A NEW METHOD FOR UPDATING SU(N) MATRICES IN COMPUTER SIMULATIONS OF GAUGE THEORIES

Nicola CABIBBO
Dipartimento di Fisica, II Università di Roma [1], *Rome, Italy*
and INFN, Sezione di Roma, Rome, Italy

and

Enzo MARINARI
Istituto di Fisica "G. Marconi", Università di Roma, Rome, Italy
and INFN, Sezione di Roma, Rome, Italy

Received 12 July 1982

We present a new method for updating SU(N) matrices in lattice gauge theories simulations. The new method has been found for the case of SU(3) to be about three times more efficient than the Metropolis method.

In computer simulations for a lattice gauge theory the state of a lattice is represented by the values of the link variables U_ℓ, which are elements of the gauge group of interest.

The simulation is composed of many steps in each of which a single link variable is updated, i.e. given a new value (possibly coinciding with the old one). Processing once each link in the lattice is called an iteration.

Updating algorithms differ as to the efficiency with which they lead to thermal equilibrium. It seems to us clear that the most efficient algorithms are those of the heat bath type [1], where the new value of the link variable is independent of the old one.

A heat bath method for the SU(2) gauge theory with Wilson action has been given by Creutz [2,3]. This algorithm consists in replacing U_ℓ with a matrix chosen with a Boltzmann distribution:

$$P(U_\ell) = Z^{-1} \exp[-\beta S(U_\ell)]\, \mathrm{d}U_\ell \,, \qquad (1)$$

where $S(U_\ell)$ is the action for the lattice considered as a function of U_ℓ, the values of other link variables being fixed at their present value, $\mathrm{d}U_\ell$ is the Haar measure on the group and

$$Z = \int \mathrm{d}U_\ell \exp[-\beta S(U_\ell)] \,. \qquad (2)$$

In the case of SU(2) a simple algorithm exists [3] for producing unitary matrices with the distribution of eq. (1).

The algorithm to produce directly $N \times N$ unitary matrices according to the distribution (1) becomes more and more complicated for increasing N. The algorithm has been explicitly built by Pietarinen [4] for $N = 3$ and applied in an actual simulation of a 10^4 lattice. The extension to larger N seems to be very difficult.

Most simulations for SU(3) and all those of larger groups have used an alternative updating method based on the Metropolis algorithm [5]. In this method the new value of U_ℓ is obtained by multiplying the old value by one of a set of matrices $\{1, u^{(i)}\ (i = 1, ..., 2M)\}$. The set contains the unit matrix, M suitably choosen unitary matrices, and their inverses. After the updating the new link variable U_ℓ can assume one of $2M + 1$ values.

In a typical application of the Metropolis algorithm for SU(3) the updating procedure is repeated L times ($L \sim 10$) on each link before proceeding to the next link. With a suitable choice of the $u^{(i)}$ – the optimum

[1] Present address: Istituto di Fisica G. Marconi, Università degli Studi, P.le Aldo Moro, 2, 00185 Rome, Italy.

0 031-9163/82/0000–0000/$02.75

is β-dependent – one obtains a thermalization of the link variable which approaches that obtained with the heat bath method. The two methods are exactly equivalent in the limit $L \to \infty$.

In this paper we propose a new method for updating SU(N) matrices which is a natural extension of the Creutz method for SU(2).

Tests executed on a 4^4 lattice in SU(3) indicate that the method is more efficient that the Metropolis method: the new method led to a 40% saving in the computer time used for one iteration, and the thermalization is achieved faster, as indicated by a flatter hysteresis cycle during fast thermal excursions.

Furthermore the new method can be applied with small programming effort to any value of N. We propose to use it for a study of some properties of the large N limit.

The new method consists of the following steps:

(a) select a set F of SU(2) subgroups of SU(N):

$$\{\mathrm{F}: \mathrm{SU}(2)_k, k = 1, ..., m\}, \tag{3}$$

such that there is no left ideal, i.e. no subset of SU(N) which is invariant under left multiplication by F, except the whole group. A minimal, simple choice of F contains the set of SU(2) subgroups which have elements of the form:

$$a_k = \begin{pmatrix} 1 & & & & 0 \\ & \ddots & & & \\ & & 1 & & \\ & & & \boxed{\alpha_k} & \\ & & 1 & & \\ & & & \ddots & \\ 0 & & & & 1 \end{pmatrix}, \tag{4}$$

where α_k is a 2×2 unitary unimodular matrix located at the kth and $(k+1)$th rows and columns. In this case $m = N - 1$.

(b) In each step of the iteration the new link variable is obtained by multiplying the previous value by m matrices belonging to the m subgroups $\mathrm{SU}(2)_k$:

$$U' = a_m a_{m-1} \ldots a_1 U,$$

$$a_k \in \mathrm{SU}(2)_k \quad (k = 1, ..., m). \tag{5}$$

It is convenient to define

$$U^{(k)} = a_k a_{k-1} \ldots a_1 U, \quad U^{(0)} = U,$$

so that

$$U' = U^{(m)}, \quad U^{(k)} = a_k U^{(k-1)}.$$

The element a_k is chosen at random (a_1 is chosen first, then a_2, etc.) with the measure:

$$\mathrm{d}P(a_k) = \mathrm{d}^{(k)} a_k \times \exp[-\beta S(a_k U^{(k-1)})]/Z_k(U^{(k-1)}), \tag{6}$$

where $\mathrm{d}^{(k)} a_k$ is the Haar measure on $\mathrm{SU}(2)_k$ and

$$Z_k(U) = \int_{\mathrm{SU}(2)_k} \mathrm{d}a \exp[-\beta S(aU)]. \tag{7}$$

Note that $Z_k(U)$ is invariant under left multiplication by an element of $\mathrm{SU}(2)_k$:

$$Z_k(bU) = Z_k(U), \quad \text{if } b \in \mathrm{SU}(2)_k. \tag{8}$$

Before describing the random choice algorithm (which is essentially equivalent to that of ref. [3]) we have to prove that the procedure outlined above leads to thermalization.

Since the set F has no left ideal it is sufficient to prove that, if U has a Boltzmann distribution, then U' will also have a Boltzmann distribution. This is simply shown by using the invariance properties of the Haar measure under left multiplication and inversion. Assume that $U^{(k-1)}$ has a Boltzmann distribution:

$$\mathrm{d}P(U^{(k-1)}) = \exp[-\beta S(U^{(k-1)})]/Z_N. \tag{9}$$

The distribution for $U^{(k)}$ will then be:

$$\mathrm{d}P(U^{(k)}) = \int_{\mathrm{SU}(2)_k} \mathrm{d}^{(k)}(\alpha) \frac{\exp[-\beta S(U^{(k)})]}{Z_k(a^{-1}U^{(k)})} \mathrm{d}(a^{-1}U^{(k)}) \times \exp[-\beta S(a^{-1}U^{(k)})]/Z_N, \tag{10}$$

and is easily seen to coincide with the Boltzmann measure.

This follows from eq. (8), and from

$$\mathrm{d}(a^{-1}U^{(k)}) = \mathrm{d}U^{(k)}, \quad \mathrm{d}^{(k)}(a) = \mathrm{d}^{(k)}(a^{-1}). \tag{11}$$

So if U has a Boltzmann distribution, $U^{(1)}, U^{(2)}, \ldots, U^{(m)} = U'$ will also have a Boltzmann distribution.

If S is identified with the Wilson action we can write:

$$S(U) = \sum_{\mathrm{p}} \mathrm{Re}\,(\mathrm{Tr}\, U \tilde{U}_{\mathrm{p}}) + \text{constant} = \mathrm{Re}\,(\mathrm{Tr}\, UR) + \text{constant}, \tag{12}$$

where the sum is over the plaquettes which contain the link being updated, and $\tilde{U}_p$ is – for each plaquette – the ordered product of the other link variables.

In order to generate the distribution in eq. (6) for the minimal set F described above we note that

$$S(a_k U) = \mathrm{Re}\,(\mathrm{Tr}\, a_k U R) = \mathrm{Re}\,(\mathrm{Tr}\, \alpha_k r_k) + \text{terms independent of } \alpha_k\,, \tag{13}$$

where r_k is the $z \times z$ submatrix of R which has the same block structure as a_k.

We can write:

$$r_k = i\boldsymbol{r}\cdot\boldsymbol{\sigma} + r_0 \mathbf{1}\,, \quad \alpha_k = i\boldsymbol{\alpha}\cdot\boldsymbol{\sigma} + \alpha_0 \mathbf{1}\,,$$
$$(\alpha_0, \boldsymbol{\alpha} \text{ real}, \alpha_0^2 + (\boldsymbol{\alpha})^2 = \mathbf{1})\,, \tag{14}$$

so that

$$\mathrm{Re}\,(\mathrm{Tr}\, \alpha_k r_k) = 2(\alpha_0\, \mathrm{Re}\, r_0 - \boldsymbol{\alpha}\cdot \mathrm{Re}\, \boldsymbol{r})\,. \tag{15}$$

The problem reduces to that of generating unit four vectors $(\alpha_0, \boldsymbol{\alpha}) = \alpha$ with the distribution

$$\mathrm{d}P(\alpha) \propto \exp(-2\beta(\alpha_0\, \mathrm{Re}\, r_0 - \boldsymbol{\alpha}\cdot \mathrm{Re}\, \boldsymbol{r}) \times \delta(\alpha_0^2 + (\boldsymbol{\alpha})^2 - \mathbf{1})\, \mathrm{d}^4\alpha\,. \tag{16}$$

A simple algorithm for doing this has been described by Creutz in ref. [3].

The most time-consuming part of the algorithm is the computation of R for each link. This is done only once at each step in the iteration. It may be convenient to capitalize on the time spent in computing R by enlarging the set F. An obvious extension would be to include in F also all the $m = [N(N-1)/2]$ SU(2) subgroups whose significant elements are at the crossing of the i, k rows and columns, with i and k not necessarily contiguous.

In order to test our method we have made some runs on a 4^4 lattice with an SU(3) gauge theory and the Wilson action.

The method [with the minimal choice for F given in eq. (4)] has been compared with an optimized Metropolis algorithm [‡1].

We have made long runs (1000 iterations) with the two methods at $\beta = 5.5$ and $\beta = 6$. The average of the last 600 iterations on each run gave the results in table 1, which indicate that the methods give equivalent results. The new method resulted in 40% saving of computer time. A significant test of efficiency of any lattice algorithm is the ability to follow rapid β variations from iteration to iteration. To test this, we have run the two methods through a thermal cycle from $\beta = 2$ to $\beta = 8$ and back in 120 iterations. The results are given in fig. 1a and 1b. We have then repeated the cycles in 240 iterations, with results given in figs. 2a and 2b. In these figures we report also the value of the average plaquette energy at $\beta = 8$ obtained in a separate long run.

We note that:

(1) The area of 240 run with the new method is half of that of the Metropolis sweep.

(2) The area of 120 run with the new method is equal to that of the 240 Metropolis sweep.

(3) The area of the 120 Metropolis sweep is not comparable to the others because of larger distance from the correct $\beta = 8$ value.

From these results we are led to conclude that our algorithm is twice as efficient, iteration by iteration, then the optimized Metropolis algorithm. Taking into account the 40% saving of CPU time per iteration, our algorithm seems to be about three times faster.

[‡1] For the Metropolis algorithm we have used a program written by E. Marinari and C. Rebbi.

Table 1
Numerical results for the mean plaquette energy and specific heat in SU(3) with Wilson action over a 4^4 lattice. The results were obtained as the average over the last 600 of 1000 iterations for each of the two methods, at each value of β. The errors were evaluated by the standard deviation of subsamples of 100 iterations. We have defined: $C_s = \beta^2(\langle E_L^2\rangle - \langle E_L\rangle^2)/(6n^4)$, n^4 being the number of sites in the lattice, and E_L the total energy of the lattice.

	$\beta = 5.5$		$\beta = 6.0$	
	$\langle E\rangle$	$\langle C_s\rangle$	$\langle E\rangle$	$\langle C_s\rangle$
new method	0.506 ± 0.003	9.2 ± 1.4	0.4027 ± 0.0006	2.9 ± 0.3
Metropolis	0.504 ± 0.003	7.3 ± 1.6	0.4029 ± 0.0009	3.1 ± 0.3

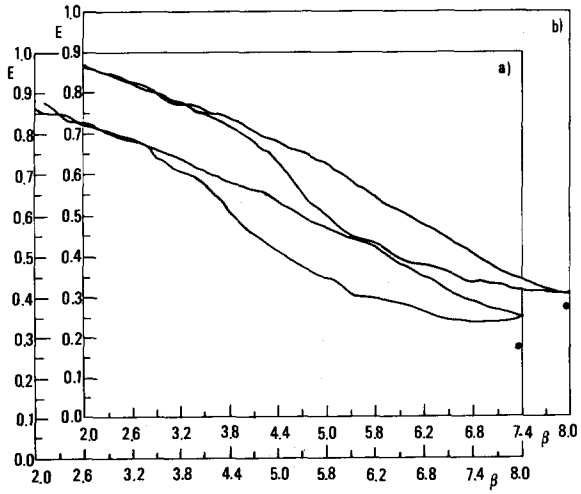

Fig. 1. (a) Thermal cycle with 120 iteration and Metropolis algorithm. The dot indicates the correct value at $\beta = 8$. (b) As in (a) new algorithm.

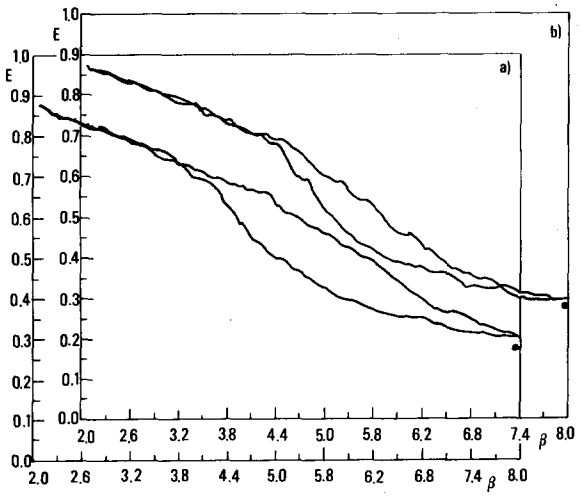

Fig. 2. (a) As in fig. 1(a), but 240 iterations. (b) As in (a) but 240 iterations, new algorithm.

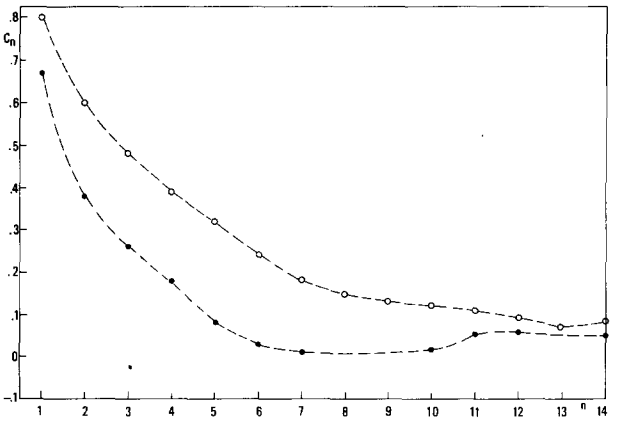

Fig. 3. Correlation function C_n [see eq. (17)]. Black dots: new method. Open circles: Metropolis.

As a further check of the relative speed of the two methods, we have computed for the 1000 iterations runs at $\beta = 6$ mentioned above the correlation function:

$$C_n = (\langle E_t E_{t+n}\rangle - \langle E_t\rangle^2)/(\langle E_t^2\rangle - \langle E_t\rangle^2)\,, \qquad (17)$$

where E_t is the average plaquette energy for the tth configuration, and $\langle\ \rangle$ indicates an average over the last 600 configurations. The results in fig. 3 indicate that both algorithms display a certain "memory" from iteration to iteration, but the new algorithm wipes out this memory in half the number of iterations.

References

[1] M. Creutz, L. Jacobs and C. Rebbi, Phys. Rev. Lett. 42 (1979) 1390.
[2] M. Creutz, Phys. Rev. Lett. 43 (1979) 553.
[3] M. Creutz, Phys. Rev. D21 (1980) 2308.
[4] E. Pietarinen, Nucl. Phys. B190 FS3 (1981) 349.
[5] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller and E. Teller, J. Chem. Phys. 21 (1953) 1087.