Monte Carlo Algorithms for Reduced Lattices, Mixed Actions, and Double-Trace Deformations

Hélvio Vairinhos

Centre for Computational Physics, Department of Physics, University of Coimbra, 3004-516 Coimbra, Portugal

Abstract

We construct efficient Monte Carlo updating algorithms for two classes of pure $SU(N)$ lattice gauge actions with non-linear dependence on the link variables. Our construction generalises the method of auxiliary variables used by Fabricius and Haan in the framework of Eguchi-Kawai models. We first review the original Fabricius-Haan method of constructing a pseudo-heatbath algorithm for fully reduced models, and discuss its extension to lattices with any number of reduced directions. We then use a similar method to construct updating algorithms for generic $SU(N)$ mixed Wilson actions. We construct explicit examples of algorithms for Wilson actions whose plaquettes are in an irreducible representation of $SU(N)$ with N-ality $k \leq 3$. We also construct updating algorithms for the lattice version of centre-stabilised $SU(N)$ Yang-Mills theories defined on $\mathbb{R}^{d-1} \times S^1$, including the case of a fully reduced compact direction. We simulate the new algorithms and show that they are, in general, significantly more efficient than their Metropolis counterparts.
# Contents

1 Introduction ................................. 1

2 Reduced lattices ............................ 5
   2.1 Fabricius-Haan method .................. 5
   2.2 Partially reduced lattices ............... 8

3 Mixed actions ............................... 9
   3.1 Linearisation of arbitrary characters .... 11
   3.2 Adjoint representation .................. 13
      3.2.1 \( \beta_A > 0 \) ...................... 14
      3.2.2 \( \beta_A < 0 \) ...................... 15
   3.3 Higher \( N \)-ality representations ......... 16
      3.3.1 \( R = 2 \pm \) ...................... 16
      3.3.2 \( R = 3 \pm \) ...................... 17
      3.3.3 \( R = (2, 1) \) ...................... 19
   3.4 Mixed Wilson actions .................... 21
   3.5 A note on reduced mixed models .......... 22
   3.6 Numerical tests .......................... 24

4 Double-trace deformations .................. 26
   4.1 CYM theory on \( \mathbb{R}^{d-1} \times S^1 \) .......... 26
   4.2 Numerical tests .......................... 30

5 Discussion ................................ 32

A Monte Carlo algorithms ..................... 34
   A.1 Reduced lattices ........................ 34
   A.2 Mixed actions ........................... 34
   A.3 Double-trace deformations ............... 35

References .................................. 38

## 1 Introduction

Simulations of pure lattice gauge theories normally involve the generation of Markov chains of gauge field configurations with the help of a local Monte Carlo algorithm. Such configurations (link variables on hypercubic lattices) are generated with respect to a probability distribution
characterised by the Boltzmann weight of the Euclidean partition function. Configurations are
generated locally, i.e. the link variables are updated individually while keeping the remaining
links fixed. A new configuration results from a sequence of local updates over the whole lattice
(one sweep). It is thus necessary to know the form of the Boltzmann probability distribution
restricted to only one link. The purpose of the local Monte Carlo (MC) algorithm is to sample
this distribution function efficiently. However, save in some special cases, the direct sampling of
the probability distributions of individual links is often impracticable. Therefore, it is essential
to find a good algorithm that samples such distributions efficiently, albeit indirectly. The ideal
MC algorithm should be ergodic, fast, and produce decorrelated data.

One straightforward possibility is to generate new link variables using a local Metropolis
algorithm \[1\]. Metropolis has the advantages of having an universal scope (it can sample any
probability distribution) and of being very easy to implement. However, it is usually slow,
decorrelates poorly, and may fail to be ergodic. Also, it requires the acceptance rates of new
link proposals to be tuned at optimal values (which must be at around 50\%, in order to avoid
‘too fast’ or ‘too slow’ an exploration of the configuration space). Therefore, despite the ad-
vantages, Metropolis is not a very efficient algorithm for simulating lattice gauge theories. In
many situations, however, it is the only known algorithm. Fortunately, for pure SU\((N)\) lattice
gauge theories with the standard Wilson action it is possible to construct a more efficient
alternative to the Metropolis algorithm.

Consider the standard SU\((N)\) Wilson action (up to an additive constant) with plaquettes
in the fundamental representation:

\[
S_F(\beta_F; [U]) = -\frac{\beta_F}{N} \sum_p \text{ReTr} \{ U_p \} \tag{1}
\]

Here \(p\) labels positively oriented plaquettes in the \(d\)-dimensional hypercubic lattice \(\Lambda\), \(U_p\) is
the plaquette operator, defined as the path-ordered product of all link variables \(U_{\mu,x} \in SU(N)\)
in the boundary of \(p\),

\[
U_p \equiv U_{\mu,x} U_{\nu,x+\hat{\mu}} U_{\mu,x+\hat{\nu}} U_{\nu,x} \tag{2}
\]

(\(\hat{\mu}\) denotes the unit lattice vector in the \(\mu\)-direction), \(\beta_F\) is the bare lattice coupling, and \(\text{Tr}\) de-
notes the character of the fundamental representation of \(SU(N)\). The probability distribution
of link configurations \([U]\) is Boltzmann:

\[
dP(\beta_F; [U]) = \mu_H[U] \exp (-S_F(\beta_F; [U])) / Z_F(\beta_F) \tag{3}
\]

where \(\mu_H\) denotes the product of the \(SU(N)\)-invariant Haar measures of all link variables,

\[
\mu_H[U] \equiv \prod_{x \in \Lambda} \prod_{\mu=1}^d dU_{\mu,x} \tag{4}
\]

and \(Z_F\) is the Euclidean partition function of the lattice gauge theory:

\[
Z_F(\beta_F) = \int \mu_H[U] \exp (-S_F(\beta_F; [U])) \tag{5}
\]
Because the Wilson action (1) is linear with respect to each link variable, the probability distribution of individual links is given by:

\[ dP(U_{\mu,x}) \propto dU_{\mu,x} \exp \left( \text{ReTr} \left\{ V^\dagger_{\mu,x} U_{\mu,x} \right\} \right) \]  

(6)

where \( V_{\mu,x} \) is the sum of all ‘staples’ that couple to \( U_{\mu,x} \) multiplied by the lattice coupling:

\[ V_{\mu,x} = \frac{\beta F N_d}{N} \sum_{\nu=1}^{d} \left( U_{\nu,x} U_{\mu,x+\nu} U^\dagger_{\mu,x+\nu} + U^\dagger_{\nu,x-\nu} U_{\mu,x-\nu} U_{\nu,x-\nu} - U^\dagger_{\nu,x} U_{\mu,x+\nu} - U_{\nu,x-\nu} U^\dagger_{\mu,x-\nu} \right) \]  

(7)

It is important to note that \( V_{\mu,x} \) does not (and should not) depend on \( U_{\mu,x} \).

In the SU(2) case, it is possible to perform a very efficient sampling of the probability distribution (6) [2, 3, 4]. It relies on the fact that any sum \( v \) of SU(2) matrices, e.g. (7), is always proportional to an SU(2) matrix. Because of this property, the probability distribution of individual SU(2) matrices is of the form:

\[ dP(u) \propto du \exp \left( \xi \text{ReTr} \left\{ (v\xi^{-1})^\dagger u \right\} \right) \]  

(8)

where \( u, v\xi^{-1} \in SU(2) \), \( \xi = \sqrt{\det(v)} \), and \( du \) is the SU(2)–invariant Haar measure. The SU(2)–projected sum of ‘staples’ \( v\xi^{-1} \) can be absorbed by the Haar measure. Consequently, the probability distribution (8) only depends on \( a_0 \equiv \text{Tr} \{ u \} \in [-1, 1] \):

\[ dP(a_0) \propto da_0 \left( 1 - a_0^2 \right)^{\frac{1}{2}} \exp \left( 2\xi a_0 \right) \]  

(9)

which can be sampled very efficiently. Since SU(2) is isomorphic to \( S^3 \), \( a_0 \) corresponds to a ‘latitude’ parameter. The traceless part of \( u \) is then uniformly distributed over the \( S^2 \) ‘equator’, i.e. with a probability distribution given by the solid angle \( d^2 \Omega \), which can be sampled trivially. The algorithms [2, 3, 4] that generate random SU(2) matrices with the probability distribution (8) are the heatbath algorithms.

There are two other very useful SU(2) algorithms for lattice gauge theory simulations: the overrelaxation and cooling algorithms. The overrelaxation algorithm [5] performs ‘large’ local changes to the gauge field configurations that keep the Boltzmann weight invariant. Because the sum \( v \) of SU(2) matrices is always proportional to an SU(2) matrix, an exact overrelaxation update of \( u \) is obtained with the proposal:

\[ u \mapsto u' = (v\xi^{-1}) \cdot u \cdot (v\xi^{-1}) \in SU(2) \]  

(10)

Using overrelaxation updates together with heatbath updates improves the efficiency of the overall MC algorithm, because (10) corresponds to take a large ‘step’ on SU(2) as compared with the typical heatbath ‘steps’. This allows a broader exploration of the configuration space around the local minimum probed by the simulation. On the other hand, the cooling algorithm [6] generates gauge field configurations that minimise the lattice action locally. Because the
sum $v$ of $SU(2)$ matrices is always proportional to an $SU(2)$ matrix, an exact cooling update of $u$ is obtained with the proposal:

$$u \mapsto u' = (v\xi^{-1}) \in SU(2)$$

(11)

The cooling of lattice gauge fields suppresses their quantum fluctuations. This is useful for the search of topological structures in the vacuum of the gauge theory, e.g. instantons.

For gauge groups larger than $SU(2)$, however, the situation is different: the sum of $SU(N)$ matrices is not proportional to an $SU(N)$ matrix, in general. Therefore, none of the algorithms described above can be directly extrapolated to $N > 2$. A way to circumvent this obstacle is to construct algorithms that only update the $SU(2)$ subgroups of $SU(N)$, and not the whole group \[7\]. Due to the lattice action (6) being linear with respect to individual links $U_{\mu,x}$, it is then possible to single out each of its $N(N-1)/2$ $SU(2)$ subgroup elements $u \subset U_{\mu,x}$, and obtain a distribution for $u$ that is of the form \[8\]. In this way, the $SU(2)$ algorithms described above can still be used. The update of all (or a subset\[4\] of all) $SU(2)$ submatrices of a link variable results in a new link that is compatible with the probability distribution (6). The algorithm that generates random $SU(N)$ matrices is known as Cabibbo-Marinari pseudo-heatbath algorithm \[7\]. Even though it doesn’t sample (6) directly, it performs much better than the Metropolis algorithm in all aspects: it provides faster equilibration times, smaller autocorrelations, and it doesn’t require any tuning. Because of all this, it has naturally become the standard algorithm for the generation of equilibrium configurations in pure lattice gauge theories\[2\].

In the same way, the overrelaxation and cooling of $SU(N)$ gauge configurations can be done by restricting such updates to the $SU(2)$ subgroups. Recently, algorithms for the overrelaxation of the full $SU(N)$ group, and not just of its $SU(2)$ subgroups, have been suggested \[9\]. They are based on the singular value decomposition of $V_{\mu,x}$, and are shown to perform better than the overrelaxation of $SU(2)$ subgroups. In the same line of thought, a cooling algorithm for the full $SU(N)$ group can easily be constructed using the singular value decomposition of $V_{\mu,x}$.

However, the $SU(N)$ updating algorithms discussed above only apply if the probability distribution of individual link variables is of the form (6), i.e. if the lattice action is linear with respect to each link. If not, the contribution to the partition function of individual $SU(2)$

---

1. The subset of $SU(2)$ subgroups must be such that the remaining subgroups do not generate a left ideal \[7\]. The minimal number of $SU(2)$ subgroups that must be updated is therefore $N - 1$, i.e. those of the form $u_{i,i+1}$ ($u_{ij}$ represents the $2 \times 2$ submatrix whose diagonal elements lie on the positions $i$ and $j$ of the diagonal of the $N \times N$ matrix it belongs to). Despite the large difference between the minimal number and the total number of $SU(2)$ subgroups, which becomes very significant for large $N$, it is recommended to update all of them. Updating too small a number of $SU(2)$ subgroups may result in a poor performance. In our simulations, we always update all the $N(N-1)/2$ $SU(2)$ subgroups of $SU(N)$.

2. An exact heatbath algorithm has been suggested for $SU(3)$ by Pietarinen \[8\]. However, it is hard to implement, and so the Cabibbo-Marinari pseudo-heatbath strategy has also been adopted as the standard heatbath algorithm for $SU(3)$.
subgroups is not of the form (8), and so the $SU(2)$ algorithms discussed above cannot be used. For the same reasons, the full $SU(N)$ overrelaxation/cooling algorithms cannot be used, too. In sum, if a lattice action is not linear with respect to the link variables, and no other efficient algorithms are known, then using a Metropolis algorithm is the only way to simulate such a theory.

In this paper, we construct MC updating algorithms for some $SU(N)$ lattice gauge actions whose dependence on the link variables is nonlinear. Our strategy is based on a generalisation of the Fabricius-Haan method of auxiliary variables [3]. We perform numerical simulations with the new algorithms and show that they are, in most cases, significantly more efficient then their Metropolis alternatives. In Section 2, we review the Fabricius-Haan method of constructing a pseudo-heatbath algorithm for the Eguchi-Kawai model. We then generalise their method, and apply it to the case of hypercubic lattices with any number of fully reduced directions. In Section 3, we apply similar methods to pure lattice gauge theories with a mixed Wilson action. We explicitly construct MC updating algorithms for Wilson actions whose plaquettes are in irreducible representations of $SU(N)$ with $N$–ality $k \leq 3$. We then study the numerical performance of these algorithms against Metropolis. In Section 4, we deal with the lattice regularisation of centre-stabilised $SU(N)$ Yang-Mills theories defined on $\mathbb{R}^3 \times S^1$. We explicitly construct MC an updating algorithm for these theories and study its numerical performance against Metropolis. We conclude our paper with a discussion on the advantages, efficiency and applicability of the new algorithms. We summarise in Appendix A all the algorithms proposed in this paper.

2 Reduced lattices

A situation where the probability distribution of individual links is not of the form (6) occurs already in pure lattice gauge theory with the standard Wilson action. If one or more lattice directions of a hypercubic lattice are reduced (i.e. have length $N_\mu = 1$ in lattice units, for some direction $\mu$), the Wilson action (1) becomes quadratic with respect to the link variables orthogonal to the reduced directions. Because of this, gauge field configurations on reduced lattices would have to be generated with a Metropolis algorithm. However, Fabricius and Haan [3] were able to construct an efficient pseudo-heatbath algorithm for the case of a fully reduced ($1^d$) lattice, also known as Eguchi-Kawai (EK) model [10].[3] They used a method of auxiliary variables, which we review below.

2.1 Fabricius-Haan method

The EK model [10] is the original proposal of a matrix model for the large $N$ limit of pure $SU(N)$ lattice Yang-Mills theories. Initially, it was believed that their planar sectors would coincide, but the spontaneous breaking of an important symmetry eliminated that hope [12].

[3] In fact, the pseudo-heatbath algorithm of Fabricius and Haan was originally constructed for $1^d$ lattices with twisted boundary conditions, also known as twisted Eguchi-Kawai (TEK) models [11]. We omit the twists factors in the lattice action because they are not relevant to our discussion.
Since then, a number of alternative reduced models have been suggested [12, 11, 13]. The goal is to find a zero-volume model of the full gauge theory in the large $N$ limit, in which the set of global symmetries relevant for the large $N$ equivalence stay intact for all values of the coupling. This is hard to achieve, however, because those symmetries are sensitive to the volume.

The action of the EK model is simply the fundamental Wilson action (1) on a $1^d$ lattice:

$$S_{\text{EK}}(\beta_F; [U]) = -\frac{\beta_F}{N} \sum_{\mu<\nu} \text{ReTr} \{ U_{\mu} U_{\nu} U_{\mu}^{\dagger} U_{\nu}^{\dagger} \}$$

(12)

Notice that the total absence of spacetime degrees of freedom in (12) makes the plaquette operators quadratic on the link variables. Therefore, the probability distribution of individual links cannot be put in the form of (6), because the sum of ‘staples’ (7) would not be independent of $U_{\mu}$. Consequently, none of the efficient updating algorithms discussed in Section 1 can be applied directly, and the Metropolis algorithm seems to be the only alternative to simulate the EK model.

Fabricius and Haan [3] circumvented this no-go by adding auxiliary degrees of freedom to the EK model. The new lattice variables are complex $N \times N$ matrices associated with the plaquettes, $\tilde{Q}_{\mu\nu} \equiv \tilde{Q}_{\nu\mu}, \forall \mu < \nu$. They are given a free dynamics completely decoupled from the gauge field. In other words, the spurious degrees of freedom $\tilde{Q}_{\mu\nu}$ are random matrices with the normal distribution, whose only effect is to multiply the partition function of the EK model by a constant factor:

$$Z_{\text{EK}}(\beta_F) \propto \int \mu_H[U] \exp (-S_{\text{EK}}(\beta_F; [U])) \times \int [d\tilde{Q}^\dagger d\tilde{Q}] \exp \left(-\frac{1}{2} \sum_{\mu<\nu} \text{Tr} \left\{ \tilde{Q}_{\mu\nu}^\dagger \tilde{Q}_{\mu\nu} \right\} \right)$$

(13)

$\mu_H[U]$ is the product of $SU(N)$–invariant Haar measures of all $d$ link variables, and $[d\tilde{Q}^\dagger d\tilde{Q}]$ is the product of standard flat measures for the auxiliary fields,

$$[d\tilde{Q}^\dagger d\tilde{Q}] \equiv \prod_{\mu<\nu} \prod_{a,b=1}^N d\text{Re}(\tilde{Q}_{\mu\nu})_{ab} d\text{Im}(\tilde{Q}_{\mu\nu})_{ab}$$

(14)

To simplify the notation, we use $\mu_\sigma$ to denote the Gaussian measure with variance $\sigma^2$ of a complex variable $z \in \mathbb{C}$,

$$\mu_\sigma(z) \equiv dz^*dz (2\pi\sigma^2)^{-1} \exp \left(-\frac{1}{2\sigma^2} |z|^2 \right)$$

(15)

or more generally, of a complex $N \times N$ matrix $A \in M(N, \mathbb{C})$,

$$\mu_\sigma(A) \equiv dA^\dagger dA (2\pi\sigma^2)^{-N^2} \exp \left(-\frac{1}{2\sigma^2} \text{Tr} \left\{ A^\dagger A \right\} \right)$$

(16)
In this notation, the Gaussian integral multiplying the EK partition function is:

$$\int \mu_1[\tilde{Q}] = 1 \quad (17)$$

where $\mu_1[\tilde{Q}] \equiv \prod_{\mu<\nu} \mu_1(\tilde{Q}_{\mu\nu})$.

The second step in the Fabricius-Haan method consists in performing a particular change of variables $(\tilde{Q}, U) \mapsto (Q, U)$ given by:

$$\tilde{Q}_{\mu\nu} = \left(\frac{\beta_F}{N}\right)^{\frac{1}{2}} (Q_{\mu\nu} - U_{\mu}U_{\nu} - U_{\nu}U_{\mu}) \quad (18)$$

This transformation keeps the integration measure $(14)$ invariant, up to a constant factor. It also cancels out the quadratic terms in the EK action, replacing them with Gaussian and linear terms. To see this, consider the effect of this change of variables on the exponent of the Gaussian term in $(13)$:

$$-\frac{1}{2} \sum_{\mu<\nu}^d \text{Tr} \{\tilde{Q}_{\mu\nu}^\dagger \tilde{Q}_{\mu\nu}\} \quad (19)$$

The change of variables produces Gaussian terms for the $Q_{\mu\nu}$, linear terms on the link variables, and quadratic terms that have the same functional form as the terms in $(12)$, but with opposite sign. Therefore, the quadratic terms in $(13)$ are cancelled out and only the Gaussian and linear terms survive. The partition function of the EK model with auxiliary variables then becomes:

$$Z_{\text{EK}}(\beta_F) = \int \mu_H[U] \mu_{\sigma}[Q] \exp \left(\frac{\beta_F}{N} \sum_{\mu<\nu}^d \text{ReTr} \{Q_{\mu\nu}^\dagger U_{\mu}U_{\nu}\} \right) \quad (20)$$

with $\sigma^2 = N/\beta_F$. Since the exponent of the Boltzmann factor in $(20)$ is linear with respect to each link variable, the Cabibbo-Marinari pseudo-heatbath can be used to update them. The update is performed with respect to the probability distribution of individual links $(6)$, where $U_{\mu,x} \equiv U_{\mu}$, and $V_{\mu,x} \equiv V_{\mu}$ is analogous to the sum of ‘staples’ $(7)$:

$$V_{\mu} = \sum_{\nu=1}^d \left(U_{\nu}^\dagger Q_{\mu\nu} + Q_{\mu\nu} U_{\nu}^\dagger\right) \quad (21)$$
Note that $V_\mu$ does not depend on $U_\mu$. In essence, the change of variables in (18) is an example of a Hubbard-Stratonovich transformation [14].

Despite the apparent coupling between the gauge and auxiliary degrees of freedom in (20), the gauge dynamics is completely unaffected by the presence of the auxiliary fields: the partition function (20) can always be transformed back to its original form (13) via the inverse of the non-singular transformations (18). Another apparent paradox resides in the fact that we are actually increasing the number if degrees of freedom that need to be updated in numerical simulations. However, all components $(\tilde{Q}_{\mu\nu})_{ab}$ of the auxiliary fields are independent normally-distributed complex numbers. These can be generated very fast with a known algorithm, like the Box-Müller transform.

Therefore, the time needed to update the auxiliary variables is negligible as compared with the time needed to update the link variables. Consequently, the Fabricius-Haan method for the EK model is actually faster than its Metropolis alternative, given in [16].

2.2 Partially reduced lattices

The Fabricius-Haan method for the EK model can easily be generalized to lattices with any number of reduced directions. In such lattices, only those plaquettes that are parallel to at least one reduced direction are quadratic with respect to the link variables. We call them reduced plaquettes. The remaining unreduced plaquettes are linear, hence the respective action terms do not need to be replaced. It then suffices to introduce one normally-distributed $N \times N$ complex matrix per reduced plaquette, i.e. $\tilde{Q}_{\mu\nu,x} \equiv \tilde{Q}_{\nu\mu,x}$ such that $N_\mu = 1$ or $N_\nu = 1$. The partition function (5) of the partially reduced lattice gauge theory is then multiplied by the Gaussian integral over those auxiliary variables, $\mu_1[\tilde{Q}]$, which again amounts to multiply the partition function by ‘1’. With the following change of variables:

$$\tilde{Q}_{\mu\nu,x} = \left(\frac{\beta_F}{N}\right)^{\frac{1}{2}} (Q_{\mu\nu,x} - U_{\mu,x}U_{\nu,x+\hat{\mu}} - U_{\nu,x}U_{\mu,x+\hat{\nu}})$$

all action terms involving reduced plaquettes are eliminated, and the partition function (5) becomes:

$$Z_F(\beta_F) = \int \mu_H[U] \mu_\sigma[Q] \exp\left(-S'_{\beta_F}(\beta_F; [Q, U])\right)$$

4 The Box-Müller transform [15] is a method for generating pairs of independent random real numbers $(g_1, g_2)$ with normal distribution $dP(g_1, g_2) = dq_1dq_2 \left(2\pi\sigma^2\right)^{-1} \exp\left(-(g_1^2 + g_2^2)/2\right)$ from a pair of random real numbers $(u_1, u_2)$ uniformly distributed over the unit interval $(0,1]$. The pair $(g_1, g_2)$ can also be understood as a random complex number $z = g_1 + ig_2$ with normal distribution over the complex plane $dP(z) = dz^*dz \left(2\pi\sigma^2\right)^{-1} \exp\left(-|z|^2/2\right)$. The Gaussian complex number $z$ is generated from the pair $(u_1, u_2)$ via the Box-Müller transform:

$$z = (-2 \ln u_1)^{\frac{1}{2}} \exp\left(i2\pi u_2\right)$$

5 An example of the Fabricius-Haan method applied to partially reduced lattices is used in [17].
where \( \sigma^2 = N/\beta_F \), and \( S'_F \) is the ‘linearised’ action:

\[
S'_F(\beta_F; [Q, U]) = -\frac{\beta_F}{N} \sum_{x \in \Lambda} \sum_{\mu < \nu}^{d} \Re \text{Tr}\{U_{\mu\nu,x}\}
- \frac{\beta_F}{N} \sum_{x \in \Lambda} \sum_{\mu < \nu}^{d} \Re \text{Tr}\{Q_{\mu\nu,x}^\dagger (U_{\mu,x}U_{\nu,x+\bar{\mu}} + U_{\nu,x}U_{\mu,x+\bar{\nu}})\}
\] (25)

The contribution from the unreduced plaquettes is unchanged, because those terms are already linear in the link variables. On the other hand, the quadratic contributions coming from the reduced plaquettes are replaced by the linear terms involving auxiliary variables. The Cabibbo-Marinari pseudo-heatbath can then be used to update the link variables. The updates are performed with respect to the probability distribution (6) of individual links, where the sum of ‘staples’ \( V_{\mu,x} \) is given by:

\[
V_{\mu,x} = \frac{\beta_F}{N} \sum_{\nu=1}^{d} \left( U_{\nu,x}U_{\nu,x+\nu}^\dagger + U_{\nu,x}^\dagger U_{\nu,x-\nu}U_{\nu,x-\nu+\nu} + Q_{\mu\nu,x}^\dagger U_{\nu,x} + U_{\nu,x}^\dagger Q_{\mu\nu,x} \right)
\] (26)

The Fabricius-Haan algorithm for partially reduced lattices is summarised in the Appendix A.1. The Fabricius-Haan algorithm for the EK model is contained as the special case for which the lattice is fully reduced.

3 Mixed actions

A suitable action for a \( SU(N) \) lattice gauge theory must be gauge-invariant and converge to the \( SU(N) \) Yang-Mills action in the continuum limit. It is well known \[11\] that class functions\[6\] on \( SU(N) \) can be used to construct lattice actions that satisfy the conditions above. Examples of class functions are the characters of the irreducible representations \( \mathcal{R} \) of \( SU(N) \). These are used to define an important class of lattice actions, namely Wilson actions whose plaquettes are in different representations of \( SU(N) \):

\[
S_{\mathcal{R}}(\beta_{\mathcal{R}}; [U]) = -\frac{\beta_{\mathcal{R}}}{d_{\mathcal{R}}} \sum_p \Re \chi_{\mathcal{R}} \{U_p\}
\] (27)

\[6\] A class function is a function \( f \) defined on a group \( G \) that is constant on the conjugacy classes of \( G \), i.e. \( f(ghg^{-1}) = f(h), \forall g, h \in G \). Here, \( g \) corresponds to a gauge transformation at a lattice site, and \( h \) corresponds to a lattice operator. Therefore, the definition of class function corresponds to the statement that \( f \) is gauge-invariant.
where $\chi_R$ is the character of the representation $\mathcal{R}$, $d_R = d_R \equiv \chi_R(1)$ is its dimension, and $\beta_R$ is its associated bare lattice coupling. Equally suitable are the lattice actions consisting of arbitrary linear combinations of irreducible characters, known as mixed Wilson actions:

$$S_{\text{mix}}(\vec{\beta}; [U]) = \sum_{\mathcal{R}} S_{\mathcal{R}}(\beta_R; [U])$$

(28)

where $\vec{\beta}$ denotes the set of independent lattice couplings $\beta_R$.

All irreducible characters of $SU(N)$ can be expressed in terms of the character of the fundamental representation, $\chi_F \equiv \text{Tr}$. In this paper we only consider explicitly those representations of $SU(N)$ with $N$–ality $k \leq 3$, whose characters are given by:

$$\chi_F(U) = \text{Tr}\{U\}$$

(29)

$$\chi_A(U) = |\text{Tr}\{U\}|^2 - 1$$

(30)

$$\chi_{(2)}(U) = \frac{1}{2} (\text{Tr}\{U\}^2 + \text{Tr}\{U^2\})$$

(31)

$$\chi_{(1,1)}(U) = \frac{1}{2} (\text{Tr}\{U\}^2 - \text{Tr}\{U^2\})$$

(32)

$$\chi_{(3)}(U) = \frac{1}{6} (\text{Tr}\{U\}^3 + 3\text{Tr}\{U\} \text{Tr}\{U^2\} + 2\text{Tr}\{U^3\})$$

(33)

$$\chi_{(1,1,1)}(U) = \frac{1}{6} (\text{Tr}\{U\}^3 - 3\text{Tr}\{U\} \text{Tr}\{U^2\} + 2\text{Tr}\{U^3\})$$

(34)

$$\chi_{(2,1)}(U) = \frac{1}{3} (\text{Tr}\{U\}^3 - \text{Tr}\{U^3\})$$

(35)

Here $U$ is a generic element of $SU(N)$, and the label in the characters denotes the Young tableau of a representation (e.g. $(2) \equiv \square$, $(1,1) \equiv \square$, etc.); $F$ denotes the fundamental representation, and $A$ denotes the adjoint representation.

It is clear that only the fundamental Wilson action is linear with respect to link variables. Consequently, the efficient updating algorithms discussed in Section 1 cannot be used directly in different representations. For this reason, all numerical studies of mixed Wilson actions have been performed using algorithms based on Metropolis, like the Cabibbo-Marinari-Metropolis and the overrelaxation-Metropolis algorithms [19], or the biased Metropolis algorithm for the $SU(2)$ fundamental/adjoint action [20]. However, it is possible to imagine that a generalisation of the Fabricius-Haan method could also be applied to mixed Wilson actions, and in that way evade the nonlinear terms coming from the $SU(N)$ characters.

The philosophy behind the Fabricius-Haan method consists in adding a minimal (but sufficient) number of normally-distributed auxiliary fields, together with an appropriate change of variables, in order to eliminate all the nonlinear terms in the lattice action. These are then
replaced by linear terms on the link variables and Gaussian terms for the auxiliary fields. The possibility of using the efficient updating algorithms of Section [1] follows immediately. We argue that, by choosing a fair number of auxiliary variables and the correct transformation rules, it is ideally possible to eliminate all nonlinear plaquette terms in arbitrary mixed Wilson actions (and eventually in any lattice action whose dependence on the plaquettes is polynomial). Some of the auxiliary fields may eventually generate new nonlinear terms that also need to be eliminated. But as long as these are ‘less nonlinear’ than the terms they replace (e.g. by depending on smaller powers of the plaquette), the total amount of ‘nonlinearity’ is reduced, the process eventually stops, and all nonlinear terms are eliminated after a sufficient number of auxiliary fields is introduced. The objective is then to keep the number of necessary auxiliary fields at a minimum. We will make these statements more precise below.

3.1 Linearisation of arbitrary characters

For each representation $\mathcal{R}$ of $SU(N)$, we wish to linearise the corresponding Wilson action \[27\] with respect to the plaquette operator, $U_p$. By ‘linearising’ the lattice action we mean replacing its nonlinear terms with linear and Gaussian terms, after adding a sufficient number $n_\mathcal{R}$ of normally-distributed auxiliary variables. Effectively, this linearisation corresponds to a Hubbard-Stratonovich transformation of the corresponding partition function.

Let us associate to each positively oriented plaquette on the lattice $n_\mathcal{R}$ complex $N \times N$ matrix variables $\tilde{Q}^{(i)}_{\mu\nu,x}$ ($i = 1, \ldots, n_\mathcal{R}$, and $\mu < \nu$) with the normal distribution $\mu_1(\tilde{Q}^{(i)}_{\mu\nu,x})$. For the negatively oriented plaquettes we define $\tilde{Q}^{(i)}_{\nu\mu,x} \equiv \tilde{Q}^{(i)\dagger}_{\mu\nu,x}$. First, we multiply the partition function of the mixed lattice gauge theory by $\int \mu_1[\tilde{Q}] = 1$, where $\mu_1[\tilde{Q}]$ is the product of Gaussian measures of all auxiliary variables. We then make a change of variables of the form:

$$\tilde{Q}^{(i)}_{p} = \sqrt{2\beta'_{\mathcal{R}}}(Q^{(i)}_{p} - h^{(i)}_{p}) \quad (36)$$

Here $p \equiv (\mu\nu, x)$ labels plaquettes, $\beta'_{\mathcal{R}} = \beta_{\mathcal{R}} N^\alpha / d_{\mathcal{R}}$ is a redefinition of the lattice coupling,$^{\text{7}}$ $Q^{(i)}_p$ are the transformed auxiliary variables, and $h^{(i)}_{p} \equiv h^{(i)}(Q^{(j)}_p; U_p)$ are functions of the plaquette $U_p$ and of a single auxiliary variable $Q^{(j)}_p$ with $j < i$. The condition $j < i$, whose origin becomes clear below, ensures that the change of variables $(\tilde{Q}, U) \mapsto (Q, U)$ is non-singular, hence invertible. In fact, its Jacobian is just a non-zero constant. In other words, the integration measure of the HS-transformed partition function is invariant under such a change of variables, up to a multiplicative constant. After the change of variables (36), all nonlinear terms coming from the Gaussian measure $\mu_1[\tilde{Q}]$ must have been exactly cancelled by the nonlinear terms in the Wilson action $[27]$. In the end, only the linear terms on $U_p$ and the Gaussian terms for $Q^{(i)}_p$ must survive.

The way we choose suitable $h$’s is the most straightforward possible. Basically, we associate one auxiliary variable $\tilde{Q}^{(j)}_p$ to each nonlinear term of $\chi_{\mathcal{R}}$. In order to eliminate them, we perform a change of variables $\tilde{Q}^{(j)}_p \rightarrow Q^{(j)}_p$ using a $h^{(j)}$ that only depends on $U_p$, i.e. $h^{(j)}_p \equiv h^{(j)}(U_p)$. If the

\(^{7}\) $\alpha$ is an integer often equalling the degree of $d_{\mathcal{R}}$ as a polynomial in $N$ minus one. See Table [1] for the definitions of $\beta_{\mathcal{R}}$ for each particular representation.
Given (38), the linear term on the r.h.s. of (39) can be rearranged as follows: 

\[ h^{(i)}(Q^{(j)}_p, U_p) = \bar{h}^{(i)}(Q^{(j)}_p, U_p) + h^{(i)}_{\text{nlin}}(U_p) \]  

(37)

The linear piece \( \bar{h}^{(i)} \), in its most general form, is given by:

\[ \bar{h}^{(i)}(Q^{(j)}_p, U_p) = A_1 U_p + A_2 U_p^\dagger + A_3 \text{Tr} \{ A_4 U_p \} + A_5 \text{Tr} \{ A_6 U_p^\dagger \} \]  

(38)

where the \( A_i \equiv A_i(Q^{(j)}_p) \) are complex \( N \times N \) matrices depending solely on one \( Q^{(j)}_p \), with \( j < i \). The change of variables (36) has the following effect on the Gaussian measures \( \mu_1(Q^{(i)}_p) \):

\[
\frac{1}{2} \left( \bar{Q}^{(i)}_p \right) \bar{Q}^{(i)}_p \stackrel{(36)}{=} -\beta R \text{Tr} \{ Q^{(i)}_p^\dagger Q^{(i)}_p \} + 2\beta R \text{ReTr} \{ Q^{(i)}_p^\dagger h^{(i)} \} - \beta R \text{Tr} \{ h^{(i)}_p h^{(i)}_p \}
\]

\[
= -\beta R \text{Tr} \{ Q^{(i)}_p^\dagger Q^{(i)}_p \} + 2\beta R \text{ReTr} \{ Q^{(i)}_p^\dagger \bar{h}^{(i)} \} + S_{\text{nlin}}(U_p)
\]  

(39)

In the expression above, \( S_{\text{nlin}} \) collects all the nonlinear terms generated by \( h^{(i)} \). We hypothesise that \( S_{\text{nlin}} \) is either the symmetric of the sum of some nonlinear terms of \( \chi_R \), or they are secondary nonlinear terms that can be cancelled out with the addition of more auxiliary variables. In other words, we assume that a linearisation of the Wilson action is possible. Given (38), the linear term on the r.h.s. of (39) can be rearranged as follows:

\[ 2\beta R \text{ReTr} \{ Q^{(i)}_p^\dagger \bar{h}^{(i)}_p \} \stackrel{(38)}{=} 2\beta R \text{ReTr} \{ g^{(i)}_p U_p \} \]  

(40)

where \( g^{(i)}_p \) are complex \( N \times N \) matrices that only depend on the auxiliary fields:

\[ g^{(i)}(Q^{(j)}) = A^\dagger_1 Q^{(i)}_p + Q^{(i)}_p A^\dagger_2 + A^\dagger_3 \text{Tr} \{ A^\dagger_4 Q^{(i)}_p \} + A_6 \text{Tr} \{ A_5 Q^{(i)}_p \} \]  

(41)

and \( A_i \equiv A_i(Q^{(j)}_p), \ j < i \). After all \( n_R \) auxiliary variables are added and the change of variables (36) is performed, the exponent in the Gaussian measure \( \mu_1(\bar{Q}) \) becomes:
\[-\frac{1}{2} \sum_{i=1}^{n_R} \sum_{p} \text{Tr} \left\{ \tilde{Q}_p^{(i)\dagger} \tilde{Q}_p^{(i)} \right\} \]

\[\text{Gaussian} \rightarrow - \beta_R \sum_{i=1}^{n_R} \sum_{p} \text{Tr} \{Q_p^{(i)} Q_p^{(i)}\} \]

\[\text{linear} \rightarrow + 2 \beta'_R \sum_{p} \text{ReTr} \{f_R^p U_p\} \]

\[\text{cancels out} \rightarrow - \beta_R' \sum_{p} \text{Re} \chi_R \{U_p\} \quad (42)\]

where \(f_R^p\) is a complex \(N \times N\) matrix that only depends on the auxiliary variables:

\[f_R^p \equiv \sum_{i=1}^{n_R} g_p^{(i)} \quad (43)\]

In sum, \(f_R^p\) encodes the nonlinear dependence of \(\chi_R\) on \(U_p\).

Once the Wilson action (27) is linearised, the efficient updating algorithms discussed in Section 1 can be applied directly. The updates are performed with respect to the probability distribution of individual links (6), where \(V_{\mu,x} \equiv V_{\mu,x}^R\) is the sum of ‘staples’:

\[V_{\mu,x}^R = 2 \beta'_R \sum_{\nu=1}^{d} \left( f_{\mu,\nu,x}^R U_{\nu,\nu+\bar{\mu}} + U_{\nu,x-\bar{\nu}} f_{\nu,\mu,x-\bar{\nu}}^R U_{\mu,x-\bar{\nu}} U_{\nu,x-\bar{\nu}+\bar{\mu}} \right) \quad (44)\]

and \(f_{\mu,\nu,x}^R \equiv f_{\nu,\mu,x}^R, \forall \mu < \nu\). It must be noted that the order in which \(f_R^p\) appears in the ‘staples’ depends on the particular choice for the initial point of the plaquette operator (2). Changing the initial point of the plaquette to any other vertex would simply result in a relocation of \(f_R^p\) within the ‘staples’. For the particular case when \(R = F\) we have \(f_F^p = 1\), so we have recovered the original algorithm.

In the next Sections we linearise explicitly the Wilson action for each representation of \(SU(N)\) with \(N\)-ality \(k \leq 3\); Table 1 summarises the changes of variables necessary for each linearisation. We then discuss the case of mixed actions, with and without reduced directions. In the end, we perform numerical tests on some of these algorithms against Metropolis, in order to check their accuracy and performance.

### 3.2 Adjoint representation

The best studied example of a mixed Wilson action is the one involving both fundamental and adjoint plaquette terms,

\[S_{F+A}(\beta_F, \beta_A; [U]) = S_F(\beta_F; [U]) + S_A(\beta_A; [U]) \quad (45)\]
The parameter space of \((\beta_F, \beta_A)\) lattice couplings is known as the fundamental/adjoint plane. This action is important, for example, in the study of the role of centre of the gauge group \(Z_N\) in colour confinement, because the adjoint character \(\chi_A(U_p)\) is invariant under centre shifts, \(U_p \mapsto zU_p, z \in Z_N\).

Due to the quadratic nature of \(\chi_A\) \((30)\), it is not possible to use the efficient updating algorithms of the fundamental Wilson action, discussed in Section 1. A Metropolis algorithm is often used instead. In particular, combining a Cabibbo-Marinari-Metropolis algorithm with overrelaxation-Metropolis sweeps results in a rather good performance, at least for \(SU(3)\) \([19]\). Recently, a biased Metropolis algorithm with heatbath efficiency has been constructed for the \(SU(2)\) case \([20]\). In this Section we construct a pseudo-heatbath algorithm for the \(SU(N)\) adjoint Wilson action,

\[
S_A(\beta_A; [U]) = -\frac{\beta_A}{N^2 - 1} \sum_p |\text{Tr}\{U_p\}|^2
\]  

for all \(N\), using the method of auxiliary variables. But because the double-trace term is always positive, the cases for positive and negative adjoint coupling \(\beta_A\) must be considered separately.

### 3.2.1 \(\beta_A > 0\)

When \(\beta_A\) is positive, the adjoint Wilson action \((46)\) is always negative. In order to cancel out the double-trace terms, it suffices to introduce one complex number per plaquette \(\tilde{z}_p\) with the normal distribution \(\mu_1(\tilde{z}_p)\). We then multiply the partition function of the adjoint Wilson theory \(Z_A(\beta_A)\) by the Gaussian integral \(\int \mu_1[\tilde{z}]=1\). Consider the change of variables:

\[
\tilde{z}_p = \sqrt{2\beta'_A} \left( z_p - \frac{1}{N} \text{Tr}\{U_p\} \right)
\]  

where \(\beta'_A = \beta_A N^2 / (N^2 - 1)\). The effect of \((47)\) on the Gaussian exponent of \(\mu_1[\tilde{z}]\) is:

\[
-\frac{1}{2} \sum_p |\tilde{z}_p|^2 \quad \Rightarrow \quad -\beta'_A \sum_p |z_p|^2
\]

\[
\text{Gaussian} \quad \Rightarrow \quad -\beta'_A \sum_p |z_p|^2
\]

\[
\text{linear} \quad \Rightarrow \quad +\frac{2\beta'_A}{N} \sum_p \text{ReTr} \{z_p^* U_p\}
\]

\[
\text{cancels out} \quad \Rightarrow \quad -\frac{\beta'_A}{N^2} \sum_p |\text{Tr}\{U_p\}|^2
\]  

The last term in the r.h.s. of \((48)\) cancels out all the adjoint terms of \((46)\) exactly, as long as \(\beta_A\) is positive. It comes from conjugating \(h^A_p = \text{Tr}\{U_p\} / N\) with itself. If \(\beta_A < 0\), then it is not possible to do the same with \(z\)-variables only. We discuss this case later. The partition function of the adjoint Wilson theory then becomes:

\[
Z_A(\beta_A) = \int \mu_1[U] \mu_{\sigma}[z] \exp \left( \frac{2\beta'_A}{N} \sum_p \text{ReTr} \{z_p^* U_p\} \right)
\]  

14
where $\mu_\sigma[z] \equiv \prod_p \mu_\sigma(z_p)$ and $\sigma^2 = 1/\beta_A'$. Since the exponent of the Boltzmann factor is now linear, the link variables can be updated using the efficient algorithms of Section 1. The updates are performed with respect to the probability distribution of individual links $[5]$, where $V_{\mu,x} \equiv V^A_{\mu,x}$ is given by (44) with $f^A_p = z_p/N$. And because $z_p$ and $f^A_p$ are just complex numbers, the cost of updating the auxiliary variables in this case is negligible as compared with link updates.

### 3.2.2 $\beta_A < 0$

In the case of negative $\beta_A$, the adjoint Wilson action (46) is positive. Therefore, the double-trace terms cannot be eliminated using (47). However, it is possible to linearise the action if we introduce different auxiliary variables, namely one complex $N \times N$ matrix per plaquette $\tilde{Q}_p$ with the normal distribution $\mu_1(\tilde{Q}_p)$. As usual, we multiply the partition function $Z_A(-|\beta_A|)$ with the Gaussian integral $\int \mu_1[\tilde{Q}] = 1$. Then we consider the change of variables:

$$\tilde{Q}_p = \sqrt{2\beta_A'} \left( Q_p - \left( U_p - \frac{1}{N} \text{Tr} \{ U_p \} \mathbb{1} \right) \right)$$  \hspace{1cm} (50)

where $\beta_A' = |\beta_A| N/(N^2 - 1)$. The effect of (50) on the Gaussian exponent of $\mu_1[\tilde{Q}]$ is:

- $\frac{1}{2} \sum_p \text{Tr} \left\{ \tilde{Q}_p^\dagger \tilde{Q}_p \right\}$  \hspace{1cm} (50)
  - Gaussian $\rightarrow -\beta_A' \sum_p \text{Tr} \{ Q_p^\dagger Q_p \}$
  - linear $\rightarrow +2\beta_A' \sum_p \text{ReTr} \left\{ (Q_p - \frac{1}{N} \text{Tr} \{ Q_p \} \mathbb{1} )^\dagger U_p \right\}$
  - cancels out $\rightarrow +\frac{\beta_A'}{N} \sum_p |\text{Tr} \{ U_p \}|^2$  \hspace{1cm} (51)

The last term in the r.h.s. of (51) carries the correct sign to cancel out the double-trace terms with negative $\beta_A$ from (46). The partition function of this theory then becomes:

$$Z_A(\beta_A) = \int \mu_H[U] \mu_\sigma[Q] \exp \left( 2\beta_A' \sum_p \text{ReTr} \left\{ (Q_p - \frac{1}{N} \text{Tr} \{ Q_p \} \mathbb{1} )^\dagger U_p \right\} \right)$$  \hspace{1cm} (52)

where $\mu_\sigma[Q] \equiv \prod_p \mu_\sigma(Q_p)$, and $\sigma^2 = 1/\beta_A'$. Individual links are then updated with respect to the probability distribution (6), with $V_{\mu,x} \equiv V^A_{\mu,x}$ given by (44), and $f^A_p$ is the matrix factor:

$$f^A_p = Q_p - \frac{1}{N} \text{Tr} \{ Q_p \} \mathbb{1}$$  \hspace{1cm} (53)

Since $Q_p$ and $f^A_p$ are $N \times N$ matrices, and not just complex numbers, it is expected for the $\beta_A < 0$ case to be less efficient than the $\beta_A > 0$ case.
3.3 Higher $N$-ality representations

The case of higher $N$-ality representations is more complicated, because there are more nonlinear terms to be eliminated, and these depend on higher powers of the link variables. This means that a larger number of auxiliary variables is needed in order to eliminate all nonlinear terms. In the following, we explicitly linearise the Wilson actions for two- and three-index irreducible representations of $SU(N)$ using the method of auxiliary variables.

3.3.1 $R = 2^\pm$

Here we consider Wilson actions with plaquettes in the symmetric $(+)$ or antisymmetric $(-)$ two-index representations of $SU(N)$, respectively $R = (2)$ or $R = (1, 1)$:

$$S_{\pm}(\beta; [U]) = -\frac{\beta_{\pm}}{N(N \pm 1)} \sum_p \text{Re} \left( \text{Tr}\{U_p\}^2 \pm \text{Tr}\{U_p^2\} \right)$$

We consider both situations of positive or negative lattice coupling $\beta_{\pm}$, whose sign we denote by $\sigma \equiv \text{sgn}(\beta_{\pm})$. One way of linearising (54) requires the addition of three auxiliary complex matrix variables per plaquette $\tilde{Q}^{(i)}_p$, $i = 1, 2, 3$, and the corresponding change of variables:

$$\tilde{Q}^{(1)}_p = \sqrt{2\beta_{\pm}} \left( Q^{(1)}_p - \frac{1}{2} \left( U_p + \frac{\sigma}{N} \text{Tr}\{U_p^\dagger\} \right) \right)$$

$$\tilde{Q}^{(2)}_p = \sqrt{2\beta_{\pm}} \left( Q^{(2)}_p - \frac{1}{2} \left( U_p \pm \frac{\sigma}{N} U_p^\dagger \right) \right)$$

$$\tilde{Q}^{(3)}_p = \sqrt{2\beta_{\pm}} \left( Q^{(3)}_p - \frac{1}{2} \left( U_p - \frac{1}{N} \text{Tr}\{U_p\} \right) \right)$$

where $\beta_{\pm} = 2|\beta_{\pm}|/(N \pm 1)$. The variable $Q^{(1)}_p$ is responsible for the elimination of the quadratic term $\text{Tr}\{U_p\}^2$ in the action, and its replacement by a linear term. However, it also generates a secondary nonlinear term of the form $|\text{Tr}\{U_p\}|^2$. The variable $Q^{(2)}_p$ eliminates the quadratic term $\text{Tr}\{U_p^2\}$ in the action and replaces it with a linear term, with no other side effects. Finally, the variable $Q^{(3)}_p$ eliminates the term $|\text{Tr}\{U_p\}|^2$ generated by $Q^{(1)}_p$. This is achieved using the transformation (57), which is similar to the one used in the elimination of negative-coupling adjoint plaquettes (50). The remaining linear terms contribute to the sum of 'staples' (44) with the matrix factor:

$$f^\pm_p = \frac{1}{2} \left( Q^{(1)}_p + \frac{\sigma}{N} \text{Tr}\{Q^{(1)}_p^\dagger\} \right) + Q^{(2)}_p \pm \frac{\sigma}{N} Q^{(2)}_p^\dagger + Q^{(3)}_p - \frac{1}{N} \text{Tr}\{Q^{(3)}_p\}$$

(58)

However, the above choice is not unique. It is also possible to linearise the Wilson action (54) by adding only two auxiliary complex matrix variables per plaquette. The corresponding
change of variables is:

\[
\tilde{Q}_p^{(1)} = \sqrt{2\beta_\pm} \left( Q_p^{(1)} - \frac{1}{2} \left( \sigma U_p^\dagger \pm \frac{1}{N} U_p + \frac{1}{N} \text{Tr} \{U_p\} \mathbb{1} \right) \right) \\
\tilde{Q}_p^{(2)} = \sqrt{2\beta_\pm} \left( Q_p^{(2)} - \left( \frac{N \pm 2}{4N} \right)^{\frac{1}{2}} \left( U_p - \frac{1}{N} \text{Tr} \{U_p\} \mathbb{1} \right) \right)
\]

(59) (60)

The variable \(Q_p^{(1)}\) is responsible for the simultaneous elimination of both nonlinear terms from \(\chi_{\pm}\). However, it also generates a \(|\text{Tr}\{U_p\}|^2\) term, which is promptly eliminated by variable \(Q_p^{(2)}\). After some algebra, we find the contribution of this change of variables to the sum of ‘staples’ (44) to be:

\[
f_p^\pm = \frac{1}{2} \left( \sigma Q_p^{(1)^\dagger} \pm \frac{1}{N} Q_p^{(1)} + \frac{1}{N} \text{Tr} \{Q_p^{(1)}\} \mathbb{1} \right) + \left( \frac{N \pm 2}{4N} \right)^{\frac{1}{2}} \left( Q_p^{(2)} - \frac{1}{N} \text{Tr} \{Q_p^{(2)}\} \mathbb{1} \right)
\]

(61)

Albeit different, both situations describe the same theories. However, it is clear that the second method, with only two auxiliary variables, is superior to the first method: the smaller the number of auxiliary variables, the more efficient we expect the corresponding MC algorithm to be.

3.3.2 \(\mathcal{R} = 3\mp\)

Here we consider Wilson actions with plaquettes in the symmetric \((\pm)\) or antisymmetric \((-)\) three-index representations of \(SU(N)\), respectively \(\mathcal{R} = (3)\) or \(\mathcal{R} = (1,1,1)\):

\[
S_\pm(\beta_\pm; U) = -\frac{\beta_\pm}{N(N^2 \pm 3N + 2)} \sum_p \text{Re} \left( \text{Tr} \{U_p\}^3 \pm 3 \text{Tr} \{U_p\} \text{Tr} \{U_p^2\} + 2 \text{Tr} \{U_p^3\} \right)
\]

(62)

We again consider both situations of positive or negative lattice coupling \(\beta_\pm\), whose sign we denote by \(\sigma \equiv \text{sgn}(\beta_\pm)\). One way of linearising (62) requires the addition of seven auxiliary complex matrix variables per plaquette \(\tilde{Q}_p^{(i)}\), \(i = 1, \ldots, 7\), and the corresponding change of variables:
\[ \tilde{Q}_p^{(1)} = \sqrt{2\beta_\pm} \left( Q_p^{(1)} - \frac{1}{\sqrt{12}N} \left( U_p \text{Tr}\{U_p\} + \sigma \text{Tr}\{U_p^\dagger\} \mathbb{1} \right) \right) \]  

\[ \tilde{Q}_p^{(2)} = \sqrt{2\beta_\pm} \left( Q_p^{(2)} - \frac{1}{2N} \left( U_p \text{Tr}\{U_p\} \pm \sigma U_p^\dagger \right) \right) \]  

\[ \tilde{Q}_p^{(3)} = \sqrt{2\beta_\pm} \left( Q_p^{(3)} - \frac{1}{\sqrt{6}N} \left( U_p^2 + \sigma U_p^\dagger \right) \right) \]  

\[ \tilde{Q}_p^{(4)} = \sqrt{2\beta_\pm} \left( Q_p^{(4)} - \frac{1}{\sqrt{12}} Q_p^{(1)} U_p^\dagger - \frac{1}{N} \text{Tr}\{U_p\} \mathbb{1} \right) \]  

\[ \tilde{Q}_p^{(5)} = \sqrt{2\beta_\pm} \left( Q_p^{(5)} - \frac{1}{2} Q_p^{(2)} U_p^\dagger - \frac{1}{N} \text{Tr}\{U_p\} \mathbb{1} \right) \]  

\[ \tilde{Q}_p^{(6)} = \sqrt{2\beta_\pm} \left( Q_p^{(6)} - \frac{1}{\sqrt{6}} Q_p^{(3)} U_p^\dagger - \frac{1}{N} U_p \right) \]  

\[ \tilde{Q}_p^{(7)} = \sqrt{2\beta_\pm} \left( Q_p^{(7)} - \left( \frac{29}{12} \right)^{\frac{1}{2}} \left( U_p - \frac{1}{N} \text{Tr}\{U_p\} \mathbb{1} \right) \right) \]  

where \( \beta_\pm = 6|\beta_\pm|N/(N^2 \pm 3N + 2) \). The variable \( Q_p^{(1)} \) eliminates the cubic term \( \text{Tr}\{U_p\}^3 \) from the action, but it also generates nonlinear terms of the form \( \text{Tr}\{Q_p^{(1)} U_p\} \text{Tr}\{U_p\} \) and \( |\text{Tr}\{U_p\}|^2 \). The variable \( Q_p^{(2)} \) eliminates the cubic term \( \text{Tr}\{U_p^2\} \text{Tr}\{U_p\} \) and generates similar nonlinear terms, namely \( \text{Tr}\{Q_p^{(2)} U_p\} \text{Tr}\{U_p\} \) and \( |\text{Tr}\{U_p\}|^2 \). The variable \( Q_p^{(3)} \) eliminates the cubic term \( \text{Tr}\{U_p^3\} \) and generates one nonlinear term of the form \( \text{Tr}\{Q_p^{(3)} U_p^2\} \). The variables \( Q_p^{(4)}, Q_p^{(5)} \) and \( Q_p^{(6)} \) eliminate the secondary nonlinear terms generated by \( Q_p^{(1)}, Q_p^{(2)} \) and \( Q_p^{(3)} \), respectively, except the term of the form \( |\text{Tr}\{U_p\}|^2 \); in fact, the variables \( Q_p^{(4)} \) and \( Q_p^{(5)} \) also produce such a term. The sum of these double-traces is eliminated by the variable \( Q_p^{(7)} \). All the seven auxiliary variables generate linear terms, whose contribution to the sum of ‘staples’ (44) is given by:

\[
 f_p^\pm = \frac{\sigma}{\sqrt{12}N} \text{Tr}\{Q_p^{(1)}\} \mathbb{1} \pm \frac{\sigma}{2N} Q_p^{(2)} + \frac{\sigma}{\sqrt{6}N} Q_p^{(3)} + \frac{1}{N} \text{Tr}\{Q_p^{(4)}\} \mathbb{1} + \frac{1}{\sqrt{6}} Q_p^{(4)} Q_p^{(1)} \\
 + \frac{1}{N} \text{Tr}\{Q_p^{(5)}\} \mathbb{1} + \frac{1}{\sqrt{2}} Q_p^{(5)} Q_p^{(2)} + \frac{1}{N} Q_p^{(6)} + \frac{1}{\sqrt{3}} Q_p^{(6)} Q_p^{(3)} \\
 + \left( \frac{29}{12} \right)^{\frac{1}{2}} \left( Q_p^{(7)} - \frac{1}{N} \text{Tr}\{Q_p^{(7)}\} \mathbb{1} \right) \]  

Just like in the previous case of two-index representations, the choice above is not unique. In fact, there is a cheaper way to eliminate all nonlinear terms, involving only five auxiliary variables. The corresponding change of variables is:
\[ \tilde{Q}^{(1)}_p = \sqrt{2\beta'_\pm} \left( Q^{(1)}_p - \frac{1}{\sqrt{12N}} \left( U_p \text{Tr}\{U_p\} + \sigma \text{Tr}\{U^\dagger_p\} \mathbf{1} \pm 3\sigma U^\dagger_p \right) \right) \]  
\[ \tilde{Q}^{(2)}_p = \sqrt{2\beta'_\pm} \left( Q^{(2)}_p - \frac{1}{\sqrt{6N}} \left( U^2_p + \sigma U^\dagger_p \right) \right) \]  
\[ \tilde{Q}^{(3)}_p = \sqrt{2\beta'_\pm} \left( Q^{(3)}_p - \frac{1}{\sqrt{12}} Q^{(1)}_p U^\dagger_p - \frac{1}{N} \text{Tr}\{U_p\} \mathbf{1} \right) \]  
\[ \tilde{Q}^{(4)}_p = \sqrt{2\beta'_\pm} \left( Q^{(4)}_p - \frac{1}{\sqrt{6}} Q^{(2)}_p U^\dagger_p - \frac{1}{N} U_p \right) \]  
\[ \tilde{Q}^{(5)}_p = \sqrt{2\beta'_\pm} \left( Q^{(5)}_p - \left( \frac{7N \pm 3}{6N} \right)^{\frac{1}{2}} \left( U_p - \frac{1}{N} \text{Tr}\{U_p\} \mathbf{1} \right) \right) \]

The variable \( Q^{(1)}_p \) eliminates simultaneously both the \( \text{Tr}\{U_p\}^3 \) and \( \text{Tr}\{U^2_p\} \text{Tr}\{U_p\} \) terms in the action, but it also generates nonlinear terms of the form \( \text{Tr}\{Q^{(1)}_p U^\dagger_p\} \text{Tr}\{U_p\} \) and \( |\text{Tr}\{U_p\}|^2 \). The variable \( Q^{(2)}_p \) eliminates the cubic term \( \text{Tr}\{U^3_p\} \) and generates one nonlinear term of the form \( \text{Tr}\{Q^{(2)}_p U^2_p\} \). The variables \( Q^{(3)}_p \) and \( Q^{(4)}_p \) eliminate the nonlinear terms generated by \( Q^{(1)}_p \) and \( Q^{(2)}_p \), respectively, except the term of the form \( |\text{Tr}\{U_p\}|^2 \) (which is also generated by \( Q^{(3)}_p \)). The sum of these double-traces is eliminated by \( Q^{(5)}_p \). In the end, the contribution of the remaining linear terms to the sum of ‘staples’ (44) is given by:

\[ f^\pm_p = \frac{\sigma}{\sqrt{12N}} \left( \text{Tr}\{Q^{(1)}_p\} \mathbf{1} \pm 3Q^{(1)}_p \right)^\dagger + \frac{\sigma}{\sqrt{6N}} Q^{(2)}_p U^\dagger_p + \frac{1}{\sqrt{12}} Q^{(3)}_p U^\dagger_p + \frac{1}{N} \text{Tr}\{Q^{(3)}_p\} \mathbf{1} \]

\[ + \frac{1}{\sqrt{6}} Q^{(4)}_p U^\dagger_p + \frac{1}{N} Q^{(4)}_p + \left( \frac{7N \pm 3}{6N} \right)^{\frac{1}{2}} \left( Q^{(5)}_p - \frac{1}{N} \text{Tr}\{Q^{(5)}_p\} \mathbf{1} \right) \]

3.3.3 \( \mathcal{R} = (2, 1) \)

Finally, we consider Wilson actions with plaquettes in the representation \( \mathcal{R} = (2, 1) \):

\[ S_{(2,1)}(\beta; U) = -\frac{\beta_{(2,1)}}{N(N^2 - 1)} \sum_p \text{Re} \left( \text{Tr}\{U_p\}^3 - \text{Tr}\{U^3_p\} \right) \]

The sign of \( \beta_{(2,1)} \) is denoted by \( \sigma \equiv \text{sgn}(\beta_{\pm}) \). One way of linearising (62) requires the addition of five auxiliary complex matrix variables per plaquette \( \tilde{Q}^{(i)}_p \), \( i = 1, \ldots, 5 \), and the corresponding change of variables:
\[ \tilde{Q}_p^{(1)} = \sqrt{2\beta'_{(2,1)}} \left( Q_p^{(1)} - \frac{1}{\sqrt{6N}} (U_p \text{Tr}\{U_p\} + \sigma \text{Tr}\{U_p^\dagger \}) \right) \]  \hspace{1cm} (78)

\[ \tilde{Q}_p^{(2)} = \sqrt{2\beta'_{(2,1)}} \left( Q_p^{(2)} - \frac{1}{\sqrt{6N}} (U_p^2 - \sigma U_p^\dagger) \right) \]  \hspace{1cm} (79)

\[ \tilde{Q}_p^{(3)} = \sqrt{2\beta'_{(2,1)}} \left( Q_p^{(3)} - \frac{1}{\sqrt{6N}} Q_p^{(1)U_p^\dagger} - \frac{1}{N} \text{Tr}\{U_p\} \right) \]  \hspace{1cm} (80)

\[ \tilde{Q}_p^{(4)} = \sqrt{2\beta'_{(2,1)}} \left( Q_p^{(4)} - \frac{1}{\sqrt{6N}} Q_p^{(2)U_p^\dagger} - \frac{1}{N} U_p \right) \]  \hspace{1cm} (81)

\[ \tilde{Q}_p^{(5)} = \sqrt{2\beta'_{(2,1)}} \left( Q_p^{(5)} - \frac{2}{\sqrt{3}} \left( U_p - \frac{1}{N} \text{Tr}\{U_p\} \right) \right) \]  \hspace{1cm} (82)

where \( \beta'_{(2,1)} = 3|\beta_{(2,1)}|N/(N^2 - 1) \). The variable \( Q_p^{(1)} \) eliminates the cubic term \( \text{Tr}\{U_p\}^3 \) in the action, and it generates nonlinear terms of the form \( \text{Tr}\{Q_p^{(1)}U_p\}\text{Tr}\{U_p\} \) and \( |\text{Tr}\{U_p\}|^2 \). The variable \( Q_p^{(2)} \) eliminates the cubic term \( \text{Tr}\{U_p^3\} \) and generates one nonlinear term of the form \( \text{Tr}\{Q_p^{(2)}U_p^2\} \). The variables \( Q_p^{(3)} \) and \( Q_p^{(4)} \) eliminate the secondary nonlinear terms generated by \( Q_p^{(1)} \) and \( Q_p^{(2)} \), respectively, except the term of the form \( |\text{Tr}\{U_p\}|^2 \) (which is also generated by \( Q_p^{(3)} \)). The sum of these quadratic terms is eliminated by the fifth variable. In the end, the contribution of the remaining linear terms to the sum of ‘staples’ \( \{44\} \) is given by:

\[ f_p^{(2,1)} = \frac{\sigma}{\sqrt{6N}} \text{Tr}\{Q_p^{(1)}\} - \frac{\sigma}{\sqrt{6N}} Q_p^{(2)} + \frac{1}{N} \text{Tr}\{Q_p^{(3)}\} + \frac{1}{\sqrt{6}} Q_p^{(4)} + \frac{1}{N} Q_p^{(4)U_p^\dagger} + 2 \left( Q_p^{(5)} - \frac{1}{N} \text{Tr}\{Q_p^{(5)}\} \right) \]  \hspace{1cm} (83)

Also in this case it is possible to find a smaller set of auxiliary variables that does the same job. In fact, only three auxiliary variables suffice to eliminate all the nonlinear terms from the action \( \{44\} \). The corresponding change of variables is:

\[ \tilde{Q}_p^{(1)} = \sqrt{2\beta'_{(2,1)}} \left( Q_p^{(1)} - \frac{1}{\sqrt{6N}} \left( U_p (\text{Tr}\{U_p\} - U_p) + \sigma (U_p + \text{Tr}\{U_p\}) \right) \right) \]  \hspace{1cm} (84)

\[ \tilde{Q}_p^{(2)} = \sqrt{2\beta'_{(2,1)}} \left( Q_p^{(2)} - \frac{1}{\sqrt{6N}} Q_p^{(1)U_p^\dagger} - \frac{1}{N} (U_p - \text{Tr}\{U_p\}) \right) \]  \hspace{1cm} (85)

\[ \tilde{Q}_p^{(3)} = \sqrt{2\beta'_{(2,1)}} \left( Q_p^{(3)} - \frac{4N - 6}{3N} \left( U_p - \frac{1}{N} \text{Tr}\{U_p\} \right) \right) \]  \hspace{1cm} (86)

The variable \( Q_p^{(1)} \) eliminates both nonlinear terms in the action, but it also generates nonlinear terms of the form \( \text{Tr}\{Q_p^{(1)}U_p\}\text{Tr}\{U_p\} \), \( \text{Tr}\{Q_p^{(1)}U_p^2\} \) and \( |\text{Tr}\{U_p\}|^2 \). The variable \( Q_p^{(2)} \)
eliminates all secondary nonlinear terms generated by $Q_p^{(1)}$, except the term $|\text{Tr}\{U_p\}|^2$, which is also generated by it. The sum of these double-traces is eliminated by $Q_p^{(3)}$. In the end, the contribution of the remaining linear terms to the sum of ‘staples’ \((44)\) is given by:

$$f_p^{(2,1)} = \frac{\sigma}{\sqrt{6N}} (Q_p^{(1)} + \text{Tr}\{Q_p^{(1)}\} \mathbb{1})^\dagger - \frac{1}{N} (Q_p^{(2)} - \text{Tr}\{Q_p^{(2)}\} \mathbb{1}) + \frac{1}{\sqrt{6}} Q_p^{(1)} Q_p^{(2)}$$

\[ + \left( \frac{4N - 6}{3N} \right)^{\frac{1}{2}} \left( Q_p^{(3)} - \frac{1}{N} \text{Tr}\{Q_p^{(3)}\} \mathbb{1} \right) \]  

\( (87) \)

### 3.4 Mixed Wilson actions

The generalisation to the case of mixed Wilson actions is trivial, because a linearised mixed Wilson action is simply the sum of linearised Wilson actions for each representation:

$$S'_{\text{mix}}(\vec{\beta}; U_{\mu,x}) = \sum_R S'_R(\beta_R; U_{\mu,x})$$  

Therefore, the probability distribution of individual links associated with a generic mixed Wilson action is given by \((3)\), where $V_{\mu,x}$ is the sum of ‘staples’:

$$V_{\mu,x} = \sum_{\nu=1}^{d} \left( f_{\mu\nu,x} U_{\nu,x} U_{\mu,x+\hat{\nu}} U_{\nu,x+\hat{\nu}}^\dagger + U_{\nu,x-\hat{\nu}}^\dagger f_{\nu\mu,x-\hat{\nu}} U_{\mu,x-\hat{\nu}} U_{\nu,x-\hat{\nu}} \right)$$  

\( (89) \)

and $f_{\mu\nu,x} \equiv f_p$ encodes the information about all representations involved:

$$f_p = 2 \sum_R \beta'_R f_p^R$$  

\( (90) \)

The possibility of using the efficient MC algorithms discussed in Section \[I\] follows immediately. The MC algorithm for a general mixed Wilson action is summarised in Appendix \[A.2\].

This straightforward approach to mixed Wilson actions may not be the most efficient, however. In this approach, each representation is treated independently, and the total number of auxiliary variables is $n_{\text{mix}} = \sum_R n_R$. But in some situations it is possible to reduce $n_{\text{mix}}$. For example, consider the mixed ‘$A + (2)$’ Wilson action. During the linearisation of $\chi(2)$, a secondary nonlinear term of the form $|\text{Tr}\{U_p\}|^2$ is generated, and it can be eliminated in simultaneous with the double-trace terms of $\chi_A(U_p)$. Therefore, only two auxiliary variables per plaquette are needed, instead of the naive $n_A + n_{(2)} = 3$. This is the same ambiguity that exists in the case of individual representations, as shown above. In sum, the process of linearisation is not unique, and may be improved by a wise choice of auxiliary variables and respective change of variables. In the end, one must always choose the set with the smallest number of auxiliary variables.
3.5 A note on reduced mixed models

A special note must be taken in the case of mixed Wilson actions on lattices with reduced directions. These may be useful to study the (non-)universality of the symmetry breaking transitions that invalidate the large $N$ equivalence in TEK models \[21\].

As usual, the construction of an efficient updating algorithm for a mixed reduced model consists in adding enough auxiliary variables in order to eliminate all the nonlinear terms in the action. In reduced models, however, the nonlinearities have two origins: the quadratic nature of the reduced plaquette operator, and the nonlinear nature of the $SU(N)$ characters. Clearly, the nonlinearities associated with $SU(N)$ characters need to be dealt with first. For that end, the linearisation of the mixed lattice action proceeds exactly as described in Sections 3.1–3.4.

The problem occurs in the final step, when dealing with the reduced plaquette operator. One would be tempted to solve the problem with the original Fabricius-Haan treatment for the EK model. The nonlinear terms to be eliminated are of the form:

$$-2\beta' R \text{ReTr} \left\{ f_{\mu\nu,x}^{R\dagger} U_{\mu,x} U_{\nu,x+\mu} U_{\mu,x}^{\dagger} U_{\nu,x}^{\dagger} \right\}$$  \hspace{1cm} (91)

which differ from the EK plaquette terms by the matrix factor $f_{\mu\nu,x}^{R}$. If one adds an auxiliary variable $\tilde{R}_{\mu\nu,x}$ and perform the change of variables

$$\tilde{R}_{\mu\nu,x} = \sqrt{2\beta' R} (R_{\mu\nu,x} - f_{\mu\nu,x}^{R\dagger} U_{\mu,x} U_{\nu,x+\mu} U_{\mu,x}^{\dagger} U_{\nu,x}^{\dagger})$$  \hspace{1cm} (92)

the linearisation with respect to the link variables is promptly achieved:

$$-\frac{1}{2} \text{Tr} \left\{ \tilde{R}_{\mu\nu,x}^{\dagger} \tilde{R}_{\mu\nu,x} \right\}$$  \hspace{1cm} (92)

Gaussian $\rightarrow$ $-\beta' R \text{Tr} \left\{ R_{\mu\nu,x}^{\dagger} R_{\mu\nu,x} \right\}$

non-Gaussian $\rightarrow$ $-\beta' R \text{Tr} \left\{ f_{\mu\nu,x}^{R\dagger} f_{\mu\nu,x}^{R} \right\}$

linear $\rightarrow$ $+2\beta' R \text{ReTr} \left\{ R_{\mu\nu,x}^{\dagger} \left( f_{\mu\nu,x}^{R\dagger} U_{\mu,x} U_{\nu,x+\mu} U_{\mu,x}^{\dagger} U_{\nu,x}^{\dagger} \right) \right\}$

cancels out $\rightarrow$ $-2\beta' R \text{ReTr} \left\{ f_{\mu\nu,x}^{R\dagger} U_{\mu,x} U_{\nu,x+\mu} U_{\mu,x}^{\dagger} U_{\nu,x}^{\dagger} \right\}$  \hspace{1cm} (93)

However, new terms of the form $\text{Tr} \left\{ f_{\mu\nu,x}^{R\dagger} f_{\mu\nu,x}^{R} \right\}$ appear in the linearised action. Even though they do not depend on the link to be updated, they give a non-Gaussian weight to the auxiliary variables. Therefore, the efficient algorithms discussed in Section 3 cannot be applied, unless such terms are eliminated too. The elimination of the term $\text{Tr} \left\{ f_{\mu\nu,x}^{R\dagger} f_{\mu\nu,x}^{R} \right\}$ is not easy. In general, $f_{\mu\nu,x}^{R}$ has a rather complicated dependence on the auxiliary variables, as can be seen in the examples derived in Sections 3.2–3.4. The only situation for which we have an easy solution to the problem (or, at least, a solution that does not undermine the efficiency of the resulting algorithm) is the case of the adjoint representation.
For positive $\beta_A$, the solution is trivial. Given that $f^{A}_{\mu\nu,x} = z_{\mu\nu,x}/N$, the non-Gaussian term is in fact Gaussian:

$$- \beta' A \text{Tr} \left\{ f^{A\dagger}_{\mu\nu,x} f^{A}_{\mu\nu,x} \right\} = - \frac{\beta' A}{N} |z_{\mu\nu,x}|^2$$

and so the problem is solved. The update of the link variables for a partially reduced adjoint Wilson action is performed with respect to the probability distribution of individual links (6), with $V_{\mu,x} \equiv V^A_{\mu,x}$ given by:

$$V^A_{\mu,x} = \frac{2\beta' A}{N} \sum_{\nu=1}^{d} \left( z_{\mu\nu,x} U_{\nu,x} U_{\mu,x+\hat{\nu}} U_{\nu,x+\hat{\mu}}^\dagger + z^{\ast}_{\mu\nu,x-\hat{\nu}} U_{\mu,x-\hat{\nu}} U_{\nu,x-\hat{\nu}} U_{\nu,x-\hat{\mu}}^\dagger \right)$$

$$+ 2\beta' A \sum_{\nu=1}^{d} \left( \frac{1}{N} z_{\mu\nu,x} R_{\mu\nu,x} U_{\nu,x+\hat{\mu}}^\dagger + U_{\nu,x}^\dagger R_{\mu\nu,x} \right)$$

The first term in the r.h.s. of the equation above is the contribution from the unreduced plaquettes, and the second term is the contribution from the reduced plaquettes.

For negative $\beta_A$, $f^{A}_{\mu\nu,x}$ is a matrix. In order to eliminate this term, we first expand it in terms of the $Q$-variables:

$$- \beta' A \text{Tr} \left\{ f^{A\dagger}_{\mu\nu,x} f^{A}_{\mu\nu,x} \right\}$$

$$\text{Gaussian} \rightarrow - \beta' A \text{Tr} \left\{ Q_{\mu\nu,x}^\dagger Q_{\mu\nu,x} \right\}$$

$$\text{nonlinear} \rightarrow + \frac{\beta' A}{N} \left| \text{Tr} \left\{ Q_{\mu\nu,x} \right\} \right|^2$$

This term is clearly non-Gaussian. However, it is the sum of a Gaussian term and a double-trace term that can easily be eliminated. For that end, we introduce yet another auxiliary variable $M_{\mu\nu,x}$ with the normal distribution $\mu_1(M_{\mu\nu,x})$, associated with positively-oriented reduced plaquettes (for negatively-oriented plaquettes we define $\tilde{M}_{\nu\mu,x} \equiv M^{\dagger}_{\mu\nu,x}$). Let us perform the change of variables:

$$\tilde{M}_{\mu\nu,x} = \sqrt{2\beta' A} \left( M_{\mu\nu,x} - \frac{1}{N} \text{Tr} \left\{ Q_{\mu\nu,x} \right\} U_{\mu,x+\hat{\nu}} U_{\nu,x+\hat{\mu}}^\dagger \right)$$

The effect of (97) on the Gaussian exponent in $\mu_1(M_{\mu\nu,x})$ generates the following terms:
\[ - \frac{1}{2} \text{Tr} \left\{ \tilde{M}_{\mu \nu, x} \tilde{M}_{\mu \nu, x} \right\} \]

Gaussian \rightarrow - \beta' A \text{Tr} \left\{ M_{\mu \nu, x} \tilde{M}_{\mu \nu, x} \right\}

linear \rightarrow + \frac{2 \beta' A}{N} \text{Re} \text{Tr} \left\{ \text{Tr} \left\{ Q_{\mu \nu, x} \right\} M_{\mu \nu, x} U_{\mu, x} + \hat{\nu} U_{\nu, x}^\dagger + \hat{\mu} \right\}

cancels out \rightarrow - \beta' A \left| \text{Tr} \left\{ Q_{\mu \nu, x} \right\} \right|^2 \]

the last of which cancels out the nonlinear term coming from the non-Gaussian piece, thus solving the problem for negative \( \beta_A \). The sum of 'staples' for the probability distribution of individual links (44) is then given by:

\[ V_{\mu, x}^A = 2 \beta' A \sum_{i=1}^{d} \left( f^{A}_{\mu \nu, x} U_{\mu, x} + \hat{\nu} U_{\nu, x} + \hat{\mu} \right) + U_{\nu, x}^\dagger f^{A\dagger}_{\mu \nu, x} U_{\mu, x} + \hat{\nu} U_{\nu, x} + \hat{\mu} \]

\[ + 2 \beta' A \sum_{i=1}^{d} \left( f^{A}_{\mu \nu, x} R_{\mu \nu, x} U_{\nu, x} + \hat{\nu} R_{\nu, x} + \hat{\mu} \right) + \frac{1}{N} \text{Tr} \left\{ Q_{\mu \nu, x} \right\} M_{\mu \nu, x} U_{\nu, x} + \hat{\mu} \]

where \( f^{A}_{\mu \nu, x} \) is defined as in (53).

For higher representations, the analogous of the expansion (96) of the non-Gaussian term results in a higher number of complicated nonlinear terms. In order to eliminate them, many more auxiliary terms would have to be introduced. This would certainly render the resulting algorithms inefficient and useless. Although unlikely, it is not a priori impossible that a clever choice of a smaller set of auxiliary variables would result in viable updating algorithms for higher-representation reduced models.

### 3.6 Numerical tests

We simulated some of the new algorithms proposed above, with the purpose of comparing them with the Metropolis algorithm. We considered mixed ‘F + R’ Wilson actions in \( d = 4 \), i.e. the Wilson action (27) with plaquettes in the fundamental representation, \( F \), and in another representation \( R \neq F \) of \( SU(N) \). We simulated each theory with both a Metropolis algorithm and the new MC algorithm proposed in Section 3.4. We tested both algorithms for their compatibility, by checking if the expectation values of gauge-invariant observables coincide in both cases. We also tested them for their relative efficiency, by comparing the magnitude of the autocorrelations in the Markov chains they generate.

For the thermal Metropolis updates we used a variant of the (1-hit) Cabibbo-Marinari-Metropolis (CMM) algorithm described in [19], and appropriately adapted to an arbitrary
representation $\mathcal{R}$. In our CMM algorithm, new link proposals are generated via Cabibbo-Marinari updates with respect to the fundamental part of the action only, in which $\beta_F$ is replaced with the tuning parameter of the Metropolis algorithm $\beta_M$ ($\beta_M < \beta_F$). The link proposal is then accepted or rejected a la Metropolis with respect to the full ‘$F + \mathcal{R}$’ action. The acceptance rates are tuned to stay in the range 40-60%. For the overrelaxation updates, we adapted the overrelaxation-Metropolis algorithm also described in [19]. In this algorithm, $SU(2)$– or $SU(N)$–overrelaxed link variables are accepted or rejected with respect to the $S_\mathcal{R}$ part of the action only. Acceptance rates for the overrelaxation-Metropolis algorithm stayed well above 85%.

For both the Metropolis and the new algorithms, each configuration update consisted of one thermal update followed by 5 overrelaxation updates. For each configuration, we evaluated the characters $\chi_F$ and $\chi_\mathcal{R}$ of the plaquette, in order to estimate their expectation values. In each simulation we performed $O(10^5)$ measurements, after discarding the initial 2,000 configurations for equilibration. The simulation parameters and measured observables, together with their naive confidence intervals, are shown in Table 2.

The parameters of the simulations of the ‘$F + A$’ action were chosen to coincide with those of [19]. In this way, we could compare our results with the literature. We considered two particular values for the adjoint lattice coupling, one positive and one negative, that were also considered in that article. We obtained compatible results for both plaquette characters, which is good evidence that our new algorithm reproduces accurately the ‘$F + A$’ lattice gauge theory. For higher representations, we performed simulations of the ‘$F + \mathcal{R}$’ theories on a $8^4$ lattice. In each of these cases, the observables calculated for a particular value of the lattice coupling also matched in both algorithms. There is, however, some discrepancies in the last significant digits of each observable. This is probably due to the fact that the confidence intervals shown in Table 2 do not take autocorrelations into account, i.e. they are underestimated.

We also measured the autocorrelations of the Markov chains generated by both types of algorithms. We calculated the values of the fundamental trace of the plaquette $u_i$ as a function of the MC time $i$, and then used them to estimate the normalised autocorrelation function:

$$C(t) = \frac{\langle u_{i+t}u_i \rangle - \langle u_i \rangle^2}{\langle u_i^2 \rangle - \langle u_i \rangle^2}$$

From $C(t)$ we estimated the time-dependent integrated autocorrelation time:

$$\tau_{\text{int}}(t) = 1 + 2 \sum_{i=1}^{t} C(i)$$

The estimator of the integrated autocorrelation time, $\tau_{\text{int}}$, is the plateau value of $\tau_{\text{int}}(t)$ when $t \to \infty$. In terms of these autocorrelations, the new pseudo-heatbath algorithms performs significantly better than their Metropolis counterparts, as can be seen in Figs.1–12. The graphs show that the new algorithms decorrelate faster than Metropolis, for all representations, both in MC time and in effective CPU time.
We also observed that, except for the case of the adjoint representation, one pseudo-heatbath update takes longer to perform (in CPU time) than one Metropolis hit. This is not surprising, taking into account the large number of auxiliary variables that some representations require. Only for the adjoint case, which requires only one auxiliary variable per plaquette, were the pseudo-heatbath updates faster than Metropolis. In particular, this difference was largest for $\beta_A > 0$, as expected. Nevertheless, the fast decorrelation of the new algorithms overshadows the disadvantage of slower updates, making them a superior alternative to Metropolis.

4 Double-trace deformations

Recently, Ünsal and Yaffe suggested [13] a strategy to prevent the spontaneous breaking of the centre symmetry in pure $SU(N)$ Yang-Mills theories on manifolds with small compactified directions, namely $R^{d-k} \times (S^1)^k$. The strategy consists in deforming the Yang-Mills action with a centre-stabilising potential, which contains double-trace operators dependent on the holonomies wrapping the compact directions. This mimics the effective potential contribution of adjoint fermions with periodic boundary conditions in the compact direction, which are known to stabilise the centre symmetry [22]. The gauge theories deformed by such double-trace terms are called centre-stabilised Yang-Mills (CYM) theories.

The breaking of the centre symmetry is the reason why the EK model (and most of its variants) is not equivalent to pure Yang-Mills theories in the large $N$ limit. CYM theories, however, are believed to be completely volume-independent in the large $N$ limit. If this is true, then it would be possible to represent large $N$ Yang-Mills theory defined on $R^d$ by the zero volume limit of the compact directions of CYM theories. Taking the volume of the compactified directions to zero with no consequences for the large $N$ equivalence has clear analytical and numerical advantages.

In this section we consider the lattice regularisation of CYM theories defined on manifolds with a single compactified direction, $R^{d-1} \times S^1$. Due to the double-trace terms, the CYM lattice action is highly nonlinear with respect to the link variables, hence Metropolis seem to be the only possible algorithm to simulate such theories. However, we will show that it is possible to linearise the CYM lattice action using the method of auxiliary variables, and from there obtain a rather efficient MC algorithms.

4.1 CYM theory on $R^{d-1} \times S^1$

The lattice action for the $SU(N)$ CYM theory defined on a manifold with only one compact direction, namely $R^{d-1} \times S^1$, is given by:

$$S_{\text{CYM}} (\vec{\alpha}, \beta_F; [U]) = S_F (\beta_F; [U]) + S_{\text{def}} (\vec{\alpha}; [U])$$  \hspace{1cm} (102)

where $S_F$ is the fundamental Wilson action [1] and $\vec{\alpha} \equiv (\alpha_1, \ldots, \alpha_{[N/2]})$ are free parameters of the CYM model that must be positive. The deformation potential $S_{\text{def}}$ is a sum of double-traces:
\[ S_{\text{det}} (\vec{\alpha}; [U]) = \frac{1}{N_t^{d-1}} \sum_{x \in \Lambda_\perp} \sum_{n=1}^{[N/2]} \alpha_n |\text{Tr} \{ \Omega_x^n \}|^2 \]  

(103)

where \( N_t \) is the size of the compact direction \((S^1)\) in lattice units, \( x \) labels the lattice sites on the \((d-1)\)-dimensional lattice \( \Lambda_\perp \) (the discretization of \( \mathbb{R}^{d-1} \)), and \( \Omega_x \) is the holonomy (Polyakov loop) wrapping the compact direction, \( \hat{d} \):

\[ \Omega_x = \mathcal{P} \prod_{i=0}^{N_t-1} U_{d,x+i\hat{d}} = U_{d,x} U_{d,x+d} \cdots U_{d,x+(N_t-1)d} \]  

(104)

The CYM action (103) is highly nonlinear with respect to the link variables, especially for large \( N \) (because it contains terms that depend up to the \([N/2]\)-th power of the link variable). In addition, \( S_{\text{det}} \) contains \( O(N^1) \) terms, which makes the theory even harder to simulate in that limit. From these facts, we should expect to use a relatively large number of auxiliary variables in order to linearise the whole action. Let us consider the following set:

\[ \tilde{R}_{n,x} , \quad 1 \leq n \leq K \]

\[ \tilde{Q}^{(m)}_{n,x} , \quad 1 \leq m < n \leq K \]

\[ \tilde{Q}^{(K)}_{n,x} , \quad 2 \leq n \leq K \]

where \( K = \lfloor N/2 \rfloor \). We then multiply the partition function of the CYM theory by the Gaussian integrals \( \int \mu_1 [\tilde{R}, \tilde{Q}] = 1 \). We now show that the following change of variables linearises the CYM action:

\[ \tilde{R}_{n,x} = \left( \frac{2N\alpha_n}{N_t^{d-1}} \right)^\frac{1}{2} \left( R_{n,x} - \left( \Omega_x^n - \frac{1}{N} \text{Tr} \{ \Omega_x^n \} \mathbb{1} \right) \right) \]  

(105)

\[ \tilde{Q}^{(m)}_{n,x} = \left( \frac{2N\alpha_n}{N_t^{d-1}} \right)^\frac{1}{2} \left( Q^{(m)}_{n,x} - (Q^{(m-1)}_{n,x} \Omega_x^1 + \Omega_x^{n-m}) \right) \]  

(106)

\[ \tilde{Q}^{(K)}_{n,x} = \left( \frac{2N\alpha_n}{N_t^{d-1}} \right)^\frac{1}{2} \left( Q^{(K)}_{n,x} - \frac{1}{N} \text{Tr} \{ R_{n,x} \} \Omega_x^1 \right) \]  

(107)

where we define \( Q^{(0)}_{n,x} \) to be the traceless part of \( R_{n,x} \):

\[ Q^{(0)}_{n,x} \equiv R_{n,x} - \frac{1}{N} \text{Tr} \{ R_{n,x} \} \mathbb{1} \]  

(108)

The \( \tilde{R} \)-variables cancel out the double-trace terms from the CYM action. We can see this when we apply the change of variables (105) on the exponent of the Gaussian measure \( \mu_1 [\tilde{R}] \):
\[ - \frac{1}{2} \sum_{n=1}^{K} \text{Tr} \left\{ \tilde{R}_{n,x}^{\dagger} \tilde{R}_{n,x} \right\} \quad (105) \]

\[
\text{Gaussian} \rightarrow - \frac{N}{N_{d-1}^t} \sum_{n=1}^{K} \alpha_{n} \text{Tr} \left\{ R_{n,x}^{\dagger} R_{n,x} \right\} \\
\text{linear} \rightarrow + \frac{2N}{N_{d-1}^t} \alpha_{1} \text{ReTr} \left\{ Q_{1,x}^{(0)} \Omega_{x} \right\} \\
\text{nonlinear} \rightarrow + \frac{2N}{N_{d-1}^t} \sum_{n=2}^{K} \alpha_{n} \text{ReTr} \left\{ Q_{n,x}^{(0)} \Omega_{x}^{n} \right\} \\
\text{cancels out} \rightarrow + \frac{1}{N_{d-1}^t} \sum_{n=1}^{K} \alpha_{n} |\text{Tr} \left\{ \Omega_{x}^{n} \right\}|^2 \quad (109)\]

The nonlinear terms in the r.h.s. of the expression above are cancelled by the \( \tilde{Q} \)-variables:

\[ - \frac{1}{2} \sum_{n=2}^{K} \sum_{m=1}^{n-1} \text{Tr} \left\{ \tilde{Q}_{n,x}^{(m)} \tilde{Q}_{n,x}^{(m)} \right\} \quad (106) \]

\[
\text{Gaussian} \rightarrow - \frac{2N}{N_{d-1}^t} \sum_{n=2}^{K} \sum_{m=1}^{n-1} \alpha_{n} \text{Tr} \left\{ Q_{n,x}^{(m)} \Omega_{x}^{n} \right\} \\
\text{Gaussian} \rightarrow - \frac{N}{N_{d-1}^t} \sum_{n=2}^{K} \alpha_{n} \text{Tr} \left\{ Q_{n,x}^{(n-1)} \Omega_{x}^{n-1} \right\} \\
\text{non-Gaussian} \rightarrow - \frac{N}{N_{d-1}^t} \sum_{n=2}^{K} \alpha_{n} \text{Tr} \left\{ Q_{n,x}^{(0)} \Omega_{x}^{0} \right\} \\
\text{linear} \rightarrow - \frac{2N}{N_{d-1}^t} \sum_{n=2}^{K} \sum_{m=1}^{n-1} \alpha_{n} \text{ReTr} \left\{ Q_{n,x}^{(m)} \Omega_{x}^{(m-1)} \Omega_{x}^{\dagger} \right\} \\
\text{linear} \rightarrow + \frac{2N}{N_{d-1}^t} \sum_{n=2}^{K} \alpha_{n} \text{ReTr} \left\{ Q_{n,x}^{(n-1)} \Omega_{x} \right\} \\
\text{cancels out} \rightarrow - \frac{2N}{N_{d-1}^t} \sum_{n=2}^{K} \alpha_{n} \text{ReTr} \left\{ Q_{n,x}^{(0)} \Omega_{x}^{n} \right\} \quad (110)\]

The only remaining pathological terms that need to be eliminated are non-Gaussian. This situation is similar to the case of reduced mixed models with a negative adjoint coupling \( \beta_A \) (see Section 3.5). Hence let us expand the non-Gaussian piece in terms of the \( R \)-variables:
\[- \frac{N}{N^d - 1} \sum_{n=2}^{K} \alpha_n \text{Tr} \{ Q^{(0)\dagger}_n Q^{(0)}_n \} \]

\[\text{Gaussian } \rightarrow \quad - \frac{N}{N^d - 1} \sum_{n=2}^{K} \alpha_n \text{Tr} \{ R^{\dagger}_{n,x} R_{n,x} \} \]

\[\text{nonlinear } \rightarrow \quad + \frac{1}{N^{d-1}} \sum_{n=2}^{K} \alpha_n |\text{Tr} \{ R_{n,x} \}|^2 \quad (111)\]

In order to eliminate the nonlinear term in (111), we use the auxiliary variables $\tilde{Q}^{(K)}_{n,x}$:

\[- \frac{1}{2} \sum_{n=2}^{K} \text{Tr} \left\{ \tilde{Q}^{(K)\dagger}_{n,x} \tilde{Q}^{(K)}_{n,x} \right\} \quad (110)\]

\[\text{Gaussian } \rightarrow \quad - \frac{N}{N^d - 1} \sum_{n=2}^{K} \alpha_n \text{Tr} \{ Q^{(K)\dagger}_{n,x} Q^{(K)}_{n,x} \} \]

\[\text{linear } \rightarrow \quad + \frac{2N}{N^d - 1} \sum_{n=2}^{K} \alpha_n \text{ReTr} \left\{ \frac{1}{N} \text{Tr} \{ R_{n,x} \} Q^{(K)\dagger}_{n,x} Q^{(K)}_{n,x} \right\} \]

\[\text{cancels out } \rightarrow \quad - \frac{1}{N^d - 1} \sum_{n=2}^{K} \alpha_n |\text{Tr} \{ R_{n,x} \}|^2 \quad (112)\]

All nonlinear terms cancel out and are replaced by Gaussian and linear terms. The applicability of the efficient updating algorithms discussed in Section 1 follows immediately.

With auxiliary variables, the partition function of the CYM theory becomes:

\[Z_{\text{CYM}}(\bar{\alpha}, \beta_F) = \int \mu_H[U] \mu[R, Q] \exp \left( -S_F(\beta_F; [U]) + \frac{2N}{N^d - 1} \sum_x \text{ReTr} \{ f_x \Omega_x \} \right) \quad (113)\]

where $\mu_H[U]$ is the usual product of $SU(N)$-invariant Haar measures of the link variables, and $\mu[R, Q]$ is the product of Gaussian measures of the auxiliary variables,

\[\mu[R, Q] \equiv \prod_{x \in \Lambda \perp d} \prod_{n=2}^{K} \left( \mu_{\sigma_n}(R_{n,x}) \mu_{\sigma_{K,n}}(Q^{(K)}_{n,x}) \prod_{m=1}^{n-1} \mu_{\sigma_{m,n}}(Q^{(m)}_{n,x}) \right) \quad (114)\]

with $\sigma$'s given by:

\[\sigma_1^2 = 2\sigma_n^2 = 2\sigma_{m,n}^2 = \sigma_{n-1,n}^2 = \sigma_{K,n}^2 = \frac{N^{d-1}}{2N\alpha_n}, \quad 2 \leq n \leq K, \quad 1 \leq m \leq n - 2 \quad (115)\]

and $f_x$ is a matrix factor that encodes all the information about the deformation terms, which is obtained from the linear terms generated by the auxiliary variables:

\[f_x = \alpha_1 Q^{(0)}_{1,x} + \sum_{n=2}^{K} \alpha_n \left( Q^{(n-1)}_{n,x} + \sum_{m=1}^{n-1} Q^{(m)}_{n,x} Q^{(m-1)}_{n,x} + \frac{1}{N} \text{Tr} \{ R_{n,x} \} Q^{(K)\dagger}_{n,x} \right) \quad (116)\]
The exponent of the Boltzmann factor in the partition function is now fully linear. Hence each link variable can be updated with respect to a probability distribution \( (6) \), where \( V_{\mu,x} \) is analogous to the typical sum of ‘staples’. If the link variable to be updated is parallel to a non-compact direction, then \( V_{\mu,x} \) contains only the contribution \( V_{\mu,x}^F \) from the neighboring plaquettes; if the link variable is parallel to the compact direction, then it will also contain the contribution \( V_{x}^{\text{def}} \) from the deformation terms:

\[
V_{\mu,x} = \begin{cases} 
\frac{\beta_F}{N} V_{\mu,x}^F, & \text{if } \mu = 1, \ldots, d - 1 \\
\frac{\beta_F}{N} V_{\mu,x}^F + \frac{2N}{N_t^2} V_{x}^{\text{def}}, & \text{if } \mu = d
\end{cases}
\]  

(117)

The plaquette contribution \( V_{\mu,x}^F \) is given by \( (26) \), which already takes into account the possibility of a fully reduced compact direction (i.e. \( N_t = 1 \)). On the other hand, the contribution from the double-trace terms is given by:

\[
V_{x}^{\text{def}} = \left( \mathcal{P} \prod_{i=0}^{t-1} U_{d,x+i\hat{d}} \right) \dagger \cdot f_x \cdot \left( \mathcal{P} \prod_{i=t+1}^{L_d-1} U_{d,x+i\hat{d}} \right) \dagger
\]  

(118)

This term resembles the Hermitian conjugate of the Polyakov loop wrapping the compact direction and starting at \( x = (x, t) \), except that the link variable \( U_{d,x} \) is replaced with \( f_x \). When implementing the algorithm, one may find convenient to redefine the Polyakov loop \( \Omega_x \) to start at \( x \) before the link \( U_{d,x} \) is updated,

\[
\Omega_x \rightarrow \mathcal{P} \prod_{i=0}^{N_t-1} U_{d,x+i\hat{d}} = U_{d,x+t\hat{d}} \cdots U_{d,x+(N_t-1)d} U_{d,x} U_{d,x+d\hat{d}} \cdots U_{d,x+(t-1)d}
\]  

(119)

where \( i' \equiv i + t \mod N_t \). The ‘staple’ contribution coming from the deformation terms is then given by:

\[
V_{x}^{\text{def}} = f_x \cdot \left( \mathcal{P} \prod_{i=1}^{N_t-1} U_{d,x+i\hat{d}} \right) \dagger
\]  

(120)

If the compact direction is fully reduced, then \( V_{x}^{\text{def}} \) is simply given by:

\[
V_{x}^{\text{def}} = f_x
\]  

(121)

The MC algorithm for lattice CYM theories on \( \mathbb{R}^{d-1} \times S^1 \) is summarised in Appendix A.3.

4.2 Numerical tests

We performed numerical simulations of the MC algorithm described above and compared it with Metropolis. We considered the lattice regularised version of \( SU(5) \) CYM theory defined on \( \mathbb{R}^3 \times S^1 \). We simulated the theory on two different lattices, namely on \( 10^3 \) and \( 10^3 \) lattices.
The purpose was to test the new algorithm in both situations of an unreduced and a fully reduced compact direction.

For the gauge group $SU(5)$, the CYM action has $|5/2| = 2$ distinct deformation terms. The $\alpha_n$ parameters attached to them supposedly interpolate between different phases of the Yang-Mills theory at fixed coupling. In particular, for large values of $\beta_F$, the CYM theory should interpolate between the deconfining regime for small $\alpha_n$ (where the centre symmetry associated with the compact direction is spontaneously broken) and a confining regime for large $\alpha_n$ (where the centre symmetry is intact). This behaviour is a result of the competition between the $S_F$ and the $S_{\text{def}}$ terms in the CYM action. For this reason we chose to perform our simulations at $\beta_F = 25.0$, which is located in the deconfining ($Z_N$-broken) regime of the fundamental Wilson action. For the $10^31$ lattice, we chose values for $\alpha_n$ that would put the CYM theory on two different phases. Specifically, we chose a ‘small’ $\vec{\alpha} = (0.20, 0.05)$ and a ‘large’ $\vec{\alpha} = (0.60, 0.10)$. For the $10^33$ lattice, we only chose one value, $\vec{\alpha} = (2.80, 0.20)$.

For the thermal Metropolis updates, we used a variant of the (1-hit) Cabibbo-Marinari-Metropolis (CMM) algorithm described in [19] and adapted to CYM theories. In our CMM algorithm, new link proposals are generated using an appropriately tuned $S_F$ action, and are then subjected to an accept/reject step with respect to the full CYM action. The acceptance rates are tuned to stay within a range of 40-60%.

For both the Metropolis and the new algorithm, each configuration update consists of only one thermal update (no overrelaxation updates were performed). For each configuration we evaluated the CYM action $S_{\text{CYM}}$, which was then used to estimate its expectation value $\langle S_{\text{CYM}} \rangle$. We also evaluated plaquette and Polyakov loop traces. In each simulation we performed 398,000 measurements, after discarding the initial 2,000 configurations for equilibration. The simulation parameters and measured observables, together with their naive confidence intervals are shown in Table 3.

Both algorithms agree in terms of the measured values of $\langle S_{\text{CYM}} \rangle$, as can be seen in the first and third rows of Table 3. In the second row there is a clear discrepancy in the last significant digits, but this is very likely due to the fact that autocorrelations were not taken into account in the evaluation of the confidence intervals. From Fig.14 it is possible to see that $\bar{\tau}_{\text{int}}$ is very large in this example, which means that the confidence intervals on the second row of Table 3 are highly underestimated.

In terms of autocorrelations, the pseudo-heatbath algorithm for the $N_t = 1$ CYM theory performs much better than the (optimally tuned) Metropolis algorithm (see Figs.13–16). However, in the case $N_t = 3$, the pseudo-heatbath algorithm does not show an improvement over Metropolis (see Figs.17–18). A possible reason for this behaviour could be an excessive number of auxiliary variables in the $N_t = 3$ CYM theory, whose update could easily undermine the efficiency of the pseudo-heatbath algorithm. The smaller the number of auxiliary variables, the more efficient and faster the algorithm is. In sum, the new updating algorithm for CYM theories on $\mathbb{R}^3 \times S^1$ are most efficient when the compact direction is fully reduced. Fortunately, this is also the most interesting case for a CYM theory, as long as large $N$ volume-independence holds nonperturbatively for any volume.

Finally, Fig.19 provides a qualitative check for the compatibility of the results from the Metropolis and the pseudo-heatbath algorithms. The graphs show the MC histories of the
complex trace of the Polyakov loop, for Metropolis and pseudo-heatbath simulations of the $SU(5)$ CYM theory on a $10^3 \times 1$ lattice. The simulation with a ‘small’ deformation parameter $\vec{\alpha} = (0.20, 0.05)$ results in a deconfining vacuum ($\langle \text{Tr} \{\Omega_x\} \rangle \neq 0$), as expected for a CYM theory whose action is dominated by plaquette terms with a large $\beta_F$. The simulation with ‘large’ deformation parameters $\vec{\alpha} = (0.60, 0.10)$ results in a confining vacuum ($\langle \text{Tr} \{\Omega_x\} \rangle = 0$), as expected for a CYM theory whose action is dominated by the centre-stabilising double-trace terms. Both situations were correctly captured by the Metropolis and the pseudo-heatbath algorithm, which reinforces the validity of the latter.

5 Discussion

In this paper we constructed new algorithms for the update of the link variables for two classes of pure lattice gauge theories with nonlinear actions. The theories under consideration were (i) pure $SU(N)$ lattice gauge theories with a generic mixed Wilson action, and (ii) the lattice regularisation of centre-stabilised $SU(N)$ Yang-Mills (CYM) theories defined on $\mathbb{R}^{d-1} \times S^1$.

We used a generalisation of the Fabricius-Haan method of auxiliary variables in order to construct such algorithms. By adding enough extra degrees of freedom to the lattice gauge theory in question, we were able to get rid of the nonlinear terms in its action, and replace them with linear terms on the link variables. In this way it is possible to perform pseudo-heatbath, overrelaxation or cooling updates on the links of the nonlinear theories, just like in the standard $SU(N)$ lattice gauge theory with the fundamental Wilson action.

As a test for the accuracy of the new algorithms, we evaluated numerically the expectation values of some gauge-invariant observables and compared them with a Metropolis evaluation. Both quantitative and qualitative results showed a match between the outputs of the new pseudo-heatbath and Metropolis algorithms (modulo the autocorrelation correction of the confidence levels).

We also showed numerically that the new algorithms are more efficient than Metropolis. Despite the new updates, in general, being slower than Metropolis hits in CPU time (because of the large number of auxiliary variables that some lattice theories require), they also decorrelate the gauge configurations very fast. Therefore, when taking autocorrelations into account, the new algorithms perform much better than Metropolis.

The method of auxiliary variables is rather general and may be applied to other lattice theories with polynomial dependence on the link variables, for example the 3D Georgi-Glashow model\textsuperscript{8} the TEK–reduced 2D principal chiral model, etc. In relation to the CYM theories discussed in this paper, the method can also be extended to the more complicated case of centre-stabilised Yang-Mills theories defined on manifolds with multiple compactified directions, namely $\mathbb{R}^{d-k} \times (S^1)^k$. In particular, it could be used to construct an efficient MC algorithm for the zero-volume limit of CYM theories compactified on a $d$–torus, also known as deformed Eguchi-Kawai (DEK) models. An efficient algorithm for the DEK model would be very helpful in establishing nonperturbatively (via numerical simulations) if its centre symmetry is indeed intact, as claimed by Ünsal and Yaffe \cite{unsal-yaffe}, thus providing the first problem-free

\textsuperscript{8} We thank G. Bali for this suggestion.
matrix model representation of the planar sector of pure $SU(N)$ Yang-Mills theories on $\mathbb{R}^d$. We leave the construction of appropriate algorithms and the study of these theories to later publications.

**Acknowledgements**

We are very grateful to Mike Teper and Barak Bringoltz for useful discussions and feedback in the early stage of this project. We are also grateful for the hospitality at CERN, where part of this work was done.

Our lattice calculations were carried out on the Milipeia cluster at the University of Coimbra, and on PCs equipped with Intel® Pentium IV and Core Duo processors.

HV is supported by FCT (Portugal) under the grant SFRH/BPD/37949/2007.
A Monte Carlo algorithms

Here we summarise the MC algorithms for the different $SU(N)$ lattice gauge theories discussed in the paper.

A.1 Reduced lattices

MC algorithm for the fundamental Wilson action (1) on partially reduced lattices:

1. For each reduced plaquette $(\mu\nu,x)$:
   (a) Generate one random complex $N \times N$ matrix, $\tilde{Q}_{\mu\nu,x}$, with normal distribution $\mu_1(\tilde{Q}_{\mu\nu,x})$, using the Box-Müller transform (22).
   (b) Construct new auxiliary variables:
   \[ Q_{\mu\nu,x} = \left( \frac{N}{\beta F} \right)^{\frac{1}{2}} \tilde{Q}_{\mu\nu,x} + U_{\mu,x}U_{\nu,x+\hat{\mu}} + U_{\nu,x}U_{\mu,x+\hat{\nu}} \quad (122) \]

2. For each link variable $U_{\mu,x}$:
   (a) Construct the sum of ‘staples’:
   \[ V_{\mu,x} = \frac{\beta F}{N} \sum_{\nu=1}^{d} \left( U_{\nu,x}U_{\mu,x+\hat{\nu}}U_{\nu,x+\hat{\mu}} + U_{\nu,x-\hat{\nu}}U_{\mu,x-\hat{\nu}} \right) \]
   \[ + \frac{\beta F}{N} \sum_{\nu=1}^{d} \left( Q_{\mu\nu,x}U_{\nu,x+\hat{\mu}} + U_{\nu,x}Q_{\mu\nu,x} \right) \quad (123) \]
   where $Q_{\mu\nu,x} \equiv Q_{\nu\mu,x}$.
   (b) Update $U_{\mu,x}$ with respect to the probability distribution (6) using the Cabibbo-Marinari pseudo-heatbath algorithm.

A.2 Mixed actions

MC algorithm for generic mixed Wilson actions (28) on lattices with unreduced directions. The case of partially reduced directions is discussed in Section 3.5.

1. For each plaquette $p \equiv (\mu\nu,x)$:
   (a) Construct $U_p = U_{\mu,x}U_{\nu,x+\hat{\mu}}U_{\mu,x+\hat{\nu}}U_{\nu,x}$
   (b) For each irreducible representation $\mathcal{R}$ contributing to the mixed Wilson action:
      i. Generate $n_\mathcal{R}$ random complex $N \times N$ matrices $\tilde{Q}_p^{(i)}$ with the normal distribution $\mu_1(\tilde{Q}_p^{(i)})$, using the Box-Müller transform (22).
ii. Construct new auxiliary variables using the appropriate $\beta'_R, h^{(i)}_p$ (see Table 1):

$$Q^{(i)}_p = \frac{1}{\sqrt{2\beta'_R}} \bar{Q}^{(i)}_p + h^{(i)}_p \quad (124)$$

iii. Construct $f^R_p$ using the appropriate $g^{(i)}_p$ (see Table 1):

$$f^R_p = \sum_{i=1}^{n_R} g^{(i)}_p \quad (125)$$

(c) Construct $f_p$:

$$f_p = 2 \sum_R \beta'_R f^R_p \quad (126)$$

2. For each link variable $U_{\mu,x}$:

(a) Construct the sum of 'staples':

$$V_{\mu,x} = \sum_{(\nu \neq \mu)}^{d} (f_{\mu,\nu,x} U_{\mu,x+\nu} U^\dagger_{\nu,x+\mu} + U^\dagger_{\nu,x-\nu} f_{\nu,\mu,x-\nu} U_{\mu,x-\nu} U_{\nu,x-\nu+\mu}) \quad (127)$$

where $f_{\nu,\mu,x} \equiv f^\dagger_{\mu,\nu,x} \equiv f_p$.

(b) Update $U_{\mu,x}$ with respect to the probability distribution (6) using any of the following algorithms:

- Cabibbo-Marinari pseudo-heatbath
- $SU(2)$ or $SU(N)$ overrelaxation
- $SU(2)$ or $SU(N)$ cooling

**A.3 Double-trace deformations**

MC algorithm for lattice CYM theories on $\mathbb{R}^{d-1} \times S^1$, with the possibility of a fully reduced ‘compact’ direction.

1. For each $x \in \Lambda$ ($\sim \mathbb{R}^{d-1} \times S^1$):
   (a) If $N_t = 1$, do step (1) from the algorithm A.1

2. For each $x \in \Lambda_\perp$ ($\sim \mathbb{R}^{d-1}$):
   (a) Construct $\Omega_x = U_{d,x} U_{d,x+d} \cdots U_{d,x+(N_t-1)d}$

   (b) Generate a random complex $N \times N$ matrix, $\tilde{R}_{1,x}$, with normal distribution $\mu_1(\tilde{R}_{1,x})$, using the Box-Müller transform (22).
(c) Construct the new auxiliary variables:

\[ R_{1,x} = \left( \frac{N_1^d-1}{2N\alpha_1} \right)^{1/2} \tilde{R}_{1,x} + \Omega_x - \frac{1}{N} \text{Tr}\{\Omega_x\} \mathbb{1} \]  

(128)

(d) Construct \( Q^{(0)}_{1,x} = R_{1,x} - \frac{1}{N} \text{Tr}\{R_{1,x}\} \mathbb{1} \).

(e) For \( n = 2, \ldots, K \):

i. Construct \( \Omega^n_x \)

ii. Generate random complex \( N \times N \) matrices \( \tilde{R}_{n,x} \) and \( \tilde{Q}_{n,x}^{(K)} \) with normal distributions \( \mu_1(\tilde{R}_{n,x}) \) and \( \mu_1(\tilde{Q}_{n,x}^{(K)}) \), respectively, using the Box-Müller transform (22).

iii. Construct the new auxiliary variables:

\[ R_{n,x} = \left( \frac{N_d^d-1}{2N\alpha_n} \right)^{1/2} \tilde{R}_{n,x} + \Omega^n_x - \frac{1}{N} \text{Tr}\{\Omega^n_x\} \mathbb{1} \]  

(129)

\[ Q_{n,x}^{(K)} = \left( \frac{N_d^d-1}{2N\alpha_n} \right)^{1/2} \tilde{Q}_{n,x}^{(K)} + \frac{1}{N} \text{Tr}\{R_{n,x}\} \Omega^n_x \]  

(130)

iv. Construct \( Q^{(0)}_{n,x} = R_{n,x} - \frac{1}{N} \text{Tr}\{R_{n,x}\} \mathbb{1} \).

v. For \( m = 1, \ldots, n - 1 \):

A. Generate a random complex \( N \times N \) matrix \( \tilde{Q}_{n,x}^{(m)} \) with the normal distribution \( \mu_1(\tilde{Q}_{n,x}^{(m)}) \), using the Box-Müller transform (22).

B. Construct the new auxiliary variables:

\[ Q_{n,x}^{(m)} = \left( \frac{N_d^d-1}{2N\alpha_n} \right)^{1/2} \tilde{Q}_{n,x}^{(m)} + Q_{n,x}^{(m-1)} \Omega^n_x + \Omega^n_x - m \]  

(131)

(f) Construct \( f_x \):

\[ f_x = \alpha_1 Q^{(0)}_{1,x} + \sum_{n=2}^{K} \alpha_n \left( Q_{n,x}^{(n-1)} + \sum_{m=1}^{n-1} Q_{n,x}^{(m-1)} Q_{n,x}^{(m)} + \frac{1}{N} \text{Tr}\{R_{n,x}\} Q_{n,x}^{(K)} \right) \]  

(132)

3. For each link variable \( U_{\mu,x} \):

(a) Construct the sum of ‘staples’ \( V_{\mu,x} \) as in the step (2a) of the algorithm [A.1] and multiply it by \( \beta_F/N \):

\[ V_{\mu,x} \leftarrow \frac{\beta_F}{N} V_{\mu,x} \]  

(133)

(b) If \( \mu = d \), then :
i. Construct $V_{x}^{\text{def}}$:

\[
\begin{aligned}
\text{If } N_t > 1, \quad V_{x}^{\text{def}} &= \left( \mathcal{P} \prod_{i=0}^{t-1} U_{d,x+i\hat{d}} \right)^{\dagger} \cdot f_x \cdot \left( \mathcal{P} \prod_{i=t+1}^{N_t-1} U_{d,x+i\hat{d}} \right)^{\dagger} \\
\text{If } N_t = 1, \quad V_{x}^{\text{def}} &= f_x
\end{aligned}
\]  

(134)

where $x \equiv (x, t)$.

ii. Add $V_{x}^{\text{def}}$ to $V_x$:

\[
V_x \leftarrow V_x + \frac{2N}{N_t^{d-1}} V_{x}^{\text{def}}
\]  

(136)

(c) Update $U_{\mu,x}$ with respect to the probability distribution \(\langle 6 \rangle\) using any of the following algorithms:

- Cabibbo-Marinari pseudo-heatbath
- $SU(2)$ or $SU(N)$ overrelaxation
- $SU(2)$ or $SU(N)$ cooling
References

[1] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller and E. Teller, *Equation of State Calculations by Fast Computing Machines*, J. Chem. Phys. 21, 1087–1092 (1953).

[2] M. Creutz, *Monte Carlo Study Of Quantized SU(2) Gauge Theory*, Phys. Rev. D21, 2308–2315 (1980).

[3] K. Fabricius and O. Haan, *Heat Bath Method For The Twisted Eguchi-Kawai Model*, Phys. Lett. B143, 459 (1984).

[4] A. D. Kennedy and B. J. Pendleton, *Improved Heat Bath Method For Monte Carlo Calculations In Lattice Gauge Theories*, Phys. Lett. B156, 393–399 (1985).

[5] F. R. Brown and T. J. Woch, *Overrelaxed Heat Bath and Metropolis Algorithms for Accelerating Pure Gauge Monte Carlo Calculations*, Phys. Rev. Lett. 58, 2394 (1987).

[6] M. Creutz, *Overrelaxation and Monte Carlo Simulation*, Phys. Rev. D36, 515 (1987).

[7] M. Teper, *Instantons in the Quantized SU(2) Vacuum: A Lattice Monte Carlo Investigation*, Phys. Lett. B162, 357 (1985).

[8] E.-M. Ilgenfritz, M. L. Laursen, G. Schierholz, M. Muller-Preussker and H. Schiller, *First Evidence for the Existence of Instantons in the Quantized SU(2) Lattice Vacuum*, Nucl. Phys. B268, 693 (1986).

[9] N. Cabibbo and E. Marinari, *A New Method For Updating SU(N) Matrices In Computer Simulations Of Gauge Theories*, Phys. Lett. B119, 387–390 (1982).

[10] P. de Forcrand and O. Jahn, *Monte Carlo Overrelaxation for SU(N) Gauge Theories*, (2005), hep-lat/0503041.

[11] T. Eguchi and H. Kawai, *Reduction Of Dynamical Degrees Of Freedom In The Large N Gauge Theory*, Phys. Rev. Lett. 48, 1063 (1982).

[12] G. Bhanot, U. M. Heller and H. Neuberger, *The Quenched Eguchi-Kawai Model*, Phys. Lett. B113, 47 (1982).
[13] M. Ünsal and L. G. Yaffe, *Center-Stabilized Yang-Mills Theory: Confinement and Large N Volume Independence*, Phys. Rev. **D78**, 065035 (2008), 0803.0344.

[14] R. Stratonovich, *On a Method of Calculating Quantum Distribution Functions*, Soviet Phys. Doklady **2**, 416 (1958).

J. Hubbard, *Calculation of Partition Functions*, Phys. Rev. Lett. **3**, 77 (1959).

[15] G. E. P. Box and M. E. Müller, *A Note on the Generation of Random Normal Deviates*, Ann. Math. Statist. **29**(2), 610–611 (1958).

[16] M. Okawa, *Monte Carlo Study Of The Eguchi-Kawai Model*, Phys. Rev. Lett. **49**, 353 (1982).

[17] W. Bietenholz, J. Nishimura, Y. Susaki and J. Volkholz, *A Non-Perturbative Study Of 4D U(1) Non-Commutative Gauge Theory: The Fate Of One-Loop Instability*, JHEP **10**, 042 (2006), hep-th/0608072.

[18] I. Montvay and G. Münster, *Quantum Fields On A Lattice*, Cambridge, UK: Univ. Pr. (1994) 491 p. (Cambridge Monographs on Mathematical Physics).

[19] M. Hasenbusch and S. Necco, *SU(3) Lattice Gauge Theory With A Mixed Fundamental And Adjoint Plaquette Action: Lattice Artefacts*, JHEP **08**, 005 (2004), hep-lat/0405012.

[20] A. Bazavov, B. A. Berg and U. M. Heller, *Biased Metropolis-Heat-Bath Algorithm For Fundamental-Adjoint SU(2) Lattice Gauge Theory*, Phys. Rev. **D72**, 117501 (2005), hep-lat/0510108.

[21] A. González-Arroyo, private communication.

[22] P. Kovtun, M. Unsal and L. G. Yaffe, *Volume Independence In Large Nc QCD-Like Gauge Theories*, JHEP **06**, 019 (2007), hep-th/0702021.
| $k$ | $R$ | $\text{Aux.}$ | $h_p^{(i)}$ | $g_p^{(i)}$ | $\beta'_R$ |
|-----|-----|-------------|-------------|-------------|-------------|
| 0   | $\beta_A > 0$ | $z_p$ | $z_p$ | $\frac{\beta_A N^2}{N^2 - 1}$ |     |
|     | $\beta_A < 0$ | $Q_p$ | $U_p - \frac{1}{N} \text{Tr}(U_p) \mathbb{1}$ | $Q_p - \frac{1}{N} \text{Tr}(Q_p) \mathbb{1}$ | $\frac{|\beta_A| N}{N^2 - 1}$ |     |
| 2   | $\pm$ | $Q_p^{(1)}$ | $\frac{1}{2} (U_p + \sqrt{\frac{2}{3}} \text{Tr}(U_p^2) \mathbb{1})$ | $\frac{1}{2} (Q_p^{(1)} + \frac{1}{N} \text{Tr}(Q_p^{(1)}) \mathbb{1})$ | $\frac{2|\beta_p|}{N^2}$ |     |
|     | $\pm$ | $Q_p^{(2)}$ | $\frac{1}{2} (U_p + \sqrt{\frac{2}{3}} U_p^2)$ | $\frac{1}{2} (Q_p^{(2)} + \frac{1}{N} Q_p^{(2)})$ |     |     |
|     | $\pm$ | $Q_p^{(3)}$ | $\frac{1}{2} (U_p - \sqrt{\frac{2}{3}} \text{Tr}(U_p) \mathbb{1})$ | $\frac{1}{2} (Q_p^{(3)} - \frac{1}{N} \text{Tr}(Q_p^{(3)}) \mathbb{1})$ |     |     |
| 3   | $\pm$ | $Q_p^{(1)}$ | $U_p - \frac{1}{N} \text{Tr}(U_p) \mathbb{1}$ | $\frac{1}{2} (U_p + \frac{1}{N} U_p^2 + \frac{1}{N} \text{Tr}(U_p) \mathbb{1})$ | $\frac{3|\beta_p|}{N^2}$ |     |
|     | $\pm$ | $Q_p^{(2)}$ | $\frac{1}{2} (U_p + \frac{1}{N} U_p^2 + \frac{1}{N} \text{Tr}(U_p) \mathbb{1})$ | $\frac{1}{2} (Q_p^{(2)} + \frac{1}{N} Q_p^{(2)} + \frac{1}{N} \text{Tr}(Q_p^{(2)}) \mathbb{1})$ | $\frac{6|\beta_p|}{N^2}$ |     |
|     | $\pm$ | $Q_p^{(3)}$ | $\frac{1}{2} (U_p - \frac{1}{N} \text{Tr}(U_p) \mathbb{1})$ | $\frac{1}{2} (U_p - \frac{1}{N} U_p^2 + \frac{1}{N} \text{Tr}(U_p) \mathbb{1})$ | $\frac{3|\beta_p|}{N^2}$ |     |
|     | $\pm$ | $Q_p^{(4)}$ | $\frac{1}{2} (U_p - \frac{1}{N} U_p^2 + \frac{1}{N} \text{Tr}(U_p) \mathbb{1})$ | $\frac{1}{2} (U_p - \frac{1}{N} U_p^2 + \frac{1}{N} \text{Tr}(U_p) \mathbb{1})$ | $\frac{6|\beta_p|}{N^2}$ |     |
|     | $\pm$ | $Q_p^{(5)}$ | $\frac{1}{2} (U_p - \frac{1}{N} \text{Tr}(U_p) \mathbb{1})$ | $\frac{1}{2} (U_p - \frac{1}{N} U_p^2 + \frac{1}{N} \text{Tr}(U_p) \mathbb{1})$ | $\frac{3|\beta_p|}{N^2}$ |     |

Table 1: List of the expressions for $h_p^{(i)}$ and $g_p^{(i)}$, useful in the construction of MC algorithms for mixed Wilson actions with plaquettes in irreducible representations of $SU(N)$ with $N$-ality $k \leq 3$. The symbol ‘+’ stands for ‘symmetric representation’, either (2) or (3), while ‘−’ stands for ‘antisymmetric representation’, either (1, 1) or (1, 1, 1). In $N$-ality $k \geq 2$, two different algorithms are suggested for each representation; they differ in the number of auxiliary variables. For the adjoint representation, the cases of positive and negative $\beta_A$ are considered separately. Here, $p$ labels plaquettes in the hypercubic lattice, $\sigma$ denotes the sign of the lattice coupling, $\sigma \equiv \text{sgn}(\beta_R) = \pm 1$, and $\beta'_R$ is the redefined lattice coupling.
Table 2: Expectation values of plaquette characters in the ‘F + R’ mixed Wilson action, using a Cabibbo-Marinari-Metropolis algorithm (M) and a pseudo-heatbath algorithm proposed in this paper (H). For the mixed fundamental/adjoint Wilson action, we also include plaquette values that were calculated by Hasenbusch and Necco with a Cabibbo-Marinari-Metropolis algorithm [19]. The statistical errors in the table do not take autocorrelations into account, which is likely the reason for some discrepancies in the last significant digits.

| N | lattice | F + R | β_F | β_R | 〈1/N ReTr_FU_p⟩ | 〈1/β ReTr_RU_p⟩ | N_meas | Algor. |
|---|---------|-------|-----|-----|----------------|-----------------|--------|-------|
| 3 | 12^38   | F + A | 4.0 | 2.0 | 0.626587(7)    | 0.487024(7)     | 398,000 | M     |
|   |         |       |     |     | 0.626689(6)    | 0.487145(7)     | 398,000 | H     |
|   |         |       |     |     | ~ 0.6267       |                 |        |       |
| 3 | 16^4   | F + A | 9.25 | -3.56 | 0.53828(2)    | 0.36850(2)     | 398,000 | M     |
|   |         |       |     |     | 0.53816(2)    | 0.36834(3)     | 398,000 | H     |
|   |         |       |     |     | ~ 0.5383       | ~ 0.3685       |        |       |
| 4 | 8^4    | F + (2) | 10.665 | 0.3556 | 0.743870(5)  | 0.494378(7)   | 398,000 | M     |
|   |         |       |     |     | 0.743913(2)  | 0.494446(3)   | 398,000 | H     |
| 4 | 8^4    | F + (1,1) | 10.665 | 1.0668 | 0.671709(4)  | 0.587947(5)   | 398,000 | M     |
|   |         |       |     |     | 0.671735(3)  | 0.587978(3)   | 398,000 | H     |
| 5 | 8^4    | F + (3) | 16.665 | 4.1487 | 0.620798(5)  | 0.159693(5)   | 180,000 | M     |
|   |         |       |     |     | 0.620836(6)  | 0.159729(5)   | 180,000 | H     |
| 5 | 8^4    | F + (1,1,1) | 16.665 | 1.1853 | 0.571895(6)  | 0.428673(7)   | 180,000 | M     |
|   |         |       |     |     | 0.571923(7)  | 0.428706(8)   | 180,000 | H     |
| 5 | 8^4    | F + (2,1) | 16.665 | 4.7413 | 0.656146(5)  | 0.316318(7)   | 180,000 | M     |
|   |         |       |     |     | 0.656167(4)  | 0.316345(6)   | 180,000 | H     |

Table 3: Expectation values of the SU(5) CYM action for different compactification radii and α_n, using a Cabibbo-Marinari-Metropolis algorithm (M) and the pseudo-heatbath algorithm (H) proposed in this paper. The statistical errors in the table do not take autocorrelations into account, which is likely the reason for some discrepancies in the last significant digits. The discrepancy in the second row is due to a very slow equilibration of the Metropolis algorithm due to large correlations (see Fig.15).

| N | lattice | β | α_1 | α_2 | 〈S_{CYM}〉 | N_meas | Algor. |
|---|---------|---|-----|-----|-------------|--------|-------|
| 5 | 10^31  | 25.0 | 0.2 | 0.05 | -4.14109(9)   | 398,000 | M     |
|   |         |     |     |     | -4.14113(4)    | 398,000 | H     |
| 5 | 10^31  | 25.0 | 0.6 | 0.1  | -4.39116(20)  | 398,000 | M     |
|   |         |     |     |     | -4.39230(6)    | 398,000 | H     |
| 5 | 10^33  | 25.0 | 2.8 | 0.2  | -4.39339(2)   | 398,000 | M     |
|   |         |     |     |     | -4.39341(2)    | 398,000 | H     |
Figure 1: Estimator of the autocorrelation function, $C(t)$, vs. the CPU time, $t_{CPU}$, for the $SU(3)\,'F + A'$ mixed Wilson action, simulated on a $12^4$ lattice with $(\beta_F, \beta_A) = (4.00, 2.00)$, via a Cabibbo-Marinari-Metropolis algorithm (squares) and a Fabricius-Haan-type pseudo-heatbath algorithm (circles).

Figure 2: Estimator of the $t$-dependent integrated autocorrelation time, $\tau_{int}(t)$, vs. the MC time, $t_{MC}$, for the $SU(3) \,'F + A'$ mixed Wilson action, simulated on a $12^3$ lattice with $(\beta_F, \beta_A) = (4.00, 2.00)$, via a Cabibbo-Marinari-Metropolis algorithm (squares) and a Fabricius-Haan-type pseudo-heatbath algorithm (circles).
Figure 3: Estimator of the autocorrelation function, $C(t)$, vs. the CPU time, $t_{CPU}$, for the $SU(4) \ 'F + (2)'$ mixed Wilson action, simulated on a $8^4$ lattice with $(\beta_F, \beta_{(2)}) = (10.665, 0.3556)$, via a Cabibbo-Marinari-Metropolis algorithm (squares) and a Fabricius-Haan-type pseudo-heatbath algorithm (circles).

Figure 4: Estimator of the $t$-dependent integrated autocorrelation time, $\tau_{int}(t)$, vs. the MC time, $t_{MC}$, for the $SU(4) \ 'F + (2)'$ mixed Wilson action, simulated on a $8^4$ lattice with $(\beta_F, \beta_{(2)}) = (10.665, 0.3556)$, via a Cabibbo-Marinari-Metropolis algorithm (squares) and a Fabricius-Haan-type pseudo-heatbath algorithm (circles).
Figure 5: Estimator of the autocorrelation function, $C(t)$, vs. the CPU time, $t_{CPU}$, for the $SU(4)$ ‘$F + (1, 1)$’ mixed Wilson action, simulated on a $8^4$ lattice with $(\beta_F, \beta_{(1,1)}) = (10.665, 1.0668)$, via a Cabibbo-Marinari-Metropolis algorithm (squares) and a Fabricius-Haan-type pseudo-heatbath algorithm (circles).

Figure 6: Estimator of the $t$-dependent integrated autocorrelation time, $\tau_{int}(t)$, vs. the MC time, $t_{MC}$, for the $SU(4)$ ‘$F + (1, 1)$’ mixed Wilson action, simulated on a $8^4$ lattice with $(\beta_F, \beta_{(1,1)}) = (10.665, 1.0668)$, via a Cabibbo-Marinari-Metropolis algorithm (squares) and a Fabricius-Haan-type pseudo-heatbath algorithm (circles).
Figure 7: Estimator of the autocorrelation function, $C(t)$, vs. the CPU time, $t_{CPU}$, for the $SU(5)$ ‘$F + (3)$’ mixed Wilson action, simulated on a $8^4$ lattice with $(\beta_F, \beta_{(3)}) = (16.665, 4.1487)$, via a Cabibbo-Marinari-Metropolis algorithm (squares) and a Fabricius-Haan-type pseudo-heatbath algorithm (circles).

Figure 8: Estimator of the $t$-dependent integrated autocorrelation time, $\tau_{int}(t)$, vs. the MC time, $t_{MC}$, for the $SU(5)$ ‘$F + (3)$’ mixed Wilson action, simulated on a $8^4$ lattice with $(\beta_F, \beta_{(3)}) = (16.665, 4.1487)$, via a Cabibbo-Marinari-Metropolis algorithm (squares) and a Fabricius-Haan-type pseudo-heatbath algorithm (circles).
Figure 9: Estimator of the autocorrelation function, $C(t)$, vs. the CPU time, $t_{CPU}$, for the $SU(5)$ ‘$F + (1, 1, 1)$’ mixed Wilson action, simulated on a $8^4$ lattice with $(\beta_F, \beta_{(1, 1, 1)}) = (16.665, 1.1853)$, via a Cabibbo-Marinari-Metropolis algorithm (squares) and a Fabricius-Haan-type pseudo-heatbath algorithm (circles).

Figure 10: Estimator of the $t$-dependent integrated autocorrelation time, $\tau_{int}(t)$, vs. the MC time, $t_{MC}$, for the $SU(5)$ ‘$F + (1, 1, 1)$’ mixed Wilson action, simulated on a $8^4$ lattice with $(\beta_F, \beta_{(1, 1, 1)}) = (16.665, 1.1853)$, via a Cabibbo-Marinari-Metropolis algorithm (squares) and a Fabricius-Haan-type pseudo-heatbath algorithm (circles).
Figure 11: Estimator of the autocorrelation function, $C(t)$, vs. the CPU time, $t_{CPU}$, for the $SU(5)$ ‘$F + (2,1)$’ mixed Wilson action, simulated on a $8^4$ lattice with $(\beta_F, \beta_{(2,1)}) = (16.665, 4.7413)$, via a Cabibbo-Marinari-Metropolis algorithm (squares) and a Fabricius-Haan-type pseudo-heatbath algorithm (circles).

Figure 12: Estimator of the $t$-dependent integrated autocorrelation time, $\tau_{int}(t)$, vs. the MC time, $t_{MC}$, for the $SU(5)$ ‘$F + (2,1)$’ mixed Wilson action, simulated on a $8^4$ lattice with $(\beta_F, \beta_{(2,1)}) = (16.665, 4.7413)$, via a Cabibbo-Marinari-Metropolis algorithm (squares) and a Fabricius-Haan-type pseudo-heatbath algorithm (circles).
Figure 13: Estimator of the autocorrelation function, $C(t)$, vs. the CPU time, $t_{CPU}$, for the $SU(5)$ CYM theory, simulated on a $10^3$ lattice with $\beta_F = 25.0$ and $(\alpha_1, \alpha_2) = (0.20, 0.05)$, via a Cabibbo-Marinari-Metropolis algorithm (squares) and a Fabricius-Haan-type pseudo-heatbath algorithm (circles).

Figure 14: Estimator of the $t$-dependent integrated autocorrelation time, $\tau_{int}(t)$, vs. the MC time, $t_{MC}$, for the $SU(5)$ CYM theory, simulated on a $10^3$ lattice with $\beta_F = 25.0$ and $(\alpha_1, \alpha_2) = (0.20, 0.05)$, via a Cabibbo-Marinari-Metropolis algorithm (squares) and a Fabricius-Haan-type pseudo-heatbath algorithm (circles).
Figure 15: Estimator of the autocorrelation function, $C(t)$, vs. the CPU time, $t_{CPU}$, for the $SU(5)$ CYM theory, simulated on a $10^3 1$ lattice with $\beta_F = 25.0$ and $(\alpha_1, \alpha_2) = (0.60, 0.10)$, via a Cabibbo-Marinari-Metropolis algorithm (squares) and a Fabricius-Haan-type pseudo-heatbath algorithm (circles).

Figure 16: Estimator of the $t$-dependent integrated autocorrelation time, $\tau_{int}(t)$, vs. the MC time, $t_{MC}$, for the $SU(5)$ CYM theory, simulated on a $10^3 1$ lattice with $\beta_F = 25.0$ and $(\alpha_1, \alpha_2) = (0.60, 0.10)$, via a Cabibbo-Marinari-Metropolis algorithm (squares) and a Fabricius-Haan-type pseudo-heatbath algorithm (circles).
Figure 17: Estimator of the autocorrelation function, $C(t)$, vs. the CPU time, $t_{\text{CPU}}$, for the $SU(5)$ CYM theory, simulated on a $10^3 x 3$ lattice with $\beta_F = 25.0$ and $(\alpha_1, \alpha_2) = (2.80, 0.20)$, via a Cabibbo-Marinari-Metropolis algorithm (squares) and a Fabricius-Haan-type pseudo-heatbath algorithm (circles).

Figure 18: Estimator of the $t$-dependent integrated autocorrelation time, $\tau_{\text{int}}(t)$, vs. the MC time, $t_{\text{MC}}$, for the $SU(5)$ CYM theory, simulated on a $10^3 x 3$ lattice with $\beta_F = 25.0$ and $(\alpha_1, \alpha_2) = (2.80, 0.20)$, via a Cabibbo-Marinari-Metropolis algorithm (squares) and a Fabricius-Haan-type pseudo-heatbath algorithm (circles).
Figure 19: Trace of the holonomy, $\Omega_x$, around the compact direction of $\mathbb{R}^3 \times S^1$, in simulations of the $SU(5)$ CYM theory on a $10^8$ lattice with $\beta_F = 25.0$, using either a Fabricius-Haan-type pseudo-heatbath algorithm (left) or a Cabibbo-Marinari-Metropolis algorithm (right).