MAR 12 1986

ANL-HEP-CP-86-09
January, 1986
CONF-851170--1

THE LOOK-AHEAD FERMION ALGORITHM*

ANL-HEP-CP--86-09

DE86 007494

Michael Grady
High Energy Physics Division
Argonne National Laboratory
Argonne, Illinois 60439

ABSTRACT

I describe a fast fermion algorithm which utilizes pseudofermion fields but appears to have little or no systematic error. Test simulations on two-dimensional gauge theories are described. A possible justification for the algorithm being exact is discussed.

---

*Talk presented at the Lattice Gauge Theory--A Challenge in Large Scale Computing Conference, Univ. of Wuppertal, Nov. 5-7, 1985

Work supported by the U.S. Department of Energy, Division of High Energy Physics, Contract W-31-109-ENG-38.

INTRODUCTION

I would like to describe a new fermion algorithm with which I have had a great deal of success in two-dimensional applications.[1] It is based on locally updating a small number of pseudofermion fields each time a gauge field is updated. I call it a look-ahead algorithm because the pseudofermion update is tentative. It is made after a trial new gauge field link is selected, but before it is accepted or rejected. I will attempt to show how such a procedure eliminates an important source of systematic error which occurs in the standard pseudofermion technique of Fucito et al.[2,3]

The algorithm has many desirable features. In one and two-dimensional tests, which include the Schwinger Model, I have found no detectable systematic error in the results. This is true even after very high statistics have been achieved through long simulations. A corollary to the lack of systematic error is a lack of dependence on the gauge field hit size. This is the maximum amount that the new trial gauge field link is allowed to differ from its old value in an update. In the standard pseudofermion algorithm, the systematic error increases with increasing hit size, forcing one to adopt rather small hit sizes. This has a detrimental impact on the performance of the algorithm, since it leads to very high correlations between adjacent sweeps. Consequently, one has to perform many sweeps before one has the equivalent of a single uncorrelated measurement; thus the amount of information extracted per sweep is relatively small when compared with the pure gauge theory algorithm, for which the hit size may be optimized for maximum algorithm performance. A limitation to small hit sizes also means that many more sweeps need to be performed to equilibrate lattices before any measurements can take place. It is also more probable that the system will get stuck in a metastable state if the hit size is small. Thus the advantages of an algorithm with complete freedom to set the hit size are great.

On the face, the look-ahead algorithm appears to be an approximation which is exact in several limits which I shall discuss. However, the lack of detectable systematic errors in the numerical examples studied to date lends one to speculate whether the algorithm could possibly be exact, i.e. completely free of systematic error. This could be due to a balancing of errors between the probability of a transition $A \to A'$ and that of the transposed process $A' \to A$. I will try to indicate how this could work in what follows. Of course, although it would be very nice to have a fast exact algorithm, one does not need such an algorithm to make progress. If the systematic errors are of sufficiently high order that they affect physical measurements by, say, less than 10% of the total fermionic effect on a system, for reasonable values of parameters, then such an algorithm would be practical for most lattice gauge theory applications. The look-ahead algorithm is fast. It runs only a factor of 3-12 times slower than the corresponding pure gauge theory algorithm in the two-dimensional models studied, depending on the type of interaction. This is with largely unoptimized programs. Realistic four-dimensional problems should not be more than another factor of two slower. Finally, the look-ahead algorithm is simple to implement compared to many of the improved algorithms being suggested, a feature it shares with the standard pseudofermion algorithm.

THE ALGORITHM

The generic action for a system of gauge fields interacting with Fermi fields is

$$S_f(\bar{\psi},\psi,A) = S_0(A) + \sum_{ij} \bar{\psi}_i O_{ij}(A)\psi_j \ , \tag{1}$$

where $S_0(A)$ is the pure gauge action which contains a multiplicative parameter β. $O_{ij}(A)$ is a local operator which depends on A, such as the Dirac operator. In what follows, I shall assume that $O_{ij}(A)$ is positive definite. This is usually not an important restriction, since one can generally use a second order fermion formalism which satisfies the requirement. The partition function is given by

$$Z_f = \int dA\ d\bar{\psi}\ d\psi\ \exp\left(-S_f(\bar{\psi},\psi,A)\right) \ . \tag{2}$$

The Grassmann Fermi fields are then integrated to give

$$Z_f = \int dA\ \exp\left(-S_0(A)\right)\det O(A) \ . \tag{3}$$

Eq. (3) is the starting point for numerical simulations, since it is impractical to represent the Grassmann fields themselves on a computer, for all but very small lattices. One could also imagine an analogous system of a complex Bose field interacting with the gauge field through the same interaction matrix $O_{ij}(A)$. This system is formed through the substitution:

$$\bar{\psi}_i \to \phi_i^* \ , \qquad \psi_i \to \phi_i \ . \tag{4}$$

So one has what I will refer to as the boson/gauge system, with action given by

$$S_b(\phi^*,\phi,A) = S_0(A) + \sum_{ij} \phi_i^* O_{ij}(A)\phi_j \ , \tag{5}$$

and partition function

$$Z_b = \int dA\ d\phi^* d\phi\ \exp\left(-\ S_b(\phi^*,\phi,A)\right)\ . \tag{6}$$

Integration of the ϕ fields yields

$$Z_b = \int dA\ \exp\left(-S_0(A)\right)\left(\det\ O(A)\right)^{-1}\ . \tag{7}$$

Thus one sees that the difference between Bose and Fermi statistics is simply the inversion of the determinant factor occurring in (3). Note that the fields ψ_i and ϕ_i carry the same spin, thus the system (5) violates the spin-statistics theorem. This will not be of any consequence, however, as the lattice partition function (6) is still well defined. Unlike the fermion case, for the boson system one can simulate the partition function (6), including the ϕ fields, with Monte Carlo techniques. It is much less practical to use the integrated form (7), due to the non-local determinant factor. However, for the fermion case, one is forced to use the integrated form. This is the primary source of difficulty in incorporating fermions into Monte Carlo simulations. The pseudofermion approach is to somehow use the ϕ fields and the system (6) in an unconventional way so as to simulate the fermion determinant occurring in (3), rather than the inverse determinant which would result from a straightforward simulation of the Gaussian integral.

I restrict consideration to the Metropolis et al. Monte Carlo technique.[4] One produces a chain of configurations for the Fermion/Gauge system in the following way. Choose a gauge field A' which differs from the current gauge configuration, A, on just one link, j. Then accept A' with the probability

$$P(A \to A') = \mathrm{Min}\left(1, \exp(-S_0(A') + S_0(A))\Delta\right) \tag{8}$$

$$\Delta \equiv \det O(A')/\det O(A) \ . \tag{9}$$

If rejected the old configuration A is retained. Thus one needs only a fast way of computing the determinant ratio, Δ, of two operators which differ only locally. For the boson/gauge system one needs Δ^{-1} in place of Δ. It can be written

$$\Delta^{-1} = \frac{\int d\phi^* d\phi \, \exp\left(-\phi^* O(A')\phi\right)}{\int d\phi^* d\phi \, \exp\left(-\phi^* O(A)\phi\right)} = \langle \exp\left(-\phi^* \left(O(A')-O(A)\right)\phi\right)\rangle_{\phi(A)} \tag{10}$$

where the notation has been streamlined by replacing the explicit summations in the action with implied matrix-vector multiplication. $\langle \ \rangle_{\phi(A)}$ is defined to be an expectation value in the ensemble with probability distribution $P(\phi,A) \propto \exp\left(-\phi^* O(A)\phi\right)$. This is the distribution which is naturally generated through Monte Carlo simulation of the Gaussian ϕ field. Note that in a standard simulation of the boson system (6), one would not use the full expectation value, Δ^{-1}, for each A field update, but rather the operator inside the expectation value evaluated in a single ϕ configuration. The correctness of both formulations can be verified by detailed balance. They basically differ by an interchange of order of integration. In the latter case the expectation value is evaluated a bit at a time, with an additional determination each time the chain reencounters the transition $A \to A'$.

For the fermion/gauge system one needs Δ rather than Δ^{-1}. One could simply write

$$\Delta = \langle \exp\left(-\phi^* \left[O(A') - O(A)\right]\phi\right)\rangle^{-1}_{\phi(A)} . \quad (11)$$

This is basically the strategy of the standard pseudofermion algorithm. However, this leads to a systematic error when the expectation value is incompletely evaluated at each update. This is simply because the average of the inverse is not the inverse of the average. If one has a number of determinations of some quantity x, then

$$\langle \frac{1}{x} \rangle \neq \frac{1}{\langle x \rangle} \quad (12)$$

if the variance is nonzero; in fact, the left-hand side will always be greater. Another way to represent Δ is to write the same ratio of gaussians as before (10) but interchange A with A', giving

$$\Delta = \langle \exp\left(\phi^* \left[O(A')-O(A)\right]\phi\right)\rangle_{\phi(A')} . \quad (13)$$

This differs from the boson expression Δ^{-1} (10) by the sign in the expectation value and in the distribution over which it is taken:

$$P(\phi,A') = \exp\left(-\phi^* O(A')\phi\right) . \quad (14)$$

Suppose, for the moment, that one had an efficient way of producing this distribution. Then, I claim, in analogy with the Boson algorithm, that instead of using the full expectation value Δ in the Metropolis probability (8), one can instead use the argument of (13) evaluated for a single member of the distribution (14), without introducing a systematic error. The average transition probability $P(A \to A')$ will be the ϕ dependent transition probability $P(A \to A', \phi)$ averaged over the ϕ distribution.

$$P(A \to A') = \frac{\int d\phi\, P(\phi, A') P(A \to A', \phi)}{\int d\phi\, P(\phi, A')}$$

$$= \frac{\int d\phi\, \exp\left(-\phi^* O(A')\phi\right) \mathrm{Min}\left(1, \exp\left(-S_0(A') + S_0(A) + \phi^*(O(A') - O(A))\phi\right)\right)}{\int d\phi\, \exp\left(-\phi^* O(A')\phi\right)}$$

$$= \det O(A') \exp\left(-S_0(A')\right) \int d\phi\, \mathrm{Min}\left(\exp(S_0(A') - \phi^* O(A')\phi)\right. ,$$
$$\left. \exp\left(S_0(A) - \phi^* O(A)\phi\right)\right) \tag{15}$$

up to a numerical factor. The expression within the integral is symmetric under interchange of $A \leftrightarrow A'$. Now, the equilibrium distribution is given by the detailed balance equation

$$P_{eq}(A) P(A \to A') = P_{eq}(A') P(A' \to A) \; . \tag{16}$$

This can be seen to be solved by

$$P_{eq}(A) = \exp\left(-S_0(A)\right)\det O(A) \ , \tag{17}$$

which is the desired distribution for the fermion/gauge system. So the systematic error has been eliminated, but at the expense of needing a ϕ distribution equilibrated to the new trial gauge field rather than the old one. How is this distribution to be obtained? One could just equilibrate the entire lattice of ϕ fields to A', each time a gauge field is updated, but this is impractically slow. I suggest, rather, equilibrating only a small neighborhood of ϕ fields about the link, j, being updated. This neighborhood is defined as those ϕ fields which occur with nonzero coefficients in the expression $\phi^*\left(O(A')-O(A)\right)\phi$, i.e. those which contribute directly to the Metropolis probability. More distant neighbors contribute only indirectly via feedback through the probability distribution. I now define the algorithm explicitly. The steps are as follows:

(1) Choose A' different from A at one site j.

(2) Perform a look-ahead update of ϕ fields within the immediate neighborhood of A_j to $\exp\left(-\phi^* O(A')\phi\right)$. Fields must be equilibrated, either with several rounds of updates, or a single multivariate heat bath update.[5]

(3) Accept or reject A according to the Metropolis probability $P(A \rightarrow A',\phi) = \mathrm{Min}\left(1, \exp(-S_0(A') + S_0(A) + \phi^*\left(O(A')-O(A)\right)\phi\right)$, using, for the ϕ fields, those obtained in (2).

(4) It turns out that the ϕ distribution gets distorted by the Metropolis decision in (3), so one must, at this point, update the neighboring ϕ fields again, <u>if</u> A' was accepted. If A' was rejected in (3) then the original ϕ fields are retained.

(5) Go on to the next link.

Note that step (4) is slightly different from the algorithm of Ref. 1. There I also updated the ϕ fields when the A' update was rejected. I eventually found a small systematic error using that algorithm. It also does not satisfy the conditions of the more detailed justification presented later.

What can one say about this algorithm? First, it is exact in zero dimensions (field theory at a point). This is because there is only one ϕ field, and a single heat bath look-ahead update achieves the needed $\exp\left(-\phi^{*}O(A')\phi\right)$ distribution for the ϕ field. This is in contrast to the standard pseudofermion algorithm which already has serious systematic errors in zero dimensions. Second, it may be viewed as the first term in a systematic approximation based on equilibrating larger and larger neighborhoods of the gauge field, A_j, being updated, becoming exact as the neighborhood approaches the full lattice. Third, it is also exact as the gauge field hit size goes to zero, for essentially the same reasons that the standard pseudofermion algorithm is. The hit size is the maximum amount that the new gauge field A_j' is allowed to differ from the old field A_j. This limit is very useful for testing the algorithm. One merely has to run the simulation for several different hit sizes, and look for a hit size dependence in measured quantities. An exact algorithm should be independent of the hit size. If the algorithm has a small systematic error, it can usually be compensated for by extrapolating several runs to zero hit size, or simply by running with a relatively small hit size for which the systematic error is correspondingly small. In the systems studied so far, I have found no evidence for a hit size dependence using the look-ahead algorithm. Even a completely open hit, one in which the new field A_j' is completely unrestricted, seems to produce correct results. For instance, in the spinless Schwinger Model (Table I), internal

Table 1. Spinless Schwinger Model

Algorithm	Internal Energy	$\langle\bar{\psi}\psi\rangle$	$\langle\bar{\psi}UU\psi\rangle$
look-ahead, $\eta=1$	0.5597(10)	---	---
look-ahead, $\eta=2$	0.5595(5)	---	---
look-ahead, $\eta=2\pi$	0.5591(3)	0.4035(6)	.0594(6)
exact	0.5589(7)	0.4036(5)	.0590(4)
quenched	0.5450(14)	0.4066(8)	.0627(6)
no look-ahead, $\eta=1$	0.5713(15)	---	---
no look-ahead, $\eta=2$	0.5847(18)	---	---
no look-ahead, $\eta=2\pi$	0.6600(7)	0.3948(6)	.0536(4)

Internal energy, $\langle\bar{\psi}\psi\rangle$, and $\langle\bar{\psi}UU\psi\rangle$ (a two lattice spacing correlation function) for the look-ahead algorithm with three different hit sizes (η). $\eta = 2\pi$ corresponds to a completely open hit. Values are also given for the exact determinant calculating algorithm, and for the quenched or pure gauge system. The last three entries are for an algorithm which had no look-ahead update; the ϕ fields in a neighborhood were updated only after each gauge field update. This last algorithm is similar to the standard pseudofermion algorithm with only one ϕ configuration used for averages. $\langle\bar{\psi}\psi\rangle$ and $\langle\bar{\psi}UU\psi\rangle$ were measured from a sample of gauge configurations using a separate pseudofermion Monte Carlo. Gauge configurations were not stored for every run, so this measurement was not performed in every case. Simulations are for $\beta = 1.02$, $m = 0.02$ on a 4×4 lattice. Errors are from binned correlations, with a bin size of 1000 sweeps.

energies for runs with an open hit ($-\pi < A_j' < \pi$), and runs with a fairly restricted hit ($A_j - 0.5 < A_j' < A_j + 0.5$), as well as runs with an exact algorithm, agree within statistical errors of about 5% of the fermionic effect on the system. By this I mean the difference between the measured quantity with the fermions present, and the same quantity in the quenched approximation, where the fermion determinant is set to unity. The spinless Schwinger model is an unphysical two-dimensional model of a U(1) gauge field, A, interacting with a ψ field which has Fermi statistics but the interactions of a scalar field. Because of its relative simplicity, this model is a good testing ground for lattice fermion algorithms, since the main problem of such algorithms is getting the Fermi statistics right. The precise nature of the interaction matrix O_{ij} is of secondary importance. Of course, one should then go on to test more realistic models, if trials with the simplest models are successful.

The excellent accuracy of the algorithm, for quantities which depend on the gauge fields, is all the more remarkable if one observes the behavior of the running ϕ fields of the simulation. I do not propose using these pseudofermion fields for anything but performing the gauge field simulations; however, it is interesting to monitor $\langle\phi^*\phi\rangle$ from these running fields, and compare it to the same quantity one gets if one takes a sample of gauge fields from the simulation and equilibrates the pseudofermion fields to each gauge configuration using a separate Monte Carlo. This is a standard matrix inversion technique used to measure $\langle\bar{\psi}\psi\rangle$. (For the spinless Schwinger model $\langle\bar{\psi}\psi\rangle = \langle\phi^*\phi\rangle$; for the Schwinger Model with Kogut-Susskind fermions discussed later $\langle\bar{\psi}\psi\rangle = 2m\langle\phi^*\phi\rangle$.[6]) There is a very noticeable lag in the value of $\langle\phi^*\phi\rangle_r$, from the running fields of the simulation which increases in an apparently linear fashion with the hit size. For example a run with an open hit gave $\langle\phi^*\phi\rangle_r = 0.3721 \pm 0.0003$ and one with the restricted hit of ± 0.5 gave $\langle\phi^*\phi\rangle_r = 0.3999 \pm 0.0003$. On the other hand, the relaxing pseudofermion run on the sample of fixed gauge fields from the open hit run gave $\langle\phi^*\phi\rangle =$

0.4035 ± 0.0006 and the same procedure applied to a sample of gauge fields from a run with an exact algorithm which calculates the full determinant analytically at each update gave $\langle\phi^*\phi\rangle = 0.4036 \pm 0.0005$. The agreement of these last figures is another indication that the algorithm is producing a correct, or nearly correct, gauge field distribution. The lag in $\langle\phi^*\phi\rangle_r$ for an open hit is a large effect, 8% in absolute terms and 1000% in terms of the fermionic effect as earlier defined. However, this appears to affect the gauge fields produced by the simulation very little, if at all (i.e. within the ~5% statistical error). There would be no lag in $\langle\phi^* \phi\rangle_r$ if the whole lattice were equilibrated at each update, so it is clearly an effect of not equilibrating the more distant neighbors. This would seem to indicate a very high degree of cancellation of errors as far as the gauge fields are concerned. It should be noted that there certainly is room for such cancellations in a Monte Carlo algorithm. One would expect that not equilibrating more distant neighbors would result in a reduction of the Metropolis transition probability $P(A \to A')$. This is because the more distant neighbors will retain memory of A, and likely weight the system toward keeping the old configuration. However, the same effect will also reduce $P(A' \to A)$. If the amount of reduction is the same, i.e. if each probability $P(A \to A')$ is augmented by an extra factor $f(A,A')$, satisfying $f(A,A') = f(A',A)$, then the equilibrium distribution will be unaffected, since the detailed balance equation (16) is unchanged upon inclusion of such a factor.

Details of simulations of the spinless Schwinger model are given in Ref. 1. Runs have since been extended to approximately 500,000 sweeps on a 4×4 lattice for the data presented in Table I. Each run consumed 15-20 hours of CPU time on a VAX 780. I have also studied the Schwinger model with

Kogut-Susskind fermions,[7] which have the correct spin interactions, although there is still the doubling problem to contend with. Here lattices up to 70 × 70 have been studied, with modest amounts of CPU time on a CRAY XMP. The Schwinger Model is complicated by a symmetry breaking effect. In the continuum, there are an infinity of vacuums distinguished by a parameter θ which does not appear in the Lagrangian.[8] Although, in the massive model, they are not degenerate, they are nevertheless stable. This means that on an infinite lattice there will be no tunneling between different θ-vacua. However, on a finite lattice, there is tunneling among a band of θ's centered around $\theta = 0$, which is, perturbatively, the minimum energy vacuum. Not surprisingly, the tunneling rate increases with increasing hit size, resulting in a hit size dependence of results. I find that these effects can be eliminated by, for each hit size, extrapolating results to $\theta = 0$. On a lattice, θ can be obtained by measuring the expectation value of the imaginary part of the plaquette variable, which, to lowest order, can be equated with the continuum field strength. This is true since θ can be interpreted as an overall background electric field. Larger values of θ can be induced in the simulations by adding a term $\alpha \cdot \mathrm{Im}$ (plaquette) to the action, and varying the parameter α. Several runs for each hit size were performed, some with $\alpha \neq 0$. Physical measurements such as internal energy are then plotted vs. $\langle \mathrm{Im}(\text{plaquette}) \rangle$, and linearly extrapolated to $\theta = 0$ as in Fig. 1. Although the raw data for widely different hit sizes does not appear to agree, extrapolations to $\theta = 0$ do agree within statistical errors which, in this case, are again around 5% of the fermionic effect on the system. Thus, once the symmetry breaking effect is understood and controlled, possible systematic errors of the algorithm are again below an acceptably small limit, even for an

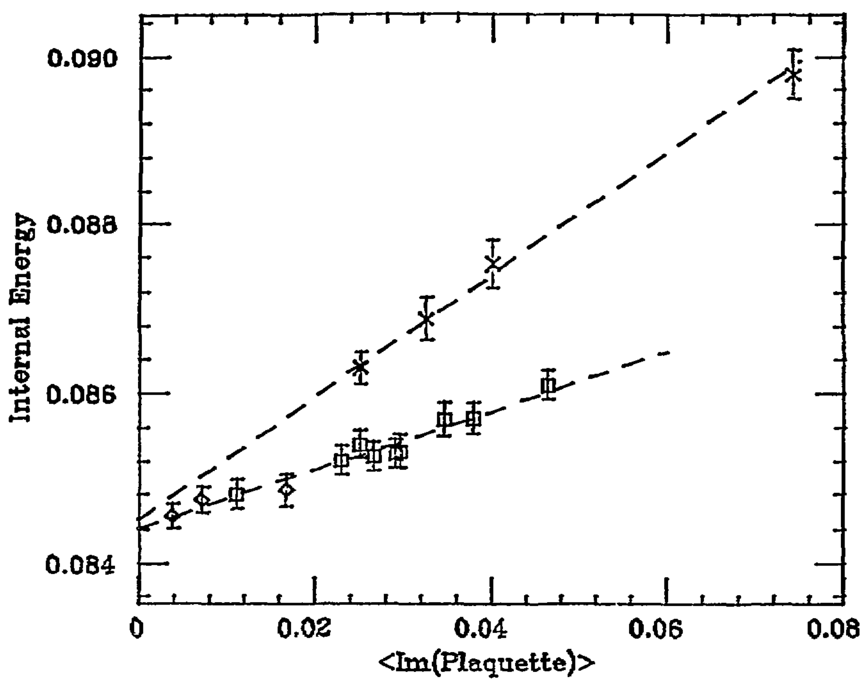

Fig. 1. Extrapolation to $\theta = 0$ for the Schwinger Model with Kogut-Susskind fermions. Data are for $\beta = 6$, $M = 0.07$ on 32×32 (x,□) and 70×70 (◇) lattices. Data plotted as x are for an open hit size, $\eta = 2\pi$, and as □ and ◇ for the restricted hit size $\eta = 1$. Although the data as taken show a hit size dependence, their extrapolations to $\theta = 0$ do not. This suggests that the observed hit size dependence is due to increased vacuum tunneling, and not a systematic error in the algorithm.

open hit size. A detailed report on the Schwinger model with Kogut-Susskind fermions will appear soon.

The look-ahead algorithm shows promise in that possible systematic errors are apparently well under control, at least for the two-dimensional gauge theories studied. Of course, it must be checked that the same is true for non-abelian four-dimensional theories, which will be done soon. On the surface there does not appear to be any feature of the algorithm which is inherently low-dimensional. One may therefore hope with some confidence that it will work well enough to be of practical value in four dimensions, especially since the performance in two-dimensions is far above the minimum requirements of practicability. Finally, I should mention the speed of the algorithm which is, of course, of primary importance. As an example, for the Schwinger model with Kogut-Susskind fermions, one must update four pseudo-fermion fields for each gauge field update, and then, about half of the time (if the gauge field was accepted) these four fields must be updated again. So one has an average of six pseudofermion updates per gauge field update. In four dimensions one will have twelve pseudofermion updates per gauge field update. Thus, one can guess roughly, that the program with fermions will run about one order of magnitude slower than the corresponding pure gauge theory algorithm. Considering the high quality of gauge configurations obtained, due to the freely adjustable hit size, this is extremely fast for a fermion algorithm. In actual programs I have achieved about one-half of this guessed performance; however the current programs have some inefficiencies which, when corrected, should nearly double the speed. For non-abelian theories this ratio may actually go down, since, for an SU(N) theory, the number of color components of the gauge field increases faster with N than do those of the pseudofermion field.

A POSSIBLE JUSTIFICATION

I have attempted to make the previous sections self-contained in that they describe a fast fermion algorithm which is an approximation to a much slower exact algorithm, and which works well in two-dimensional examples. One needn't really analyze the situation further than this. The method can simply be tested in four dimensions, either by varying the hit size and looking for consistency or by comparing results to those obtained using other established methods. However, the complete lack of a detectable systematic error in the examples tested so far, even for a completely open hit size, is intriguing. Could it be that the cancellation of errors which appears to be occurring is exact, making what appears to be an approximation into an exact algorithm? I wish to spend some time pursuing this question. The analysis will be along unconventional lines, as the usual sort of detailed balance proofs do not seem to shed much light on this question. The reason for this is that the ϕ distribution in the fermion/gauge system is dependent on the history of the gauge field updates, as the decision whether to update ϕ fields is contingent upon acceptance of the previous A field update. Also, unlike the algorithm for the boson/gauge system, which uses a single joint probability distribution for both A and ϕ updates, the pseudofermion fields are updated to a different probability function than are the gauge fields. Thus the ϕ fields are constantly chasing the A fields, and don't attain their true equilibrium values. They do as the hit size, and thus the speed of changes in A, is reduced. On the other hand, they reach a sort of equilibrium distribution in that operators which depend on ϕ achieve stable values. Apparently, for a given hit size, the ϕ distribution achieves a stable, uniform, lag behind the true equilibrium distribution. As mentioned before, a symmetric bias will not upset the balance between the transitions $A \to A'$ and $A' \to A$. One needs to know to what extent bias in the transition probabilities due to the lag in the ϕ distribution is symmetric.

My approach follows from the observation that, neglecting for a moment the pure gauge piece of the action, the probability ratios $P(A \to A')/P(A' \to A)$ for the fermion/gauge system and for the analogous boson/gauge system are simply the inverses of each other

$$\frac{P_f(A \to A')}{P_f(A' \to A)} = \frac{P_b(A' \to A)}{P_b(A \to A')} = \frac{\det O(A')}{\det O(A)} \; .(18)$$

The probabilities themselves differ by interchanging the identity of the old and new gauge configurations, A and A', i.e.

$$P_f(A \to A') = P_b(A' \to A) \; .(19)$$

This is true both for the integrated form of the algorithms, using explicit determinants as above, and for the ϕ-dependent probabilities used in the look-ahead algorithm and the standard boson algorithm.

One can view the production of a chain of gauge configurations from P_f as being closely related to the backward construction of chains for the bosonic system from P_b, i.e. the determination of probable past states of the bosonic chain, given the present state. Past and present, of course, refer to Monte Carlo time. The basic idea of what follows is to interpret the look-ahead algorithm as a procedure which produces a representative chain of configurations for the <u>bosonic</u> system, out of an ensemble of possible chains for a specific correct algorithm. This chain ends with a specified final configuration and can, through applying the look-ahead procedure, be extended indefinitely into the past. The ϕ distribution is viewed as a dependent distribution, produced by the specified boson algorithm which is specially chosen so that it is possible to reconstruct past updates. If one can really reconstruct the ϕ distributions that go with past gauge configurations according to probabilities that correspond to a correct boson algorithm, then the usual detailed balance arguments will apply, implying that the ϕ fields are simulating the determinant in the gauge field transition probabilities without bias. Thus the overall philosophy is not to directly analyze the fermion algorithm. Rather, one first establishes a relationship between it and a correct algorithm for the analogous boson system. Then one can make use of the extensive knowledge one has of the boson algorithm, including full detailed balance.

It is interesting to first consider the idea of these "backward" Monte Carlos in general, without the complications of the ϕ fields. The backward Monte Carlo is defined to be the procedure one gets upon interchanging the roles of the new and old gauge fields in the Metropolis transition probability. For a system with a desired equilibrium distribution $P_{eq}(A) = \exp(-\beta H(A))$, the normal Metropolis probability would be

$$P(A \to A') = \mathrm{Min}\big(1,\ \exp(-\beta H(A'))/\exp(-\beta H(A))\big)\ . \qquad (20)$$

One can easily check that the given distribution solves the detailed balance condition (16). For the backward Monte Carlo one uses a transition probability

$$P'(A \to A') \equiv P(A' \to A) \ . \tag{21}$$

Interestingly, this does not yield a nonsensical or totally random equilibrium distribution, but rather

$$P'_{eq}(A) = \exp\big(\beta H(A)\big) = P_{eq}(A)^{-1} \ . \tag{22}$$

The equilibrium distribution produced by the backward Monte Carlo is the negative temperature distribution for the original system, i.e. a distribution with all of the relative equilibrium probabilities inverted. Actually, one wants the pure gauge piece of the probability distribution to be the same for the two systems; however, it too will be inverted under $A \leftrightarrow A'$. This can be corrected by simultaneously taking $\beta \to -\beta$ in S_0. Thus the actual relationship between the fermion/gauge and boson/gauge transition probabilities is

$$P_f(A \to A', \beta) = P_b(A' \to A, -\beta) \ . \tag{23}$$

If the transition probabilities are the same, then why are the forward and backward chains, and equilibrium distributions so different? The reason is two-fold. First, there is a selection effect, in that, for the forward constructed chain an initial state is given and the Monte Carlo produces a distribution of final states. In the backward case, a final state is given and an initial state distribution is sought. The second, related, reason is that the backward transition probabilities are not exactly the transpose of the forward ones; they differ in the diagonal elements, i.e. the probability of rejection. It is useful to consider $P(A \to A')$ as a matrix operating on probability distributions, with indices A and A'. For the boson system

$$M_{bA'A} \equiv P_b(A \rightarrow A') \ . \tag{24}$$

A particular state A_0 is represented by the distribution $\delta(A - A_0)$. What is of interest is the eigenvector of M with eigenvalue 1, which is the equilibrium distribution. It turns out that this is the largest eigenvalue, so that $P_{eq}(A)$ can be obtained by raising M to a high power and operating on an arbitrary initial distribution

$$P_{eq}(A) \propto \lim_{N\rightarrow\infty} M^N_{AA'} P_0(A') \ . \tag{25}$$

This is simply a restatement of the Monte Carlo procedure. Now, although it is true that

$$M_{fA'A} = M_{bAA'} = M^T_{bA'A} \ , \quad A' \neq A \ , \tag{26}$$

where T represents the transpose, the diagonal elements are not the same. $M_{fA'A}$ is defined so that it has columns which sum to unity

$$M_{fAA} = 1 - \sum_{A'\neq A} M_{fA'A} \ , \tag{27}$$

whereas

$$M^T_{bAA} = M_{bAA} = 1 - \sum_{A'\neq A} M_{bA'A} = 1 - \sum_{A'\neq A} M_{fAA'} \ . \tag{28}$$

In other words, M_f is column normalized whereas M^T_b has the same off-diagonal elements but has diagonal elements chosen so that it is row normalized

(normalized in the sense of an arithmetic sum to unity). So

$$M_f = M_b^T + D \ , \tag{29}$$

where D is a diagonal matrix. The extra term D is really due to the same selection effect mentioned above, but occurring at each updating step. The rejection probability is different if the initial state is fixed and the final state is to be determined, or vice versa. Therefore it is this selection effect, occurring at each stage in the process, which makes for the difference in the forward and backward chains.

The next step is to introduce ϕ updates in between the A updates, and to use the ϕ-dependent probabilities for the A updates, $P(A \rightarrow A', \phi)$. The ϕ updates will be heat bath updates for complete ϕ neighborhoods of a given link, as defined earlier. The idea is to choose the ϕ updates so that both the forward and backward (i.e. look-ahead) algorithms end up with the same ϕ distributions along the chain, for fixed sequences of A configurations. Then, since it is known that the ϕ fields in the forward algorithm correctly simulate the determinant factor, the same will be true of the backward algorithm. There are two requirements for the ϕ updates. First, the ϕ distribution must be insensitive to $P(A \rightarrow A)$ since this differs between the two chains. This can be done by forbidding an update of a ϕ neighborhood if the previous A field update was rejected. Consider, for example, an algorithm depicted as follows:

$$-A_1 - \phi_1 - A_2 - \phi_2 - A_3 - \phi_3- \ . \tag{30}$$

"A_1" means to update the field A_1; "ϕ_1" to update the ϕ neighborhood of A_1, etc. The states of the chain are represented by the dashes and can be labelled by the previous update. The above requirement can be met by considering the A_1 - ϕ_1 update as a unit, rather than as two independent updates, with probability $P(A,\phi \rightarrow A',\phi')$. The rule can be implemented by augmenting this probability with a factor

$$\left(1 + \delta_{AA'}(\delta_{\phi\phi'} - 1)\right) . \tag{31}$$

This prevents ϕ from being changed if A is not changed. It does not upset detailed balance since it is symmetric under interchange of primed and unprimed configurations. The second requirement is that the ϕ updates are invertible in the sense that later updates can be undone, uncovering earlier states of the chain, and reproducing the past ϕ configurations. One wants forward and backward produced chains that contain no gauge update rejections to have identical probabilities except for end selection effects. It turns out that this is all that is needed to make use of the known ϕ distribution for the forward algorithm in demonstrating the lack of systematic error for the backward (fermion) algorithm.

On the face, the ϕ updates in (30) are not invertible since, e.g. the ϕ_2 update only partially overlaps the ϕ_1 update. Thus the last time some of the fields in the ϕ_2 neighborhood were updated was much further back on the chain. It would be impossible to recreate the conditions of such an "old" update. However, it is not necessary to exactly recreate these conditions. This can be seen as follows. There is a very large amount of freedom in choosing a boson algorithm, in that the order of ϕ and A updates is immaterial, as far as the equilibrium distribution is concerned. The ensembles of Monte Carlo chains produced by different algorithms will, however, differ in detectable ways, such as correlations along the chain or in average rejection probability. Thus some of the information in a particular ensemble of chains is irrelevant. The nature and extent of this irrelevant information can be characterized by considering artificial combinations of pieces of chains produced by one algorithm, which are constructed so that they belong to the ensemble of a different algorithm. Consider the algorithm where the ϕ neighborhood of a gauge field link is updated both immediately before and immediately after the gauge field is updated.

$$\begin{array}{l} -\phi_{1b} - A_1 - \phi_{1a} - \phi_{2b} - A_2 - \phi_{2a} - \phi_{3b} - A_3 - \phi_{3a} - \\ -\phi'_{1b} - A'_1 - \phi'_{1a} - \phi'_{2b} - A'_2 - \phi'_{2a} - \phi'_{3b} - A'_3 - \phi'_{3a} - \\ -\phi''_{1b} - A''_1 - \phi''_{1a} - \phi''_{2b} - A''_2 - \phi''_{2a} - \phi''_{3b} - A''_3 - \phi''_{3a} - . \end{array} \qquad (32)$$

The subscript "b" refers to a "before" hit and "a" to an "after" hit. One constructs an artificial chain as follows. Choose among the ensemble a chain (primed) which happens to be in the same state after the ϕ'_{2a} update as the double-primed chain is after the ϕ''_{3b} update. Then find another (unprimed) which is in the same state after ϕ_{1a} as the primed chain is after ϕ'_{2b}. This procedure can be extended indefinitely. The artificial chain is made by making the indicated jumps between chains, i.e. of the states

$$-A_1 - \phi_{1a} - A'_2 - \phi'_{2a} - A''_3 - \phi''_{3a} - . \qquad (33)$$

Does this artificial chain have the correct equilibrium distribution? Surprisingly, the answer is yes, because the transition probabilities and the order of updates in the artificial chain are the same as in an algorithm where the ϕ neighborhood is updated only after the corresponding gauge field update (30). Thus the artificial chain (33) does not belong to the ensemble of chains from which it was created (32) but it nevertheless does belong to an ensemble associated with a correct algorithm.

The concept of the artificial chain is the final ingredient needed to complete the argument. Although it does not appear possible to recreate ϕ updates for past positions on chains of types (30) or (32), it is possible for the artificial chain. Suppose one is given the state of the chain ϕ''_{3a}. One wishes to have the ϕ distribution at position ϕ''_{3b}, for use in the gauge field update probability

$$P(A''^{old}_3 \rightarrow A''_3, \phi) \quad . \tag{34}$$

One can achieve this ϕ distribution by inserting the trial old value of the gauge field into the gauge configuration and simply performing the update ϕ''_{3b}. This is just the look-ahead update procedure described earlier. If one wanted to stay on the double-primed chain, then one would have difficulty in undoing the ϕ''_{3b} hit, since the last time some of those fields were updated was long ago. Use of the artificial chain obviates this problem by allowing a skip to a place on a different chain which is reversible. The artificial chain argument is just making explicit the extent to which information carried by the ϕ fields is irrelevant to producing the correct equilibrium distribution. It is telling us that the ϕ_3 update in (30) can be erased, uncovering the ϕ distribution that exists at the point ϕ_2 simply by re-updating the ϕ_3 fields to the conditions that existed before the A_3 update.

Recapitulating, one considers the ensemble of all Monte Carlo chains produced by the algorithm (30) for the boson/gauge system. For this system, detailed balance in the ϕ fields can be used to show that if the ϕ fields are ignored, which is equivalent to integrating over the ϕ distribution, the effective binary gauge field transition probability for $A \neq A'$ is, from the bosonic analog of (15),

$$P_b(A \to A') = \det O(A)\exp\left(S_0(A)\right)f(A,A') \ ; \qquad f(A,A') = f(A',A) \ . \tag{35}$$

The look-ahead algorithm reproduces almost the same ensemble of chains, but works backward from a given configuration rather then the usual forward direction. The forward and backward produced chains differ only in the probability of retaining the same gauge configuration, i.e. of rejecting an update, because the normalization conditions of forward and backward transition probabilities are necessarily different. However, since neither algorithm updates a ϕ field if the corresponding gauge field update is rejected, the dependent ϕ distributions are not sensitive to this difference. The artificial chain concept was used to prove that the results of individual ϕ updates used by the two algorithms are the same. Since the look-ahead algorithm, therefore, produces the same ϕ distribution as the normal boson algorithm, the average binary transition probabilities (35) must also be the same. Thus, after integrating out the ϕ distribution one obtains an average fermion transition probability

$$P_f(A \to A',\beta) = P_b(A' \to A,-\beta) = \det O(A')\exp\left(-S_0(A')\right)f(A,A') \tag{36}$$

which, using (16), gives

$$P_{feq}(A) = \exp\left(-S_0(A)\right)\det O(A) \tag{37}$$

the desired fermion equilibrium distribution. Also, it is now clear why the running ϕ distribution does not appear to be in equilibrium with the fermion gauge distribution. Rather it is in equilibrium with the corresponding gauge distribution for the negative β boson/gauge system, which has a different weight for a given gauge configuration because of the different probability of retention compared to the fermion case.

The above justification is relatively complex and admittedly unconventional. I do not consider it at the level of a rigorous proof, although I think it contains most of the necessary conceptual ingredients for one. I am still searching for possible flaws in the logic and a clearer and more precise notation in which to express the relevant ideas. The look-ahead algorithm is in a new class of algorithms in which an auxiliary field is updated according to a different function than is the main field one wishes to simulate. For this reason, detailed balance in the auxiliary field is not satisfied. Relaxing the full detailed balance condition allows for a greater range of algorithms, some of which might be able to simulate distributions which cannot be simulated by local algorithms which are fully balanced. In the absence of analytical proofs, such algorithms may still be used by testing them empirically, either against themselves, such as looking for hit size dependences, or against slower, but more easily justifiable, algorithms. I hope to have convinced the reader, however, that analysis of such algorithms, although difficult, is not impossible. It is hoped that further progress in this area will result in a completely rigorous justification of the look-ahead algorithm, or a variant of it.

REFERENCES

1. M. Grady, Phys. Rev. D32 (1985) 1496, and Argonne Report No. ANL-HEP-PR-84-69 (1984) unpublished.
2. F. Fucito, E. Marinari, G. Parisi, and C. Rebbi, Nucl. Phys. B180 (1981) 369.
3. A variation of the pseudofermion algorithm which is more similar in spirit to the one discussed here is given in G. Bhanot, U. M. Heller, and I. O. Stamatescu, Phys. Lett. 129B (1983) 440.
4. N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, J. Chem Phys. 21 (1953) 1087.
5. I thank B. Bunk and M. Blairon for suggesting that a multivariate heat bath be used here. The heat bath method is discussed in M. Creutz, Phys. Rev. D21 (1980) 2308.
6. E. Marinari, G. Parisi, and C. Rebbi, Nucl. Phys. B190 [FS3] (1981) 734.
7. T. Banks, S. Raby, L. Susskind, J. Kogut, D. R. T. Jones, P. N. Scharbach, and D. K. Sinclair, Phys. Rev. D15 (1977) 1111; L. Susskind, Phys. Rev. D16 (1977) 3031.
8. S. Coleman, R. Jackiw, and L. Susskind, Ann. of Phys. 93 (1975) 267; S. Coleman, Ann. of Phys. 101 (1976) 239.