UNIVERSITÀ DEGLI STUDI DI MILANO-BICOCCA

Dipartimento di Fisica "Giuseppe Occhialini"

Corso di Laurea Magistrale in Fisica

Computational Physics Laboratory
Metropolis and Hybrid Monte Carlo Algorithms

Giuseppe Cavallaro

Academic Year 2019/2020

Contents

1 Introduction

1.1 Path Integral

We want to formulate the path integral for a quantum system in one dimension. In order to do that we need the quantum evolution operator, which has to be evolved over a discretized time and then the limit at continuum should be taken. We have to consider that the evolution of a quantum system can be described using both the Heisenberg and the Schroedinger pictures. The two formalisms are equivalent, if we set

$$|x\rangle_H = |x, t = 0\rangle_S \,, \qquad \hat{O}_H(0) = \hat{O}_S$$

we obtain the equality

$${}_S\langle x, t|\hat{O}_S|x, t\rangle_S = {}_H\langle x|\hat{O}_H(t)|x\rangle_H$$

where $\hat{O}_S$ and $\hat{O}_H(t)$ are operators in Schroedinger and Heisenberg formalism. The system's dynamics is defined by the retarded propagator

$$\hat{G}(t, t_0) \equiv \theta(t - t_0) e^{-i\hat{H}(t-t_0)} \tag{1}$$

The $\hat{H}$ operator has the following eigenstates and eigenvalues

$$\hat{H}E_n = E_n \, |E_n\rangle \,, \qquad u_n(x) = \langle x|E_n\rangle$$

and using the identity relation $I = \sum_n |E_n\rangle \langle E_n|$ we obtain

$$\hat{G}(x, t; x_0, t_0) = \theta(t - t_0) \sum_n e^{-iE_n(t-t_0)} u_n(x) {u^*}_n(x_0) \tag{2}$$

or equivalently after performing a Fourier Transformation

$$\tilde{\hat{G}}(t, t_0; z) = i \sum_n \frac{1}{z - E_n} u_n(x) {u_n}^*(x_0) \tag{3}$$

$\hat{G}$ contains all the information about the system's dynamics: $\hat{H}$ has real eigenvalues and so $\hat{G}$ has singularities along the real axis. Bounded states are poles for $Re(z) < 0$ and $Im(z) = 0$. $\hat{G}(x_N, T; x_0, t_0)$ is the probability amplitude that a system is measured at position x_N at time t_N , if it was at x_0 at time t_0 . The convolution property

$$\hat{G}(x_N, T; x_0, t_0) = \int dx_1 \hat{G}(x_N, T; x_1, t_1) \hat{G}(x_1, t_1; x_0, t_0)$$

states that the probability amplitude is the sum of all possible products of probability amplitudes, where, for possible, one intends over all the possible positions x_1. If we iterate the procedure N times, summing over all the possible paths, discretize time on a lattice with spacing $a = T/N$ where $T = (t - t_0)$ we can define the evolution operator as the product of evolutions over each time interval and so the retarded propagator can be written (using the Trotter Formula) as

$$\hat{G}(t, t_0) = e^{-iaN\hat{H}} = e^{-iaN(\frac{p^2}{2m} + V(\hat{x}))}$$

$$\simeq [e^{-ia\frac{p^2}{2m}} e^{-iaV(\hat{x}))}]^N =$$

$$= e^{\frac{-iaV(x)}{2}} [e^{\frac{-iaV(x)}{2}} e^{\frac{-iap^2}{2m}} e^{\frac{-iaV(x)}{2}}]^N e^{\frac{-iaV(x)}{2}}$$

We can now define the transfer operator $\hat{\tau}_a$, that evolves the system by a time interval $\delta t = a$ as

$$\hat{\tau}_a = e^{\frac{-iaV(x)}{2}} e^{\frac{-iap^2}{2}} e^{\frac{-iaV(x)}{2}} \tag{4}$$

so

$$\hat{G}(t, t_0) = e^{\frac{iaV(x)}{2}} [\hat{\tau}_a]^N e^{\frac{-iaV(x)}{2}} \tag{5}$$

The transfer operator is a unitary operator so we can define an operator $\hat{\bar{H}}$ such that

$$\hat{\tau}_a = e^{-ia\hat{\bar{H}}}, \qquad \hat{\bar{H}} = \hat{\bar{H}}^\dagger \tag{6}$$

We define $|\epsilon_n\rangle$ the transfer operator eigenstates so

$$\hat{\tau}_a |\epsilon_n\rangle = e^{-ia\epsilon_n} |\epsilon_n\rangle \tag{7}$$

and we have

$$\epsilon_n = E_n + o(a^2) \tag{8}$$

where the difference between the retarded propagator eigenvalues and the transfer operator ones is linked to the discretizazion errors.

We now perform the Wick rotation to pass from the Minkowski retarded propagator to the Euclidean one, so that the integrals involved are well defined and convergent.

$$\hat{G}_E(t, t_0) = \hat{G}(-it, -it_0), \qquad t_E = it \tag{9}$$

t_E is the euclidean time. The Euclidean propagator contains all the information on the dynamics of the system. The time interval is as before $[0, T]$, with $t_E = an$,

$n = 0, 1, ..., N$. Therefore under $a \to -ia_E$ the Euclidean transfer operator becomes

$$\hat{T}_E^a = e^{-\frac{a_E}{2}V(\hat{x})} e^{-\frac{p^2}{2m}a_E} e^{-\frac{a_E}{2}V(\hat{x})} \tag{10}$$

We can modified relations (6), (7) and (8) in this way

$$\hat{T}_a = e^{-a\hat{\tilde{H}}} = e^{-a\hat{H}} + o(a^2) \tag{11}$$

$$\hat{T}_a \left|\epsilon_n\right\rangle = e^{-a\epsilon_n} \left|\epsilon_n\right\rangle \tag{12}$$

$$\epsilon_n = E_n + o(a^2) \tag{13}$$

Since

$$\hat{T}_a^N = \hat{T}_a \int dx_1 \left|x_1\right\rangle \left\langle x_1\right| \hat{T}_a \int dx_2 \left|x_2\right\rangle \left\langle x_2\right| \hat{T}_a ... \tag{14}$$

for each time interval a we are going to evaluate

$$\left\langle x_i\right| \hat{T}_a \left|x_{i+1}\right\rangle = \left\langle x_i\right| e^{-\frac{a_E}{2}V(\hat{x})} e^{-\frac{p^2}{2m}a_E} e^{-\frac{a_E}{2}V(\hat{x})} \left|x_{i+1}\right\rangle \tag{15}$$

By definition

$$e^{\frac{a}{2}V(\hat{x})} \left|x_i\right\rangle = e^{-\frac{a}{2}V(x_i)} \left|x_i\right\rangle \tag{16}$$

The kinetic energy can be written as

$$\left\langle x_i\right| e^{-\frac{p^2}{2m}a} \left|x_{i+1}\right\rangle = \left(\frac{m}{2\pi a}\right)^{1/2} e^{-\left(\frac{x_i - x_{i+1}}{2a}\right)^2 m} \tag{17}$$

Putting together the two terms (16) and (17) we obtain that

$$\left\langle x_i\right| \hat{T}_a \left|x_{i+1}\right\rangle = \left(\frac{m}{2\pi a}\right)^{\frac{1}{2}} e^{[\frac{m}{2}\left(\frac{x_i - x_{i+1}}{a}\right)^2 + \frac{1}{2}V(x_i) + \frac{1}{2}V(x_{i+1})]} \tag{18}$$

We can finally turn back to the Euclidean propagator and write it as a path integral in the following way

$$\begin{aligned}\left\langle x_N\right| \hat{G}(T) \left|x_0\right\rangle &\simeq e^{\frac{aV(x_N)}{2}} \left\langle x_N\right| \hat{T}_a \int dx_{N-1} \left|x_{N-1}\right\rangle \left\langle x_N\right| \hat{T}_a \int dx_{N-2} ... \hat{T}_a \left|x_{N-1}\right\rangle \\ &= \int \prod_{i=1}^{N-1} dx_i e^{\frac{a}{2}V(x_N)} [\prod_{j=0}^{N-1} \left\langle x_{j+1}\right| \hat{T}_a \left|x_j\right\rangle] e^{-\frac{a}{2}V(x_0)}\end{aligned}$$

$$\left(\frac{m}{2\pi a}\right)^{N/2} \int \prod_{i=0}^{N-1} dx_i \prod_{j=0}^{N-1} e^{-a[\frac{m}{2}(\frac{x_{j+1}-x_j}{a})^2+V(x_j)]}$$

So we define an Euclidean Lagrangian density as

$$\mathcal{L}_E(x_{i+1}, x_i) = \frac{m}{2}\left(\frac{x_{i+1}-x_i}{a}\right)^2 + V(x_i)$$

and call Euclidean action

$$\mathcal{S}_E = a\sum_{i=0}^{N-1} \mathcal{L}_E(x_{i+1}, x_i)$$

then

$$G_E(x_N, T; x_0, 0) = \left(\frac{m}{2\pi a}\right)^{N/2} \int \prod_{i=1}^{N-1} dx_i e^{-\mathcal{S}_E} \tag{19}$$

This is the retarded propagator expressed as a path integral of a quantistic system in the Euclidean and we come back to Minkowski performing $a_E \to ia$ and we obtain

$$G(x_N, T; x_0, 0) = \left(\frac{m}{2\pi i a}\right)^{N/2} \int \prod_{i=1}^{N-1} dx_i e^{i\mathcal{S}} \tag{20}$$

1.2 Partition Function and Numerical Methods

The final goal is simulating a quantum field, scalar and real, using numerical and non perthurbative methods. We need to implement a scalar lattice field theory $\lambda\phi^4$. In order to do that we will use the path integral on a lattice. We define a generic potential $V(\phi)$, the action of the field ϕ is

$$\mathcal{S} = \int d^4x \mathcal{L}(\phi, \partial_\mu\phi), \qquad \mathcal{L} = \frac{1}{2}\partial_\mu\phi\partial^\mu\phi - \frac{m^2}{2}\phi^2 - V(\phi) \tag{1}$$

We define the lattice with N_i point for d dimensions

$$\Sigma_d = \{\vec{x}|\vec{x} = a\vec{n} \qquad n_i = 0, ..., N_i - 1\} \qquad i = 1, ..., d$$

Now we can define an Hamiltonian operator on this lattice:

$$\hat{H} = \hat{H}_0 + \hat{U}$$

$$\hat{H}_0 = a^d \sum_{x\in\Sigma_d} \frac{1}{2}\Pi(\vec{x}^2)$$

$$\hat{U} = a^d \sum_{x\in\Sigma_d} \left[\frac{1}{2} \sum_{j=1}^{d} \left(\frac{\phi(\vec{x}+aj) - \phi(\vec{x})}{a} \right)^2 + \frac{m^2}{2} + V(\phi(\vec{x})) \right]$$

where $\Pi(\vec{x}) = \frac{\partial_0 \mathcal{L}}{\partial_0 \phi}$.

We can define the partition function as

$$Z_a = \left(\frac{m}{2\pi a}\right)^{N/2} \int \prod_{i=0}^{N-1} dx_i e^{-\mathcal{S}_E}$$

When periodic boundary conditions are imposed, then Z_a can be easily written in operatorial form as follows

$$Z_a = \left(\frac{m}{2\pi a}\right)^{N/2} \int \prod_{i=0}^{N-1} dx_i e^{-\mathcal{S}_E} = Tr[\hat{G}_E(T, t_0)] = Tr[\hat{T}_a^N] \tag{2}$$

It follows that

$$Z_a = \sum_n e^{-aN\epsilon_n}$$

From Z_a we can extract the energy eigenvalues. In quantum mechanics, we are interested in calculating matrix elements of operators, for example those between eigenstates of the hamiltonian $\hat{H}$. It is possible to extract the matrix elements using the path integral formalism. Now we define the two-point correlation function

$$C(t_k, t_j) \frac{\int \prod_{i=0}^{N-1} dx_i [e^{-\mathcal{S}\hat{O}_1(x_k)\hat{O}_2(x_j)}]}{\int \prod_{i=0}^{N-1} dx_i [e^{-\mathcal{S}}]} = \frac{\int \prod_{i=0}^{N-1} dx_i [e^{-\mathcal{S}\hat{O}_1(x_k)\hat{O}_2(x_j)}]}{Z_a} \tag{3}$$

Monte Carlo methods enter here as we extract trajectories with probability distribution given by e^{-S}. We know that under suitable limits, (3) is equal to the T-product of operators

$$\lim_{T\to+\infty} C(t_k, t_j) = \langle \epsilon_0 | \, \hat{T}(\hat{O}_1 \hat{O}_2) \, | \epsilon_0 \rangle$$

and

$$\lim_{a\to 0} \langle \epsilon_0 | \hat{T}(\hat{O}_1 \hat{O}_2) | \epsilon_0 \rangle = \langle E_0 | \hat{T}(\hat{O}_1 \hat{O}_2) | E_0 \rangle$$

Along with the correlation functions path integrals are used to calculate expectations values of observables

$$\left\langle \hat{O} \right\rangle = \frac{1}{Z} \int [\mathcal{D}x] e^{-\mathcal{S}} \hat{O}(x) \tag{4}$$

where $[\mathcal{D}x]$ is the integration over all the trajectories. Monte Carlo methods are used to extract paths with probability $\frac{1}{Z} e^{-\mathcal{S}}$. The Central Limit Theorem assures that the best estimate of an integral

$$I = \int dx [f(x) P(x)]$$

is given by

$$\bar{I} = \frac{1}{N_{conf}} \sum_{i=0}^{N_{conf}-1} f(x_i)$$

where x_i are the paths generated with MC methods according to the distribution probability $P(x)$ (in our case $\frac{1}{Z} e^{-\mathcal{S}}$) and N_{conf} is the number of the extracted trajectories.

We will implement Markov Chain Monte Carlo methods to generate large samples of system configurations, distributed according to $\frac{1}{Z} e^{-\mathcal{S}}$. This willbe used to numerically evaluate the integrals involved in the scalar field theory.

2 Metropolis

2.1 Detailed Balance

We have to implement numerical methods in order to evaluate the correlation functions and the expectation values of observables. Such methods are part of the Monte Carlo methods, calculational techniques which make use of random numbers to construct samples distributed according to a probability density. In our work we implement a Markov Chain Monte Carlo (MCMC) algorithm. In MCMC methods we generate samples from a target distribution π by the means of a Markov chain whose stationary probability density is indeed π. A Markov chain is a sequence of random variables $(x_0, x_1, x_2, ..., x_n)$ such that x_i depends only on x_{i-1}. We can say that a sequence of events is a Markov Chain if

$$P(x_{j_0}, ..., x_{j_n}) = a_{j_o} P_{j_o j_1} ... P_{j_{n-1} j_n}$$

where a_{j_0} is an arbitrary first element of the chain, with

$$a_{j_0} \geq 0, \qquad \sum_{j_0} a_{j_0} = 1$$

and

$$P_{j_0 J_1} \geq 0, \qquad \sum_{j_i} P_{j_0 j_i} = 1$$

To each Markov chain is associated a transition matrix, P where every matrix elements satisfy

$$P_{ij} \geq 0, \qquad \forall i, j, \qquad \sum_{j} P_{ij} = 1$$

.

If we perform n steps, with $\lim_{n\to+\infty}$ the Ergodic Theorem states that if we have an ergodic Markov chain then

$$\lim_{n\to+\infty} [P]_{ij}^{(n)} = \pi_j$$

A Markov chain is said to be ergodic when each state h_j in the configuration space is persistent (the probability to extract it again after the first extraction is 1), not null (its mean recurrence time is $\mu_j < +\infty$) and aperiodic. If the Markov chain is ergodic, it will converge to its stationary distribution independently of its initial conditions, thanks to the ergodic theorem. A sufficient but not necessary condition to guarantee ergodicity is to demand that the detailed balance condition

$$\pi_i P_{ij} = \pi_j P ji \tag{1}$$

is satisfied. This condition requires that the probabilities of moving from state x_i to x_j and from x_j to x_i are the same. The desired asymptotic distribution is $\pi = \frac{1}{Z} e^{-\mathcal{S}}$. Metropolis Algorithm and hamiltonian Monte Carlo are methods that generate ergodic Markov chains.

2.2 The Algorithm

We consider a potential

$$V(\phi) = \frac{g_0}{4!}\phi^4$$

and then define ϕ, m^2 and g_0 in this way in order to semplify the action $\mathcal{S}$:

$$\phi = \frac{\sqrt{2k}}{a^{(d-1)1/2}}\phi,$$
$$m^2 = \frac{1}{a^2}\left[\frac{1-2\lambda}{k} - 2(d+1)\right],$$
$$g_0 = \frac{6\lambda}{a^{3-d}k^2}$$

so

$$\mathcal{S}(\phi) = \sum_{x\in\Lambda}\left[-2k\sum_{\mu=0}^{d}\phi(x)\phi(x+\hat{\mu}a) + \phi(x)^2 + \lambda(\phi(x)^2-1)^2\right] \tag{1}$$

where Λ is the lattice where we define x-points.

Now, the Metropolis algorithm works in this way:

- Generate any random initial path ϕ_0
- Given the current field configuration, say, ${\phi'}_n(x)$ at Markovian time n, propose a new field configuration: ${\phi'}_n(x) = \phi_n(x) + \delta_{x,x'}\Delta(r-\frac{1}{2})$ which is different from the $\phi_n(x)$ in the point x' only and ${\phi'}_n(x)$ which differs from $\phi_n(x)$ only at the space-time point x'. In this equation, $r \in [0,1)$ is a uniformly distributed random number, while $\Delta > 0$ is a parameter that controls the amount of change of the field at x'. In practical applications: $0 < \Delta < 1$.
- Compute $\Delta\mathcal{S} = \mathcal{S}(\phi') - \mathcal{S}(\phi)$ and then accept the new configuration ϕ' with probability $P_A(\phi \to \phi') = min[1, e^{-\delta\mathcal{S}}]$. If the proposed configuration is accepted, then the next configuration in the Markov chain is: $\phi_{n+1} = \phi'(x)$. Otherwise, if rejected, we shall have: $\phi_{n+1}(x) = \phi(x)_n$

We can now repeat this steps for every x' or, alternatively, we can choose x' randomly V times. Once V values of x' have been considered we call this a *sweep* over the lattice. After each sweep we can compute the observables $\mathcal{O}_i$. Fields depends on the previous one configuration so they will be correlated. In order to avoid that we can extract observables only from n measurements (cycles) or we can binning data during the analysis. We can measure the correlation in a case of of time-homogeneous Markov chains with the auto-correlation function

$$\Gamma_{\mathcal{O}}(t) = \langle\mathcal{O}_i\mathcal{O}_{i+t}\rangle - \langle\mathcal{O}\rangle^2 \tag{2}$$

In actual simulations, we can estimate the auto-correlation function as:

$$\hat{\Gamma}_{\mathcal{O}}(t) = \frac{1}{N-t}\sum_{i=1}^{N-t}((\mathcal{O}_i - \hat{\mathcal{O}})(\mathcal{O}_{i+t} - \hat{\mathcal{O}})), \qquad \langle\Gamma_{\mathcal{O}}(t)\rangle = \Gamma_{\mathcal{O}}(t)$$

Then will we analyze errors with the Jacknife Methods, which is a useful resampling technique which we used for the standard deviation estimation, when dealing with non primary variables. It allows to avoid all the derivative part calculations as would be in the standard procedure of uncertainty propagation.

2.3 Results

We want to study:

- Magnetization:

$$m = \frac{1}{V}\langle M\rangle, \qquad M = \sum_{x\in\Lambda}\phi(x), \qquad V = \prod_{\mu=0}^{d} L_\mu$$

- Absolute magnetization:

$$|m| = \frac{1}{V}\langle |M|\rangle$$

- Square magnetization

$$m^2 = \frac{1}{V}\langle M^2\rangle$$

- Susceptibility:

$$\chi = \frac{1}{V}\left[\langle M^2\rangle - \langle |M|\rangle^2\right]$$

- Binder cumulant:

$$B = \frac{\langle M^4\rangle}{\langle M^2\rangle^2}$$

where the expectation value $\langle ...\rangle$ refers to the path integral expectation value:

$$\langle\hat{\mathcal{O}}\rangle = \frac{1}{Z}tr\left[e^{-L_0\hat{H}}\hat{\mathcal{O}}\right] = \frac{1}{Z}tr\left[e^{-(L_0-x_0)\hat{\mathcal{H}}}\hat{\mathcal{O}}e^{-x_0\hat{H}}\right] \simeq \frac{1}{Z_a}\int D\phi e^{-\mathcal{S}_E(\phi)}\mathcal{O}(\phi(x))$$

.

Now we define

$$nsweep = 10^8, \qquad ntherm = 10^5, \qquad naccu = 10^4$$

where $nsweep$ is the number of proposed $\phi(x)$. $ntherm$ is the number of proposed $\phi(x)$

during the thermalization time, this time is used to reach equilibrium, after that we can calculate observables. *naccu* is the bin size and we use that in order to uncorrelate data.

We start our simulation with a random path. We can see by looking at the value of the action (see figure (1)) that, at the beginning, the new paths are not in equilibrium. After few iterations the value of the action settles down and there are just statistical fluctuations. We say that we are in thermal equilibrium. By taking measurements over many such paths, values of observables may then be calculated.

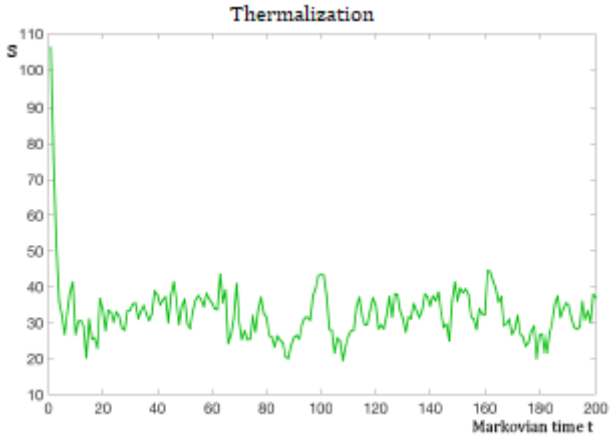

Figure 1: The action value vs the markovian time

Then we compute the observables and their errors with jackkife. We use an acceptance probability of the fields generated of $P_a \sim 90\%$

$$|m| = 0.1928162 \pm 0.0005139$$

$$m^2 = 3.5773481 \pm 0.0179827$$

$$\chi = 1.2051763 \pm 0.0053361$$

$$B = 2.5893372 \pm 0.0075389$$

Then we want to study the autocorrelation of the data, with $naccu = 10^4$ we bin data with a size of 10000 and binning data is a method to decrease the autocorrelation as we can see in the plot (see figure (2)), with the error independent by the bin size

Then we can study the autocorrelation function, so we reduce the bin size and we can vary the Δ in order to see difference in the plots. Δ is the difference of every fields generated from the previous one and is quite understandable that if we increase this value the autocorrelation decrease because every fields will be less correlated to the previous ones. As we say before, the bin size reduce the autocorrelation. (See Figure

(3),(4),(5)).

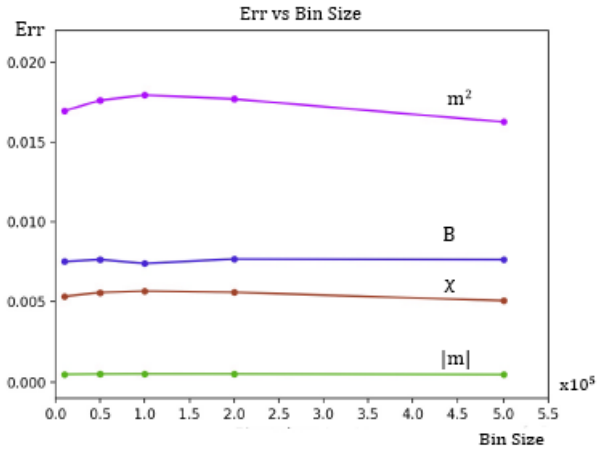

Figure 2: Errors vs the bin size

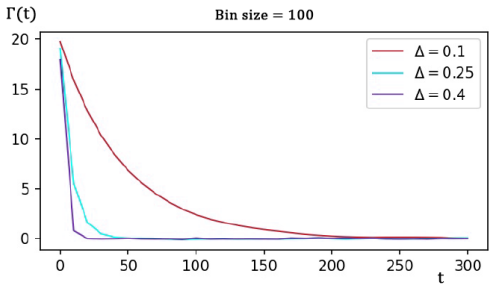

Figure 3: Autocorrelation for Bin Size 100

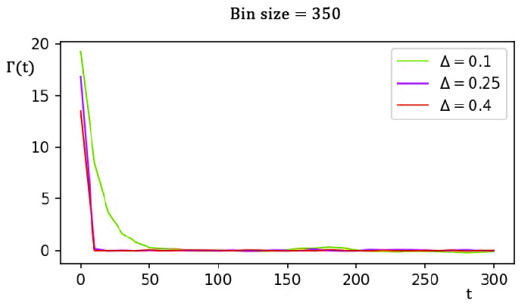

Figure 4: Autocorrelation for Bin Size 350

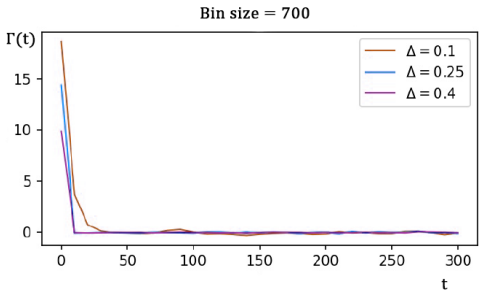

Figure 5: Autocorrelation for Bin Size 700

3 Hybrid Monte Carlo (HMC)

3.1 Theoretical Briefing

As in the previous case the calculation of the observables is linked to the path integral formalism. The expectation value of an operator $\langle\hat{\mathcal{O}}\rangle$ is given by

$$\langle\hat{\mathcal{O}}\rangle = \frac{1}{Z}\int [\mathcal{D}\phi]\,\hat{\mathcal{O}}(\phi)e-\mathcal{S}$$

Z is the system partition function as defined before. We can now calculate observables averaged over the generated sample (we need an algorithm that generate field configurations according to probability distribution $P_S(\phi) = \frac{1}{Z}e^{-\mathcal{S}}$) as

$$\bar{O} = \frac{1}{N_{conf}}\sum_{i=1}^{N_{conf}} O(\phi_i)$$

Metropolis algorithm is inefficient to study a $\lambda\phi^4$ system since local updates explore the configuration space slowly, i.e. there are long autocorrelations. Moreover, since the action is extensive, even if we simultaneously updated the field, the difference between the initial and the final action, $\Delta\mathcal{S}$, would be considerable and the acceptance rate would be low. In order to avoid that, we will use a different algorithm, which involves parallel updates of the field at all lattice sites, followed by an accept/reject step that decides for the whole configuration. The algorithm is known as Hybrid Monte Carlo (HMC).

3.2 HMC Implementation

We introduce an auxiliary field $\pi(x)$,a new parameter τ, the simulation time, and makes use of the hamiltonian dynamics equations to evolve the field ϕ over τ. The new Hamiltonian will be

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

The partition function Z now depends on π

$$Z_H = \int [\mathcal{D}\phi][\mathcal{D}\pi]e^{-H(\phi,\pi)} \tag{2}$$

and the two fields, ϕ and π, both acquire dependance on τ

$$\phi(x) \to \pi(x,\tau), \qquad \phi(x) \to \phi(x,\tau)$$

We can now write the motion equations:

$$\frac{\partial\phi(\tau,x)}{\partial\tau} = \frac{\partial H(\pi,\phi)}{\partial\pi(x)} = \pi(\tau,x)$$

$$\frac{\partial\pi(\tau,x)}{\partial\tau} = -\frac{\partial H(\pi,\phi)}{\partial\phi(x)} = -\frac{\partial S(\phi)}{\partial\phi(x)}$$

$$\frac{dH(\pi,\phi)}{d\tau} = 0$$

We can see that H does not depend on τ so the τ dependance does not modify the path integral.

We can write now the probability

$$P_H(\phi,\pi) = \frac{e^{-H(\phi,\pi)}}{Z_H} \tag{3}$$

and considering observables ϕ-dependant only

$$P_S(\phi) = \frac{e^{-S(\phi)}}{Z} \tag{4}$$

The HMC algorithm would be ideal if we were able to solve the equations of motion exactly. In general, we solve them using some numerical method, as the leapfrog integrator in our case, and integration errors occur. However it becomes exact if we add a Metropolis step at the end of each trajectory. We call it the accept/reject step. This procedure works if two requirements are satisfied: time-reversibility and conservation of the phase space volume.

Now we can see the steps for the algorithm implementation:

- We generate the auxiliary field $\pi(x)$ following a Gaussian distribution. In order to do so, we use a random number generator in ranlxd.c which extracts random variables uniformly distributed. Then we use the Box-Muller procedure to convert x1, x2, flatly distributed numbers, to y1, y2, which are distributed according to the gauss distribution

$$y_1 = \sqrt{-ln(1-x_1)}cos(2\pi(1-x_2)),$$

$$y_2 = \sqrt{-ln(1-x_1)}sin(2\pi(1-x_2))$$

- We compute the Hamiltonian $H(\pi,\phi)$ and then we make fields evolve using the Leap Frog integrator. N_{step} is the step's number of the integrator, τ_0 the total

evolution time (Markovian time) and $\delta\tau = \frac{\tau_0}{N_0}$ the single integration Leap Frog step. The leapfrog integrator is

$$I_{LPF}(N_{step}, \delta\tau) = \left[I_\pi \left(\frac{\delta\tau}{2} \right) I_\phi(\delta\tau) I_\pi \left(\frac{\delta\tau}{2} \right) \right]$$

that will be applied to $\begin{pmatrix} \pi(x) \\ \phi(x) \end{pmatrix}$. where

$$I_\pi(\delta\tau) \begin{pmatrix} \pi(x) \\ \phi(x) \end{pmatrix} = \begin{pmatrix} \pi'(x) \\ \phi(x) \end{pmatrix}, \qquad \pi'(x) - \pi(x) = -\delta\tau \frac{\partial S(\phi)}{\partial \phi(x)}$$

$$I_\phi(\delta\tau) \begin{pmatrix} \pi(x) \\ \phi(x) \end{pmatrix} = \begin{pmatrix} \pi(x) \\ \phi'(x) \end{pmatrix}, \qquad \phi'(x) - \phi(x) = \delta\tau\pi(x)$$

- We can now compute the new Hamiltonian after the Leap Frog $H(\pi', \phi')$ and $\Delta H = H(\pi', \phi') - H(\pi, \phi)$. Then, it will accepted the proposed configuration ϕ' with probability $P_A = min[1, e^{-\Delta H}]$. Otherwise $\phi' = \phi$

At the beginning of every cycle the field $\pi(x)$ is renewed and generated again so it will be independent from the previous one, while $\phi(x)$ will be part of a Markov Chain, so there is correlation.

3.3 Checks

Now we will check some properties of the algorithm.

3.3.1 Reversibility and Area Preserving

Time-reversibility means that if we perform one trajectory $((\phi, \pi) \to (\phi', \pi'))$ and then change the sign $\pi' \to -\pi$ and use the same algorithm to run the trajectory back $((\phi', -\pi') \to (\phi'', \pi''))$ and if it be reversible we obtain the starting point

$$H_{diff} = H(\phi'', \pi'') - H(\phi, \pi) = 0$$

We can verify that, obtaining results compatibile with the computer rounding errors:

$nstep$	$\lvert\phi'' - \phi\rvert$
10	$7.34 * 10^{-17}$
15	$7.92 * 10^{-17}$
20	$7.26 * 10^{-17}$
25	$1.2 * 10^{-16}$
30	$1.06 * 10^{-16}$

We can verified a phase space area-preserving property if it is valid this relation about the exponential of the change in the hamiltonian ΔH is on average

$$\langle e^{-\Delta H(\phi,\pi)} \rangle = \frac{1}{Z_H} \int [\mathcal{D}\phi][\mathcal{D}\pi] e^{-H(\phi,\pi)} e^{-\Delta H(\phi,\pi)} = \int [\mathcal{D}\phi_0][\mathcal{D}\pi_0] e^{-H(\phi_0,\pi_0)} = 1$$

We can see it from the graph, the points fluctuate around 1

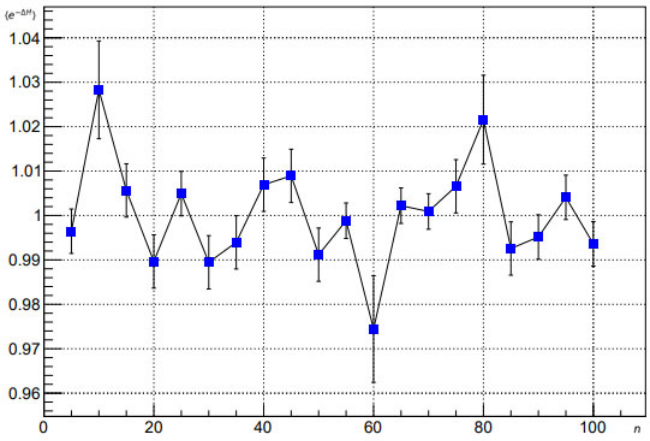

Figure 6: $e^{-\Delta H}$ vs $nstep$ with the accept-reject mechanism for L=4

So we know that the Leap Frog is implemented correctly and the phase space measure $\mathcal{D}(t_0)\phi\mathcal{D}\pi(t_0) = \mathcal{D}\phi\mathcal{D}\pi$ is conserved.

3.3.2 Hamiltonian Conservation

Hamilton Equations cannot be solved exactly so the hamiltonian is not conserved exactly, but it is in the limit $\delta\tau \to 0$. We expect that $|\Delta H|$ approaches zero linearly with $\delta\tau^2$ (see figure (6)).

3.3.3 Accept/reject

Leap Frog make the evolution of observables $\sim O(\delta\tau^2)$. We introduce an accept-reject step in order to avoid that and make observables indepent of $\delta\tau^2$. We can see that from the evolution of the $\langle m^2 \rangle$ (see figure (7)). The two case converge for low values of $\delta\tau^2$

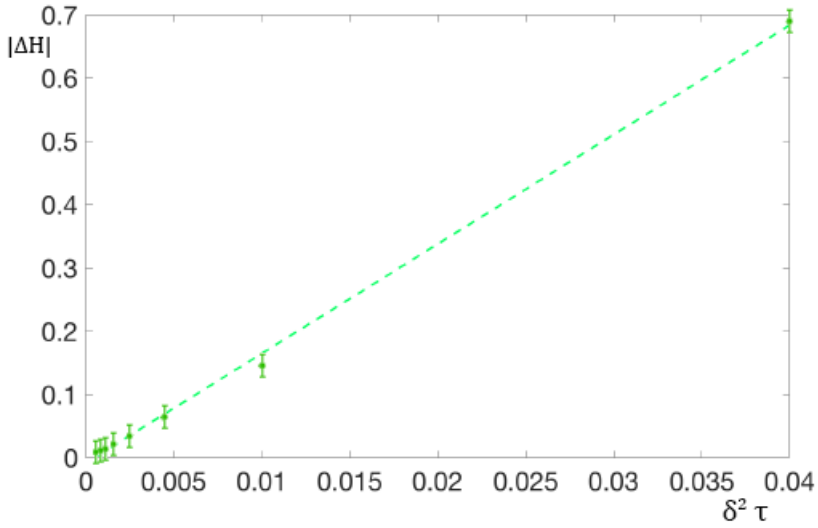

Figure 7: ΔH vs $\delta^2\tau$ with a linear fit $y = (17.3 \pm 0.3)x - (0.008 \pm 0.004)$

but, while with accept-reject is constant, the one without has big difference between the values.

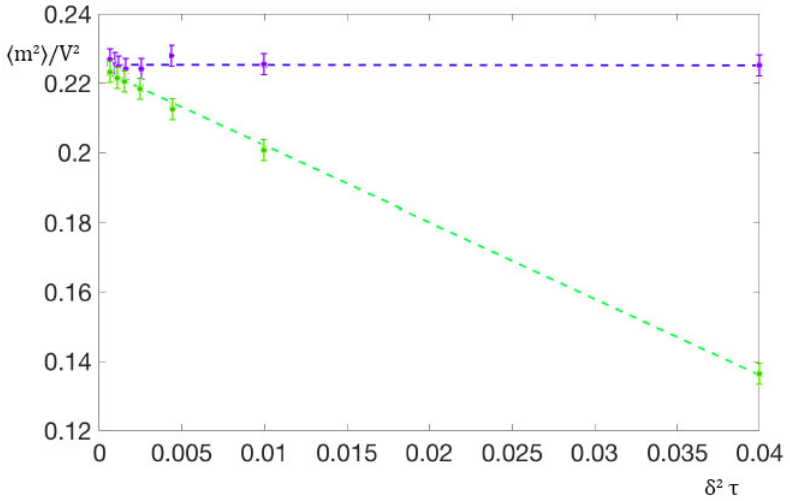

Figure 8: $\frac{\langle m^2 \rangle}{V^2}$ vs $\delta^2\tau$ with (in purple) and without (in green) the accept-reject mechanism. There are two linear fits in purple $y = (-0.006 \pm 0.004)x + (0.225 \pm 0.001)$ and in green $y = (-2.203 \pm 0.002)x + (0.224 \pm 0.001)$

3.4 Results

We perform a study of the spontaneous symmetry breaking in $\lambda\phi^4$ theory. Data are collected once thermalization is done and the binning procedure is implemented in order to uncorrelated data. We'll work with a fixed value of $\lambda = 1.145$ and will vary the k values with $k \in [0.15, 0.23]$ and the L values with $L \in [4, 16]$. For the algorithm we define these parameters:

- **ntherm**: number of thermalization cycles. Is put 5000 in the algorithm
- **ntraj**: the Metropolis nsweep, so algorithm will be repeated as a number of cycles equal to ntraj. Is put 100000 in the algorithm
- **naccu**: as in the Metropolis case is the bin size. Is put 1000 in the algorithm
- **tlenght**: is equal to 1 in the algorithm, is the τ_0 length in the Leap Frog integrator.
- **nstep**: is the steps number of the Leap Frog integrator ($\delta\tau = tlenght/nstep$)

It can be seen that as L gets higher, the phase transition (of the second order) becomes more sharp and that spontaneous symmetry breaking occurs for $0.18 < k < 0.19$ where k corresponding to the Curie temperature. We compute the same observables of the Metropolis case $|m|, m^2, B$ (figure (9),(10),(11)). In order to find the exact value of k the binder cumulant is plotted at different values of k since, according to theory, it's constant for all values of L for $T = T_C$. From the plot 11 we find $k = 0.186$.

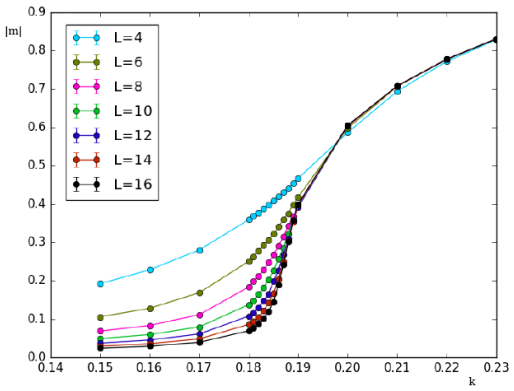

Figure 9: $\frac{\langle |m| \rangle}{V}$ vs k for different L

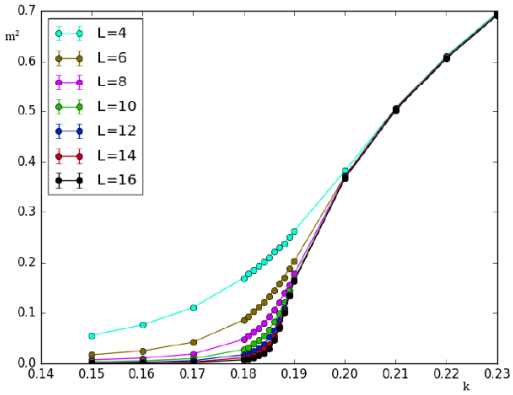

Figure 10: $\frac{\langle m^2 \rangle}{V}$ vs k for different L

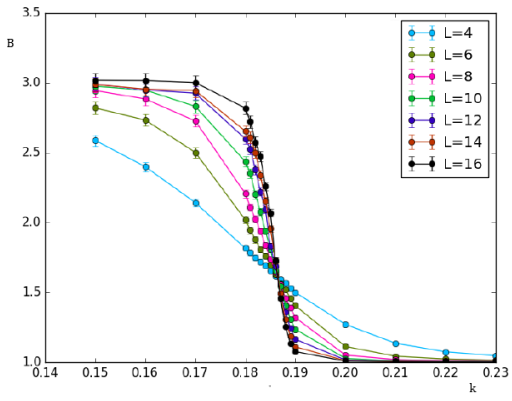

Figure 11: B vs k for different L

The χ plot (figure (11)) in the infinite volume limit has a divergence for $k = k_C$ (where call k_C in the Curie Temperature) and that in the infinite volume it should diverge like a Dirac delta.

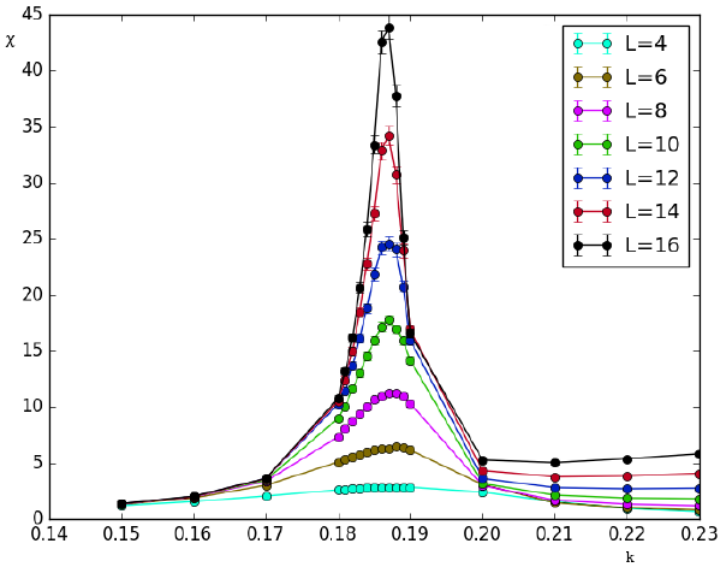

Figure 12: χ vs k for different L

The shape of the function goes like a Dirac delta with increasing maximum and decreasing width for growing L.

We can do a exponential fit (see figure (12)) for the maximum of χ in order to find a critical exponent $\frac{\gamma}{\nu}$. The k taken in order to compute the $max\chi$ is $k = 0.186$ for all the L considered.

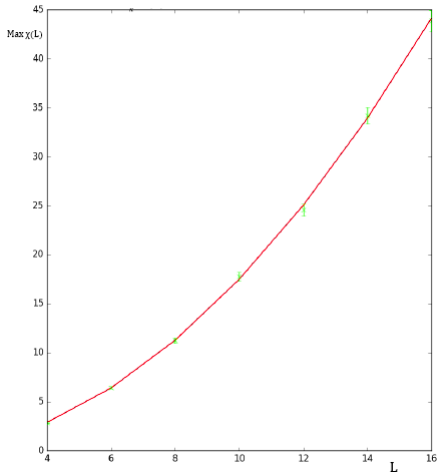

Figure 13: Exponential Fit for the plot $MAX\chi$ vs L

From the exponential fit $y = ax^b$ we obtain

- $a = 0.18743 \pm 0.0045428$
- $b = 1.96977 \pm 0.010748$
- $\frac{\chi^2}{\nu} = 0.282$

So $\frac{\gamma}{\nu} = 1.97 \pm 0.01$ and the parameter of the reduced chi-square measure the goodness of the fit with $\simeq 0.3$. This is in accord to the theory where the expected pattern is $|T - T_C|^{-\frac{\gamma}{\nu}}$ and we know that $\gamma \sim 1.24$ and $\nu \simeq 0.63$.