Monte Carlo methods for lattice QCD

Alessandro Cotellucci

July 23, 2019

Abstract

Monte Carlo methods are very powerful methods based on stochastical process. In this relation will be introduced the basic ideas behind Monte Carlo and its use to evaluate integrals and mean value. The second chapter will be focussed on the application for quantum mechanics which is an intrinsic probabilistic theory, in particular the use for lattice QCD for nuclear physics.

Contents

1 Monte Carlo methods **1**
1.1 Idea behind Monte Carlo methods 1
1.2 Markov chain 2
1.3 Metropolis algorithm 3

2 Application for lattice QCD **4**
2.1 Path integrals 4
2.1.1 Metropolis algorithm for path integrals 6
2.1.2 Harmonic Oscillator program 7
2.2 Lattice QCD 10
2.2.1 Wilson loop 12
2.2.2 Static Quark Potential 14

Chapter 1

Monte Carlo methods

The Monte Carlo method is a numerical method based on the generation of random numbers to obtain numerical results, for example they can be used to quadrature problems or to explore the Hilbert space of a system.

1.1 Idea behind Monte Carlo methods

Let's consider the basic application of Monte Carlo method: the evaluation of integrals. If we consider the figure 1.1 we can see the random sampling of point in the box around a circle, the basic idea is that according to the classical interpretation of probability theory the probability for a point to fall into the circle is given by the area of the circle on the area of the box:

$$\frac{A_{Circle}}{A_{Box}} = \frac{N_{Circle}}{N_{Box}} = P(circle). \tag{1.1}$$

By the frequentist interpretation of probability theory we get that the probability, of a point, to fall into the circle is given by the number of points in the circle over the total number of points (in the high total number limit). A sophisticated method can be obtained considering the mean value of a quantity $h(x)$ on a general probability distribution function $f(x)$:

$$\langle h \rangle = \int h(x) f(x) dx \sim \sum h(x_{rand}) \tag{1.2}$$

in this case generating the x variables considering the probability distribution function $f(x)$ we can approximate the integral to the sum of $h(x)$ computed in the random distributed x.
This method can be generalized to any kind of integral by choosing a suitable $f(x)$:

$$\int h(x) dx = \int f(x) \frac{h(x)}{f(x)} dx. \tag{1.3}$$

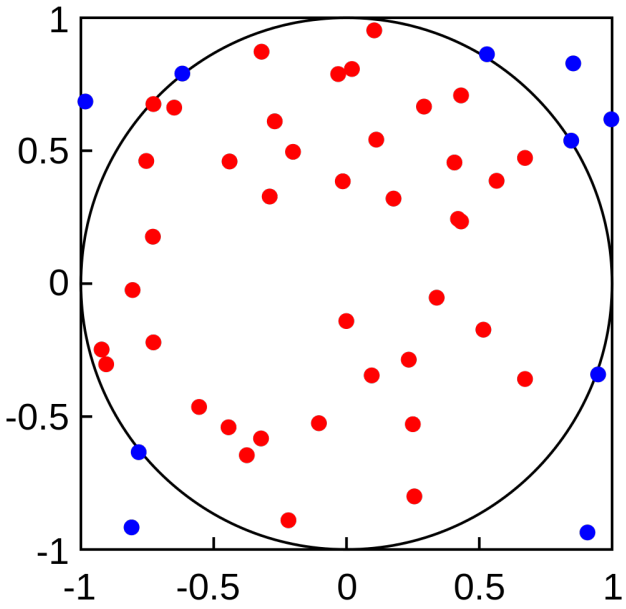

Figure 1.1: Evaluation of a circle area using Monte Carlo method (https://en.wikipedia.org/wiki/Monte_Carlo_integration/media/File:MonteCarloIntegrationCircle.svg)

This method is very comfortable for quantum mechanical calculations because the mean value of an observable (for example the Hamiltonia H) on the state $|\psi\rangle$ will be given by:

$$\langle\psi|H|\psi\rangle = \frac{\int \overline{\psi}(x)H(x)\psi(x)dx}{\int \overline{\psi}(x)\psi(x)dx} = \int \frac{|\psi(x)|^2}{\int |\psi(x)|^2 dx}\frac{H(x)\psi(x)}{\psi(x)}dx. \tag{1.4}$$

This can be applied to all the possible observable.

1.2 Markov chain

Let's consider a general set of discrete states $s_1, s_2, ..., s_N$ we would like to have a rule to go from a state s_i to a state s_j, to do this we introduce the transition matrix T_{ji} which represents the transition probability going from i to j. The transitions matrix are stochastic because they fulfilling this conditions:

- they are bounded $0 \leq T_{ji} \leq 1$
- $\sum_j T_{ji} = 1$

Some transition probabilities may be 0, and the transition matrix may also contain non zero diagonal terms which represent the probability of remaining in the initial state. If all term of a column are zero but a term we call this stopping state.
There are two kind of Markov chain:

- Endles chain: we want to explore the whole space of state so we don't want stopping states
- No disconnected chain: We want to be able to reach every point from each point (in a network frame work we want a connected graph of the chain)

All this conditions can be generalized to set of continuum states by considering the transition matrix as a two dimensional function still stochastic:

- bounded $0 \leq T(x, x') \leq 1$
- $\int T(x, x')dx' = 1$

1.3 Metropolis algorithm

The Metropolis algorithm aim is to find a stochastic matrix T_{ji} defining a random walk which gives rise to a given probability density function $P_i(x)$.
Let's define a new stochastic and symmetric matrix S_{ji} such that:

$$\begin{aligned} T_{ji} &= S_{ji} && if\ P_j \geq P_i,\ i \neq j \\ T_{ji} &= S_{ji}P_j/P_i && if\ P_j < P_i,\ i \neq j \\ T_{ii} &= S_{ii} + \sum_k S_{ki}(1 - P_k/P_i) && for\ k \in \{P_k < P_i\} \end{aligned} \tag{1.5}$$

Let's assume to start from a state s_i, using the auxiliary transition matrix we select a candidate s_j for the next state. If the probability P_j is grater than P_i the new state is accepted. If the probability P_j is smaller than P_i the new state is accepted with probability P_j/P_i. In the rejected case we stay in the s_i state.
Metropolis algorithm is of asymptotic nature so it has meaning only after many samples. Because it is prone to strong correlations between successive steps. The main reason is that generally the probability distribution function is peaked around some maximal value so successive random numbers are quite close and to sample the full domain where P_i is defined we have to carry out many moves (for a physical application we do many moves before taking out evaluations). It is normally assumed that a rate of acceptance of 50% is sufficient.

Chapter 2

Application for lattice QCD

In this chapter we will give a short description of Metropolis algorithm for solving path integrals and we will apply this to gluonic QCD computing first the Wilson Loops and then using this code to compute the so called 'Static Quark potential', the potential between a quark and an anti-quark in the non relativistic limit.

2.1 Path integrals

According to the path integrals formulation of quantum mechanics the quantization of a unidimensional system is given by:

$$\langle x_f|e^{-H(t_f-t_i)}|x_i\rangle = \int Dx(t)e^{-S[x]}. \tag{2.1}$$

Where $\int Dx(t)$ denotes the integral over all the possible paths with the condition: $x(t_i) = x_i$, $x(t_f) = x_f$. With the classical action for the system:

$$S[x] = \int_{t_i}^{t_f} dt \left[\frac{m\dot{x}(t)^2}{2} + V(x(t))\right]. \tag{2.2}$$

We are using Euclidian path integrals applying the Wick's rotation.
We can easily compute the path integral discretizing the time interval introducing:

$$a \equiv \frac{t_f - t_i}{N}, \tag{2.3}$$

we get the discretized time steps:

$$t_j = t_i + ja \qquad for\ j = 0, 1, 2, ..., N \tag{2.4}$$

and the discretized set of $x(t)$:

$$x = x(t_0), x(t_1), ..., x(t_N). \tag{2.5}$$

Setting periodic boundary's condition $x_0 = x_N = x$ the integration becomes:

$$\int Dx(t) \sim A \int_{-\infty}^{\infty} dx_1 dx_2 ... dx_{N-1} \tag{2.6}$$

with:

$$A \equiv \left(\frac{m}{2\pi a}\right)^{N/2}. \tag{2.7}$$

Using the discretizing action:

$$S_{lat}[x] \equiv \sum_{j=0}^{N-1} \left[\frac{m}{2a}(x_{j+1} - x_j)^2 + aV(x_j)\right], \tag{2.8}$$

we get:

$$\langle x_f | e^{-H(t_f - t_i)} | x_i \rangle = \int_{-\infty}^{\infty} dx_1 ... dx_{N-1} e^{-S_{lat}[x]}. \tag{2.9}$$

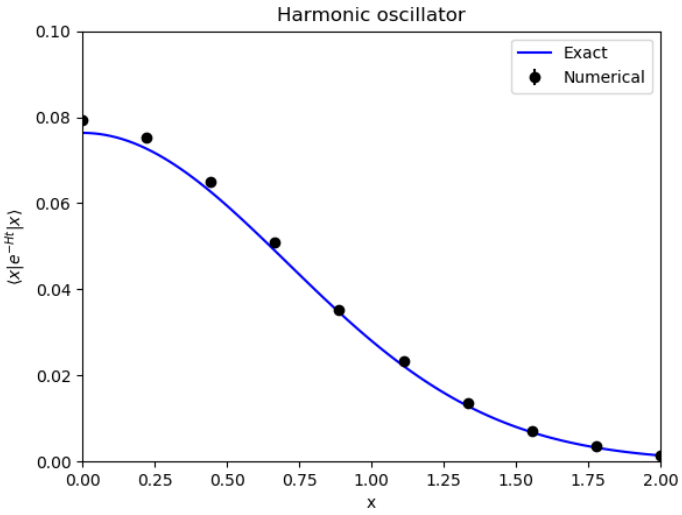

Figure 2.1: Euclideal unitary mass harmonic-oscillator propagator at large time ($T = 4$) with $a = 1/2$. The path integral is approximated using an 8 dimensional integral computed using the packet *vegas* integrating from -5 to 5(*PathIntegral.py*).

2.1.1 Metropolis algorithm for path integrals

Let's consider an observable $\Gamma[x]$, in this case the mean value of the observable on the path integral is given by:

$$\langle \Gamma[x] \rangle = \frac{\int Dx(t)\Gamma[x]e^{-S[x]}}{\int Dx(t)e^{-S[x]}}. \tag{2.10}$$

We can notice that the mean value is a weighted average over the path with weight $exp(-S[x])$, so we can generate a number (N_{cf}) of random configurations:

$$x = x_0^{(\alpha)}, ..., x(t_{N-1}^{(\alpha)}) \qquad \alpha = 1, 2, ..., N_{cf}, \tag{2.11}$$

in such way the probability distribution is given by:

$$P[x^{(\alpha)}] \propto e^{-S[x^{(\alpha)}]}. \tag{2.12}$$

Computing the mean value of $\Gamma[x]$ becomes:

$$\langle \Gamma[x] \rangle \sim \frac{1}{N_{cf}} \sum_{\alpha=1}^{N_{cf}} \Gamma[x^{(\alpha)}], \tag{2.13}$$

with the standard deviation:

$$\sigma_\Gamma^2 = \frac{\langle \Gamma^2 \rangle - \langle \Gamma \rangle^2}{N_{cf}}. \tag{2.14}$$

Considering this we can now define the steps of the Metropolis algorithm for the evaluation of the mean values:

- generate a random number s, with uniform probability between $-\epsilon$ and ϵ;
- replace x_j with $x_j + s$ and compute the change in action (we consider only the terms of the action affected by the x_j);
- if $\Delta S < 0$ keep the new value for x_j and proceed to the next site;
- if $\Delta S > 0$ generate a random number b uniformly distributed between 0 and 1, restore the old value if $exp(-\Delta S) < b$.

Using the Metropolis algorithm we have the problem already mentioned of correlation between successive steps, to avoid this we first termalize the lattice applying $5 * N_{cor}$ times the upload before beginning the simulation and then before every computation we apply N_{cor} update.

2.1.2 Harmonic Oscillator program

For a first application of this methods we considered an Harmonic oscillator of unitary mass with discretized action:

$$S_{lat}[x] \equiv \sum_{j=0}^{N-1} \left[\frac{1}{2a}(x_{j+1} - x_j)^2 + a\frac{x^2}{2} \right], \tag{2.15}$$

and with discretization $a = 1/2$, $N = 8$ and $\epsilon = 1.4$. As function computed we considered the propagator:

$$G(t) \equiv \langle x(t_2)x(t_1) \rangle, \tag{2.16}$$

because in the large t limit the only leading therm gives us:

$$log(G(t)/G(t+a)) \to (E_1 - E_0)a = \Delta E a \tag{2.17}$$

in this way we have the energy difference between the ground state and the first exited state. In the lattice framework computing $G(t)$ means compute G_n which is:

$$G_n = \frac{1}{N} \sum_j \langle x_{(j+n) mod N} x_j \rangle. \tag{2.18}$$

The result of this simulation is given in figure 2.2. To get a bigger sample of simulation we used the bootstrap method which consist in a casual rearranging of the ensemble of $G^{(\alpha)}$ to get more fictitious simulations.
A way to improve the simulation is to improve the order of approximation of the derivative in the action, considering the definition (2.2) of the action we can by integrating by part obtain:

$$S[x] = \int dt \left[-\frac{1}{2} m x(t) \ddot{x}(t) + V(x(t)) \right]. \tag{2.19}$$

by considering the central approximation of the second order derivative to the second order, we get:

$$S_{imp}[x] \equiv \sum_{j=0}^{N-1} a \left[-\frac{1}{2} x_j (\Delta^{(2)} - a^2 (\Delta^{(2)})^2 / 12) x_j + V(x_j) \right], \tag{2.20}$$

with the definition:

$$\Delta^{(2)} x_j \equiv \frac{x_{j+1} - 2x_j + x_{j-1}}{a^2}. \tag{2.21}$$

This improvement of the action gives an additional oscillation mode called "numerical ghost" $\omega^2 \approx (2.6/a)^2$. This mode appear at high energies but it affects also our calculation at low energy because the numerical result ignored the possibility that spurius

states might be induced by the numerical analysis. This states have negative norm and this is why in the figure 2.3 the numerical solution is below the exact result for low t.

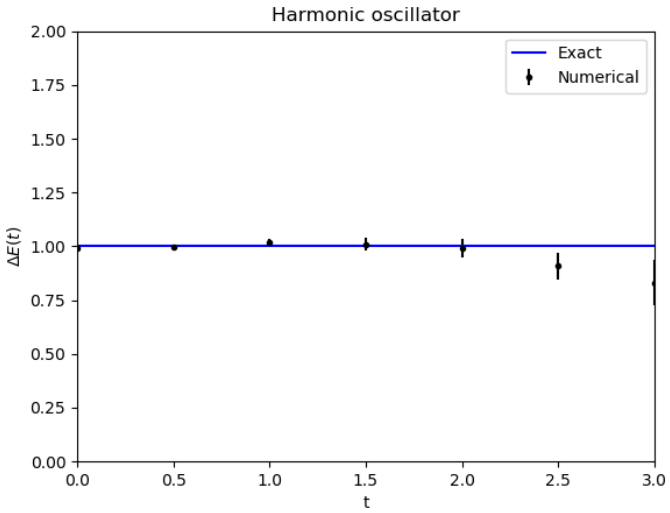

Figure 2.2: Simulation of the ΔE for an Harmonic oscillator (*MetropolisBootstrap.py*)

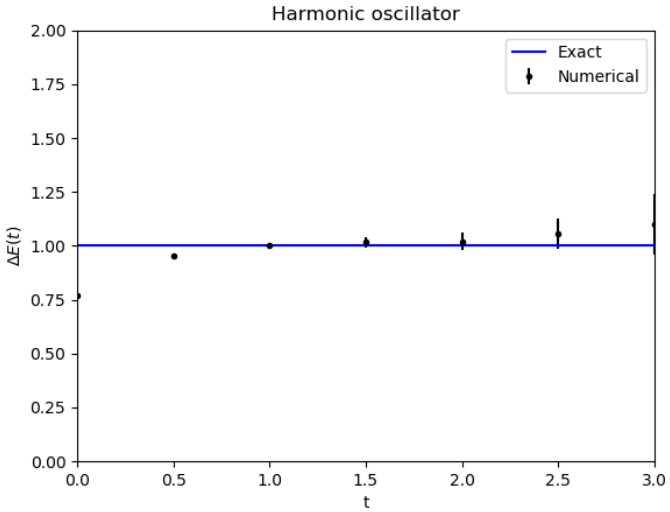

Figure 2.3: Simulation of the ΔE for an Harmonic oscillator, using the improved action we get the "numerical ghost"(*Metropolis_Enhanced.py*)

This "numerical ghost" can be removed performing a change of variables:

$$x_j = x'_j + \frac{a^2}{12}\Delta^{(2)}x'_j - \frac{a^2}{12}\omega_0^2 x'_j, \tag{2.22}$$

which gives us as result in the action

$$S'_{imp}[x'] \equiv \sum_{j=0}^{N-1} a\left[-\frac{1}{2}x'_j\Delta^{(2)}x'_j + V'(x'_j)\right], \tag{2.23}$$

with:

$$V'_{imp}(x') = \frac{1}{2}m\omega_0^2 x_j'^2\left(1 + \frac{(a\omega_0)^2}{12}\right). \tag{2.24}$$

The result of this kind of configuration is given by the figure 2.4 in which we can see that the "numerical ghost" has disappeared.

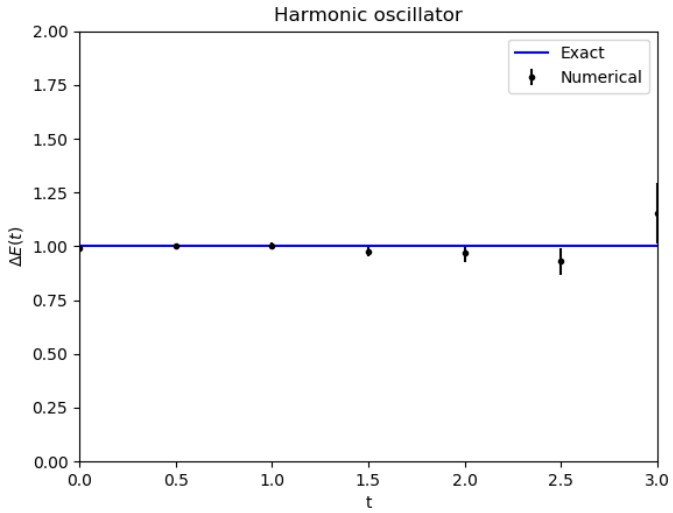

Figure 2.4: Simulation of the ΔE for an Harmonic oscillator, using the substitution we avoid the "numerical ghost" (this result is given by the code *Metropolis_improved_action.py*)

2.2 Lattice QCD

The Quantum Chromo Dynamics is a non abelian gauge theory with symmetry group $SU(3)$ which describe the interactions among quarks mediated by the gluons. The gluonic field is described by the action:

$$S = \int d^4x \frac{1}{2} tr(F_{\mu\nu}F^{\mu\nu}) \tag{2.25}$$

where

$$F_{\mu\nu} \equiv \partial_\mu A_\nu - \partial_\nu A_\mu + ig[A_\mu, A_\nu]. \tag{2.26}$$

The theory is invariant under $SU(3)$ transformations such that:

$$F_{\mu\nu} \longrightarrow \Omega(x) F_{\mu\nu} \Omega^\dagger(x), \tag{2.27}$$

where $\Omega(x)$ is an arbitrary $SU(3)$ x-dependent matrix.
The discretization of the theory is made using the "link variable" $U_\mu(x)$ which are defined as:

$$U_\mu \equiv Pexp\left(-ig \int_x^{x+a\mu} A_\nu(y) dy^\nu\right) \tag{2.28}$$

where P denotes the path ordering and $A_\nu(y)$ is the gauge vector field. This definition is made to keep the gauge invariance of the theory in fact:

$$U_\mu \longrightarrow \Omega(x) U_\mu \Omega^\dagger(x + a\mu). \tag{2.29}$$

Chosen an order to go back to a point x of the lattice we use the dagger link variable $U_\mu^\dagger(x)$. A nice graphical description of the lank variable is given by the figure 2.5.

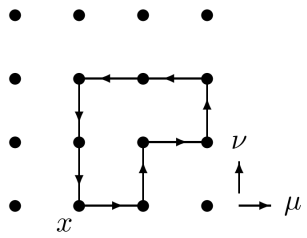

Figure 2.5: Picture of a closed path on the lattice (G. Lepage, Lattice QCD for novice, page 28)

The gauge invariance is kept because, in this case, the quark-gluon, three-gluon and four-gluon coupling are equal and the bare gluon mass is zero. If we loose gauge invariance

we have to tune all this parameters independently and this kind of simulation is very expensive. This becomes more serious considering the loop corrections which renormalize the various coupling differently without the gauge invariance.
The basis of the discretized action is given by the plaquette operator

$$P_{\mu\nu} \equiv \frac{1}{3} Re\ Tr Pexp\left(-ig \oint_{\boxdot} A_\nu(y) dy^\nu\right), \tag{2.30}$$

where $\boxdot$ denotes a closed $a \times a$ square path. Using the link variables it becomes:

$$P_{\mu\nu} \equiv \frac{1}{3} Re\ Tr(U_\mu(x) U_\nu(x + a\mu) U_\mu^\dagger(x + a\nu) U_\nu^\dagger(x)). \tag{2.31}$$

Using this operator we can define the easiest approximation for a discretized action: the Wilson action:

$$S_{Wil} = \frac{6}{g^2} \sum_{x,\mu>\nu} (1 - P_{\mu\nu}(x)). \tag{2.32}$$

Using this action we have a second order error in a in fact:

$$S_{Wil} = \int d^4x \sum_{\mu\nu} \left[\frac{1}{2} Tr\ F_{\mu\nu}^2 + \frac{a^2}{24} Tr\ F_{\mu\nu}(D_\mu^2 + D_\nu^2) F_{\mu\nu} + ...\right]. \tag{2.33}$$

We can improve the action considering the rectangular operator:

$$R_{\mu\nu} \equiv \frac{1}{3} Re\ Tr(U_\mu(x) U_\mu(x + a\mu) U_\nu(x + 2a\mu) U_\mu^\dagger(x + a\mu + a\nu) U_\mu^\dagger(x + a\nu) U_\nu^\dagger(x)). \tag{2.34}$$

Using this operator the action becomes:

$$S_{imp} = -\frac{6}{g^2} \sum_{x,\mu>\nu} \left[\frac{5}{3} P_{\mu\nu}(x) - \frac{1}{12}(R_{\mu\nu} + R_{\nu\mu})\right]. \tag{2.35}$$

In this case we have only a fourth order error in a. The action che be improved considering high order operators on the lattice.
A last improvement to do is to remove the lattice tadpole contributions that would spoil the weak-coupling perturbation theory in the lattice theory and undermine our procedure for improving the lattice discretization. The way to do this is to introduce u_0:

$$u_0 = \langle 0|P_{\mu\nu}|0\rangle^{1/4}, \tag{2.36}$$

in this way the improved action becomes:

$$S_{imp} = -\frac{6}{g^2} \sum_{x,\mu>\nu} \left[\frac{5}{3} \frac{P_{\mu\nu}}{u_0^4} - \frac{1}{12} \frac{R_{\mu\nu} + R_{\nu\mu}}{u_0^6}\right]. \tag{2.37}$$

2.2.1 Wilson loop

A possible first calculation on the lattice is given by the Wilson loop, which is trace of a product of link variables on a closed loop on the lattice:

$$W(C) \equiv \frac{1}{3} Tr P e^{-ig \oint_C A dx} = \frac{1}{3} Tr(U_\mu(x)...U_\nu^\dagger). \tag{2.38}$$

The calculation are given for path $a \times a$ and $a \times 2a$ in the table 1.
The calculation are using the Metropolis algorithm but in this case the upload must be made using random matrix of the group $SU(3)$ because the link variables are $SU(3)$ matrix.
To generate the random matrix we start generating hermitian matrix with elements random number between -1 and 1. Then the matrix are obtained computing the Taylor series and all the elements are divided to the cubic root of the determinant to unitarize the matrix.
The calculation are given using first the Wilson action and then the improved rectangular action. The variation in action for each point are computed considering:

$$\Delta S_{Wil}(x,\mu) = -\frac{\beta}{3} Re\ Tr((U'_\mu(x) - U_\mu(x))\Gamma_\mu(x)) \tag{2.39}$$

with

$$\begin{aligned} \Gamma_\mu(x) = &\sum_{\nu \neq \mu} (U_\nu(x+\mu) U_\mu^\dagger(x+\nu) U_\nu^\dagger(x)) + \\ &(U_\nu^\dagger(x+\mu-\nu) U_\mu^\dagger(x-\nu) U_\nu(x-\nu)) \end{aligned} \tag{2.40}$$

In the computation of $\Gamma_\mu(x)$ we have to consider all the plaquette involving the $U_\mu(x)$ that are 6. For the improved action:

$$\begin{aligned} \Delta S_{imp}(x,\mu) = -\frac{\overline{\beta}}{3} \Bigg(&\frac{5}{3u_0^4} Re\ Tr((U'_\mu(x) - U_\mu(x))\Gamma_\mu(x)) - \\ &\frac{1}{12u_0^6} Re\ Tr((U'_\mu(x) - U_\mu(x))\Gamma_{imp\ \mu}(x))), \end{aligned} \tag{2.41}$$

In this case we have to consider all the rectangular involving $U_\mu(x)$ that are 12.

$$\begin{aligned} \Gamma_{imp\ \mu}(x) = &\sum_{\nu \neq \mu} (U_\mu(x+\mu) U_\nu(x+2\mu) U_\mu^\dagger(x+\mu+\nu) U_\mu^\dagger(x+\nu) U_\nu^\dagger(x)) + \\ &(U_\mu(x+\mu) U_\nu^\dagger(x+2\mu-\nu) U_\mu^\dagger(x+\mu-\nu) U_\mu^\dagger(x-\nu) U_\nu(x-\nu)) + \\ &(U_\nu(x+\mu) U_\nu(x+\mu+\nu) U_\mu^\dagger(x+2\nu) U_\nu^\dagger(x+\nu) U_\nu^\dagger(x)) + \\ &(U_\nu^\dagger(x+\mu-\nu) U_\nu^\dagger(x+\mu-2\nu) U_\mu^\dagger(x-2\nu) U_\nu(x-2\nu) U_\nu(x-\nu)) + \\ &(U_\nu(x+\mu) U_\mu^\dagger(x+\nu) U_\mu^\dagger(x-\mu+\nu) U_\nu^\dagger(x-\mu) U_\mu(x-\mu)) + \\ &(U_\nu^\dagger(x+\mu-\nu) U_\mu^\dagger(x-\nu) U_\mu^\dagger(x-\mu-\nu) U_\nu(x-\mu-\nu) U_\mu(x-\nu))). \end{aligned} \tag{2.42}$$

The $a \times a$ Wilson loop is computed as mean value in all the direction and for all sites of:

$$W(a \times a) = \frac{1}{3} Re\ Tr(U_\mu(x) U_\nu(x + a\mu) U_\mu^\dagger(x + a\nu) U_\nu^\dagger(x)). \tag{2.43}$$

The $a \times 2a$ Wilson loop is computed as mean value in all the direction and for all sites of:

$$\begin{aligned} W(a \times 2a) = & \frac{1}{3} Re\ Tr(U_\mu(x) U_\nu(x + a\mu) \\ & U_\nu(x + a\mu + a\nu) U_\mu^\dagger(x + 2a\nu) U_\nu^\dagger(x + a\nu) U_\nu^\dagger(x)). \end{aligned} \tag{2.44}$$

Another change respect to the old update program is to update the link variable of a site many times (10) before go to the next site to avoid correlations.
The results of the simulation are given in the table 2.1 with the parameters: $\beta = 5.5$, $a = 2.5$, $N = 8$, $\epsilon = 0.24$, $N_{cor} = 50$, $N_{cf} = 10$, $\overline{\beta} = 1.719$, $u_0 = 0.797$.
To use the improved action the parameter *imp* must be changed into $True$ in the function *Update* (default value is $False$).

Type of WL	Wilson Action	Improved Action
$a \times a$	0.500 ± 0.001	0.540 ± 0.001
$a \times 2a$	0.263 ± 0.001	0.283 ± 0.001

Table 2.1: Table of results of the program ($Wilson\_loop.py$)

2.2.2 Static Quark Potential

A first application of the computation of the Wilson loop is given by the estimation of the "static quark potential" which is the energy between interacting quark and anti-quark with infinite mass:

$$V(r) = \sigma r - b/r + c. \tag{2.45}$$

The energy is given considering the Wilson loop $r \times t$ because in the large t limit:

$$W(r,t) \longrightarrow e^{-V(r)t} \tag{2.46}$$

This means that we can compute the potential with:

$$W(r,t)/W(r,t+a) \longrightarrow aV(r). \tag{2.47}$$

There is a substitution that can be used to increased the results. This is to replace the link variables in the r direction with:

$$\overline{U}_\mu(x) \equiv \left(1 + \epsilon a^2 \Delta^{(2)}\right)^n U_\mu(x), \tag{2.48}$$

where

$$\Delta^{(2)} U_\mu(x) \equiv \sum_\rho \Delta_\rho^{(2)} U_\mu(x) \tag{2.49}$$

with $\Delta_\rho^{(2)}$ is the discretized gauge-covariant derivative:

$$\begin{aligned} \Delta_\rho^{(2)} U_\mu(x) \equiv \frac{1}{u_0^2 a^2} (U_\rho(x) U_\mu(x + a\rho) U_\rho^\dagger(x + a\mu) - 2u_0^2 U_\mu(x) \\ + U_\rho^\dagger(x - a\rho) U_\mu(x - a\rho) U(x - a\rho + a\mu)) \end{aligned} \tag{2.50}$$

The low momentum components are not affected by the substitution but, for the high momentum the substitution acts as an ultraviolet cutoff. By suppressing the high momentum gluons we suppress contributions from gluonic excitations and we increase the convergence. This is done by the function $SubstitutionU$ using $\epsilon = 1/12$ and $n = 1$, increasing the number of n will increase the convergence.
The figure 2.6 show my results using the same parameters of the Wilson loop calculation. The calculation is made considering only the planar Wilson loops, a possible way to improve the estimation is to include also the non planar loops to get non integer value of r/a
According to our periodic boundaries condition the maximum time and space distance on the lattice is $4a$ and this is the reason of the few points, but we can see the shape of the potential, in fact the potential is linearly increasing for increasing r so we recover the confinement. The Coulomb-like behaviour for low r is harder to see by the fact that have been used few points and a high value for a.

The few number of points is given by the fact that for other points the large-t limit doesn't holds.
As we can see by the two figure the use of the improved action decrease the error of the estimation.

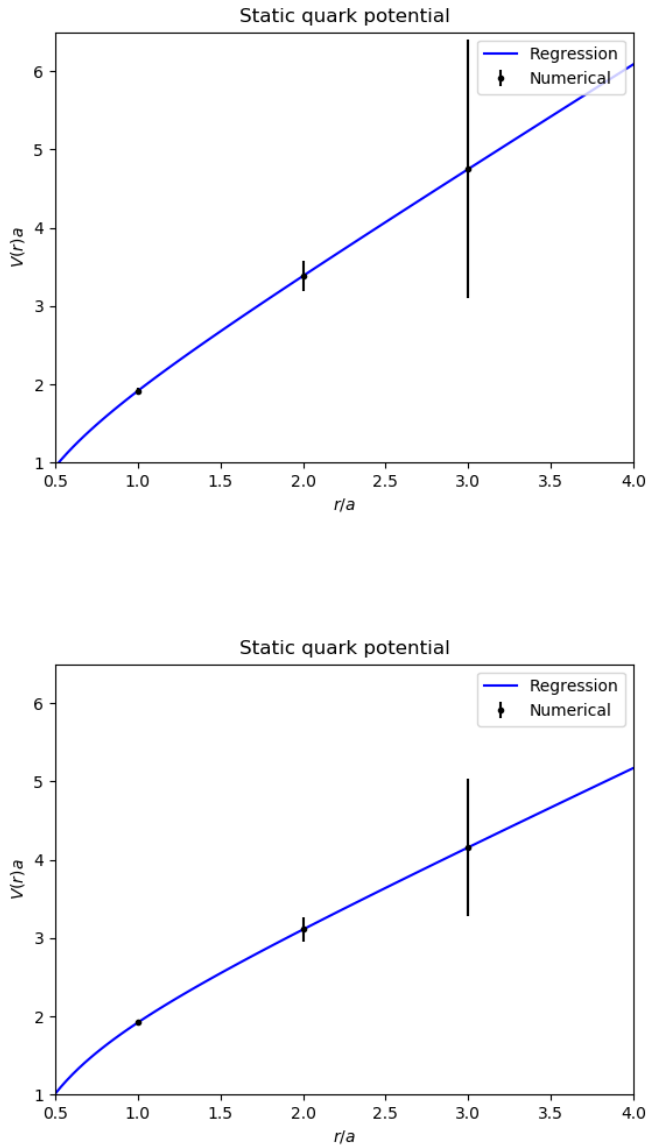

Figure 2.6: Result for the static quark potential using the Wilson action (up) and the improved action (down) (*Static_Quark.py*)

Bibliography

[1] Rafael Guardiola, *Monte Carlo Methods in Quantum Many-Body Theories*, Valencia.

[2] G. Peter Lepage, *Lattice QCD for Novice*, Ithaca NY.

[3] Christine Davies, *Lattice CQD-A guide for people who want results*, Glasgow.

[4] G. Peter Lepage, *Redesigning Lattice QCD*, Ithaca NY, 2018.

[5] C. Gattringer, C. B. Lang, *Quantum Chromodynamics on the Lattice: An Introductory Presentation*, Springer Nature, 2009.

[6] Y. Makeenko, *A Brief Introduction to Wilson Loops and Large N*, 2009.