PHYSICAL REVIEW D, VOLUME 63, 054505

Noisy Monte Carlo algorithm reexamined

T. Bakeyev
Joint Institute for Nuclear Research, 141980 Dubna, Russia

Ph. de Forcrand
Institut für Theoretische Physik, ETH-Hönggerberg, CH-8093 Zürich, Switzerland
and CERN, Theory Division, CH-1211 Geneva 23, Switzerland

(Received 8 August 2000; published 1 February 2001)

We present an exact Monte Carlo algorithm designed to sample theories where the energy is a sum of many couplings of decreasing strength. Our algorithm, simplified from that of Lin, Liu, and Sloan, avoids the computation of almost all nonleading terms. We illustrate its use by simulating SU(2) lattice gauge theory with a five-loop action, and discuss further applications to full QCD.

DOI: 10.1103/PhysRevD.63.054505 PACS number(s): 11.15.Ha, 11.10.Wx, 11.30.Rd, 12.38.Gc

I. INTRODUCTION

When sampling by Monte Carlo simulation the partition function $Z=\int \Pi\, dU\, e^{-H(\{U\})}$, the most common algorithm is that of Metropolis [1]. At each step, starting from the current configuration $\{U\}$, a candidate configuration $\{U'\}$ is proposed, and it is accepted with a probability,

$$P_{\rm acc}=\min(1,e^{-(H(\{U'\})-H(\{U\}))}). \tag{1}$$

This acceptance test is realized by comparing the right-hand side of (1) to a random number uniformly distributed in [0, 1]. This seems like a waste of information. Why compute $H(\{U'\})$ *exactly*, then compare it with a random number? It should be sufficient to *estimate* it. Indeed, this logical proposition has been studied several times [2,3,4]. Two difficulties have been identified, both caused by the nonlinear relationship between the energy H and the probability $\propto e^{-H}$: (i) What is needed is an unbiased estimate of e^{-H}, which must be obtained from unbiased estimate(s) of H; (ii) To be interpreted as a probability, the noisy estimator of $P_{\rm acc}$ must be bounded, and in particular stay positive. Difficulty (i) was overcome in Ref. [3], which however showed that violations of (ii) caused intolerable systematic errors unless the amount of noise in the estimate of H was minuscule. Difficulty (ii) was overcome in [4], which showed that exact results could be obtained even in the presence of a large amount of noise in the estimate of H. Reference [4], however, introduces an infinite number of auxiliary variables for each term in the Hamiltonian, and tests of the method are performed on a toy model with five degrees of freedom only. Here, we simplify the method of [4] by introducing only one auxiliary variable per term in H. Moreover, we separate H into a leading part to be calculated exactly, and a sum of small correction terms which we treat stochastically. This separation is essential; because stochastic estimates are used for correction terms only, large amounts of noise can be tolerated. As a consequence, our algorithm is a very efficient approach to the simulation of complicated Hamiltonians.

Consider a generic Hamiltonian of the type

$$H=\sum_{k=0}^{m} c_k W_k\,, \tag{2}$$

where as k increases, $|c_k|$ decreases and the successive terms W_k typically become less and less local. For instance, in a spin model $\{\vec{\sigma}_i\}$, W_0 would be the nearest-neighbor interaction $\Sigma_{\langle ij\rangle}\vec{\sigma}_i\cdot\vec{\sigma}_j$, W_1 would represent next-nearest-neighbor interactions, etc. Here, we will illustrate our method for lattice gauge theory. In that context, W_k are the traces of Wilson loops of increasing size: $W_0=\Sigma_{x,\mu,\nu}{\rm Tr}\,\Pi_4 U$ around elementary plaquettes, $W_{1,2,3}$ correspond to different geometries of six-link loops, etc. It is often the case that one would like to study a Hamiltonian of type (2) resulting from an expansion, be it perturbative [5], nonperturbative [6], or based on the fixed point of a renormalization group transformation [7]. In all these situations, the expansion is truncated to a maximal order m dictated by technical reasons. As k increases in (2), the number of geometrically equivalent terms grouped into W_k increases exponentially. In a spin model on a hypercubic lattice in d dimensions, each spin has $2d$ nearest-neighbors (these interactions are grouped into W_0), $d(d-1)/2$ next-nearest neighbors (grouped into W_1), $d(d-1)(d-2)/6$ 3rd-neighbors, etc. This combinatoric explosion normally makes the simulation of extended Hamiltonians prohibitively expensive. This is the reason for a truncation to very low m, often taken to be 1 or 2. However, in most cases, the couplings c_k in (2) decrease exponentially with k, so that the overall Hamiltonian is dominated by W_0, with small corrections. In lattice field theory, this is actually required if the Hamiltonian is to make sense and tend to a local operator as the continuum limit is approached. By making use of stochastic methods to estimate the correction terms W_k, $k\geqslant 1$, we aim at postponing the combinatoric explosion of the simulation costs incurred when including higher terms W_k. This opens the possibility of studying numerically much more complicated Hamiltonians, including higher-order correction terms. In lattice field theory, these correction terms are crucial to suppress discretization errors and form the building blocks of so-called ''improvement'' strategies. Also, the inclusion of higher-order terms can be

very useful in the approaches to the fermion determinant simulations involving the loop expansion [8].

We present our method in Sec. II, and illustrate it in Sec. III with simulations of a five-loop perturbatively improved action for SU(2) lattice theory. We conclude with prospective applications of our method, in particular for dynamical fermion QCD simulations.

II. NOISY MONTE CARLO ALGORITHM: THE METHOD

Given the Hamiltonian (2), let us suppose that the terms $c_k W_k$ are nonpositive starting from $k=1$:

$$k \geqslant 1: \quad c_k W_k(U) \leqslant 0 \quad \forall U. \tag{3}$$

This can be easily arranged by adding to each term of the Hamiltonian a nonessential constant. Here U are the fields of the model under consideration. The key idea of the method is to estimate the contribution of the terms $W_k(U)$, $k \geqslant 1$ stochastically by introducing auxiliary fields. This will lead to a significant reduction of computational effort if the coefficients c_k, $k \geqslant 1$ are small enough. In all cases, the algorithm remains exact.

We introduce auxiliary fields σ_k, $k \geqslant 1$ (associated with the terms W_k), which can take two values: 0 and 1. Using the identity

$$a+b = \sum_{\sigma=0,1} [a * \delta_{\sigma,0} + b * \delta_{\sigma,1}] \tag{4}$$

we represent the probability e^{-H} in the form

$$e^{-H} = P_0[U] * P_1[U,\sigma], \tag{5}$$

where

$$P_0[U] = e^{-c_0 W_0(U)};$$

$$P_1[U,\sigma] = \prod_{k=1}^{m} \sum_{\sigma_k=0,1} [\delta_{\sigma_k,0} + \delta_{\sigma_k,1}(e^{-c_k W_k(U)} - 1)]. \tag{6}$$

The right-hand side of (5) can be interpreted as the joint probability distribution for the original fields of the model and the new σ fields. Because of the inequalities (3) this distribution is well defined: $P_1[U,\sigma] \geqslant 0 \ \forall \{U,\sigma\}$, and the probabilities for σ_k to take value 0 or 1 when the U fields are fixed lie in the interval [0, 1]:

$$p_{\sigma_k=0} = e^{c_k W_k(U)}; \quad p_{\sigma_k=1} = 1 - e^{c_k W_k(U)}. \tag{7}$$

This means that our algorithm has no probability bound violations, which plagued previous attempts to construct an efficient stochastic algorithm [2,3].

One can easily see why the introduction of auxiliary σ fields can be useful. Starting from the current $\{U_1,\sigma\}$ configuration, a candidate configuration $\{U_2\}$ distributed with the weight $P_0[U_2]$ is proposed, and accepted with probability

$$P_{\rm acc} = \min\left(1, \frac{P_1[U_2,\sigma]}{P_1[U_1,\sigma]}\right) = \min\left(1, \prod_{k:\sigma_k=1} \frac{e^{-c_k W_k(U_2)} - 1}{e^{-c_k W_k(U_1)} - 1}\right). \tag{8}$$

Since the terms $c_k W_k(U)$ contribute in $P_{\rm acc}$ only if $\sigma_k = 1$, the amount of computational work is greatly reduced if the configurations with $\sigma_k = 0$ are dominating. That is certainly the case when the absolute values of the coupling coefficients $|c_k|$ are small. The probabilities for σ_k to be unity, averaged over $\{U\}$ configurations, are negligible then. Indeed, to leading order in c_k the average probability $p_{\sigma_k=1}$ from Eq. (7) can be written as

$$\langle p_{\sigma_k=1} \rangle \approx -c_k \langle W_k(U) \rangle \approx 0 \quad \text{if} \quad c_k \approx 0. \tag{9}$$

Expression (9) also suggests that one should try to make $|\langle W_k \rangle|$ as small as possible, using the freedom one has to shift W_k by a constant. This goal should remain compatible, however, with inequalities (3); otherwise, probability bound violations will appear for p_{σ_k} and $P_{\rm acc}$ in Eqs. (7) and (8).

Actually, the violation of conditions (3) is not completely forbidden. As it was pointed out in Ref. [4], one can address the problem of the lower probability-bound violations by redefining the measure. If the distribution $P_1[U,\sigma]$ in (5) can be negative for some configurations $\{U,\sigma\}$, one can effectively simulate with the probability distribution $P_0[U] * |P_1[U,\sigma]|$ instead and include the sign $\mathrm{sgn}(P_1)$ into the observable expectation value:

$$\langle O \rangle = \frac{\langle O \, \mathrm{sgn}(P_1) \rangle_{\|}}{\langle \mathrm{sgn}(P_1) \rangle_{\|}}, \tag{10}$$

where by $\langle \, \rangle_{\|}$ we denote the averages with respect to distribution $P_0[U] * |P_1[U,\sigma]|$. Sometimes the admission of very rare sign violations can substantially decrease the probability $p_{\sigma_k=1}$. However, one should be very careful in using this trick: As the volume of the system increases, one needs an exponential growth of statistics to estimate $\langle \mathrm{sgn}(P_1) \rangle_{\|}$ within the same accuracy. In the following, we shall always assume fulfillment of the inequalities (3).

After updating the U fields, one should also update the σ fields to preserve ergodicity. This requires the calculation of probabilities (7). At this point the reader might say: ''Okay, one saves computational effort by not calculating some terms W_k in expression (8) while estimating $P_{\rm acc}$. Nevertheless, one must calculate these terms when updating the σ fields! So does one gain anything in the end?'' The answer is ''yes'' for the following two reasons.

First, the terms W_k, for which it is reasonable to use the stochastic estimation, usually couple many degrees of freedom (this is due to the usual nonlocality of weakly coupled terms, which serve as corrections to more local leading terms in the Hamiltonian). If one uses usual local algorithms (without introducing stochastic σ variables), one should estimate the term W_k each time one updates a degree of freedom to which it couples. Contrary to that, if one uses a noisy Monte Carlo algorithm, the probabilities (7) should be calculated only once per σ update.

Secondly, the variables σ_k can be refreshed infrequently, the more so as the associated coupling c_k gets smaller. This will be demonstrated in the next section on a particular example. This slow dynamics of the auxiliary σ fields does not imply slow dynamics of the physically relevant U fields.

Up to now we were quite generic, showing that the noisy Monte Carlo (NMC) method can be potentially very effective for the variety of theories, where the energy (2) is a sum of couplings of decreasing strength. In the next section we illustrate these ideas on a particular example: a five-loop perturbatively improved SU(2) Yang-Mills model.

III. FIVE-LOOP SU(2) GAUGE THEORY

We consider a five-loop SU(2) gauge action in four dimension $(4d)$:

$$s=\sum_{i=1}^{5} c_i \frac{1}{m_i^2 m_i^2} S_{m_i,n_i}, \tag{11}$$

where the indices $(m_i,n_i)=(1,1),(2,2),(1,2),(1,3),(3,3)$ for $i=1,\ldots,5$ denote the planar, fundamental loops of size $m\times n$

$$S_{m_i,n_i}=\sum_{x,\mu,\nu}\left(-2*\mathrm{sgn}\,(c_i)-\frac{\mathrm{Tr}}{2}\left(\begin{array}{c}\nu\\ x\end{array}\!\!\square\!\!\begin{array}{c}\\ \mu\end{array}+\begin{array}{c}\nu\\ \square\\ x\ \ \mu\end{array}\right)\right) \tag{12}$$

The Gibbs factor is $\exp(-(\beta/2)S)$. Note that in Eq. (12) we have arranged the constant term $-2*\mathrm{sgn}(c_i)$ to ensure the condition (3) for elementary action terms corresponding to each loop:

$$\forall\{i,\mu,\nu,x,U\}\quad S_{i,\mu,\nu,x}\equiv\frac{1}{m_i^2 n_i^2}c_i\left(-\mathrm{sgn}\,(c_i)-\frac{\mathrm{Tr}}{2}\begin{array}{c}\nu\\ x\end{array}\!\!\square\!\!\begin{array}{c}\\ \mu\end{array}\right)\le 0\ . \tag{13}$$

Using the results of [9], one can construct a one-parameter set of actions that have no $\mathcal{O}(a^2)$ and $\mathcal{O}(a^4)$ corrections:

$$\begin{aligned} &c_1=(19-55c_5)/9,\quad c_2=(1-64c_5)/9,\\ &c_3=(-64+640c_5)/45,\quad c_4=1/5-2c_5. \end{aligned} \tag{14}$$

Here we take $c_5=1/20$ (the same action was used in the context of improved cooling in Ref. [10]) and $\beta=2.4$.

Following the ideas of Sec. II, we estimate the contribution of all loops except the plaquette stochastically. For each loop $l\equiv\{\mu,\nu,x\}$ of sort $2\leqslant i\leqslant 5$ we introduce the auxiliary variable $\sigma_i(l)=0,1$, and rewrite the contribution of this loop to Gibbs factor in the form

$$e^{-(\beta/2)S_{i,\mu,\nu,x}}=\sum_{\sigma_i(l)=0,1}\left[\,\delta_{\sigma_i(l),0}+\delta_{\sigma_i(l),1}(e^{-(\beta/2)S_{i,\mu,\nu,x}}-1)\right]. \tag{15}$$

The resulting distribution of $\{U,\sigma\}$ fields is used for the generation of independent $\{U\}$ configurations. We shall say that for a given $\{\sigma\}$ configuration the loop $\{l,i\}$ is ''active'' if $\sigma_i(l)=1$.

TABLE I. Average value of σ_i field for each loop of sort i.

Loop	1×2	1×3	2×2	3×3
$\langle\sigma\rangle$	0.0753	0.0199	0.0202	0.0018

Let us describe the updating procedure in the $\{U,\sigma\}$ configuration space. Consider first the local updating of the gauge fields U when the σ fields are fixed. The proposal value $U^{\mathrm{new}}_{x,\mu}$ at a given link $\{x,\mu\}$ is generated by heatbath with respect to the measure

$$P_0[U]\propto\exp\left(-\frac{\beta}{2}c_1 S_{1,1}[U]\right), \tag{16}$$

where $S_{1,1}$ is the plaquette action [see (12)], and then accepted with probability

$$P_{\mathrm{acc}}=\min\left(1,\frac{P_1[U^{\mathrm{new}}_{x,\mu},\sigma]}{P_1[U^{\mathrm{old}}_{x,\mu},\sigma]}\right), \tag{17}$$

where

$$P_1[U,\sigma]=\prod_{\substack{(l,i)\ni\{x,\mu\}\\ \sigma_i(l)=1}}(e^{-(\beta/2)S_{i,l}[U]}-1)\,. \tag{18}$$

Only active loops that contain the given link $\{x,\mu\}$ contribute to the expression in the right-hand side of (17).

After each N_i updates of fields U on the entire lattice we update the σ fields of sort i. For each loop l we assign the values 0, 1 to the variable $\sigma_i(l)$ with the following probabilities:

$$p_{\sigma_i(l)=0}=\exp\left(\frac{\beta}{2}S_{i,l}[U]\right);\quad p_{\sigma_i(l)=1}=1-\exp\left(\frac{\beta}{2}S_{i,l}[U]\right). \tag{19}$$

Due to the absence of interaction between different σ variables, the probabilities (19) depend only on the gauge configuration, so that σ variables can be updated independently.

In our simulations, we have measured the average values of σ_i, $2\leqslant i\leqslant 5$ that are listed in Table I. They are quite small, and very close to the perturbative estimate (9). This shows that one can avoid the computation of almost all of the extended ''staples'' in the U update.

Performing numerical simulations for the five-loop model (11,14) with auxiliary σ fields, we were mainly interested in the efficiency of our new NMC algorithm. One can estimate the efficiency of the NMC method by comparing it with the updating procedures which are commonly used now for the simulation of multiloop actions like (11). In the following we label these usually applied techniques with the collective name ''usual Monte Carlo'' (UMC) method, to contrast it with the NMC method.

We compare the computer times needed to get the same results with NMC and UMC algorithms as follows. First, we make an analytic estimation of the total computational cost of one update of the U fields for both algorithms in units of

TABLE II. Integrated autocorrelation times for average loop traces in units of U updates for UMC algorithm (first column) and for NMC algorithm with different frequencies of σ updates for 1×2, 1×3, 2×2 and 3×3 loops (other columns). The last row presents the naive gain for the NMC algorithm (26).

Number of U updates per 1 σ update	UMC no σ	1 for all	5 for all	5 for 1×2 15 for 1×3, 2×2 105 for 3×3	10 for all	10 for 1×2 30 for 1×3, 2×2 210 for 3×3	20 for all	30 for all	40 for all	50 for all
$\tau_{\rm int}(1\times1)$	0.7(1)	1.9(2)	2.3(1)	2.5(2)	3.1(2)	3.2(2)	4.3(4)	4.5(4)	3.8(2)	5.1(4)
$\tau_{\rm int}(1\times2)$	0.8(1)	2.6(3)	2.8(2)	3.2(2)	4.3(4)	3.9(3)	5.2(4)	5.6(5)	5.7(4)	7.7(8)
$\tau_{\rm int}(1\times3)$	0.8(1)	2.7(3)	2.8(2)	3.2(2)	4.3(4)	3.9(3)	5.1(4)	5.4(5)	5.4(4)	7.4(8)
$\tau_{\rm int}(2\times2)$	1.0(1)	3.4(5)	3.3(3)	3.7(3)	4.7(4)	4.7(3)	5.3(4)	5.8(6)	5.7(4)	7.5(8)
$\tau_{\rm int}(2\times3)$	1.4(3)	4.2(7)	3.8(3)	4.2(3)	5.5(5)	5.7(4)	5.7(5)	6.2(6)	6.3(5)	7.7(8)
$\tau_{\rm int}(3\times3)$	1.8(4)	5.0(8)	4.5(5)	5.0(5)	5.8(6)	6.4(5)	5.9(5)	6.3(6)	6.3(5)	7.5(8)
$r_{\rm gain}^{\rm naive}$	1	7.2	14.8	18.3	17.9	20.6	18.3	20.8	21.1	21.4

matrix (link) multiplications. Second, we extract from numerical simulations the integrated autocorrelation times for different observables, in units of U update. The computer time needed to estimate any given observable is proportional to the product of the computational cost per update and the autocorrelation time.

For NMC, the average computational cost of one update of the U fields on the entire lattice is equal to

$$t_U^{\rm NMC}=t_U^{\rm pl}+4V\sum_{i=2}^{5} n_{\rm staple}(i)*n_{\rm mult}(i)*\langle\sigma_i\rangle, \qquad (20)$$

where $t_U^{\rm pl}$ is the cost for generating the proposal configuration with measure (16) (i.e., the update cost for the elementary plaquette action), $4V$ is the number of links on the lattice, $n_{\rm staple}(i)$ is the number of ''staples'' which the loops of sort i form for each link, $n_{\rm mult}(i)$ is the number of matrix multiplications needed to estimate the contribution of one staple of sort i, and the factor $\langle\sigma_i\rangle$ accounts for the fact that one needs to calculate the contribution of active loops only. One can easily check that

$$n_{\rm staple}(i)=\frac{3}{2}*P_i*s_i\,; \quad n_{\rm mult}(i)=P_i\,, \qquad (21)$$

where $P_i\equiv2(m_i+n_i)$ is the perimeter of loop i and s_i is a symmetry factor: $s_i=1$ for square loops and $s_i=2$ for rectangular loops. Then we have

$$t_U^{\rm NMC}=t_U^{\rm pl}+6V\sum_{i=2}^{5} P_i^2 s_i\langle\sigma_i\rangle. \qquad (22)$$

On the other hand, the computational cost of one update of the σ_i fields on the entire lattice is given by

$$t_{\sigma_i}=6Vs_iP_i\,. \qquad (23)$$

Here $6Vs_i$ is the number of loops of a given sort on the lattice, and the perimeter P_i of the loop appears again as the number of matrix multiplications $n_{\rm mult}(i)$ needed to calculate the probabilities (19). Since we update the σ_i fields only once per each N_i updates of the U fields, the total computational cost per U update for the NMC method is

$$t_{\rm tot}^{\rm NMC}=t_U^{\rm NMC}+\sum_{i=2}^{5}\frac{t_{\sigma_i}}{N_i}=t_U^{\rm pl}+6V\sum_{i=2}^{5} P_is_i\left(\frac{1}{N_i}+P_i\langle\sigma_i\rangle\right). \qquad (24)$$

Let us note that the computational cost $t_{\rm tot}^{\rm UMC}$ for UMC of one U update is approximately equal to the right-hand side of expression (24) in the limit $N_i\to\infty$ and $\langle\sigma_i\rangle\to1$:

$$t_{\rm tot}^{\rm UMC}=t_U^{\rm pl}+6V\sum_{i=2}^{5} P_i^2 s_i\,. \qquad (25)$$

Indeed, in the limit when all σ are set equal to 1 and not updated, we recover the usual algorithm [certainly one should correct the expressions (17) and (18) for $P_{\rm acc}$ in this case].

Now we can compare the performance of our NMC algorithm with that of UMC. The naive gain in efficiency from using NMC does not depend on the observable measured, and is equal to the ratio between the computational costs (25) and (24):

$$r_{\rm gain}^{\rm naive}\equiv\frac{t_{\rm tot}^{\rm UMC}}{t_{\rm tot}^{\rm NMC}}=\frac{t_U^{\rm pl}+6V\Sigma_{i=2}^{5}P_i^2s_i}{t_U^{\rm pl}+6V\Sigma_{i=2}^{5}P_is_i(1/N_i+P_i\langle\sigma_i\rangle)}. \qquad (26)$$

Now, one should also take into account the increase of autocorrelation times coming from the introduction of auxiliary variables σ in the NMC algorithm, so that the real gain is

$$r_{\rm gain}^{\rm real}\equiv r_{\rm gain}^{\rm naive}*\frac{\tau_{\rm int}^{\rm UMC}}{\tau_{\rm int}^{\rm NMC}}, \qquad (27)$$

where $\tau_{\rm int}^{\rm UMC}$ and $\tau_{\rm int}^{\rm NMC}$ are integrated autocorrelation times for UMC and NMC, respectively. Note that $\tau_{\rm int}^{\rm NMC}$ is a function of the updating frequencies $1/N_i$ of the σ_i fields. Like $\tau_{\rm int}$, the ratio (27) will also depend on the observable.

In Table II, we present the autocorrelation times for averaged traces of six different loops in units of U updates. In the first column we show the results for the Usual Monte Carlo, and in other columns for the NMC algorithm with different frequencies of σ updates for 1×2, 1×3, 2×2, 3×3 loops. In the last row we present the naive gain (26).

Let us make one useful remark. It is not necessary to keep the same updating frequencies $1/N_i$ for all sorts i of loops. Actually it is even impractical. The computational cost of U update coming from the loop of sort i is proportional to the average value of σ_i, which is, in turn, proportional to the coupling (9). As the coupling decreases, we should expect a reduction of the computational effort for the corresponding terms in the action. That is not the case for the cost of σ_i update: it does not depend on the coupling and even increases with the nonlocality of the action term [factor P_i in expression (23)]. In order for the work in the σ and in the U updates coming from loops of sort i to remain comparable, one should keep the updating frequencies $1/N_i$ proportional to $\langle\sigma_i\rangle$:

$$\frac{1}{N_i}\sim P_i\langle\sigma_i\rangle. \tag{28}$$

Due to the small influence of weakly coupled terms on the dynamics of the system, one can expect only insignificant changes in the autocorrelation behavior as N_i increases. These considerations are distinctly demonstrated in Table II, where in two columns we present the results for updating frequencies of σ fields varying in accordance with (28).

Table II gives an impressive demonstration of the benefits which come from using the NMC algorithm. The naive gain increases substantially as we decrease the frequencies of σ updates, while the autocorrelation times grow rather slowly. That is particularly visible for the runs where the updating frequencies for σ fields are adjusted as per Eq. (28). For such runs we can infer that the ''real gain'' $\mathcal{O}$ (4–6) in computer time (27) for the observables measured is large enough for a convincing demonstration of the possible advantages coming from using the NMC algorithm.

Let us make a conclusion for this section. We have applied our NMC algorithm for the five-loop model (11), (14). We have shown that with this algorithm a significant gain in efficiency is obtained in comparison with usual updating techniques. Finally, we note that the action (11) is a relatively simple one, and one can expect a much greater gain for more complicated highly-improved actions with many nonlocal weakly coupled terms.

IV. DISCUSSION

Let us summarize our algorithm: (a) Separate the Hamiltonian (or action) into a dominant term c_0W_0, to be calculated exactly, and correction terms $\Sigma_{k=1}^{m}c_kW_k$, to be estimated stochastically. (b) Shift the correction terms to guarantee $c_kW_k\leqslant 0$. (c) Introduce auxiliary local variables $\sigma_k(l)$, through identity (6) (here l runs through all the elementary ''bonds'' which form W_k, e.g., loops in gauge theory). (d) Update the auxiliary variables σ_k by heat bath. (e) To update the original variables U, propose a new value U', sampled from the distribution $\propto e^{-c_0W_0}$, and accept it with the Metropolis probability

$$\min(1,\Pi_{k\geqslant 1;\sigma_k=1}(e^{-c_kW_k(U')}-1)/(e^{-c_kW_k(U)}-1)).$$

The essential advantage of our algorithm appears in step (e): only the terms W_k whose associated σ_k is equal to 1 need to be computed. Since on average $\langle\sigma_k\rangle$ goes to zero with c_k, the computation of almost all correction terms can be avoided.

To avoid simply shifting the cost of the algorithm to step (d), we propose to refresh the variables σ_k infrequently, more so as the associated coupling $|c_k|$ gets smaller. We have pointed out that this introduction of slow dynamics for the σ_k does *not* enforce slow dynamics for the system, since $W_k(l)$ will fluctuate regardless of the value of $\sigma_k(l)$. Our numerical study of Sec. III confirms this statement.

Let us now speculate on possibilities to use our algorithm to simulate a Hamiltonian with a very large number of terms. A specific example we have in mind is the case of full QCD, where the measure is, for two flavors of Wilson quarks,

$$\frac{1}{Z}e^{-S_g(U)}\det{}^2(1-\kappa M(U)), \tag{29}$$

where S_g is the local gauge action, $M(U)$ is a hopping matrix connecting nearest neighbors on a $4d$ hypercubic grid, and Z normalizes the distribution. The determinant can be turned into $\exp(\mathrm{Tr}(\mathrm{Log}(1-\kappa M(U))))$, then the logarithm expanded around 1, giving the loop expansion of the measure above

$$\frac{1}{Z}e^{-S_g(U)-2\Sigma_{l=4}^{\infty}(k^l/l)\mathrm{Tr}\,M(U)^l}. \tag{30}$$

$\mathrm{Tr}\,M(U)^l$ can be represented as a sum over all closed nonbacktracking loops of length l on the $4d$ hypercubic lattice. The number of types of contributing loops n_l is bounded by $(2d-1)^l=7^l$, because of the branching factor at each hop. Although this upper bound is not saturated, it is clear that the multiplicity of terms of a given length l grows exponentially:

$$n_l\sim F_1(l)\alpha^l;\quad \alpha<7, \tag{31}$$

where $F_1(l)$ is a rational function of l and α^l is the leading exponential ascend of the number of loops of length l in the limit of large l. For this reason, it seems that sampling numerically the distribution (30) is a disastrous idea. The action contains an infinite number of terms, of exponentially growing multiplicity. Instead, other strategies are being used, based on the transformation of the determinant (29) into a Gaussian integral.

Nevertheless, the coupling κ^l/l decreases exponentially as l increases. Therefore, the auxiliary variables σ_l associated in our approach with various loops of length l will take value 0 almost always. From (9) one gets

$$\langle\sigma_l\rangle\sim c_l\sim F_2(l)k^l\gamma^l, \tag{32}$$

where the exponentially growing factor γ^l comes from the average trace of Dirac matrices along the loops of length l; $F_2(l)$ is again a rational function.

If one arranges the updating frequencies for each loop i as per Eq. (28), one can expect that the average computer time needed for estimation of contribution of all loops of length l behaves at large l as

$$t_l \sim n_l l^2 \langle \sigma_l \rangle \sim F(l) * (\alpha\gamma\kappa)^l. \tag{33}$$

(It was pointed out above that one should not expect a significant growth of autocorrelation coming from slow dynamics for σ_l.) For $\kappa < \kappa_{ca} = 1/\alpha\gamma$, the computational cost t_l decreases exponentially with l and the total computational cost of the algorithm $t = \Sigma_l t_l$ converges to some finite value. Our rough estimation from fitting n_l and average trace of Dirac matrices in the interval $4 \leqslant l \leqslant 12$ gives $\alpha \approx 5.4$, $\gamma \approx 1.4$, and therefore $\kappa_{ca} \approx 0.13$.

In the regime $\kappa < \kappa_{ca}$, we are in an interesting situation where the influence of very large loops is negligible because their associated coupling in the effective action is extremely small. Therefore, truncating the loop expansion above a certain order will introduce a statistically unobservable bias. [Note that a similar truncation strategy in the number of auxiliary variables could also be adopted in the approach of Ref. [4] to the estimation of the fermionic determinant.] Equivalently, one can freeze the associated σ variables at the value zero or update them with arbitrarily low frequency. In spite of this extremely (or infinitely) slow dynamic mode of the σ's, the dynamics of the gauge fields are not affected. Note that the cost of our algorithm grows linearly with the volume V of the system. This is better than alternative approaches to the simulation of full QCD: Hybrid Monte Carlo (cost $\propto V^{5/4}$) and MultiBoson (cost$\propto V(\mathrm{Log}\, V)^2$) [11]. In addition, the stepsize, or typical change at each update of a gauge link U, does not seem restricted *a priori* for small quark mass, unlike in the two alternative approaches above. Unfortunately, the possible high efficiency of our algorithm is counterweighted by its extreme programming complexity.

A less speculative use of our algorithm for full QCD consists of truncating the loop expansion Eq. (30) to some order $l_{\max}$, and to represent the higher orders with the MultiBoson approach [12]. This strategy, called ''UV-filtered MultiBoson,'' has already been used successfully [8]. However, in Ref. [8] the loop expansion is truncated to its lowest term $l = 4$, because the exact evaluation of larger loops is too time consuming. With our stochastic approach, these larger loops can be estimated at low cost. We expect this composite strategy to be particularly efficient.

ACKNOWLEDGMENTS

We acknowledge communications with T. DeGrand, M. Ilgenfritz, and U. Wenger. T.B. was supported by INTAS under Grant No. 96-0370 and Russian Basic Research Fund under Grant No. 99-01-00190.

[1] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, J. Chem. Phys. **21**, 1087 (1953).
[2] A. D. Kennedy and J. Kuti, Phys. Rev. Lett. **54**, 2473 (1985).
[3] G. Bhanot and A. D. Kennedy, Phys. Lett. **157B**, 70 (1985).
[4] L. Lin, K. F. Liu, and J. Sloan, Phys. Rev. D **61**, 074505 (2000).
[5] K. Symanzik, Nucl. Phys. **B226**, 187 (1983).
[6] M. Alford, W. Dimm, G. P. Lepage, G. Hockney, and P. B. Mackenzie, Phys. Lett. B **361**, 87 (1995).
[7] P. Hasenfratz and F. Niedermayer, Nucl. Phys. **B414**, 785 (1994).
[8] Ph. de Forcrand, Nucl. Phys. B (Proc. Suppl.) **73**, 822 (1999); C. Alexandrou, Ph. de Forcrand, M. D'Elia, and H. Panagopoulos, Phys. Rev. D **61**, 074503 (2000).
[9] M. García Pérez, A. Gonzalez-Arroyo, J. Snippe, and P. van Baal, Nucl. Phys. **B413**, 535 (1994).
[10] Ph. de Forcrand, M. García Pérez, and I.-O. Stamatescu, Nucl. Phys. **B499**, 409 (1997).
[11] Ph. de Forcrand, in *Multi-scale Phenomena and Their Simulation*, Proceedings of the International Conference, Bielefeld, Germany, 1996, edited by F. Karsch *et al.* (World Scientific, Singapore, 1997), pp. 169–178, hep-lat/9702009.
[12] M. Lüscher, Nucl. Phys. **B418**, 637 (1994); A. Slavnov, hep-th/9611154.