



SU(2) QCD via Monte Carlo method

Santoni Michele

Faculté des Sciences, Département de Physique, Aix-Marseille Université (AMU), Marseille, France

This study explores SU(2) Quantum Chromodynamics (QCD) using Monte Carlo methods within the Lattice QCD framework, to investigate non-perturbative phenomena like confinement and asymptotic freedom. After outlining the theoretical background, including gauge discretization and Wilson loops, the running coupling equation is derived to connect lattice formulation with the main features of QCD. Successively, Monte Carlo simulations, using the Metropolis algorithm in Python, are implemented to analyze initialization effects, Plaquette dependence on β and the string tension, extracted from Wilson loops. The results align with theoretical expectations, offering insights into the confinement mechanism in SU(2) gauge theory and its broader implications for QCD.

Key words: SU(2) QCD, Monte Carlo simulations, Lattice QCD, Wilson loops, Confinement, Running coupling, Asymptotic freedom, Gauge theory, String tension, Computational methods.

1 Introduction

QCD is the dominant theory of strong interactions, based on quarks and gluons which are supposed to be the basic d.o.f. for building up the hadronic matter. The theory has proved itself in predicting phenomena involving large momentum transfer: in this regime the coupling constant is small and perturbation theory is the principal tool.

On the other hand, at the scale of the hadronic world, $\sim 1 \text{ GeV}$, the perturbative methods break down due to the high value of the coupling constant. In this domain, lattice QCD provides a non-perturbative numerical tool for performing predictions. Not only, LQCD¹ can also be used to address issues like the mechanism for confinement and chiral symmetry breaking, the role of topology and the equilibrium properties of QCD at finite temperature.

In this paper, the theoretical framework and an explicit application to SU(2) LQCD are presented. Specifically speaking: in Sec.(2), a brief sum up of the theoretical background is presented; in Sec.(3), the theoretical framework for the undergoing case is shortly described, showing also the gauge action in the studied case; in Sec.(4), it is possible to find a small presentation of the modality of coding; in Sec.(5), the results of the simulation are reported, both via graph and numerical values. Lastly, in Appendix(6) one can find additional considerations to the presented material.

2 Theoretical Set up

Lattice QCD calculations provide a non-perturbative framework for implementing quantum field theory using the Feynman path integral formalism. These computations follow the same principles as an exact analytical solution of the field theory, assuming such calculations were feasible. In this direction, the numerical implementation of the path integral approach in a lattice framework requires a rigorous procedure:

1. Discretization of space-time: several possibilities are available, in the present case it has been proceeded by an hypercubic lattice:

$$\begin{aligned} \text{spacing} : & \text{ "a" } = \text{ "a_S" } = \text{ "a_T" } \\ \text{size} : & N_S \times N_S \times N_S \times N_T \end{aligned} \quad (1)$$

2. The transcription of the gauge and fermion degrees of freedom, Sec.(2.1);
3. Construction and description of the action, Sec.(3.1);

A more wide and rigorous mathematical introduction is presented in the Ref. [3], Sec.(1) to (7).

2.1 Lattice transcription of $\psi(\mathbf{x})$ and $A_\mu(\mathbf{x})$

The lattice transcription of the matter fields $\psi(\mathbf{x})$ is straightforward, since the quark field is represented

¹Lattice QCD

by anticommuting Grassmann variables, defined at each site of the lattice. They belong to the fundamental representation of SU(3), with a set of precise rules of integration:

$$\int d\psi d\bar{\psi} = \int d\psi d\bar{\psi} \psi = \int d\psi d\bar{\psi} \bar{\psi} = 0$$

$$\int d\psi \psi = \int d\bar{\psi} \bar{\psi} = i \quad \int d\psi d\bar{\psi} \psi \bar{\psi} = 1$$

Since the fermion action is linear in both ψ and $\bar{\psi}$, these rules can be used to integrate over them and the path integral reduces to one over only the gauge degrees of freedom. In conclusion, it turns out that in practice one does not have to worry about transcribing Grassmann variables on the lattice and implementing the Pauli exclusion principle.

At the opposite, the construction of gauge fields is less intuitive. In the continuum limit, the gauge fields $A_\mu(x)$ is a 4-vector with Lorentz indices, that mediates interactions between fermions. To transcribe, it is useful to notice that in the continuum limit a fermion moving from site x to y , in presence of a gauge field $A_\mu(x)$, gains a phase factor given by the path-ordered product:

$$\psi(y) = P e^{ig \int_x^y A_\mu(x) dx_\mu} \psi(x). \quad (2)$$

Hence, at each link² it is associated a discrete version of the path-ordered product:

$$U(x, x + \hat{\mu}) \equiv U_\mu(x) = e^{iagA_\mu(x + \frac{\hat{\mu}}{2})}, \quad (3)$$

where the "average field" A_μ is defined at the midpoint of the link and U is a 3×3 unitary matrix with unit determinant.

Moreover, the path ordering in Eq.(3) also specifies:

$$U(x, x - \hat{\mu}) \equiv U_{-\mu}(x) = e^{-iagA_\mu(x - \frac{\hat{\mu}}{2})} = U^\dagger(x - \hat{\mu}, x). \quad (4)$$

which is the link in the opposite direction.

2.2 Local Gauge Symmetry

In the framework of LQCD, gauge symmetry refers to the local invariance of the theory under transformations from the gauge group, which in the case of QCD is SU(3) for the color charge. This symmetry ensures that the physical observables are independent of arbitrary local changes in the color "phase" at each lattice point.

²The connection between 2 points in the lattice separated by the minimal possible distance.

On the lattice, gauge symmetry is preserved by associating the gauge fields (or link variables) with the edges (or links) between neighboring lattice sites, rather than with the sites themselves. These link variables are elements of the SU(3) group and represent parallel transporters of the color charge between lattice points in a certain direction.

Maintaining gauge symmetry on the lattice is essential, as it guarantees the consistency of the lattice formulation with the fundamental symmetries of QCD in the continuum limit.

The effect of a local gauge transformation $V(x)$ on the variables $\psi(x)$ and U is defined to be

$$\begin{aligned} \psi(x) &\rightarrow V(x)\psi(x) \\ \bar{\psi}(x) &\rightarrow \bar{\psi}(x)V^\dagger(x) \\ U_\mu(x) &\rightarrow V(x)U_\mu(x)V^\dagger(x + \hat{\mu}) \end{aligned} \quad (5)$$

where $V(x)$ is a matrix in the same representation as the U . As a consequence of these definitions, there are two types of gauge-invariant objects that one can construct on the lattice:

- **A string:** consisting of a path-ordered product of links capped by a fermion and an anti-fermion, as shown in Fig.(1)(a). Such strings are called Wilson/Polyakov lines.
- **Closed Wilson loops**, as shown in Fig.(1)(b). The simplest example is the Plaquette, a 1×1 loop, where its value is calculated as:

$$W_{\mu\nu}^{1 \times 1} = \text{Re Tr} (U_\mu(x)U_\nu(x + \hat{\mu})U_\mu^\dagger(x + \hat{\mu} + \hat{\nu})U_\nu^\dagger(x)). \quad (6)$$

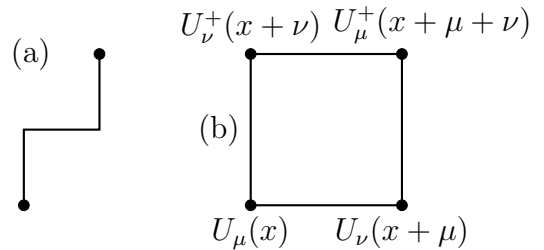


Figure 1: The two gauge-invariant quantities. (a) An ordered string of U 's capped by a fermion and an anti-fermion, and (b) closed Wilson loops.

A gauge invariant action, therefore, has to be built out of loops and strings³.

³Note that these two basic gauge invariant objects can be of arbitrary size and shape. Furthermore, they can be taken to lie in any representation of color SU(3). The only necessary constraint is that it give the familiar continuum theory in the limit $a \rightarrow 0$.

3 Case understudy

The goal of the project is to apply Monte Carlo methods to SU(2), which is a simpler, but meaningful case with respect to SU(3) of the real QCD. As matter of facts, this "reduction" brings to the fact that there are only 3 types of gluons with two different colors; even so, the system still presents the main features of QCD, which are confinement and asymptotic freedom. Lastly, SU(2) toy model is very instructive for understanding the idea behind LQCD and it represents a valid base for extension to the SU(3) work case.

Considering this, in the next sections: the basic theoretical tools for SU(2) are presented, Sec.(3.1); the running coupling equation is computed and the contact with the lattice theory is made, Sec.(3.2); the connection between string tension and Wilson loop is constructed, Sec.(3.3). Moreover, in Appendix, Sec.(6.1), the explicit computation for the gauge action in the SU(2) case has been reported.

3.1 Undergoing case: SU(2) QCD

The classical SU(2) Yang-Mills theory is a consequence of the Lagrangian density term:

$$\mathcal{L} = \frac{1}{4} F_{\mu\nu}^a F^{\mu\nu a}, \quad (3.1)$$

where the internal symmetry index a goes from one to three ⁴ and $F_{\mu\nu}^a$ is defined via the potential A_μ as:

$$F_{\mu\nu}^a = \partial_\mu A_\nu^a - \partial_\nu A_\mu^a + ie_0 \epsilon^{abc} A_\mu^b A_\nu^c. \quad (3.2)$$

Using the Pauli matrices, one obtains a convenient matrix formulation:

$$A_\mu = \frac{1}{2} i \sigma^a A_\mu^a \quad \text{with} \quad A_\mu^a = \text{Tr}(\sigma^a A_\mu) \quad (3.3)$$

$$F_{\mu\nu} = \partial_\mu A_\nu - \partial_\nu A_\mu + ie_0 [A_\mu, A_\nu], \quad (3.5)$$

$$\mathcal{L} = \frac{1}{2} \text{Tr}(F_{\mu\nu} F^{\mu\nu}). \quad (3.6)$$

Then, it is straightforward to connect the lattice theory with the Lagrangian in Eq.(3.6) via:

$$U_i = \exp \left[-ie_0 (x_i^\mu - x_i^\nu) A_\mu \left(\frac{x_i^\mu + x_i^\nu}{2} \right) \right], \quad (3.7)$$

where x_i^μ are the coordinates of the site i .

Considering a Plaquette⁵ in the (μ, ν) plane and Taylor expanding A_μ about the center of this Plaquette, one finds with a little of algebra

$$S = -\frac{1}{2} \text{Tr}(U_\mu U_\nu U_\mu^{-1} U_\nu^{-1}) = \frac{1}{4} e_0^2 a^4 \text{Tr}(F_{\mu\nu} F^{\mu\nu}) + \mathcal{O}(a^5), \quad (3.8)$$

with no implied sum over μ and ν , " a " is the lattice spacing and " e_0 " is the coupling constant. The full computation for the action is reported in Appendix, Sec.(6.1).

Consequently, by combining the Eq.(3.8) with the integral over 4D space-time of Eq.(3.6), and hence making the comparison between the two actions form, it is possible to conclude that

$$\frac{1}{4} e_0^2 \beta = 1 \quad (3.9)$$

which re-establishes the classical theory when the lattice spacing, a , goes to zero.

3.2 The running coupling equation

The running coupling e_R is defined in terms of a physical correlation function at a mass scale μ . Moreover, an ultraviolet cut-off point is introduced at length scale a in the formula. Consequently, the coupling relation appears:

$$e_R(e_0, \mu, a) = e_0 + \mathcal{O}(e_0^3). \quad (3.10)$$

To recover correctly the quantum theory in the continuum limit, when one takes $a \rightarrow 0$, e_0 must be adjusted such that e_R remains finite.

Proceeding the analysis, one has that the running coupling function is given by:

$$\gamma(e_R) = \mu \frac{\partial}{\partial \mu} e_R(e_0, \mu, a). \quad (3.11)$$

where, to be physically well-defined, it is needed that $\gamma(e_R)$ stays finite as $a \rightarrow 0$.

Looking specifically at the SU(2) case, the perturbative expansion of $\gamma(e_R)$ starts as:

$$\gamma(e_R) = -\frac{11e_R^3}{24\pi^2} + \mathcal{O}(e_R^5). \quad (3.12)$$

Since e_R is held constant while a vanishes, differentiating Eq.(3.10) gives:

⁵In lattice QCD, a Plaquette is the smallest closed loop on the lattice, formed by the product of gauge link variables around an elementary square. It represents the discrete version of the gauge field strength tensor $F_{\mu\nu}$ from the continuum theory and is a fundamental building block in the formulation of lattice gauge actions.

⁴the number of generators

$$0 = a \frac{d}{da} e_R(e_0, \mu, a) = \frac{\partial e_R}{\partial e_0} \frac{de_0}{da} + a \frac{\partial e_R}{\partial a}. \quad (3.13)$$

Via dimensional analysis:

$$a \frac{\partial e_R}{\partial a} = \mu \frac{\partial e_R}{\partial \mu} = \gamma(e_R) \quad (3.14)$$

And then by combining all the previous equations, from Eq.(3.10) to Eq.(3.14), the final result is:

$$a \frac{de_0}{da} = \frac{11e_0^3}{24\pi^2} + O(e_0^5) \quad (3.15)$$

Notice that in the case in which e_0 becomes sufficiently small, the $O(e_0^5)$ term can be neglected and it is possible to integrate Eq.(3.15), obtaining:

$$e_0^2 = \frac{12\pi^2}{11 \ln(\tilde{a}/a)} + O((\ln a)^{-2}) \quad (3.16)$$

However, this analysis does not determine the integration constant \tilde{a} .

Due to the fact that one needs to make the comparison with Monte Carlo simulations, it is necessary to rewrite Eq.(3.16) in a more convenient form:

$$a^2 \underset{e_0 \rightarrow 0}{\sim} \tilde{a}^2 \exp\left(-\frac{24\pi^2}{11e_0^2}\right) = \tilde{a}^2 \exp\left(-\frac{6\pi^2}{11}\beta\right) \quad (3.17)$$

This is the prediction of asymptotic freedom in non-Abelian gauge theories.

3.3 String tension and Wilson Loop

In lattice gauge theory, Wilson loops are fundamental objects of study, as it is described in Sec.(2.2).

For large separations, the interaction energy between two static sources in the fundamental representation of the gauge group grows linearly with distance. Thus, for sufficiently large contours, one expects a trend like:

$$\ln W(C) = -KA(C) + O(p(C)), \quad (3.20)$$

where $A(C)$ is the minimal enclosed area and $p(C)$ represents the perimeter of the contour.

In this scenario, the string tension, K , is a fundamental quantity that characterizes the confining force between static quark-antiquark pairs and corresponds to the coefficient of the linear term in the static quark-antiquark potential. By measuring A in physical units, A results to be the lattice spacing squared times the number of Plaquettes $N_\square(C)$ forming the enclosed area. This allows us to express:

$$\ln W(C) = -(a^2 K)N_\square(C) + O(p(C)) \quad (3.21)$$

As a consequence of this, via the analysis of the decay behavior of $W(C)$ with respect to N_\square , one can determine the dimensionless quantity $a^2 K$ as a function of the bare coupling e_0 .

Furthermore, the string tension is a physical quantity and it provides a reference scale for the continuum limit of the theory. Therefore, it is natural to adopt a renormalization scheme where K remains fixed while $a \rightarrow 0$, adjusting e_0 accordingly. Given that K is inherently non-perturbative, this renormalization scheme does not directly follow the perturbative approach of Eq. (3.10). However, asymptotic freedom ensures that the result in Eq. (3.17) remains valid⁶, predicting that for large β :

$$a^2 K \sim \exp\left(-\frac{6\pi^2\beta}{11}\right) \quad (3.22)$$

Notably, the only singularity in Eq. (3.22) is when the coupling vanishes and this further supports the statement that string tension is a purely non-perturbative quantity.

Lastly, observe that in the small β regime, Wilson's strong-coupling expansion is applicable and the first non-vanishing term in this expansion is:

$$W(C) \sim \left(\frac{1}{4}\beta\right)^{N_\square} \quad (3.23)$$

Consequently, it is expected:

$$a^2 K \sim -\ln\left(\frac{1}{4}\beta\right) \quad (3.24)$$

4 The code

The simulation of the lattice has been done using "CoLab" and programming in "Python". In this specific case, the aim was reproducing results from Ref. [1] from the 1970s, hence there was not any particular request for efficiency and processing speed. However, if there is the interest to study the system at higher sizes, some huge improvements could be seen using compiled languages, such as C and C++. This topic is described in Appendix, Sec.(6.2), where the comparison between Python and C/C++ is used as reference.

5 Outcomes

In this section, the outcomes of the study are analyzed, and several configurations have been implemented to compose a complete overview on the process: in Sec.(5.1), it is studied the behavior of the

⁶The result is actually independent on the renormalization scheme

system given two different initialization of the lattice; in Sec.(5.2), it is studied the trend of the Plaquette values considering different β s; in Sec.(5.3), the trend for different Wilson loops sizes are studied with respect to β ; in Sec.(5.4), the "string tension" is described and analyzed via fitting of the Wilson loops values.

5.1 "Hot" and "Cold" initialization

The first study is related to the convergence of the process, both in "Hot"⁷ and "Cold"⁸ configuration. The code has been running for the same value of " β ", equal to 2.3, for 50 times varying the size of the lattice.

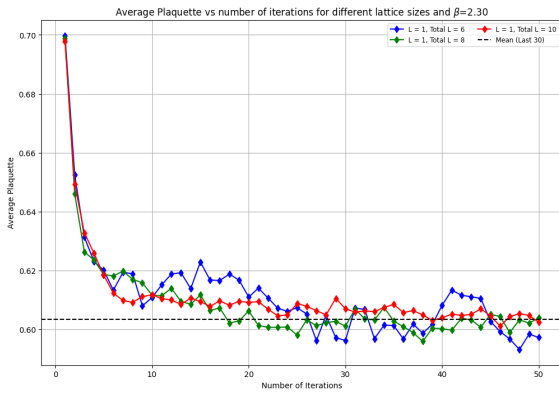


Figure 2: The image shows the behavior of the average Plaquette, having set an initial configuration in the cold state, for lattice size equal to 6 (Blue), 8(Green) and 10(Red) and 50 iterations in total.

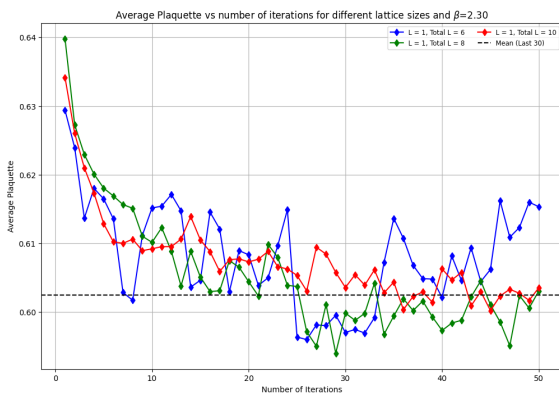


Figure 3: The image shows the behavior of the average Plaquette, having set an initial configuration in the hot state, for lattice size equal to 6 (Blue), 8(Green) and 10(Red) and 50 iterations in total.

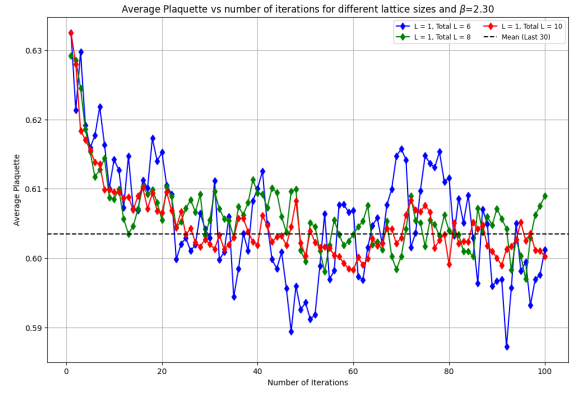


Figure 4: The image shows the behavior of the average Plaquette, having set an initial configuration in the hot state, for lattice size equal to 6 (Blue), 8(Green) and 10(Red) and 100 iterations in total.

In Fig.(2) and in Fig.(3) are presented the graph of the average Plaquette with respect to the number of iterations, for three different lattice size: 6^4 , 8^4 and 10^4 . Moreover, the second case, the hot one, has also been implemented 100 times, see Fig.(4), in order to make better explicit the "convergence". As matter of facts, one main feature is evident and clear in all the graphs: the Plaquette passes through an initial instable phase and then its value oscillates around an average quantity. This phase is called "Thermalization" and, as it is shown, it is completely independent on the lattice size and type of initialization; this phase is a consequence of the initial conditions and the points are just outliers with a small weight in the path integral, and hence a higher value in the action.

Besides, the systems seems to asymptotically converge to a fixed value, but in reality they will continue to oscillate around an average quantity. The result for just one iteration has no real physical meaning.

However, just for giving an idea, the results for the apparent convergence are reported in Tab.(1), where the mean has been done excluding the initial 20 points, to neglect the thermalization phase.

Lattice	6^4	8^4	10^4
<i>Mean Cold</i>	0.6037	0.6038	0.6023
<i>Mean Hot 50</i>	0.6055	0.6006	0.6043
<i>Mean Hot 100</i>	0.6037	0.6040	0.6032

Table 1: Table of the values where the mean Plaquette is oscillating; the average is taken after the thermalization, that is to say excluding the initial 20 points.

⁷This correspond to a completely random initialization of the lattice using SU(2) matrices.

⁸This correspond to an initialization of the lattice using everywhere the identity matrix.

Lastly, it should be highlighted that as the lattice size grows, the range of the fluctuations decreases;

the reason is that bigger lattices have more Plaquette values over which the average is done, since there are more lattice sites where the computation could be done. As a consequence of this, one expects to see very tiny fluctuations as the lattice size increases.

5.2 Plaquette convergence with respect to β

The second aspect of the study has proceeded in the investigation of the convergence: repeating the process with different values of beta, but keeping fixed the lattice size ($L=8$) and the number of iterations (to 50), the convergence of the system has been explored.

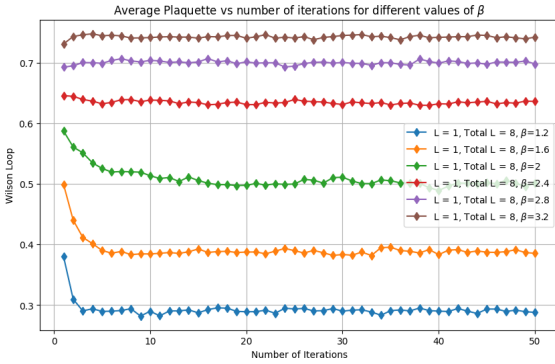


Figure 5: The image shows the trends of the Plaquette with respect to iterations for different values of β (precisely: 1.2, 1.6, 2, 2.4, 2.8 and 3.2), but with the same lattice size (8), same number of iterations and same initialization, the cold one.

In Fig.(5), it is possible to see the behavior of the system for different values of beta: for higher value of beta, the system is converging to a higher values in Plaquette, approaching one, and viceversa. Besides, for smaller values of β the initial phase of thermalization is more emphasized with respect to the higher ones; this is just a consequence of the initial conditions of the system, as already stated in previous section.

5.3 Wilson loop with respect to β

In Fig(6), the behavior of the Wilson's loops for different values of β has been explored. The graph shows different behavior of the Wilson loops depending on their size; moreover, all the loops approach unity inversely with β at low temperature⁹.

⁹As matter of facts, it is possible to make a parallelism between the "Euclidean Field Theory" and the "Classical Statistical Mechanics" by identifying $\beta = 1/k_b T$, where "T"

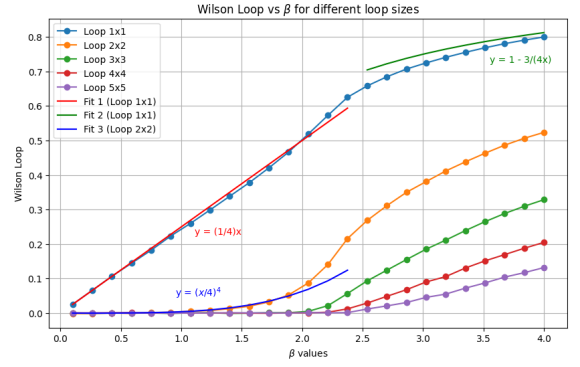


Figure 6: The image shows the trend of the different (different in size) Wilson loops with respect to β , overlapping also the expected behavior for the 1×1 and 2×2 at low and high values.

Lastly, some functions have been fitted on the graph to underline the trend. In facts, when β goes to zero, one is in the strong-coupling regime and expects to see, from Eq.(3.23), that:

$$W(\square) \underset{\beta \rightarrow 0}{\sim} \frac{1}{4}\beta, \quad (4.1)$$

$$W(2 \times 2) \underset{\beta \rightarrow 0}{\sim} \left(\frac{1}{4}\beta\right)^4 \quad (4.2)$$

On the other hand, in the weak-coupling limit one gets:

$$W(\square) \underset{\beta \rightarrow \infty}{\sim} 1 - \frac{3}{4\beta}. \quad (4.3)$$

These trends are confirmed by the tendency of the data.

It is relevant to notice not only that the behavior is not completely follow in the weak limit, but also the points on the graph do not present any error bars. In facts, taking into account the error bars, it is possible to show the compatibility between Eq.(4.3) and the generated points. In facts, at higher values in β the errors are bigger as a consequence of the structure of the code for generating matrices.

5.4 String tension

In the context of SU(2) Lattice Gauge Theory, the string tension is a fundamental quantity that characterizes the confining force between static quark-antiquark pairs. It essentially determines how the potential between these color charges grows with

is the temperature and k_b is the Boltzmann constant. Hence, small values of "T" means higher values of β . For more about it, Ref. [3], from pag. 21 to pag.25.

distance and is a key indicator of confinement in QCD.

In this specific case, to extract the string tension from the data, at a given value of β , one performs a fit over the loops using:

$$W(S) = \exp[-(A + B \times S + C \times S^2)] \quad (7)$$

where S is the loop side.

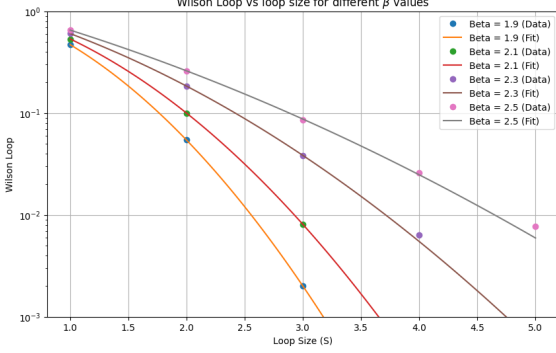


Figure 7: The image shows the fit of the Wilson loops values with respect to Eq.(7). Each point has been generated in lattice size of order 10, doing 80 iterations and considering the Wilson loops from 1×1 to 5×5 ; this procedure has been done for four different values of β : 1.9, 2.1, 2.3 and 2.5.

Note that the points at higher dimension in Wilson loop, 4×4 and 5×5 , are relatively distant from the expected curve. This is related on one side to the absence of error bars and on the other due to the fact that bigger Wilson loop are characterized by bigger errors, as a consequence of the propagation in the multiplication of the lattice elements.

Finally, using both the data from the lattice size 8^4 and 10^4 , it is possible to study the running of the coupling in LQCD.

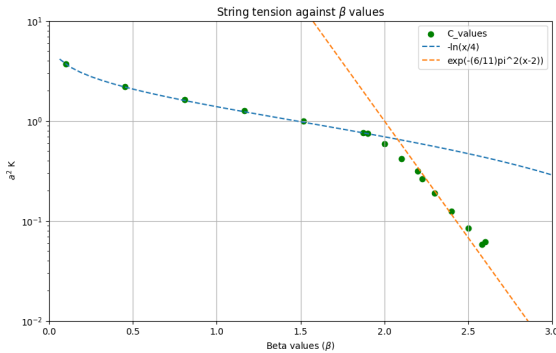


Figure 8: The image shows the cut-off square times the string tension against the β values which are equivalent to $4/e_0^2$, Eq.(3.9). On the graph are also reported the strong (blue) and weak (orange) coupling limits.

In Fig.(8), all these results for the string tension are presented, also plotting the strong-coupling result of Eq.(4.1) and the weak-coupling conclusion of Eq.(4.3). The graph has been made using both data from lattice size 8^4 and 10^4 , performing 80 iterations and varying β between 0 and 3, generating more points in the middle region, where is present a changing in the trend.

This last analysis is very crucial, as it allows the determination of the renormalization scale of the coupling in terms of the string tension.

Considering the expected asymptotic trend from Eq.(3.22) and recalling Eq.(3.9), one can solve for e_0^2 in the relation and gain

$$\frac{e_0^2}{4\pi} \underset{a \rightarrow 0}{\sim} \frac{3\pi}{11 \ln(1/a\Lambda)}, \quad (5.3)$$

where the renormalization scale is given by

$$\Lambda \approx \sqrt{K} \exp\left(-\frac{6\pi^2}{11}\right) \approx \frac{1}{200} \sqrt{K}. \quad (5.4)$$

As it was expected from Sec.(3.3), K provides a reference scale for the continuum limit of the theory.

5.5 Conclusions

On the whole, Sec.(5) has proved the great efficiency and relevance of LQCD approach for SU(2), proving the presence of asymptotic freedom for the bare coupling constant in a renormalization scheme based on confinement. This latter proof strongly suggests that SU(2) non-Abelian gauge theory exhibits both confinement and asymptotic freedom, and it is a framework which can be used for further expansion into the SU(3) case. Lastly, the reproduction of the asymptotic-freedom prediction permits to emphasize the link between the lattice formulation and the more conventional perturbative approaches.

6 Appendix

6.1 Gauge Action

The aim is to prove the equivalence in the Eq.(3.8), such that the link between the theory and the lattice approach is well-justified. The best choice that one can do is to express gauge action in terms of closed loops.

Starting from the simplest Wilson loop, the 1×1 Plaquette $W_{\mu\nu}^{1 \times 1}$:

$$W_{\mu\nu}^{1 \times 1} = U_\mu(x)U_\nu(x + \hat{\mu})U_\mu^\dagger(x + \hat{\nu})U_\nu^\dagger(x), \quad (8)$$

$= e^{iae_0 A_\mu(x+\frac{\hat{\mu}}{2})} e^{iae_0 A_\nu(x+\hat{\mu}+\frac{\hat{\nu}}{2})} e^{-iae_0 A_\mu(x+\hat{\nu}+\frac{\hat{\mu}}{2})} e^{-iae_0 A_\nu(x+\frac{\hat{\nu}}{2})}$. In Monte Carlo simulations, Python can be

Now, by expanding about $x + \frac{\hat{\mu}+\hat{\nu}}{2}$ one gets:

$$e^{iae_0(A_\mu)} e^{iae_0(A_\nu+a\partial_\mu A_\nu+\dots)} e^{-iae_0(A_\mu+a\partial_\nu A_\mu+\dots)} e^{-iae_0(A_\nu)} \quad (9)$$

Using the Baker–Campbell–Hausdorff formula:

$$e^x e^y = e^{x+y+\frac{1}{2}[x,y]+\dots}$$

where x and y are 2 operators, one obtains:

$$\begin{aligned} &\approx e^{iae_0(A_\mu+A_\nu+a\partial_\mu A_\nu)-iae_0(A_\mu+A_\nu+a\partial_\nu A_\mu)+a^2 e_0^2[A_\mu,A_\nu]+\dots} \\ &= e^{ia^2 e_0(\partial_\mu A_\nu-\partial_\nu A_\mu-ie_0[A_\mu,A_\nu]+\dots)} \\ &= e^{+ia^2 e_0 F_{\mu\nu}} \\ &\approx 1 + ia^2 e_0 F_{\mu\nu} - \frac{a^4 e_0^2}{2} F_{\mu\nu} F^{\mu\nu} + \mathcal{O}(a^6) + \dots \end{aligned} \quad (10)$$

Hence, taking half¹⁰ of the trace and subtracting it to one:

$$\begin{aligned} &1 - \frac{1}{2} \text{Tr}[U_\mu(x) U_\nu(x+\hat{\mu}) U_\mu^\dagger(x+\hat{\nu}) U_\nu^\dagger(x)] \\ &\approx 1 - \frac{1}{2} \text{Tr}[1 + ia^2 e_0 F_{\mu\nu} - \frac{a^4 e_0^2}{2} F_{\mu\nu} F^{\mu\nu}] \\ &= 1 - \frac{1}{2} \left[\text{Tr}(1) + ia^2 e_0 \text{Tr}(F_{\mu\nu}) - \frac{a^4 e_0^2}{2} \text{Tr}(F_{\mu\nu} F^{\mu\nu}) \right] \\ &= \frac{a^4 e_0^2}{4} \text{Tr}(F_{\mu\nu} F^{\mu\nu}) + \dots \end{aligned}$$

Considering the real and imaginary parts of the loop one observes that:

$$\text{Re Tr}(1 - W_{\mu\nu}^{1\times 1}) = \frac{a^4 e_0^2}{2} F_{\mu\nu} F^{\mu\nu} + \mathcal{O}(a^5), \quad (11)$$

$$\text{Im}(W_{\mu\nu}^{1\times 1}) = a^2 e_0 F_{\mu\nu}. \quad (12)$$

where so far the indices μ and ν are uncontracted.

6.2 Computational effort

The computational speed difference between Python and C/C++ depends on multiple factors, including the nature of the task, compiler optimizations, and hardware efficiency. However, general benchmarks indicate:

- C/C++ is typically 10x to 100x faster than Python for computational tasks due to its compiled nature, whereas Python is an interpreted language. See Ref. [5] [6];

¹⁰”half” due to the normalization chosen, given by: $1/N$ where N is the one in SU(N)

20x–50x slower than C++ but can be optimized with JIT compilation (e.g., Numba). See Ref. [7] [8];

- Matrix operations using NumPy reduce the performance gap to 2x–3x slower than C++, as NumPy calls C-optimized BLAS/LAPACK libraries. See Ref. [9] [10].

Hence, making a rough estimation it is possible to suppose a difference of the order $10^2 - 10^3$ in the computational time for a single iteration between the two languages, which is not negligible at all.

6.3 Errors estimation

In all the previous analysis, the effects of errors has not been presented and clearly this is an unrealistic scenario. There are several sources of errors, like the ones related to all the approximations, and making an estimation is quite difficult. One possibility is to apply the Jack Knife method, which is a resampling technique commonly used in lattice gauge theory to estimate errors, particularly for quantities like string tension, Wilson loops and correlation functions. The advantages is that particularly effective for handling correlated data, which arises naturally in Monte Carlo simulations. To have a full and rigorous explanation for the Jack Knife method see Ref. [4].

Suppose to have N independent measurements of an observable, such as the Wilson loop expectation value, and form N subsets by leaving out one data point at a time. For each subset i , one computes the jackknife estimation of the observable by averaging over the remaining $N - 1$ data points:

$$O_i = \frac{1}{N-1} \sum_{j \neq i} O_j \quad (13)$$

The jackknife error (standard deviation) is obtained as:

$$\sigma_O^2 = \frac{N-1}{N} \sum_{i=1}^N (O_i - \bar{O})^2 \quad (14)$$

This method reduces bias in nonlinear estimators (e.g., effective mass from correlation functions), handles correlations between successive Monte Carlo samples efficiently and performs well on small datasets, which are common in lattice QCD calculations.

References

- [1] Michael Creutz, *Monte Carlo Study of Quantized $SU(2)$ Gauge Theory*, Physical Review D, vol. 21, no. 8, pp. 2308–2315, 1980.
- [2] Julius Kuti, Janos Polonyi, Kornél Szlachányi, *Monte Carlo Study of $SU(2)$ Gauge Theory at Finite Temperature*, Physics Letters B, vol. 98, no. 3, pp. 199–202, 1981.
- [3] Rajan Gupta, *Introduction to Lattice QCD*, arXiv:hep-lat/9807028v1, doi: [10.48550/arXiv.hep-lat/9807028](https://doi.org/10.48550/arXiv.hep-lat/9807028).
- [4] H. Friedl and E. Stampfer, "Jackknife Resampling," *Encyclopedia of Environmetrics*, June 2001. doi: [10.1002/9780470057339.vaj001](https://doi.org/10.1002/9780470057339.vaj001).
- [5] L. Prechelt, "An Empirical Comparison of C, C++, Java, Perl, Python, REXX, and Tcl," *IEEE Transactions on Software Engineering*, vol. 26, no. 6, pp. 662–676, 2000. doi: [10.1109/32.877841](https://doi.org/10.1109/32.877841).
- [6] C. McGeoch, *A Guide to Experimental Algorithmics*. Cambridge University Press, 2012.
- [7] M. Lis and K. Zielinski, "Comparing the Performance of Monte Carlo Simulations in Python and C++," *Journal of Computational Physics*, vol. 368, pp. 151–164, 2018. doi: [10.1016/j.jcp.2018.03.021](https://doi.org/10.1016/j.jcp.2018.03.021).
- [8] M. Harris, "Accelerating Python with Numba," *NVIDIA Developer Blog*, 2019. [Available online](#).
- [9] S. Van Der Walt, S. C. Colbert, and G. Varoquaux, "The NumPy Array: A Structure for Efficient Numerical Computation," *Computing in Science Engineering*, vol. 13, no. 2, pp. 22–30, 2011. doi: [10.1109/MCSE.2011.37](https://doi.org/10.1109/MCSE.2011.37).
- [10] G. H. Golub and C. F. Van Loan, *Matrix Computations*. Johns Hopkins University Press, 4th ed., 2013.