PROCEEDINGS OF SCIENCE

Efficiency study of overrelaxation and stochastic overrelaxation algorithms for SU(3) Landau gauge-fixing

Jesuel Marques,[*] **Matheus Cerqueira and Tereza Mendes**

Instituto de Física de São Carlos, Universidade de São Paulo,
C.P. 369, 13560-970, São Carlos, SP, Brazil

E-mail: jesuel.leal@usp.br, mcerqueira@usp.br, mendes@ifsc.usp.br

As part of our study of two-point functions in SU(3) lattice gauge theory, we have carried out a comparative analysis of Landau Gauge Fixing algorithms, which complements similar existing studies for the SU(2) case. We present the results of our optimization analysis for the Landau Gauge Fixing overrelaxation and stochastic overrelaxation algorithms. By studying the distribution of necessary sweeps for gauge-fixing of a sample of configurations, we obtain the optimal choice of parameters for these algorithms, as well as their dynamic critical exponent.

The 38th International Symposium on Lattice Field Theory, LATTICE2021 26th-30th July, 2021
Zoom/Gather@Massachusetts Institute of Technology

[*] Speaker

1. Introduction

In studies of the gauge-dependent Green's functions of QFTs on the lattice, a necessary step is the gauge-fixing procedure. Given a configuration of link variables $U_\mu(n)$, in order to fix it to Landau gauge, we consider the following functional [1, 2]

$$\mathcal{E}[g] \equiv \frac{1}{dN_c|\Lambda|}\sum_{\mu=1}^{d}\sum_{n\in\Lambda}\frac{1}{2}\mathrm{tr}\left[U_\mu^{(g)}(n) + U_\mu^{(g)\dagger}(n)\right], \tag{1}$$

where $U_\mu^{(g)}(n) \equiv g(n)U_\mu(n)g^\dagger(n+\hat{\mu})$ is the gauge-transformed link variable, in which $\hat{\mu}$ denotes the unit vector in the direction μ. The number of space-time dimensions is d, Λ is the set of lattice sites, and $|\Lambda|$ is their number. The number of colors is N_c. If we find a maximum of this functional, with respect to $\mathcal{G} \equiv \{g(n)\}$, keeping $\{U_\mu(n)\}$ fixed, we will have that a discretized version of the Landau gauge condition is obeyed

$$\nabla_\mu A_\mu^b(n) = \sum_{\mu=1}^{d}\left(A_\mu^b(n) - A_\mu^b(n-\hat{\mu})\right) = 0, \tag{2}$$

for all points n. A discretization of the gauge field is

$$A_\mu(n) = \frac{U_\mu(n) - U_\mu^\dagger(n)}{2i} - \frac{1}{N_c}\mathrm{tr}\left[\frac{U_\mu(n) - U_\mu^\dagger(n)}{2i}\right]. \tag{3}$$

In practice, the gauge-fixing procedure is executed in an iterative fashion. In order to be considered "Landau gauge-fixed", $\nabla_\mu A_\mu^b(n)$, or some other measure of the quality of the fixing, needs to be smaller than a given tolerance. One such quantity is

$$e_2 = \frac{1}{|\Lambda|}\sum_{n\in\Lambda}\sum_{b=1}^{N_c^2-1}\left(\nabla_\mu A_\mu^b(n)\right)^2, \tag{4}$$

and the tolerance for this will be taken to be 10^{-12} here.

2. Algorithms

2.1 Los Alamos algorithm for SU(2) (LASU(2))

The Los Alamos algorithm [3] is a local algorithm to implement a gauge-fixing procedure. To explain its principle, we can define a local version of the $\mathcal{E}$ functional, which exhibits the dependence on the gauge transformation for a specific point n

$$\mathcal{E}[g(n)] \equiv \frac{1}{dN_c|\Lambda|}\mathrm{tr}[g(n)h(n)] \equiv \frac{1}{dN_c|\Lambda|}\mathrm{tr}[w(n)], \tag{5}$$

with

$$h(n) = \sum_{\mu=1}^{d}\left(U_\mu(n)g^\dagger(n+\hat{\mu}) + U_\mu^\dagger(n-\hat{\mu})g^\dagger(n-\hat{\mu})\right). \tag{6}$$

In this algorithm, we sweep through the lattice and maximize Eq. 5 for each point in turn. In the case of the SU(2) gauge group, we can take advantage of the fact, valid for this group, that $h(n)$, being the sum of SU(2) matrices, is proportional to an element of the group, with the constant of proportionality given by $\sqrt{\det h(n)}$. We can then define a projection to SU(2) by

$$\mathcal{P}_{\mathrm{SU}(2)}\left[h(n)\right] = \frac{h(n)}{\sqrt{\det h(n)}}. \tag{7}$$

The matrix $g(n)$ which maximizes Eq. 5 is then $\mathcal{P}_{\mathrm{SU}(2)}\left[h^{\dagger}(n)\right]$. Another way of writing this update is

$$g^{\mathrm{new}}(n) = A(n) g^{\mathrm{old}}(n), \tag{8}$$

and for the Los Alamos algorithm

$$A^{\mathrm{LA}}(n) = \mathcal{P}_{\mathrm{SU}(2)}\left[h^{\dagger}(n) g^{\mathrm{old}\dagger}(n)\right] = \mathcal{P}_{\mathrm{SU}(2)}\left[w^{\mathrm{old}\dagger}(n)\right], \tag{9}$$

with $w(n)$ defined by Eq. 5. We sweep through the lattice multiple times, updating the values of $g(n)$ following Eq. 9, and, at the end of each sweep, we measure a quantity indicative of how close we are to the gauge fixing condition.

2.2 (Pseudo)Los Alamos algorithm for SU(3) (LA)

Ideally, we would like to have $\mathcal{E}[g(n)]$ placed at its maximum after each update, as it happens in SU(2). However, in this case, it is not straightforward to have a closed analytical formula for the update. The problem is that, for SU(3), $w(n)$ (or $h(n)$) is not proportional to an element of the group. The best we can do, using the same logic as in the SU(2) case, is to promote a division of $w(n)$ in submatrices, like in the Cabibbo-Marinari trick for the pseudo heat-bath for SU(N) [4, 5]. Then, we try an update as in Eq. 8 with $A(n) = T(n)S(n)R(n)$, where each factor is a SU(2) matrix embedded in a larger SU(3) matrix (with the n dependence omitted for brevity)

$$R = \begin{pmatrix} r_0 + i r_3 & r_2 + i r_1 & 0 \\ -r_2 + i r_1 & r_0 - i r_3 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \; S = \begin{pmatrix} s_0 + i s_3 & 0 & s_2 + i s_1 \\ 0 & 1 & 0 \\ -s_2 + i s_1 & 0 & s_0 - i s_3 \end{pmatrix}, \; T = \begin{pmatrix} 1 & 0 & 0 \\ 0 & t_0 + i t_3 & t_2 + i t_1 \\ 0 & -t_2 + i t_1 & t_0 - i t_3 \end{pmatrix}, \tag{10}$$

with $\sum_{i=0}^{3} r_i^2 = \sum_{i=0}^{3} s_i^2 = \sum_{i=0}^{3} t_i^2 = 1$. The matrices $R(n)$, $S(n)$ and $T(n)$ will then depend on $w(n)$, where each entry w_{ij} of $w(n)$ is in general a complex number.

Firstly, we can keep $S(n) = T(n) = \mathbb{1}$ and update only with the matrix $R(n)$. In this case, we can ask ourselves what is the maximum of $\mathrm{Re}\,\mathrm{tr}[R(n)w(n)]$, for a given $w(n)$, as a function of the constants r_0, r_1, r_2 and r_3, under the constraint that the sum of its squares needs to be equal to 1, to satisfy the definition of the group. This can be solved as a problem of maximization with constraints, using, for example, Lagrange multipliers. The answer is

$$\begin{aligned} r_0 &= \frac{\mathrm{Re}[w_{11} + w_{22}]}{\lambda_R} & r_1 &= -\frac{\mathrm{Im}[w_{12} + w_{21}]}{\lambda_R} \\ r_2 &= -\frac{\mathrm{Re}[w_{12} - w_{21}]}{\lambda_R} & r_3 &= -\frac{\mathrm{Im}[w_{11} - w_{22}]}{\lambda_R}, \end{aligned} \tag{11}$$

where $\lambda_R = \sqrt{\text{Re}\,[w_{11}+w_{22}]^2 + \text{Im}\,[w_{12}+w_{21}]^2 + \text{Re}\,[w_{12}-w_{21}]^2 + \text{Im}\,[w_{11}-w_{22}]^2}$. This is equivalent to the Los Alamos update in SU(2), if the relevant submatrix of $w(n)$ is proportional to an SU(2) matrix. Iteratively, we can now use $w'(n) = R(n)w(n)$ for the next update corresponding to $S(n)$. Then the s_i's follow the same formula as in Eq. 11, changing $r \to s$, $w \to w'$ and the matrix indices $2 \to 3$. Finally, for $T(n)$, in an analogous way, $s \to t$, $w' \to w''$ and $1 \to 2$. The Lagrange multipliers λ_T and λ_S should be such that $\sum_{i=0}^{3} s_i^2 = \sum_{i=0}^{3} t_i^2 = 1$, as was the case with λ_R. In the end, we have $A_1(n) = T(n)S(n)R(n)$. We point out that the resulting $A_1(n)$ is not the matrix which maximizes[1] Re $\text{tr}[A(n)w^{\text{old}}(n)]$. Each step increases the value of this quantity, but the maximum will be attained only asymptotically after some update "hits" at the same lattice site, $A^{\text{LA}}(n) = A_{\#\text{hits}}(n)A_{\#\text{hits}-1}(n)\ldots A_1(n)$. However, the extra work to make these "hits" may not translate into an overall gain of time in the convergence of the algorithm and an optimization study is necessary in order to determine how many "hits" are worth making. This analysis is presented in section 3.

We note that the formulas presented here are also found in [5], but with a few minus sign mistakes and, perhaps, a confusing explanation.

2.3 Overrelaxation (OR)

One way of accelerating the convergence process is by using overrelaxation [6]. In this method, we use a fixed power ω of the update matrix determined with the pure LA algorithm at each site, instead of simply updating by the $A(n)$, as calculated in the last section. The exponent is taken $1 \le \omega \le 2$. Now one must determine the optimal parameters ω and number of "hits" which make the convergence fastest.

In practice, the ω power of the matrix $A(n)$ is obtained by truncating the binomial sum

$$A^{\omega}(n) = \sum_{m=0}^{\infty} \frac{\Gamma(\omega+1)}{\Gamma(\omega+1-m)m!}(A(n)-\mathbb{1})^m, \tag{12}$$

at its second term and projecting onto SU(3). This works because, as the gauge-fixing advances, the update matrix gets close to the identity matrix very fast. The projection onto SU(3) is done by using Eq. 4.27 of [7], in which one normalizes the first row of the matrix, then constructs the second row by a Gram-Schmidt orthogonalization procedure, and the last row is given by the cross product of the conjugated first and second rows.

2.4 Stochastic overrelaxation (SOR)

An alternative way of accelerating the convergence is by making stochastic updates [2, 3], with the aim of jumping away from regions of little variation of the global functional, Eq. 1. We update

[1] One could try to obtain analytically the SU(3) matrix which does maximize the functional locally using Lagrange multipliers. Parametrizing $A(n) = T(n)S(n)R(n)$, with T, R and S given by Eqs. 10, we can treat $\mathcal{E}[g(n)]$ as a function of the r_i, s_i and t_i, with the constraints $\sum_i r_i^2 = \sum_i s_i^2 = \sum_i t_i^2 = 1$ all at the same time. Applying the method here, one generates a complicated system of coupled non-linear equations which we were not able to solve analytically, even with the help of Mathematica®. A numerical solution which maximizes the local functional can be obtained in Mathematica® in this way and the iterative method presented here converges to this solution as the number of "hits" increase.

the gauge transformation using

$$A^{\mathrm{SRE}}(n) = \begin{cases} A^{\mathrm{LA}}(n) & \text{with probability} \quad 1-p, \\ \left(A^{\mathrm{LA}}(n)\right)^2 & \text{with probability} \quad p, \end{cases} \tag{13}$$

where $A^{\mathrm{LA}}(n)$ is the LA update and p is a tunable parameter to be optimized. Then, with probability $1-p$, the local functional goes to its maximum and with probability p, we hope to jump to another region, exploring better the space of possible gauge transformations without modifying the value of the functional too much. In fact, for SU(2) one can show that the value of the functional does not change at all when doing the "square" update (i. e. $\mathrm{Re}\,\mathrm{tr}[(A^{\mathrm{LA}})^2 g^{\mathrm{old}} h] = \mathrm{Re}\,\mathrm{tr}[g^{\mathrm{old}} h]$), by using the fact that $h(n)$ is proportional to an SU(2) matrix and the unitarity of $g(n)$.

For SU(3), we do not have $\mathrm{Re}\,\mathrm{tr}[(A^{\mathrm{LA}})^2 g^{\mathrm{old}} h] = \mathrm{Re}\,\mathrm{tr}[g^{\mathrm{old}} h]$. However, we can verify in practice that, even though the equality is not satisfied exactly, it is satisfied on average over many updates and the use of the same scheme as in Eq. 13 for SU(3) is much more efficient than simply using the LA algorithm[2]. The occasional decreases in the functional after the stochastic update are compensated by a higher mobility in the space of gauge transformations, as will be demonstrated in Section 3.

3. Analysis of the parameters and conclusion

An important aspect of the efficiency is the typical number of sweeps needed to fix the gauge. This needs to be as small as possible when comparing different versions of the algorithm. In order to do this, we considered the distribution of the sweeps needed to arrive at the gauge-fixing condition over a large enough sample of configurations. We analyzed the same 200 pure-gauge SU(3) configurations of 4^4, 8^4, 10^4 lattices and 100 configurations of a 16^4 lattice generated with a standard heat-bath algorithm, with different parameters of the gauge-fixing algorithms, as explained in the previous sections. As a quantity indicative of the typical number of sweeps needed to gauge-fix the configuration, we chose the median of the distribution. This is because a small number of configurations demand an anomalously high number of sweeps to gauge-fix, and this increases the value of the mean of the data, preventing it from representing the most typical values of sweeps. It is noteworthy to say that the results presented here were duplicated using different programs written by two of us, which provides an independent check. Let us note that some studies of efficiency of the algorithms, like the one presented in [5], consider only one or a handful of configurations and observe the behaviour of the number of sweeps to gauge-fix them when varying the parameters. By taking a substantial number of configurations, we hope to have a more robust result based on an actual statistical distribution.

As our first result, we note that not using OR or SOR makes the program slower to gauge-fix even if the parameters ω and p are not optimal. So, we will not consider the pure LA in the analyzes of this paper. We also verified that independently of β or the side of the lattice, the optimal number of "hits" is 2 for all versions of the algorithm. This can be seen, for example, in Fig. 1 for the SOR algorithm with the 4^4 and 8^4 configurations. Increasing the number of "hits" clearly matters when changing from 1 hit to 2 hits, which is to say that the typical number of sweeps drops significantly

[2] The LA algorithm can be thought of as OR with $\omega = 1$ or SOR with $p = 0$.

when p is in the optimal region. But more than 2 “hits” seem to make very little difference. The optimal region for the parameter p also does not change much when increasing the number of hits. We conclude, then, that making more than 2 “hits” is a waste of computational resources, since it will not translate into a smaller median of the distribution. Although we only show here the results for the two smallest lattices, the same conclusion was obtained for all the lattices and also for the OR version of the algorithm, so, from now on, we will only focus on the optimization for ω and p. To obtain a critical exponent from the data, which illustrates how much more work has to be done when the number of sites increases, we use configurations obtained with “constant physics”, i. e., keeping Na constant, where N is the number of lattice sites in any direction. Following [7, 8], the physical spacing is parametrized, for $5.7 < \beta < 6.92$, by

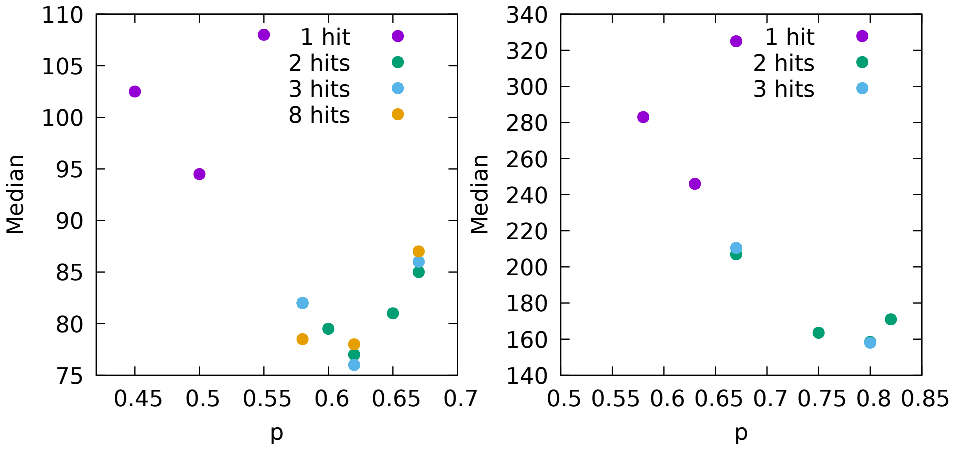

Figure 1: Optimization of # of “hits” for SOR on a 4^4 (left) and 8^4 (right) lattice.

$$a = r_0 \exp(f(\beta)), \quad f(\beta) \equiv -1.6804-1.7331(\beta-6.0)+0.7849(\beta-6.0)^2-0.4428(\beta-6.0)^3, \quad (14)$$

for the standard pure gauge Wilson plaquette action. The Sommer parameter is $r_0 \approx 0.5$ fm. The results for the median and our optimal parameters for gauge-fixing with configurations generated with β calibrated to keep the physical volume constant are shown in Fig. 2.

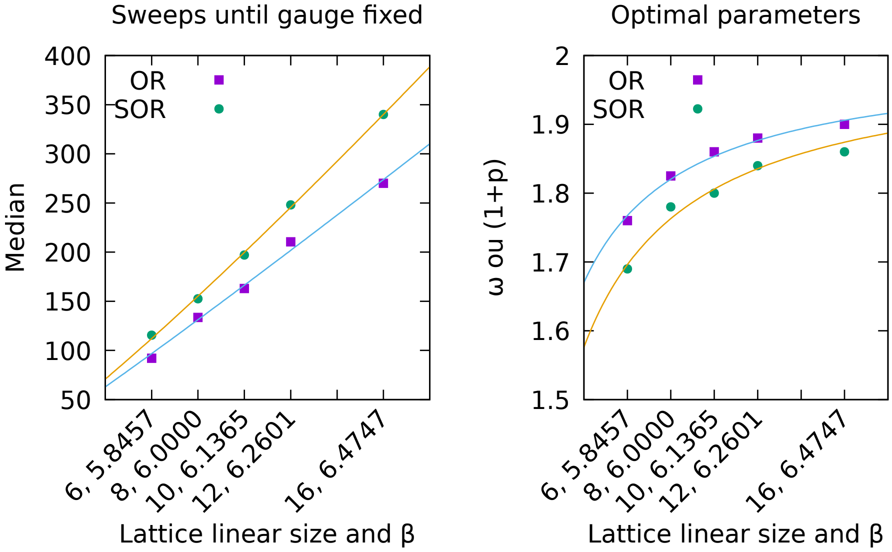

Figure 2: Left: Medians of the distribution of sweeps to arrive at a gauge-fixed configuration for OR and SOR with the optimal values of p and ω as a function of the lattice size and two "hits". Right: Optimal parameters as a function of the lattice size. The fits are explained in the text.

The fit for the median of the number of sweeps was done with a power law $f(N) = AN^z$, resulting in $A = 14(2)$ and $z = 1.06(5)$ for OR and $A = 14.7(8)$ and $z = 1.13(2)$ for SOR, which shows that OR is more efficient, given the smaller critical exponent. The fit shows that we get an almost linear dependence of the number of sweeps with the size of lattice side, much better than the quadratic dependence expected for pure LA [2]. For the dependence of the optimal parameters with the size of the lattice, we used a function ω or $(1 + p) = 2/(1 + C/N)$, based on [2], resulting in $C = 0.78(1)$ and $C = 1.07(3)$ for OR and SOR respectively. We observe that, as the number of sites grows, the values of the optimal parameters get closer to the limiting cases of $\omega = 2$ and $p = 1$, showing that the mobility in the space of possible gauge transformations is very important for an efficient gauge-fixing on large lattices.

Acknowledgments

Jesuel Marques has been supported by grant #2019/10913-0 of the São Paulo Research Foundation (FAPESP). Matheus Cerqueira acknowledges support of CNPq.

References

[1] G. 't Hooft, C. Itzykson, A. Jaffe, H. Lehmann, P. K. Mitter, I. M. Singer and R. Stora, *Recent Developments in Gauge Theories. Proceedings, Nato Advanced Study Institute, Cargese, France, August 26 - September 8, 1979*, *NATO Sci. Ser. B* **59** (1980), pp.1-438.

[2] A. Cucchieri and T. Mendes, *Critical slowing down in SU(2) Landau gauge fixing algorithms at beta = infinity*, *Comput. Phys. Commun.* **154** (2003), 1-48 [hep-lat/0301019].

[3] P. de Forcrand, *Multigrid Techniques for Quark Propagator*, *Nucl. Phys. B Proc. Suppl.* **9** (1989), 516-520.

[4] N. Cabibbo and E. Marinari, *A New Method for Updating SU(N) Matrices in Computer Simulations of Gauge Theories*, *Phys. Lett. B* **119** (1982), 387-390.

[5] H. Suman and K. Schilling, *A Comparative study of gauge fixing procedures on the connection machines CM2 and CM5*, [hep-lat/9306018].

[6] J. E. Mandula and M. Ogilvie, *Efficient gauge fixing via overrelaxation*, *Phys. Lett. B* **248** (1990), 156-158.

[7] C. Gattringer and C. B. Lang, *Quantum chromodynamics on the lattice*, Lect. Notes Phys. **788**, Springer 2010.

[8] S. Necco and R. Sommer, *The N(f) = 0 heavy quark potential from short to intermediate distances*, *Nucl. Phys. B* **622** (2002), 328-346, [hep-lat/0108008].