action is proportional to the volume and this almost always leads to a rejection of the proposal. The issue is clear: there are not reasons why $V$ random numbers should lead to an action value close to the equilibrium one. The algorithm that proposes new configurations has to know about the physics of the system.

Another possibility is to propose the new field configuration locally, changing only one lattice site field value at a time and accept it with the usual probability. However, also this strategy is inefficient as well. By computing the local variation of the action, $\delta S(x)$, the interaction term contributes such a way that the same sign field values of the next-neighbor sum are more likely compared to the opposite sign values. Moreover, this term is proportional to the dimension of the system since the number of next-neighbour is $2D$. So, the use of a local Metropolis leads to field configurations that tend to remains as they are. Furthermore, a local Metropolis leads to a computational time that grows as $V$.

The algorithm implemented is called *Hybrid Monte Carlo* [12]. It generates a global proposed field through the Hamilton equations that ideally, considering the exact solutions of them, is a valid new field configuration. Since the numerical integration of Hamilton equations is never exact, each proposed field has to be accepted with the usual probability. This means that the HMC proposes new paths considering the physics of the system and then these new paths have to be accepted, using the usual probability $e^{-\Delta S}$, to take into account that the Hamilton equation solutions are not exact.

The systems considered have $L = 4, 6, 8, 10, 12, 14, 16$; bigger volumes would require too much CPU-time. The next-neighbor coupling $k$ will vary between 0.14 and 0.25 to catch the critical point at a fixed $\lambda$. The value chosen for $\lambda$ is such that the leading order scaling corrections are minimized. Such corrections, which contribute proportionally to $L^{-\omega}$ to the interesting quantities, are finite volume effects. Indeed, the expressions that define critical exponents are valid only for infinite volumes. The argument is discussed in detail in [1, 6] from which tuns out $\lambda = 1.145$ as optimum value.

# 6 Materials and methods

## 6.1 General structure of the algorithm

Before starting the computation of $\chi$ and $M$, it is required to check that the algorithm works correctly and, as in the case of the harmonic oscillator, there are many parameters to fix. The number of Monte Carlo steps that needs to thermalize the system, *ntherm*, will be fixed in the section (6.4), while the dimension of the bin, $D_{bin}$, is discussed in the section (6.6). These two parameters have the same meaning in both simulations.

The general structure of the algorithms has been chosen as much as possible similar. The idea is to organize the files in such a way it is possible to compute all interesting quantities switching easily the parameters. For this reason, the constants in the action are defined in the `infile` as well as *nstep*, whose role will be discussed later.

In the file `lattice.h` the dimension, $D = 3$, and the size of the system $L$ are defined. Clearly, the dimension of the system is always 3, but different volumes are considered in the simulation; however, it is not so useful to define the size of the system in the `infile`: this would complicate the code and there are not many switches to do with $L$.

The two folders `devel` and `main` contain all the main programs. In `devel` those required to test the algorithm and in `main` that one to compute $M, \chi$ and another quantity, the Binder cumulant $U$.

The folder `modules` contains the functions called by the main programs in `functions.c` and a file called `hopping.c`. Such file initializes the *hop* field which for each site of the lattice gives the corresponding next-neighbors indices. This is due to the fact that, instead to be defined as a $D$ dimensional array, the field $\phi$ is a simple vector of length $V$. This could seem a useless complication: actually, it gives more generality to the algorithm, allowing to change dimension easily and to simplify the computation of the action.

Given the coordinate of a lattice site in the three dimensional space, $(n_0, n_1, n_2)$ with $0 \leq n_i \leq L-1$ and $n_i \in \mathbb{N}$, the map $(n_0, \ldots, n_{D-1}) \mapsto (i)$ is defined as follows:

$$i = \sum_{k=0}^{D-1} n_k L^k \tag{6.1.1}$$

Then, given an index $i$, the hop field returns the indices of the $2D$ next-neighbors. The figure (14) represents graphically the map in $D = 2$ and suggests the next-neighbors indices.

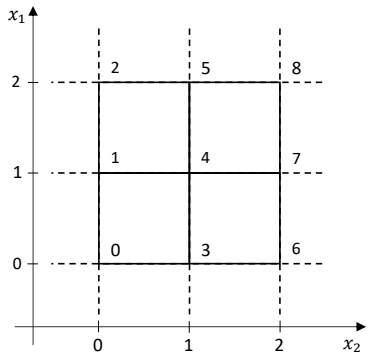



**Figure 14:** Representation of the map $(n_0, \dots, n_{D-1}) \mapsto i$.

## 6.2 Hybrid Monte Carlo

As said before, the Hybrid Monte Carlo is a global algorithm that generates the proposed field considering the physics of the system. The idea is to introduce a new field $\pi(x)$ Gaussian distributed, which is linked with $\phi$ through a fictitious time $\tau$; then it is possible to think at $\pi$ as the conjugate momentum of $\phi$. Thus, the Hamiltonian of the system can be written as follows:

$$H(\pi, \phi) = \frac{1}{2} \sum_{x \in \Lambda} \pi(x)^2 + S(\phi), \qquad \pi(x) = \frac{\delta S}{\delta \dot{\phi}} \tag{6.2.1}$$

In this Hamiltonian $S$ acts as a potential, given that the physical system is independent on the physical time. The Lagrangian does not contain a kinetic term, so it really coincides with the potential. Then it is possible to write the Hamilton equations for a little time interval $\delta\tau = \frac{\tau_0}{Nstep}$:

$$\begin{cases} \phi(\tau + \delta\tau) = \phi(\tau) + \pi(\tau)\delta\tau + O(\delta\tau^3) \\ \pi(\tau + \delta\tau) = \pi(\tau) - F(\tau)\delta\tau + O(\delta\tau^2) \end{cases} \tag{6.2.2}$$

in which $F(\tau)$ is the derivative of $S$ with respect to $\phi$, namely the force. The algorithm implemented to evolve the system for a finite time $\tau_0$, which is always fixed to 1, is called Leap Frog and is defined as follows:

$$I_{LF}(\tau_0) = \left[ I_\pi\Big(\frac{\delta\tau}{2}\Big) I_\phi(\delta\tau) I_\pi\Big(\frac{\delta\tau}{2}\Big) \right]^{N_{step}} \tag{6.2.3}$$

A natural representation of the integrator operators $I_X$ is given by matrices that act on the vector $X = (\phi, \pi)$ evolving the fields as in equation (6.2.2), just to be precise:

$$I_\phi(\delta\tau) X(0) = \begin{pmatrix} 1 & \delta\tau \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \phi(0) \\ \pi(0) \end{pmatrix} = \begin{pmatrix} \phi(\delta\tau) + \pi(\delta\tau)\delta\tau \\ \pi(0) \end{pmatrix} \tag{6.2.4}$$

In the end, the final changing is $(\phi(0), \pi(0)) \mapsto (\phi(\tau_0), \pi(\tau_0))$, so the field $\phi$ has been evolved for a time $\tau_0$ in accordance with the Hamilton equation, thus the conservation of energy is ideally preserved. Indeed, since $\delta\tau$ is a finite number,

the energy actually changes proportional to $\delta\tau^2$ at the leading order. This means that higher values of *nstep* lead to smaller violation of the energy conservation, however, it is not required to use huge values of *nstep*. To make it exact, it needs an accept/reject step acting on the proposed field $\phi(\tau_0)$, which accepts with a probability $e^{-\Delta H}$.

Finally, if the field $\phi(\tau_0)$ is not accepted, the old $\phi(0)$ can be used to compute the interesting quantities. The generation of another field $\pi(x)$ starts the next iteration and this process is repeated $N_{traj}$ times. This parameter, defined in the `infile`, is the analog of $N_{sweep}$ used in the harmonic oscillator simulation, which determines the statistical errors on the observables.

## 6.3 Gaussian momentum check

The first thing to check is that the momentum is really Gaussian distributed. The algorithm which produces random number distributed in such way is called *Box-Muller* [3]. Starting from two random numbers $x_1, x_2 \in [0, 1)$ with a flat distribution:

$$\begin{cases} y_1 = \sqrt{-2\log(1-x_1)}\cos(2\pi(1-x_2)) \\ y_2 = \sqrt{-2\log(1-x_1)}\sin(2\pi(1-x_2)) \end{cases} \tag{6.3.1}$$

then, considering 2 vectors $x_1(i), x_2(i)$ of length $V$, $\pi(i) = y_1(i)$ is Gaussian distributed.

To test this algorithm, the *Pearson* $\chi^2$ value has been computed, the $p - value = 0.57$, thus it is possible to conclude that the generation of $\pi$ works.

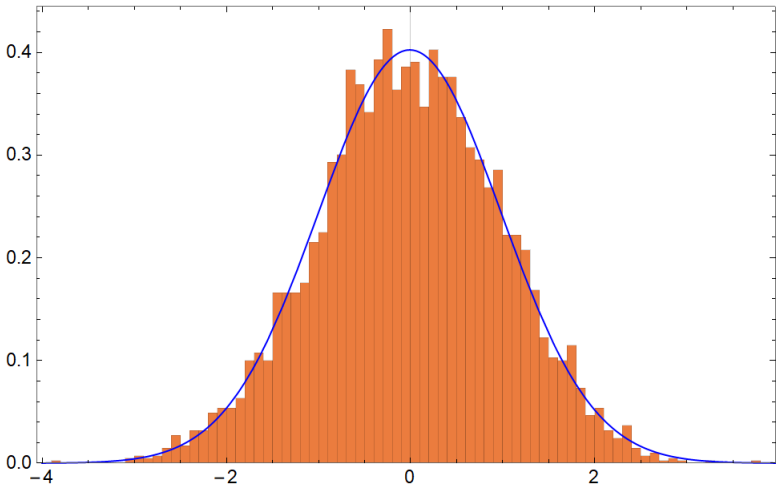



**Figure 15:** The distribution of the values of $\pi$ with $L = 16$.

## 6.4 Thermalization

The procedure to fix *ntherm* is the same for both systems. The field is initialized with two different configurations and then a certain number of HMC steps are performed. However, changing the values of the constants in the action means to consider a different system which could thermalize later. The field is initialized to 1 at every lattice site in the Cold configuration, while it is initialized with a random different value, between $-2$ and 2, at each lattice site in the Hot configuration. The Cold configuration used in this case is different from that one chosen for the harmonic oscillator: $\phi(i) = 1\,\forall i$ means that there is a magnetization, so the system is in the ordered phase and the energy is smaller than the Hot configuration. The choice $\phi(i) = 0\ \forall i$ would give a Cold configuration similar to the Hot one in terms of action, because both configurations have approximately zero magnetization. The values of the action for $L = 16$ are plotted below, the coupling has been fixed such that the system is in the disordered phase, $k = 0.165$:

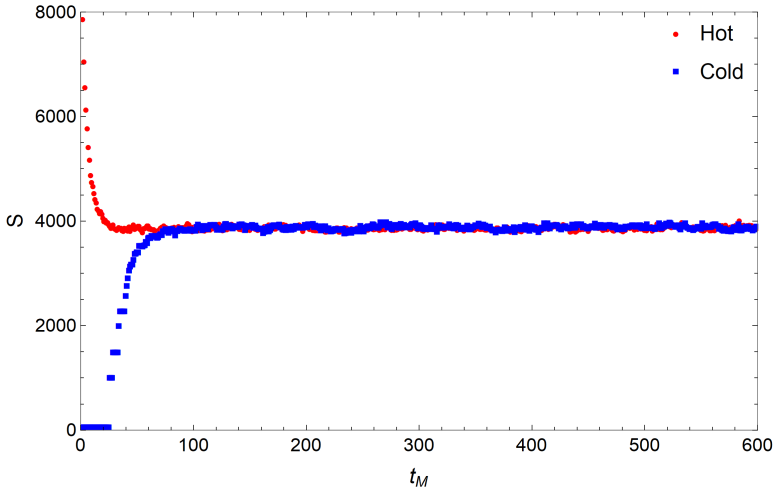



**Figure 16:** Action values for each different element of the Markov chain, $t_M$ is called Markov time. In red the data inizialized with the Hot configuration, in blue that one with the Cold one. ($L = 16$, $k = 0.165$)

From this plot it could be conclude that at $ntherm = 100$ the system has already thermalized, actually repeating the computation for $k = 0.2$, which is a value associated with the ordered phase, one notes that this is not true. At $k = 0.2$ the system thermalize around $ntherm = 400$, repeating the same computation for other values of $k$ smaller than 0.5, the value of $ntherm = 600$ has been chosen.

## 6.5 HMC tests

The choice of the Leap Frog integrator is not casual, indeed it has a reversibility property such that evolving the field forward and then backward with respect to the fictitious time $\tau$ leads exactly to the starting field. This gives a method to check the code, namely testing this symmetry with the following algorithm:

1. The field $\phi(0)$ is initialized randomly and $\pi(0)$ with a Gaussian distribution.

2. The Leap Frog evolves both fields for a time $\tau_0$ obtaining $(\phi(\tau_0), \pi(\tau_0))$.

3. The momentum is reversed $(\phi(\tau_0), \pi(\tau_0)) \to (\phi(\tau_0), -\pi(\tau_0))$.

4. The Leap Frog evolves both fields for a time $\tau_0$ obtaining $(\phi', \pi')$ .

5. The absolute value of the difference $\Delta\phi$ is computed:

$$\Delta\phi = \sum_{x\in\Lambda} |\phi'(x) - \phi(0, x)| \tag{6.5.1}$$

The result of this test is $\Delta\phi < 10^{-16}$ which is negligible, so it is possible to conclude the Leap-Frog implementation works correctly.

However, as noticed before, the violation of the energy conservation introduces a $\delta\tau$-dependence in the physical observables, so to check this behavior many tests have been performed.

The first one measures directly the violation of energy conservation. The Taylor expansion of $\Delta H$ contains only even power due to the time-reversal symmetry, so $\Delta H$ as a function of $\delta\tau^2$ has been computed without the accept/reject step.

$$\Delta H = |H(\phi(\tau_0), \pi(\tau_0)) - H(\phi(0), \pi(0))| \tag{6.5.2}$$

This quantify the energy violation since $(\phi(0), \pi(0))$ are the initial fields, while $(\phi(\tau_0), \pi(\tau_0))$ are the proposed fields. The results are plotted below:

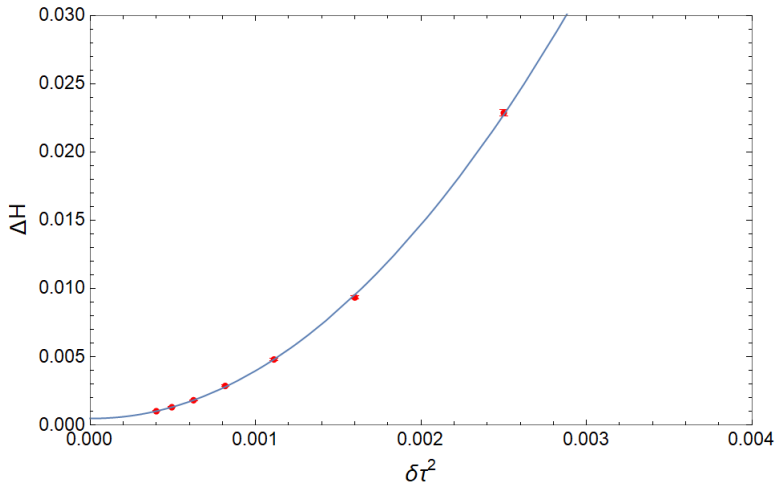



**Figure 17:** Values of $\Delta H$ as a function of $\delta\tau^2$. Fitted curve: $\Delta H = a + b\delta\tau^2 + c\delta\tau^4$. ($L = 16$, $N_{traj} = 10^6$, $k = 0.2$)

The $\chi^2$ statistic is about $10^{-6}$ with 6 degrees of freedom so the fit is really accurate and thus the expected behavior is confirmed. In order to be sure that adding an accept/reject step truly makes the HMC exact, another test is performed. Considering exact solution of the Hamilton equations and how the path integral is constructed, it is easy to show that:

$$\langle e^{-\Delta H} \rangle = \frac{1}{Z} \int D\phi D\pi \, e^{-\Delta H} \, e^{-H(\phi,\pi)} = \frac{1}{Z} \int D\phi' D\pi' \, e^{-H(\phi',\pi')} = 1 \qquad (6.5.3)$$

So it is possible to check if this identity is true at the numerical level within the statistical error.

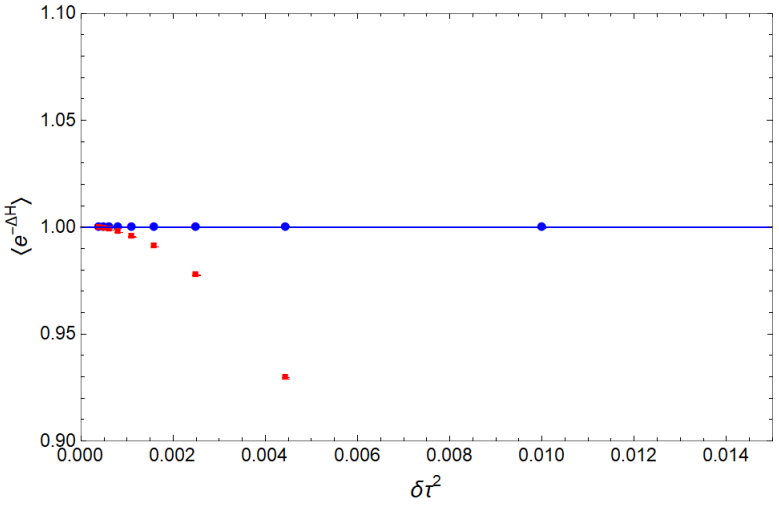



**Figure 18:** The data in red are computed without the accept/reject step while the blue ones with it. The line is a constant $y = 1$ and passes through the blue error bars. ($L = 16$, $N_{traj} = 10^6$, $k = 0.2$)

The red points depend on $\delta\tau^2$ while the blue ones do not. For instance at $\delta\tau^2 = 0.01$ with the accept/reject step, $\langle e^{-\Delta H} \rangle = 1 \pm 0.0005$. Fitting the blue data with $\alpha + \beta\delta\tau^2$ a $p - value = 0.47$ of the $\beta$ coefficient, with respect to the null hypothesis $\beta = 0$, is obtained, so there is no statistical evidence to reject the null hypothesis, thus $\langle e^{-\Delta H} \rangle$ is really independent of $\delta\tau$.

Finally, the influence of $\delta\tau^2$ on the magnetization squared has been tested at fixed $k$, measuring $\langle M^2 \rangle$ with and without the accept/reject step. The results are plotted below and to prove that there is no dependence on $\delta\tau^2$ when the computation is done with the accept/reject, it is possible to look again at the $p - value$.

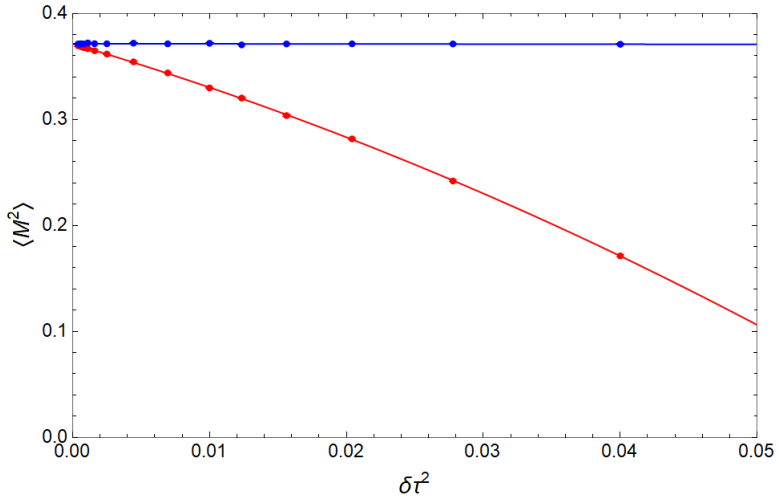



**Figure 19:** The data in red are computed without the accept/reject step, while the blue ones with it. The blue line is a constant $y = 0.3714$ and passes through the blue error bars. The red line is $a+b\delta\tau^2+c\delta\tau^4$. ($L = 8$, $N_{traj} = 10^6$, $k = 0.2$)

In this case $p - value = 0.22$, so even for $\langle M^2 \rangle$ there is no statistical evidence to reject the null hypothesis thus we conclude that $\langle M^2 \rangle$ is $\delta\tau$ independent.

## 6.6 Autocorrelation

Before to compute any physical observable data binning is necessary. In this case the auto correlation function $\Gamma$ of the squared magnetization has been computed as follows:

$$\Gamma(t_M) = \frac{1}{N_{traj}} \sum_{\alpha=1}^{N_{traj}} \frac{m_\alpha^2 m_{\alpha+t_M}^2 - \langle m^2 \rangle^2}{\langle m^4 \rangle - \langle m^2 \rangle^2} \tag{6.6.1}$$

The graph below shows the values of it for $L = 16$:

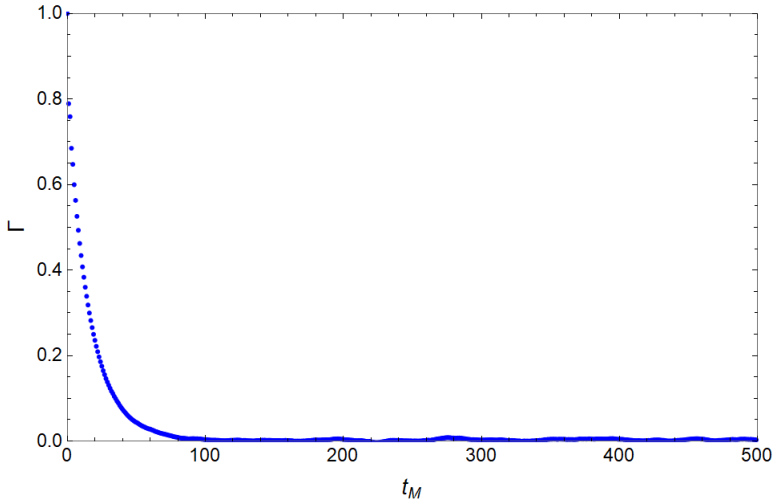



**Figure 20:** Autocorrelation $\Gamma$ as a function of the Markov time. ($L = 16$, $N_{traj} = 10^6$, $nstep = 30$, $k = 0.2$)

It is clear that $\Gamma(T_M) = 0.3$ occurs for $T_M < 100$ so $D_{bin} = 1000$ make sure that two successive bin of data are decorrelated. What makes the difference with respect to the Harmonic Oscillator simulation, as already said, is the algorithm that proposes paths. The Metropolis algorithm has a free parameter $\lambda$ which has been tuned to get the lower possible value of $T_M$, indeed the autocorrelation function of that system was dependent on $\lambda$. In this case, the HMC proposes paths knowing the physics of the systems, and for this reason it is significantly more efficient in the sense that $\Gamma$ vanishes really rapidly even for the biggest volume system. On the other hand, the auto-correlation values of the HO for N=1024 required to have $D_{bin} = 15000$.

Although $Dbin = 1000$ is enough for this simulation, dependence of $\Gamma$ with respect to the parameters of the algorithm has been investigated. Defining $T_M$ as the Markov time at which $\Gamma(T_M) = 0.3$ it is possible to compute $T_M$ as a function of the volume, $nstep$ and the coupling $k$. The figure (21) represents $T_M(L)$:

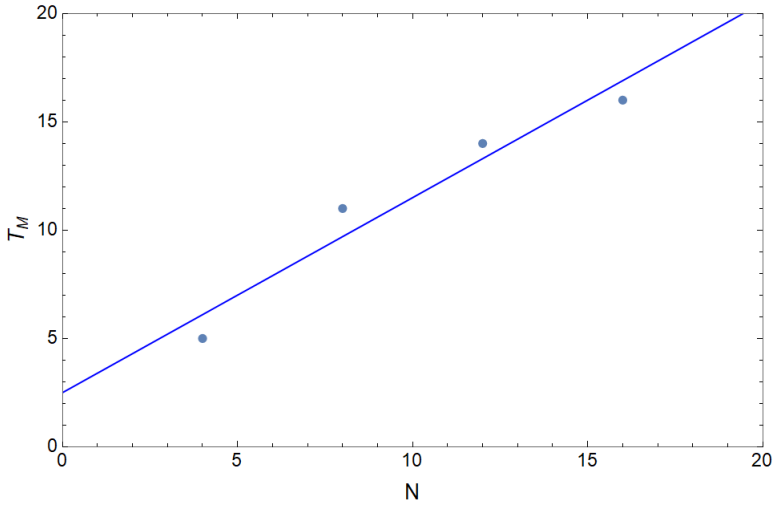



**Figure 21:** $T_M$ as a function of the lattice size $L$. ($N_{traj} = 10^6$, $nstep = 30$, $k = 0.2$)

The linear fit seems to be a good approximation instead of the quadratic growing of figure (8). Moreover, the values of $T_M$ are so small that $D_{bin} = 200$ is a good value. The dependence of $T_M$ on the number of steps used to solve the Hamilton equations is really important, as the computation time grows linearly with it.

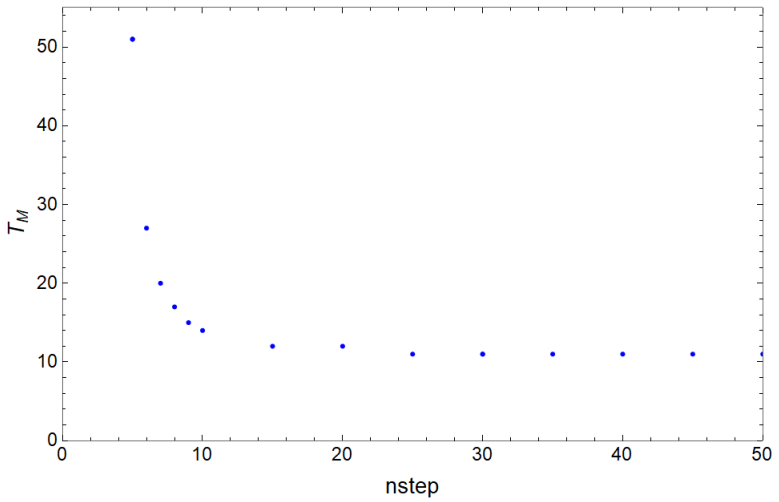



**Figure 22:** $T_M$ as function of *nstep*. ($L = 8$, $N_{traj} = 10^6$, $k = 0.2$)

From figure (22) it is clear that for small values of *nstep* $T_M$ increases exponentially, but after $nstep = 20$ the system reaches the minimum so, $nstep = 30$ has been chosen. These two plot, $T_M(L)$ and $T_M(nstep)$, represent the behaviour of the algorithm with respect to its parameters, but $T_M(k)$ has another meaning, indeed $k$ appears in the action so it is a parameter of the system, not of the algorithm. Naively, there are no reason why the autocorrelation function should depend on the value of the coupling, but actually it does. This happens because the system models a phase transition and near the critical point all lattice sites start to be correlated. This feature goes under the name of *critical slowing down* [15].

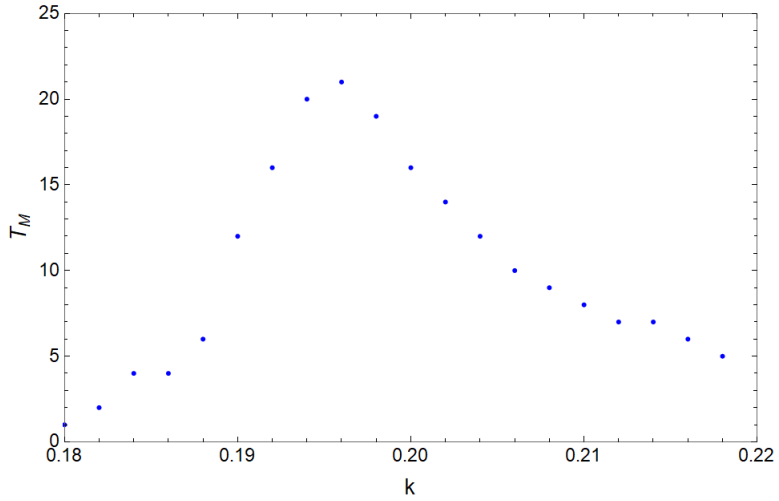



**Figure 23:** $T_M$ as function of $k$. ($L = 16$, $N_{traj} = 10^6$, $nstep = 30$)

Fortunately this effect produces only small variations on T, however considering $D_{bin} = 200$ would not be so safe. Choosing $D_{bin} = 10^3$ make sure that the the binned data are uncorrelated. Even in this case the *Jackknife method*, which is discussed in (A.1), has been used to compute $\chi$ and $U$.

# 7 Results

From statistical mechanics it is known that phase transition only occurs in systems with an infinite volume. In particular, a second order phase transition is characterized by an infinite value of the susceptibility and a magnetization that suddenly starts to grow at the critical point $k_c(\infty)$. In a system with a finite size $L$ the susceptibility will reach a maximum value at the pseudo-critical point $k_c(L)$ and the magnetization will smoothly grows around $k_c(L)$. The pseudo-critical point has a different value for each $L$ and clearly in the limit $L \to \infty$ it tends to $k_c(\infty)$.

In this section the values of $\chi$ and $M$ are presented as a function of the coupling $k$ trying to estimate the value of $k_c(L)$, $\chi(k_c(L))$ and $M(k_c(L))$. The figure below shows the measures for $\chi$ with $L = 16$.

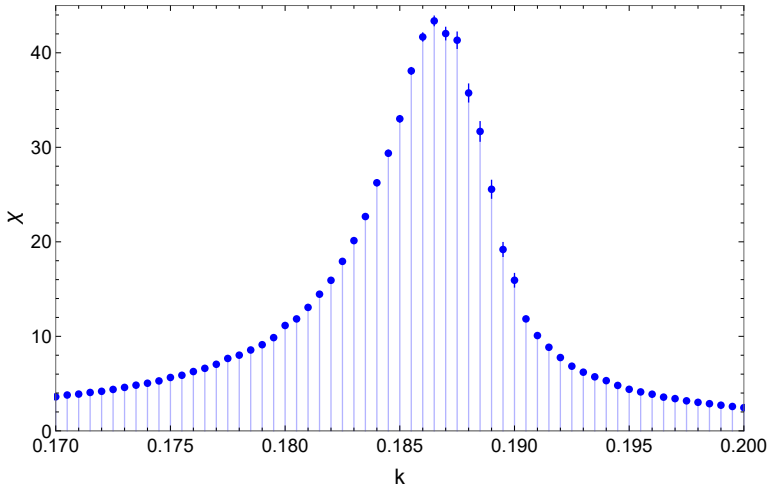



**Figure 24:** The values of the susceptibility $\chi$ as a function of the coupling $k$. The peak occurs about at $k \simeq 0.187$. ($L = 16$, $N_{traj} = 10^6$)

From these data, understanding at which values of $k$ the peak occurs is not so easy. From the plot above it is possible to notice that errors are bigger in the region close to the critical point, $0,186 < k < 0.19$. This is the same effect that produces a dependence on $k$ of $T$. A possible solution to this problem is to consider $ntraj = 10^7$ instead of $10^6$, in this way the errors decrease by a factor of about 3 since they scale with $\frac{1}{\sqrt{N_{traj}}}$. The figure below shows a zoom of the peak:

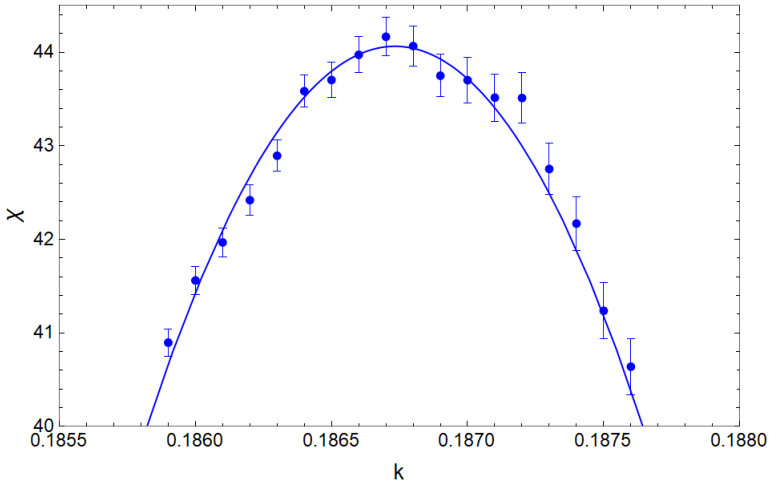



**Figure 25:** The fitted curve to data around the peak, $\chi_L = a_L k^2 + b_L k + c_L$. ($L = 16$, $N_{traj} = 10^7$)

To estimate $k_c(L)$ the *residuals bootstrap method*, which is discussed in (A.2) and [4], has been used considering the quadratic regression curve. The following table resumes the results:

| $L$ | $k_c(L)$ | $\chi(k_c(L))$ |
|---|---|---|
| 4 | 0.18861(36) | 2.832(1) |
| 6 | 0.18776(24) | 6.3561(37) |
| 8 | 0.18728(22) | 11.257(13) |
| 10 | 0.18703(17) | 17.488(27) |
| 12 | 0.18692(17) | 25.098(64) |
| 14 | 0.18679(19) | 34.03(16) |
| 16 | 0.18674(12) | 44.28(14) |

**Table 4:** Location of the pseudo-critical points and correspondent values of the susceptibility obtained with $B = 10^3$ bootstrap iterations. ($N_{traj} = 10^7$)

The estimates of $\chi(k_c(L))$ and $k_c(L)$ are required to compute the critical exponents $\nu$ and $\gamma$ that will be discussed in the next section. To compute $\beta$ the estimate of the magnetization at $k_c(L)$ needs. The following plot represents the behaviour of $M$ as a function of $k$ for $L = 4, 8, 16$:

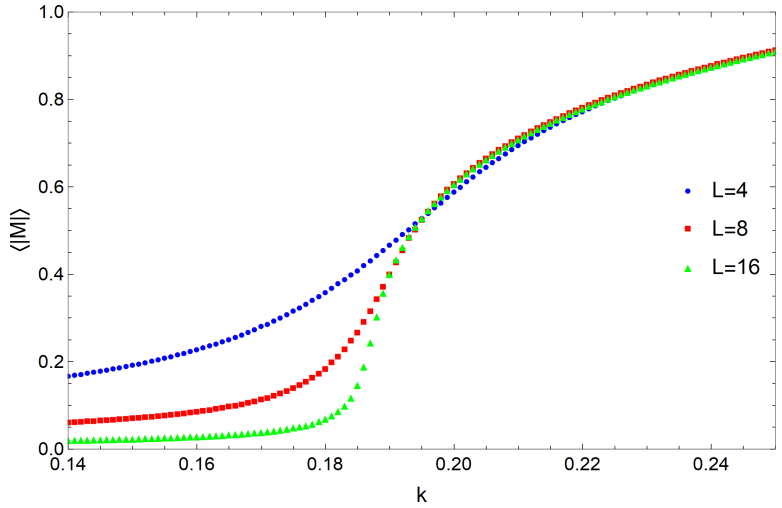



**Figure 26:** Values of the absolute value of the magnetization for $L = 4, 8, 16$. ($N_{traj} = 10^6$)

As done before, by using a linear fit instead of the quadratic one, estimates and errors of $M(k_c(L))$ has been determinate through the bootstrap method. Acting in the same way it is possible to build $B$ bootstrapped sample for $M$, and evaluate them at the corresponding $k_c^*$ one gets an estimate for $M(k_c)$. The results are presented in the table:

| $L$ | $M(k_c(L))$ |
|---|---|
| 4 | 0.4513(42) |
| 6 | 0.3742(47) |
| 8 | 0.3243(57) |
| 10 | 0.2910(59) |
| 12 | 0.2664(72) |
| 14 | 0.2454(94) |
| 16 | 0.2302(67) |

**Table 5:** Values of the absolute value of the magnetization at the pseudo-critical point with $B = 10^3$. ($N_{traj} = 10^7$)

The last interesting computed quantity is the Binder Cumulant $U$:

$$U = \frac{\langle m^4 \rangle}{\langle m^2 \rangle^2} \tag{7.0.1}$$

Such quantity is independent on $L$ at the critical point, thus it is useful to compute $k_c(\infty)$. Indeed, the crossing point of $U_L$ should correspond about to the critical point. The plot below shows the results.

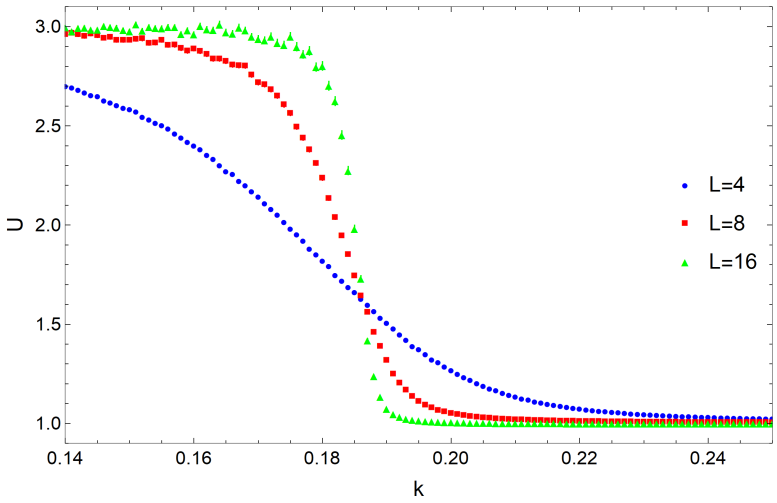



**Figure 27:** Values of the Binder cumulant for $L = 4, 8, 16$. ($N_{traj} = 10^6$)

The behaviour of $U$ is quite definite: in the disordered phase it takes the value of 3, while in the ordered phase its value is 1. To catch the crossing point of the different curves, a linear fit around the critical region is enough.

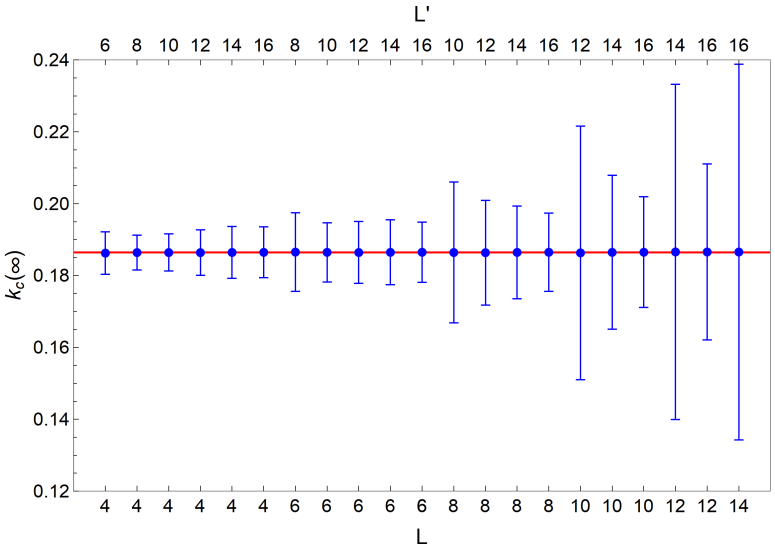



**Figure 28:** Values of $k_c(\infty)$ for each pair $L/L'$. ($N_{traj} = 10^7$)

The best estimate is $k_c(\infty) = 0.1864(45)$, so the relative error is about 2.5%. However, there is another way to compute $k_c(\infty)$ with a better accuracy that will be discussed in the next section.

## 7.1 Critical exponents

To compute the critical exponents and the critical point, it is required to focus on the behavior of the interesting quantity with respect to $L$ [14]. Considering $\gamma$ and $\nu$, which are defined for $V \to \infty$ and $t << 1$ as

$$\chi \propto |t|^{-\gamma} \qquad \xi \propto |t|^{-\nu} \qquad t = \frac{k - k_c}{k_c} \tag{7.1.1}$$

in which $\xi$ is the correlation length of an infinite volume system. In a system with a finite volume, the correlation length is cut off as it approaches $L$; the ordered phase already occurs when $\xi \simeq L$. Expressing the susceptibility in terms of $\xi$, $\chi \propto \xi^{\frac{\gamma}{\nu}}$, it turns out that even $\chi$ is cut off, it never diverges. The maximum values of $\chi$ corresponds to the maximum value of the correlation length, which means:

$$\chi(k_c(L)) \propto L^{\frac{\gamma}{\nu}} \tag{7.1.2}$$

This equation relates the measured values $\chi(k_c(L))$ with the fraction of two critical exponents, so this ratio can be obtained by comparing the maximum of $\chi$ at different volumes.

$$g = \frac{\gamma}{\nu} = \frac{\ln\left(\frac{\chi(k_c(L))}{\chi(k_c(L'))}\right)}{\ln(\frac{L}{L'})} \tag{7.1.3}$$

This is the first equation to estimate the ratio $g$. To compute these exponents, another equation needs. The precise way in which the susceptibility gets cut off

close to $k_c(L)$ is encoded in the *scaling function* $\chi_0$ [13], which depends only on the ratio between the correlation length and $L$.

$$\chi(t) = L^{\frac{\gamma}{\nu}}\chi_0(x) \qquad \text{with} \quad x = \frac{L}{\xi} \tag{7.1.4}$$

The scaling function has the following limits: for $x >> 1$ tends to a constant but for $x \to 0$ it goes as $\chi_0 \sim x^{\frac{\gamma}{\nu}}$, since the susceptibility far from $k_c$ is the same as that one for the infinite volume system. This equation contains all information about the behaviour of $\chi$ varying the systems size. However, it is a function of $\xi$ that is unknown. For this reason it is convenient to define a new scaling function, $\bar{\chi}(x) = x^{-\gamma}\chi_0(x^\nu)$, and using the relation between $\xi$ and $t$ it is possible to rewrite the equation (7.1.4) as

$$\chi(t) = L^{\frac{\gamma}{\nu}}\bar{\chi}(L^{\frac{1}{\nu}}t) \tag{7.1.5}$$

Actually, two different scaling function should be defined: one for $t > 0$ and the other for $t < 0$, since the susceptibility is not symmetric with respect to $k_c$. However, combining these two scaling function, $\bar{\chi}$ can be extended for negative values of $t$. The limit of $\bar{\chi}$ can be deduced easily from those of the old $\chi_0$, as $x \to 0$ $\bar{\chi} \to x^{-\gamma}(x^\nu)^{\frac{\gamma}{\nu}} = constant$, thus $\bar{\chi}$ is finite at the origin, which means finite at the pseudo-critical point $k_c(L)$. The main feature of this equation is that it explicitly displays all $L$-dependence for big volumes, so measuring $\bar{\chi}$ near $k_c$, for different size of the system, their values have to be equal. Actually, there are finite size corrections proportional to $L^{-\omega}$ at the leading order, due to the finite volume effects. These corrections were not taken into account.

Now considering the measured values of the susceptibility $\chi_L(k)$, and the quadratic fit, it is possible to use the equation to express the scaling function as another parabola with different coefficients. The first step is to change variable from $k$ to $t$, $\chi_L(t) = \chi_L(k_c(L)(1+t))$ and then, from $t$ to $x = tL^{\frac{1}{\nu}}$. As a result: $\bar{\chi}(x) = L^{-\frac{\gamma}{\nu}}\chi_L(L^{-\frac{1}{\nu}}x)$.

$$\begin{aligned}
&\chi_L(k) = a_L k^2 + b_L k + c_L \\
&\bar{\chi}(x) = A(a_L, k_c(L), L, \gamma, \nu)x^2 + B(b_L, k_c(L), L, \gamma, \nu)x + C(c_L, k_c(L), L, \gamma, \nu)
\end{aligned} \tag{7.1.6}$$

At the end, the result is another quadratic approximation valid only in the critical region up to errors. A possible way is to compute $\bar{\chi}$ for each $L$ considered, transforming the data not the fitted parabola, and tuning $\gamma, \nu$ and $k_c(L)$ such that near $x = 0$ these curves match as much as possible. However this is not a straightforward way to extract those values, basically because the volumes considered are not so big. To obtain $\gamma$ and $\nu$, the equality of the coefficient has been used. For example $C(L) = C(L')$ gives the same equation used to find $g$:

$$\begin{aligned}
&C(L) = L^{-\frac{\gamma}{\nu}}(a_L k_c(L)^2 + b_L k_c(L) + c_L) = L^{-\frac{\gamma}{\nu}}\chi_L(k_c(L)) \\
&B(L) = L^{-\frac{\gamma+1}{\nu}}k_c(L)(2ak_c(L) + b)
\end{aligned} \tag{7.1.7}$$

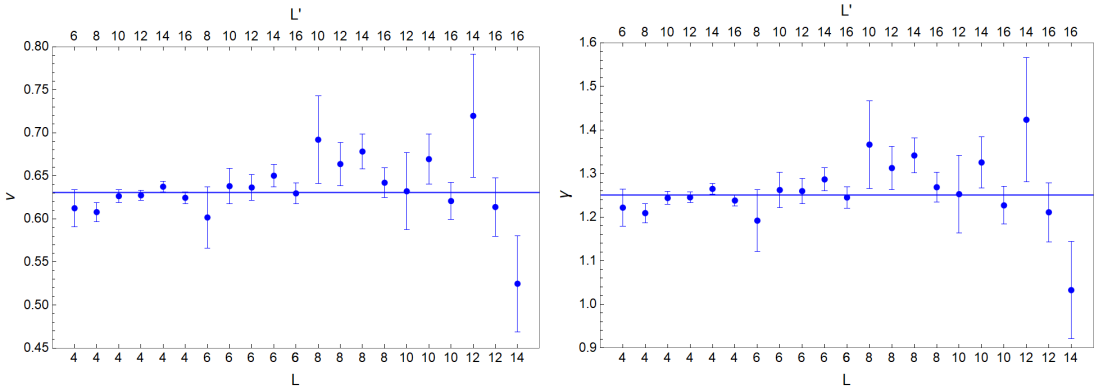
but $B(L) = 0$ since $k_c = -\frac{b}{2a}$, so this equations contains only 2 coefficients which correspond at two equation, that one already found and the following from $A(L) = A(L')$:

$$A(L) = L^{-\frac{\gamma+2}{\nu}} a_L k_c(L)^2 \quad \Rightarrow \quad f = -\frac{\gamma+2}{\nu} = \frac{\ln\left(\frac{a_{L'} k_c(L')^2}{a_L k_c(L)^2}\right)}{\ln\left(\frac{L}{L'}\right)} \tag{7.1.8}$$

Finally, inverting the relations of $\nu$ and $\gamma$ in terms of $f$ and $g$:

$$\nu = \frac{2}{g-f} = \frac{2\ln\left(\frac{L}{L'}\right)}{\ln\left(\frac{a_{L'} k_c(L')^2}{\chi(k_c(L'))}\frac{\chi(k_c(L))}{a_L k_c(L)^2}\right)} \qquad \gamma = -\nu g = \frac{\ln\left(\frac{\chi(k_c(L'))}{\chi(k_c(L))}\right)}{\ln\left(\frac{a_{L'} k_c(L')^2}{\chi(k_c(L'))}\frac{\chi(k_c(L))}{a_L k_c(L)^2}\right)} \tag{7.1.9}$$

The products in the logarithms can be written in terms of $\frac{a_L k_c(L)^2}{\chi(k_c(L))}$ which can be computed for each bootstrap sample as explained in (A.2). The estimates of these quantity can be combined together in 21 different pairs of system sizes $L/L'$. The figures below shows the results obtained for each pair of sizes: the best estimates are $\nu = 0.6308(26)$ and $\gamma = 1.2509(53)$, thus, despite considering not such large volumes, the relative error is low; it is about 0.42% .




**(a)** $\nu$ **(b)** $\gamma$

**Figure 29:** Results of the first two critical exponents. Each point corresponds to a different pair $L/L'$. ($N_{traj} = 10^7$)

The last critical exponent to compute is $\beta$. The idea is the same: the behavior of $M(k)$ can be written in terms of correlation length and it can be defined a scaling function of the magnetization $\bar{m}(x)$. Actually, there is only one equation to find, since for $t << 1$, $M \propto \xi^{\frac{\beta}{\nu}}$ and $\nu$ has been already estimate. Thus, measuring the magnetization at $k_c(L)$, at which $\xi \simeq L$, one obtains

$$\frac{M(k_c(L))}{M(k_c(L'))} = \left(\frac{L}{L'}\right)^{\frac{\beta}{\nu}} \quad \Rightarrow \quad \beta = \nu \frac{\ln\left(\frac{M(k_c(L))}{M(k_c(L'))}\right)}{\ln\left(\frac{L}{L'}\right)} \tag{7.1.10}$$

The following graph represents the results for each pair of system size:

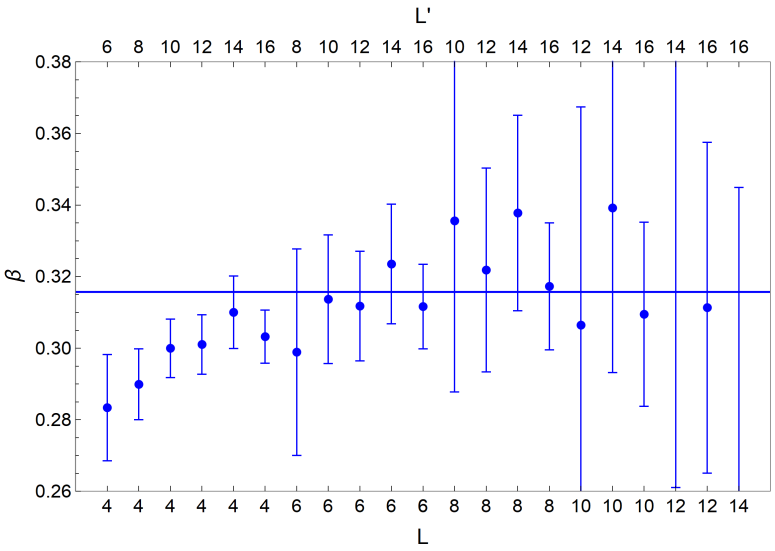



**Figure 30:** Critical exponents $\beta$. Each point corresponds to a different pair $L/L'$. The best estimate is $\beta = 0.3044(32)$. ($N_{traj} = 10^7$)

The last result presented is the location of the critical point $k_c(L)$, namely that one of the infinite volume system in which there is the real phase transition. In order to compute it, the expression of the correlation length has been considered:

$$\xi^{-\frac{1}{\nu}} \propto \frac{k - k_c(\infty)}{k_c(\infty)} \quad \overset{k \to k_c(L)}{\longrightarrow} \quad L^{-\frac{1}{\nu}} \propto \frac{k_c(L) - k_c(\infty)}{k_c(\infty)} \tag{7.1.11}$$

on the right hand side there is the definition of the critical exponent $\nu$ and, as $k$ approaches to $k_c(L)$, the correlation length approaches the size of the system. This equation is consistent with the others, indeed in the limit $L \to \infty$, $k_c(L) \to k_c(\infty)$ and both terms of the equation goes to zero. Given this relation, it is possible to obtain the value of the critical point comparing pairs of system with size $L/L'$.

$$k_c(\infty) = \frac{k_c(L) - k_c(L')\left(\frac{L}{L'}\right)^{-\frac{1}{\nu}}}{1 - \left(\frac{L}{L'}\right)^{-\frac{1}{\nu}}} \tag{7.1.12}$$

This equation leads to one estimation of $k_c(\infty)$ for each pair represented in the plot below

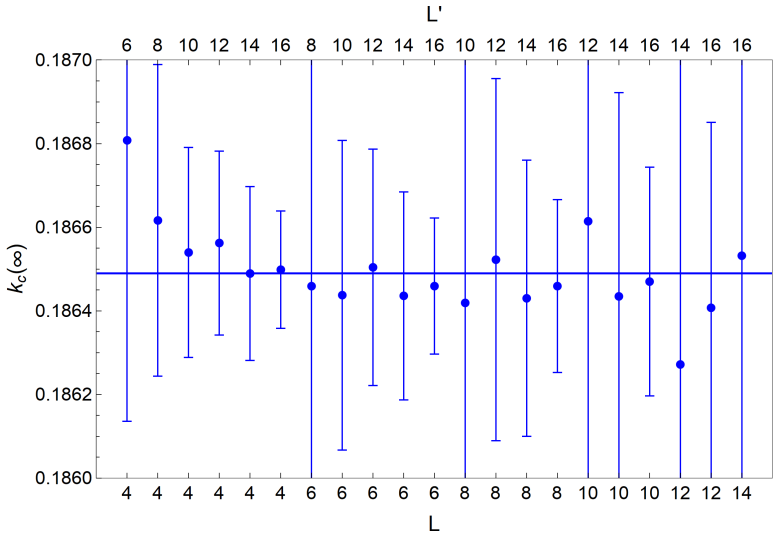



**Figure 31:** Critical point location results for each pair $L/L'$. The best estimate of the critical point is $k_c(\infty) = 0.186500(64)$. $(N_{traj} = 10^7)$

From this figure, it can be observed that, as the ratio $L/L'$ is closer to 1, the errors are bigger and bigger. This happens because the error formula has the logarithm of $L/L'$ at the denominator. This means that by using bigger volumes the errors decrease.

# 8 Discussion and conclusion

The experiment's goal is to perform the simulation of the system in order to produce the most accurate estimates possible, considering the computational power available. Clearly, considering larger volumes or a bigger number of configurations, errors decrease. The estimates that have been found are resumed in the table (6). To understand the reliability of these results, confidence intervals at $1 - \alpha = 0.99$ have been built. Given that these values come from $n = 21$ pairs of data and that the variance used is only the estimates of the true variance, the t-student distribution has been used with 20 degree of freedom in order to build the confidence intervals.

| quantity | mean | confidence interval at 0.99 |
|---|---|---|
| $k_c(\infty)$ | 0.186500(64) | (0.18632, 0.18668) |
| $\nu$ | 0.6308(26) | (0.6234, 0.6384) |
| $\gamma$ | 1.2509(53) | (1.2359, 1.2659) |
| $\beta$ | 0.3044(19) | (0.2954, 0.3134) |

**Table 6:** Confidence interval of the estimates of the critical exponents and the critical point.

Given these intervals, it is possible to compare them with the results available

in literature. The article [6], in which the optimal value of $\lambda$ has been taken, presents as best estimates: $\gamma = 1.2367(11)[9]$, $\nu = 0.6296(3)[4]$ and $k_c(\infty) = 0.1864463(4)$, the numbers in the square brackets are the systematic errors estimates. In that work the authors perform many simulation tuning $\lambda$ to obtain vanishing leading order corrections. Moreover, they estimate the exponent $\omega$ which guide these corrections $\propto L^{-\omega}$. As underlined in section (5.2), the value of $\lambda = 1.145$ has been chosen to minimize these corrections. In their work these corrections had been taken into account as systematic errors, while here they have not been computed. Nevertheless, their results are within the confidence interval in table (6) but they have smaller errors. The methods used in article [6] are more sophisticated than those used here; the volumes considered are bigger and so this justify smaller errors.

An estimate of $\beta$ can be found in [5]: $\beta = 0.3126(4)$, which is within the confidence interval in (6). However, this value sits at the extreme of the interval and other works as [7], $\beta = 0.326435(61)$, estimate a bigger $\beta$. Another possible estimate of $\beta$ can be obtained by ignoring the data coming from pairs with $L = 4$. This because corrections, that had not been taken into account, are bigger at small volumes. This estimates leads to $\beta = 0.316(6)$, which has a bigger errors because it is obtained with $n = 15$ pairs. The confidence interval in this case is (0.2989, 0.3326) which contains both result of [5] and [7].
The CPU-time of the main program goes as $N_{traj} \times V$, which is due to the averages over the lattice sites for each configuration. Further analysis could concern the computation of the leading order scaling corrections tuning $\lambda$, similarly what has been done in the article [6]. Then, more accurate estimates of the interesting quantities can be obtained.

# A Appendix

## A.1 Jackknife

This method allows to compute errors on function of the primary observable measured. Given a function $F(a)$ of the primary observable $a$, the mean value of $F$ is simply $\bar{F} = F(\bar{a})$, but the variance of it contains derivatives. Actually, computing the full expression of the variance is not wrong, but it is more expensive in terms of computational cost than using Jackknife clusters. Considering a set of $N_b = \frac{N}{D_{bin}}$ binned data $a_i$, the cluster $\{a_1^J, \ldots, a_{N_b}^J\}$ is obtained with the relation below:

$$a_i^J = \bar{a} - \frac{a_i - \bar{a}}{N_b - 1} \qquad \text{with} \quad \bar{a} = \frac{1}{N_b} \sum_{i=1}^{N_b} a_i \tag{A.1.1}$$

It is possible to show that using this set of data instead of the original one, the variance is simply:

$$\Delta F^2 = \frac{N-1}{N} \sum_{i}^{N} (F(a_i^J) - \bar{F})^2 \tag{A.1.2}$$

In the case of the harmonic oscillator, the primary observable is the correlator $\langle x(l)x(k)\rangle$ on which the energy gap and the matrix element depends. For what concerns the scalar field, the primary observable is $\langle m\rangle$: $\chi$, $U$ and $\langle M^2\rangle$ as well, contains powers of it.

## A.2 Bootstrap

This method allows to estimate average and error of quantities of interest from the empirical distribution of the sample. The alternative way to locate the peak of $\chi$, for example, is to consider the propagation of errors formula, in which variances and covariances of the parameters of the fitted curve enter. Instead use many propagation of errors formula to compute errors, it is possible to act in this way. Given $n$ quantities $A = (A_1, \ldots, A_n)$, measurable from the original data, one perform a fit over them. The second step is to compute the residuals of the fitted model, building a vector of residuals $\varepsilon_i = Y(k_i) - \chi(k_i)$. Then, the component of this vector are randomly permuted and added to the model prediction $Y(k_i)$ as follows

$$Y_b^*(k_i) = Y(k_i) + \varepsilon_b \tag{A.2.1}$$

Generating $B$ different vectors of residuals, $B$ bootstrapped samples $Y_b^*$ are obtained. Finally, fitting each $Y_b^*$ one gets $B$ quadratic fitted model. From these it is easy to compute $A_b^*$ for each $Y_b^*$ and then the mean and the error by:

$$A^* = \frac{1}{B}\sum_{b=1}^{B} A_b^* \qquad \Delta A^* = \sqrt{\frac{1}{B}\sum_{b}^{B}(A_b^* - A^*)^2} \tag{A.2.2}$$

In the scalar field simulation the interesting quantities to compute from the original data are:

$$A_L = \left(k_c(L), \chi(k_c(L)), M(k_c(L)), \frac{a_L k_c(L)^2}{\chi(k_c(L))}\right) \tag{A.2.3}$$

Thus, $k_c(L)$ and $\chi(k_c(L))$ are obtained computing the maximum of the fitted model of $\chi$. Combining these two quantity with the parameter $a_L$ gives the ratio $\frac{a_L k_c(L)^2}{\chi(k_c(L))}$. Finally, $M(k_c(L))$ is computed evaluating the linear model of $M$ at $k_c(L)$.

By this way, the propagation of errors formula have to be used only at the end, since the expressions of the interesting quantities contain $A_L$ of different lattice size.

# References


[1] H.G. Ballesteros et al. "Finite Size Scaling and "perfect" actions: the three dimensional Ising model". In: *Physics Letters B* (1998). DOI: 10.1016/s0370-2693(98)01100-9.

[2] Aron J. Beekman, Louk Rademaker, and Jasper van Wezel. "An Introduction to Spontaneous Symmetry Breaking". In: *SciPost Phys. Lect. Notes* (2019). DOI: 10.21468/SciPostPhysLectNotes.11.

[3] G. E. P. Box and Mervin E. Muller. "A Note on the Generation of Random Normal Deviates". In: *Ann. Math. Statist.* (1958). DOI: 10.1214/aoms/1177706645.

[4] Yen-Chi Chen. *Bootstrap for Regression*. 2017. URL: http://faculty.washington.edu/yenchic/17Sp_403/Lec6-bootstrap_reg.pdf.

[5] Jorge Garcia, Julio A. Gonzalo, and Manuel I. Marques. "Accurate Monte Carlo critical exponents for Ising lattices". In: *arXiv e-prints* (2002).

[6] M Hasenbusch. "A Monte Carlo study of leading order scaling corrections of phi4theory on a three-dimensional lattice". In: *Journal of Physics A: Mathematical and General* (1999). DOI: 10.1088/0305-4470/32/26/304.

[7] Martin Hasenbusch. "Finite size scaling study of lattice models in the three-dimensional Ising universality class". In: *Physical Review B* (2010). DOI: 10.1103/physrevb.82.174433.

[8] Huang Kerson. *Statistical Mechanics*. John Wiley & Sons, 1987.

[9] Avery McIntosh. *The Jackknife Estimation Method*. 2016. arXiv: 1606.00497.

[10] Nicholas Metropolis et al. "Equation of State Calculations by Fast Computing Machines". In: *jcp* (1953). DOI: 10.1063/1.1699114.

[11] M. Della Morte and L. Giusti. "Exploiting symmetries for exponential error reduction in path integral Monte Carlo". In: *Comput. Phys. Commun.* (2008). DOI: 10.1016/j.cpc.2008.10.017.

[12] Radford M. Neal. "MCMC using Hamiltonian dynamics". In: (2012). arXiv: 1206.1901.

[13] M.E.J. Newman and G.T. Barkema. *Monte Carlo Methods in Statistical Physics*. Clarendon Press, 1999, pp. 232–236.

[14] Kari Rummukainen. *Phase transitions and finite size scaling*. URL: https://www.mv.helsinki.fi/home/rummukai/simu/fss.pdf.

[15] Stefan Schaefer. "Simulations with the hybrid Monte Carlo algorithm: Implementation and data analysis". In: *Les Houches Summer School: Session 93: Modern perspectives in lattice QCD: Quantum field theory and high performance computing*. 2009.

[16] Richard Serfozo. *Basics of Applied Stochastic Processes*. Springer Berlin Heidelberg, 2009.

[17] I. Vierhaus. "Simulation of phi 4 theory in the strong coupling expansion beyond the Ising Limit". MA thesis. Humboldt-Universität zu Berlin, Mathematisch-Naturwissenschaftliche Fakultät I, 2010. DOI: http://dx.doi.org/10.18452/14138.