Monte Carlo simulation has become the primary tool for the study of nonperturbative phenomena in both quantum field theory and statistical systems. Indeed, this approach has resulted in theoretical physicists becoming rather avid users of computer time. Despite severe difficulties with including anticommuting fermionic fields in such simulations, the results have been impressive. The technique has given rise to useful quantitative information in statistical mechanics about critical phenomena and in particle physics about the confining interquark potential, glueball spectra, and the transition to a quark-gluon plasma.

These simulations have generally used a version of the old algorithm of Metropolis et al.1 A few years ago Adler2 suggested extending overrelaxation techniques for solving linear equations to Monte Carlo simulations. His approach, which was further investigated by Whitmer,3 was limited to theories where the action or energy function appearing in the Boltzmann weight is quadratic in any single individual variable. This restriction to multiquadratic actions has prevented application to the more complicated forms appearing in lattice gauge theory.

Motivated by overrelaxation ideas, I present in this paper a simple variation of the algorithm of Metropolis et al. This approach is applicable to any theory where the variables are elements of a group larger than $\mathbb{Z}_2$. The basic idea is to pick a trial change for a given variable in a region of phase space which is as far as possible from the old value without paying a severe energy penalty. This is done by approximately locating the locus of minimum energy for the variable under consideration, and then selecting the trial element on the “opposite” side of this value.

I begin with a review of the standard algorithm of Metropolis et al. Consider updating some variable $g$ whose value lies in a group $G$. While holding other variables fixed, write the Boltzmann weight for this variable as

$$P_{\text{eq}}(g) \propto \exp[-\beta H(g)].$$

In this paper I will always consider variables as elements of a group. This is quite natural in the standard formulation of lattice gauge theory. This use of group elements is for notational convenience and is not a practical restriction; real variables can always be thought of as belonging to the group of numbers under addition.

The procedure of Metropolis et al. considers a trial new value $g'$ to replace $g$. This is selected with some probability distribution $P_{T,S}(g')$. Here the subscript $g$ on $P$ is to remind us that the trial change can in general depend on the old element $g$. The trial element $g'$ is then accepted with a conditional probability

$$P_A = \min \left[ 1, \frac{P_{T,S}(g') \exp[-\beta H(g')]}{P_{T,S}(g) \exp[-\beta H(g)]} \right].$$

If this conditional probability is not met, then the change is rejected and the old value of $g$ is kept. This construction automatically satisfies the detailed balance condition

$$P(g \rightarrow g') e^{-\beta H(g)} = P(g' \rightarrow g) e^{-\beta H(g')},$$

where $P(g \rightarrow g')$ is the overall probability of taking element $g$ to $g'$. When $g' \neq g$, $P(g \rightarrow g') = P_{T,S}(g') P_A$. Equation (3) is sufficient to ensure that the equilibrium distribution of Eq. (1) is invariant under the algorithm.

That invariance plus ergodicity are necessary and sufficient conditions for a Monte Carlo algorithm to bring any ensemble towards equilibrium.

As usually implemented, $g'$ is found by multiplying $g$ by a group element $h$ which is chosen with a probability distribution peaked around the identity and with equal probability for $h$ and $h^{-1}$. For example, this can be done by choosing $h$ from a table which contains the inverse of each of its elements. This particular method of choosing the trial change has the symmetry property

$$P_{T,S}(g') = P_{T,S}(g).$$

This is convenient because the ratio of $P_T$ factors in Eq. (2) is then unity and can be ignored. The conventional approach contains an implicit parameter which represents the average distance the element $h$ lies from the identity. If this distance is too large, the trial energies will likely be large and the changes will rarely be accepted, while if $h$ always lies too close to unity, the changes will usually be accepted, but their small size will make the exploration of phase space rather slow. Lore is for a compromise with
an acceptance rate of order 50%.

The "heat bath" algorithm is a special case of this general-
ized approach of Metropolis et al., wherein the trial
element is taken randomly from the entire group manifold
but with a weighting proportional to the Boltzmann factor
of Eq. (1). In this case the factors appearing in the accep-
tance criterion of Eq. (3) cancel and the trial element is al-
ways accepted. This method is equivalent to the limit of
taking a large number of repetitions, or "hits," of any er-
godic algorithm to a single variable.

The algorithm proposed in this paper represents yet an-
other way to choose the trial change \(g'\). Suppose I have
some simple way to find a group element \(g_0\) which ap-
proximately minimizes the energy \(H(g)\). Suppose further
that \(g_0\) is obtained with no direct use of the element \(g\);
that is, \(g_0\) only has a dependence on the remaining lattice
variables. The essence of this paper is to propose selecting
the trial element for a Metropolis et al. updating to lie on
the "opposite side" of this element \(g_0\) from the old value
\(g\). In particular I consider the trial element

\[
g' = g_0 g^{-1} g_0.
\]

Note that this construction also satisfies the symmetry rela-
tion of Eq. (4). Thus, just as in the usual application of
the algorithm off Metropolis et al., the acceptance or re-
jection of this element follows solely from the comparison
of a random number with the exponential of the resulting
energy change.

To be more specific about the selection of \(g_0\), consider
the case where \(H(g)\) takes the form

\[
H(g) = E_0 - \text{Re Tr}(gM),
\]

where \(E_0\) includes contributions to the energy which do
not depend on \(g\). In the case of lattice gauge theory with the
Wilson action, \(M\) is a matrix which represents the
sum of the three sided "staples" surrounding the link to
be updated. For a nearest-neighbor spin model with the
spins being matrices from the group \(G\), \(M\) is the sum of
the group elements on sites neighboring \(g\). With this form
for the energy, a natural choice for \(g_0\) is to use the inverse
of a group element obtained by projecting \(M\) onto \(G\). As
the correctness of the algorithm does not depend on the
prescription for this projection onto the group, it is advan-
tageous to keep the procedure as computationally simple
as possible. In my tests with SU(N) lattice gauge theory, I
use a Gram-Schmidt orthogonalization process on the ma-
trix rows and then divide the last row by the determinant
of the resulting matrix.

Note that the algorithm cannot be used for variables in
the group \(Z_2\). Indeed, in this case Eq. (5) always gives
\(g' = g\) and no changes are proposed. This will also be
the case for any group when the system is totally ordered,
with \(g\) and \(g_0\) being the identity. Thus to start the algo-
rithm, some disorder must be initially present. This does
not mean that the algorithm will not be useful at low tem-
peratures; rather, the size of proposed changes automati-
cally decreases for cool systems.

In some cases, most notably with the groups SU(2) and
U(1), the matrix \(M\) is always proportional to a group ele-
ment. Projecting onto this particular element results in \(g'\)
having exactly the same energy as \(g\). In this case the

the prescription of Metropolis et al. will always accept the
change, and the algorithm is deterministic and micro-
canonical. This causes two minor complications. First,
the total energy of the system is fixed and thus will not rel-
x to any value other than where it is initially set. Second,
the algorithm is actually independent of the tempera-
ture \(\beta^{-1}\). Indeed, as with other microcanonical algo-
rithms, the temperature should be measured during the
simulation with some sort of thermometer, such as an
average kinetic energy, using an auxiliary variable with
simple dynamics, or from a dynamical equation involving
both the temperature and measurable correlation func-
tions.

This issue can be avoided if desired by putting a small
amount of randomness into \(g_0\). For example, \(g_0\) could be
the product of a deterministic estimate of the element
minimizing \(H(g)\) with a random element \(h\) chosen near
the identity. If \(h\) has a small probability of lying any-
where in the group, this would also eliminate possible
worries about ergodicity. Nevertheless, I have done lim-
ited studies which suggest that correlation times tend to in-
crease with additional noise in \(g_0\).

I now turn to some tests of the algorithm. Consider
standard four-dimensional SU(N) lattice gauge theory
with the Wilson action

\[
H = \sum_r \frac{1}{N} \text{Re Tr}_{g_r},
\]

where the sum runs over all elementary squares or pla-
quettas \(p\) on a simple hypercubic lattice. The fundamen-
tal variables are group elements associated with the links
of the lattice and the quantity \(g_r\) represents the product
of such elements around the sides of the respective plaquette
\(p\).

To simplify vectorization, all simulations presented here
used skew-periodic boundary conditions on a \(7 \times 7 \times 7 \times 6\)
lattice. The links in any given direction were updated in a
checkerboard style, with all those emanating in a positive
direction from odd sites being updated before those from
even sites. The lattices were initially equilibrated with
100 iterations of a heat-bath algorithm for SU(2) and an
optimized 10-hit algorithm of Metropolis et al. for SU(3).
Where error bars are shown, they were obtained by re-
peating the respective experiments 5 or 20 times.

A simple measure of the correlation between two lat-
tices \(U\) and \(V\) with corresponding links \((g_i)_{U}\) and \((g_i)_{V}\) is

\[
C(U,V) = \frac{1}{n_i N} \sum_l \text{Re Tr}[(g_l^{-1})_{U}(g_l)_{V}],
\]

where the sum is over all links \(l\) and \(n_i\) is the total num-er of links. This quantity is unity where \(U\) and \(V\) are the
same and vanishes for uncorrelated lattices. I will be con-
sidering lattice \(V\) obtained from \(U\) through a few applica-
tions of various Monte Carlo algorithms. The speed with
which this correlation drops to zero is then indicative of
the efficiency of the algorithm.

Figure 1 shows the falloff of this interlattice correlation
as a function of the number of Monte Carlo iterations
separating the lattices \(U\) and \(V\). Here the gauge group is
SU(2) and \(\beta = 2.3\). The correlation is shown both for
the standard heat-bath algorithm and the overrelaxation al-
gorithm. As mentioned above, for SU(2) the latter approach
FIG. 1. The correlation between two lattices as a function of the number of simulation iterations separating them. The model is SU(2) lattice gauge theory at $\beta=2.3$. The solid points are for the heat-bath algorithm while the open triangles are for the overrelaxation algorithm presented in the text.

is both microcanonical and deterministic. In this figure the heat-bath algorithm appears to give a simple monotonic decrease of the correlation, while the overrelaxation approach decorrelates somewhat faster in an oscillatory fashion.

Figure 2 shows the same correlation for the case of SU(3) at $\beta=6.0$. I determine $g_0^{-1}$ from a Gram-Schmidt orthogonalization process on the matrix $M$ interacting with the element being updated. Unlike for SU(2), the overrelaxation algorithm is now neither deterministic nor microcanonical. I find the acceptance rate for the trial changes is 57%.

For comparison Fig. 2 also shows the correlation falloff for a Metropolis et al. updating with the 10, 64, and 128 "hits" or trial changes for each element before moving to update the neighbors. Of course, as the number of such hits goes to infinity one should approach a heat-bath algorithm. These standard runs used trial elements $g'$ selected by multiplying $g$ with a matrix $h$ chosen with probability $P(h) \propto \exp(k \Re \text{Tr} h)$. I chose $k = 2\beta$ because, at least for this value of $\beta$, this empirically optimizes the correlation decrease (independent of number of hits). For this value of $k$ and $\beta$, the acceptance per hit was 30%.

For all runs in Fig. 2 the correlation appears to be monotonically decreasing, with the overrelaxation algorithm decreasing the fastest. Indeed, noting the small change in going from 64 to 128 hits, the figure suggests that the new algorithm outperforms the heat bath. This is true even with the naive method for projecting $M$ onto the group. Although the runs in this figure are consistent with approaching exponentials, it might be dangerous to assume that this continues. There could be hidden long-time correlations which emerge upon further running.

One potential difficulty with this measure of correlation is associated with the gauge invariance of the system. Indeed, a random gauge transformation will give a new lattice with zero expected correlation with the old one. This is in spite of the fact that all gauge-invariant quantities are identical on the two lattices. All the algorithms I will study here are based on updating single links at a time and do not take any special advantage of the gauge symmetries of the theory. Nevertheless, because of this worry, consider the correlation between gauge-invariant plaquette operators. In particular, define

$$C_p(U,V) = \frac{1}{n_p} \sum_p (W_p - \langle W \rangle)_U (W_p - \langle W \rangle)_V,$$  (9)

where

$$W_p = \frac{1}{N} \Re \text{Tr} g_p$$  (10)
on

on the respective lattice, $n_p$ is the total number of plaquettes, and the expectation value $\langle W \rangle$ is the plaquette averaged over the entire lattice. Because the plaquette has an expectation value, this is subtracted to obtain a vanishing correlation for independent lattices. One disadvantage of using this gauge-invariant measure of correlation is purely numerical; the corresponding numbers fall quite rapidly and are more difficult to measure than the quantity in Eq. (8).

In Fig. 3 I show the quantity from Eq. (9) for the SU(2) runs which gave rise to Fig. 1. Note that with this measure the behavior of the new algorithm is somewhat slower at decorrelating than the heat bath. There is also a hint of an even-odd oscillation in the convergence. This might be expected because for the new algorithm a double hit on the same group element would result in no change to the lattice. Figure 4 shows the corresponding quanti-
FIG. 3. The correlation between plaquettes vs iterations for the SU(2) lattices used in Fig. 1. The solid points represent heat-bath updating and the open triangles are for the overrelaxation algorithm.

ties for the SU(3) runs. Here the overrelaxation approach with one hit appears to be slightly worse at decorrelating plaquettes than the 10-hit standard algorithm of Metropolis et al.

That the new algorithm performs less well at decorrelating plaquettes than links might be expected from the fact that the SU(2) algorithm is microcanonical and the SU(3) case approximately so. Thus there is a built-in global correlation between plaquettes which appears to show up in the local plaquette-plaquette correlation.

In a sense the overrelaxation following from Eq. (5) is the maximum that one would intuitively expect to be useful. It might be interesting to consider something less extreme and choose the trial element somewhere between $g$ and $g'$. Indeed, the stochastic Langevin equation \(^5\) can be thought of as an underrelaxed algorithm of this type wherein the trial element is chosen by multiplying the old element by a driving force towards $g_0$ and then introducing a noise with width selected to make the acceptance in Eq. (2) unity to lowest order in the step size.

For an intermediate case, I have also considered picking the trial element by multiplying $g_0$ by a random element $h$ near the identity

$$g' = g_0 h .$$

The elements $h$ were selected with distribution

$$P(h) = \exp(k \text{Re} Tr h) .$$

This distribution was obtained via a separate Monte Carlo simulation. In Fig. 5 I show the acceptance rate for the

FIG. 4. The plaquette correlation for the SU(3) runs in Fig. 2. The open triangles are for the overrelaxation algorithm, while the solid points and crosses are for standard Metropolis et al. updating with 10 and 64 hits per link, respectively.

FIG. 5. (a) The acceptance rate for trial links chosen using Eq. (11) as a function of the parameter $k$ appearing in Eq. (12). The model is SU(3) lattice gauge theory at $\beta=6.0$. (b) The correlation between lattices separated by 10 iterations as a function of $k$, where the trial changes are selected as in (a).
trial changes as well as the correlation between two lattices separated by 10 sweeps as a function of the parameter \( k \). The gauge group is SU(3) and \( \beta = 6.0 \). Here only one trial change was considered for each variable before moving on to the next. Note that for a narrow region of the parameter \( k \), the decorrelation rate is quite reasonable; indeed, it is better than the standard 10-hit approach as shown in Fig. 2. This approach does not satisfy the symmetry relation of Eq. (3), so the extra ratio of trial probabilities enters nontrivially into the acceptance criterion. This is somewhat of a disadvantage because systematic errors could be introduced if the matrices \( h \) are not chosen independently with the correct distribution. At large \( k \) the acceptance becomes low because the ratio of probabilities \( P \) becomes small, while at small \( k \) acceptance suffers because most changes result in large energies.

To summarize I have presented a variation of the scheme of Metropolis et al. for Monte Carlo simulation. The approach is motivated by overrelaxation ideas; that is, I consider trial changes which lie beyond the minimum of the energy from the old value of a variable being updated. There are two intuitive arguments in favor of this idea. The simplest is that the trial element is placed rather far from the old value without a large energy penalty. Thus one might expect a rather rapid flow through phase space. A second argument is based on the overrelaxation idea as used in minimization schemes such as used in solving linear equations. The position of minimum energy for a given variable is indirectly influenced by the variable itself. When the neighbors were updated, they assumed values which tended to accommodate the position of the current variable. If, however, that variable were allowed to float, they would in general move away and one might expect that the best value for the variable being updated might lie somewhat further away than the position of lowest energy with the neighbors held fixed.

Perhaps the greatest advantage of this algorithm is its computational speed due to simplicity. For both SU(2) and SU(3) I obtained decorrelations per iteration comparable to a heat bath. Although a heat bath is rather easily implemented for SU(2), this is not the case for SU(3) and thus most lattice gauge simulations have been done with a procedure of Metropolis et al. using of order 10 trial changes on any link before proceeding. The present algorithm performs best with but a single hit; indeed, further attempts to change a given variable will just return to earlier trials. Furthermore, the construction of the trial element takes only minimally more computation than a single hit in a standard application of Metropolis et al. This advantage may be even greater for spin systems where there is substantially less overhead involved in calculating the interacting neighborhood of a variable being updated.

Note added. This algorithm and other variations of overrelaxation are discussed in a recent report of Brown and Woch.9