ALGORITHMS FOR SIMULATING FERMIONS

MICHAEL CREUTZ
Physics Department,Brookhaven National Laboratory
Upton, NY 11973,USA

ABSTRACT

I review some of the approaches to including the effects of dynamical quark loops in Monte Carlo simulations of quantum field theories. The discussion begins with a brief introduction to anticommuting variables and ends with higher order improvements to the hybrid Monte Carlo scheme. Pseudofermion, Langevin, and various combinations thereof are compared.

1. Introduction

Fermions provide some of the biggest challenges to the field of lattice gauge theory. One difficulty appears already at the level of formulating an appropriate action. Here we have the notorious doubling of species appearing in the simplest schemes incorporating chiral symmetry. The two popular schemes for handling this are the Kogut-Susskind formulation where each site carries only a single component of the Dirac spinors, and the Wilson projection operator approach where chiral symmetry is abandoned with the hope that it will be recovered in the continuum limit. Both of these schemes are mentioned in other chapters of this book. Here I will be quite generic and assume we have an acceptable lattice transcription of the Dirac equation.

The purpose of this chapter is to discuss the other primary difficulty with fermions: the extreme computational difficulties that appear on adding them to Monte Carlo simulations. While several large scale simulations are ongoing, even the most meager results strain the most advanced computational resources available. This chapter discusses the algorithms currently in use, several of which are quite clever. Nevertheless, there remains a certain awkwardness in known approaches that hints that new and better techniques will evolve.

For this discussion I will be quite generic and assume we are interested in a path integral of form

$$Z = \int (dA)(d\psi)(d\psi^*)\ \exp(-S_G(A) - \psi^* M(A)\psi). \tag{1}$$

Here the gauge fields are formally denoted A and fermionic fields ψ and ψ^*. As I will be concentrating on fermionic details, I will ignore the technicality that the gauge fields are group elements. All details of the fermionic formulation are hidden in the matrix $M(A)$. While I may call A a gauge field, the algorithms are general, and have potential applications in other field theories and condensed matter physics.

The numerical difficulties with fermionic fields stem from their being anticommuting quantities. Thus it is not immediately straightforward to place them on a computer, which likes to manipulate numbers. Indeed, the Boltzmann factor is formally an operator in a Grassmann space, and cannot be directly interpreted as a probability for Monte Carlo purposes. All algorithms in current use eliminate the fermions at the outset by a formal analytic integration. This is possible because most actions in practice are, or can easily be made, quadratic in the fermionic fields. The fermion integrals are then over generalized gaussians. Unfortunately, the resulting expressions involve the determinant of a large, albeit sparse, matrix. This determinant introduces nonlocal couplings between the bosonic degrees of freedom, making the path integrals over the remaining fields rather time consuming. Nevertheless, various tricks have been developed to minimize the pain with fermions, and in other chapters of this book you can find numerous results from such calculations. This chapter reviews some of these tricks.

2. Anticommuting variables

I begin with a brief review of Grassmann variables [1]. I start with a set $\{\psi_i\}$ of anticommuting variables

$$[\psi_i, \psi_j]_+ \equiv \psi_i\psi_j + \psi_j\psi_i = 0. \tag{2}$$

To keep the formulae simple I combine spatial dependence, internal symmetry, and spinor indices into a single index i.

These anticommutation relations can be realized via matrices. Had I just two variables, I could represent the ψ_i using two sets of independent Pauli matrices $\{\sigma\}$ and $\{\tau\}$. In fairly standard notation I could satisfy Eq. (2) by writing $\psi_1 = \sigma^+$ and $\psi_2 = \sigma^z\tau^+$. Going on to more variables I would need to increase the number of independent matrices used in this direct product. As the number of independent Grassmann variables increases, the overall matrix dimension necessary to represent the algebra increases exponentially. Because of this growth, using explicit representations for the Grassmann variables has not, at least so far, been particularly useful for computational purposes. For this reason I will proceed more formally and just treat the ψ_i abstractly.

Generalizing complex conjugation to include these variables, I adopt the convention that corresponding to each ψ_i I have another independent Grassmann variable ψ_i^*. Furthermore, I postulate

$$\begin{gathered}(\psi_i^*)^* = \psi_i \\ (\psi_1 \dots \psi_n)^* = \psi_n^* \dots \psi_1^*. \end{gathered} \tag{3}$$

If I consider just a single variable ψ, then ψ^2 must vanish, and a general function $f(\psi)$ can be expanded with two terms

$$f(\psi) = f_0 + \psi f_1. \tag{4}$$

Similarly, with n variables, a general function can be expanded as a finite polynomial with 2^n terms.

To define integration over an anticommuting variable, I demand the properties of linearity and invariance under a translation of variables. These are summarized in the axioms

$$\int d\psi (f(\psi)\alpha + g(\psi)\beta) = \left(\int d\psi f(\psi)\right) \alpha + \left(\int d\psi g(\psi)\right) \beta \tag{5a}$$

$$\int d\psi f(\psi) = \int d\psi f(\psi + \psi'). \tag{5b}$$

This is sufficient to imply that for the function in Eq. (4)

$$\int d\psi f(\psi) = K f_1 \tag{6}$$

where the normalization K is undetermined. I adopt the convention $K = i$ so that

$$\begin{aligned} &\int d\psi \ \ \psi = i \\ &\int d\psi \ \ 1 = 0 \\ &\int d\psi^* d\psi \ \ \psi^* \psi = 1. \end{aligned} \tag{7}$$

Note that under multiplicative rescaling a Grassmann integral behaves as

$$\int d\psi f(\psi a) = (\int d\psi f(\psi))a. \tag{8}$$

This can be written in the heuristic form $d(\psi a) = (d\psi)/a$.

For integration over several anticommuting variables, I have

$$\int d\psi_1 \ldots d\psi_n \psi_1 \ldots \psi_n = i^n (-1)^{n(n-1)/2}. \tag{9}$$

The analog of Eq. (8) in this case is

$$\int (d\psi) f(M\psi) = |M| \int d\psi f(\psi) \tag{10}$$

where M is an arbitrary matrix, $|M|$ its determinant, and $(d\psi)$ denotes $d\psi_1 \ldots d\psi_n$. Note that eq. (10) immediately implies the Matthews-Salam [2] formula for a fermionic Gaussian integral

$$\int (d\psi d\psi^*) \ e^{-\psi^* M \psi} = |M| \tag{11}$$

where $(d\psi d\psi^*) = d\psi_1 d\psi_1^* \dots d\psi_n d\psi_n^*$.

Eq. (11) provides an easy way out of the difficulty that our partition function is not an ordinary integral. Indeed, I explicitly integrate out the fermions to convert Eq. (1) to

$$Z = \int (dA)\ |M|\ e^{-S_G}. \tag{12}$$

This is now an integral over ordinary numbers and therefore in principle amenable to Monte Carlo attack.

For the remainder of this chapter I assume that the fermions have been formulated such that $|M|$ is positive and thus the integrand in Eq. (12) can be regarded as proportional to a probability measure. If this is not so, one can always double the number of fermionic species, using $M^\dagger$ for the extra ones, thus replacing M by $MM^\dagger$. The case where M is not positive is not yet well understood, and indeed in some such cases the path integral may not be well defined. These problems are probably closely connected with the difficulties of placing chiral fermions on the lattice. This issue is quite important; with a chemical potential present to give a background fermion density, the determinant is not in general positive. Indeed, it is not yet known how to simulate this physically important problem.

Direct Monte Carlo study of the partition function in Eq. (12) is still not practical because of the large size of the matrix M. In our compact notation, this is a square matrix of dimension equal to the number of lattice sites times the number of Dirac components times the number of internal symmetry degrees of freedom. Thus, it is typically a tens of thousands by tens of thousands matrix, precluding any direct attempt to calculate its determinant. It is, however, generally an extremely sparse matrix because most popular actions do not directly couple distant sites. All the Monte Carlo algorithms used in practice for fermions make essential use of this fact.

3. Exact local algorithms

Many current simulations make additional approximations beyond the lattice cutoff and the statistical errors inherent in Monte Carlo simulations. Such approximations are in principle not required, but save substantial computer time. Here I first discuss a few of the approaches that do not make such additional assumptions. These are in a sense "exact" because with enough Monte Carlo statistics they will give the correct correlations for the partition function. They are still approximations in the sense of having a finite lattice spacing and finite lattice size, but these approximations are already present in the bosonic simulations. Later I consider some approximate variations which are considerably faster. Finally I will show how to make some of those approximate approaches "exact" again.

Some time ago Weingarten and Petcher [3] presented a simple "exact" algorithm. They observed that by introducing auxiliary set of complex scalar fields ϕ one can rewrite Eq. (12) in the form

$$Z = \int (dA)(d\phi^* \ d\phi) \exp(-S_G - \phi^* M^{-1} \phi). \tag{13}$$

Thus a successful fermionic simulation would be possible if one could obtain configurations of fields ϕ and A with probability distribution

$$P(A, \phi) \propto \exp(-S_G - \phi^* M^{-1} \phi). \tag{14}$$

To proceed I will assume that M is a positive matrix so this distribution is well defined.

Ref. 3 notes that while M^{-1} is the inverse of an enormous matrix, one really only needs $\phi^* M^{-1} \phi$, which is just one matrix element of this inverse. Furthermore, with a local fermionic action the matrix M is extremely sparse, the nonvanishing matrix elements only connecting nearby sites. In this case there exist quite efficient iterative schemes for finding the inverse of a large sparse matrix applied to a single vector. Thus it was proposed to directly simulate the partition function in Eq. (5.1) using a Gauss-Seidel algorithm to calculate $M^{-1}\phi$. Most recent work has turned to the conjugate gradient algorithm for this inversion.

The conjugate gradient algorithm has also been quite useful for fermionic studies in the "valence" or "quenched" approximation, where the Dirac equation is solved on gauge field configurations obtained by simulations ignoring the feedback of the dynamical quarks on the gauge fields themselves [4]. This approximation, discussed many places elsewhere in this book, circumvents the problem of simulating dynamical quarks, but still imposes substantial computational demands when the lattices are large.

The conjugate gradient method to find $\xi = M^{-1}\phi$ works by finding the minimum over ξ of the function $|M\xi - \phi|^2$. The solution is iterative; starting with some ξ_0, a sequence of vectors is obtained by moving to the minimum of this function along successive directions d_i. The clever trick of the algorithm is to choose the d_i to be orthogonal in a sense defined by the matrix M itself; in particular $(Md_i, Md_j) = 0$ whenever $i \neq j$. This last condition serves to eliminate useless oscillations in undesirable directions, and guarantees convergence to the minimum in a number of steps equal to the dimension of the matrix. There are close connections between the conjugate gradient inversion procedure and the Lanczos algorithm for tridiagonalizing sparse matrices. [5]

The procedure is a simple recursion. Select some arbitrary initial pair of non-vanishing vectors $g_0 = d_0$. For the inversion problem, convergence will be improved if these are a good guess to $M^{-1}\phi$. Then generate a sequence of further vectors by iterating

$$\begin{aligned} g_{i+1} &= (Mg_i, Md_i)g_i - (g_i, g_i)M^\dagger M d_i \\ d_{i+1} &= (Md_i, Md_i)g_{i+1} - (Md_, Mg_{i+1})d_i \end{aligned} \tag{15}$$

This construction assures that g_i is orthogonal to g_{i+1} and $(Md_i, Md_{i+1}) = 0$. It should also be clear that the three sets of vectors $\{d_0, ...d_k\}$, $\{g_0, ...g_k\}$, and $\{d_0, ...(M^\dagger M)^k d_0\}$ all span the same space.

The remarkable core of the algorithm, easily proved by induction, is that the set of g_i are all mutually orthogonal, as are Md_i. For an N dimensional matrix, there can be no more than N independent orthogonal vectors. Thus, ignoring roundoff errors, the recursion in Eq. (15) must terminate in N or less steps with the vectors g and d vanishing from then on. Furthermore, as the above sets of vectors all span the same space, in a basis defined by the g_i the matrix $M^\dagger M$ is in fact tri-diagonal, with (Mg_i, Mg_j) vanishing unless $i = j \pm 1$.

To solve $\phi = M\xi$ for ξ, simply expand in the d_i

$$\xi = \sum_i \alpha_i d_i.$$

The coefficients are immediately found from the orthogonality conditions

$$\alpha_i = (Md_i, \phi)/(Md_i, Md_i).$$

Note that if I start with the solution $d_0 = M^{-1}\phi$, then I have $\alpha_i = \delta_{i0}$.

This discussion applies for a general matrix M. If M is Hermitian, then one can work with better conditioned matrices by replacing the orthogonality condition for the d_i with (d_i, Md_j) vanishing for $i \neq j$.

In practice, at least when the correlation length is not too large, this procedure adequately converges in a number of iterations which does not grow severely with the lattice size. As each step involves vector sums with length proportional to the lattice volume, each conjugate gradient step takes a time which grows with the volume of the system. Thus the algorithm of Ref. 3 is expected to require computer time which grows as the square of the volume of the lattice. Such a severe growth has precluded use of this algorithm on any but the smallest lattices. Nevertheless, it does show the existence of an exact algorithm with considerably less computational complexity than would be required for a repeated direct evaluation of the determinant of the fermionic matrix.

Here and below when I discuss volume dependences, I ignore additional factors from critical slowing down when the correlation length is also allowed to grow with the lattice size. The assumption is that such factors are common for the local algorithms treated here. In addition, such slowing occurs in bosonic simulations, and I am primarily concerned here with the extra problems presented by the fermions.

This issue of critical slowing is potentially quite important, and has been mentioned in Sokal's chapter of this book. Cluster and multigrid type algorithms will inevitably become important to speed the above matrix inversions. At present, however, in lattice gauge theory the correlation lengths are rather small, and these acceleration algorithms are not yet used in major production runs. Thus I will not mention them further in this chapter.

Two other exact, but also volume squared, methods for fermionic simulation were presented in Ref. [6]. To use a Metropolis *et. al.* [7] scheme to find a configuration of A fields with distribution

$$P_{eq}(A) \propto |M(A)| e^{-S_G(A)}, \tag{16}$$

requires knowledge of how the determinant $|M|$ changes when A is replaced by a trial value A'. Actually one only needs the ratio of the old and the new determinants, and this can be calculated as an expectation value in two ways. First, if I construct an ensemble of complex scalar fields ξ with distribution

$$P(\xi) \propto e^{-\xi^* M(A) \xi} \tag{17}$$

then I have

$$\frac{|M(A')|}{|M(A)|} = \frac{1}{\langle \exp(-\xi^*(M(A') - M(A))\xi) \rangle}. \tag{18}$$

Alternatively, if I construct the fields ξ using the trial A'

$$P(\xi) \propto e^{-\xi^* M(A') \xi}, \tag{19}$$

then I have

$$\frac{|M(A')|}{|M(A)|} = \langle \exp(-\xi^*(M(A) - M(A'))\xi) \rangle. \tag{20}$$

Both approaches involve a Monte Carlo to find the ensemble of ξ fields inside the Monte Carlo determination of the A fields. Thus they are also volume squared algorithms.

Grady [8] found an intriguing variation on the second of these two approaches. In particular, the ensemble average over the ξ fields is unnecessary in this "look ahead" scheme where the probability for ξ is determined from the trial field A'. Consider a trial change A' chosen with a probability distribution $P_{T,A}(A')$. Solely to simplify the following equations, assume that this trial probability is symmetric under interchange of A and A'. Then generate a single ξ field with probability distribution as given in Eq. (19). The prescription is to accept the change with probability

$$P_{acc} = \min[1,\ \exp(S_G - S'_G + \xi^*(M' - M)\xi)]. \tag{21}$$

Here I use the shorthand notation S_G, S'_G, M, and M' for $S_G(A)$, $S_G(A')$, $M(A)$, and $M(A')$, respectively.

To justify this procedure, consider the overall probability for taking A to A'

$$P(A \to A') = P_{T,A}(A') \frac{1}{Z_\xi} \int (d\xi^* d\xi) e^{-\xi^* M' \xi} P_{acc}. \tag{22}$$

Here I have defined the normalization factor for the ξ integral

$$Z_\xi = \int (d\xi^* d\xi) e^{-\xi^* M' \xi} \ \propto \ |M'|^{-1}. \tag{23}$$

Multiplying Eq. (22) by the equilibrium distribution in Eq. (16) and combining things gives

$$\begin{aligned} P_{eq}(A) \ P(A \to A') \propto & |M||M'| P_{T,A}(A') \\ & \int (d\xi^* d\xi) \min[e^{-S_G - \xi^* M' \xi}, \ e^{-S'_G - \xi^* M \xi}]. \end{aligned} \tag{24}$$

Remembering the symmetry of P_T, we see that this expression is symmetric under interchange of primed and non-primed variables. This symmetry is precisely the statement of detailed balance for the equilibrium distribution from Eq. (16).

The fact that a large ensemble of ξ fields is not required is a definite advantage of this approach. Nevertheless, it still contains a Monte Carlo inside a Monte Carlo to obtain the equilibrated ξ field. Thus as an exact algorithm it still requires computer time growing as the system volume squared.

The above exact approaches all require volume squared times. To avoid this, many fermionic schemes used in practice involve additional approximations. This usually involves an expansion in a step size for proposed changes. To eliminate systematic errors in principle requires an extrapolation to the limit of vanishing step. I will later return to exact algorithms and discuss how using a guided random walk can give a volume dependence intermediate between the above volume squared and the linear behavior of bosonic simulations.

The advantage of making small steps lies in the fact that a time consuming step such as a conjugate gradient inversion or a Monte Carlo generation of auxiliary fields need only be done once per sweep of all the gauge variables. In essence, this is an attempt to eliminate a Monte Carlo inside a Monte Carlo. Once one is making small steps anyway, there is no particular loss in using algorithms formulated in terms of a differential evolution. This is the basis of both the Langevin [9-10] and the microcanonical [11] methods discussed below. I begin this treatment of small-step algorithms with one of the oldest.

4. Pseudofermions

Fucito, Marinari, Parisi, and Rebbi [12] proposed a simple approximate method for calculating changes in the determinant of the matrix M. They begin by rewriting Eq. (12) in the form

$$Z = \int (dA) \ e^{-S_{PF}}. \tag{25}$$

where

$$S_{PF} = S_G(A) - \mathrm{Tr}\ \log\ M(A) \tag{26}$$

For a Metropolis *et al.* [7] updating scheme one needs to know the change in the action upon a trial change of A. As a first approximation, consider making only small changes in the gauge field and then linearizing the change in the action

$$\frac{dS_{PF}}{dA} = \frac{dS_G}{dA} - \mathrm{Tr}(M^{-1}\frac{dM}{dA}). \tag{27}$$

The quantity $\frac{dM}{dA}$ is easily calculated for a local M. The inverse of the matrix M is estimated using

$$(M^{-1})_{ij} = \langle \xi_j^*\ \xi_i \rangle \tag{28}$$

where the expectation value is over fields ξ, called pseudofermions, and distributed with weighting

$$P(\xi) \propto \exp(-\xi^* M \xi). \tag{29}$$

Note that this is the same distribution required in Eq. (17). A standard Monte Carlo simulation is used to give a set of N_c configurations of the ξ fields to estimate this expectation value. This simulation is normally done only once per full sweep of the lattice variables. This is not a major new assumption because a small-step-size approximation is already being made in using only the first derivative to calculate the changes in the action.

This algorithm, as several of the fermionic approaches discussed later, is not exact because of the approximation of small changes in A. The step size is merely a parameter in the standard applications of the Metropolis *et al.* [7] algorithm to bosons, but here it acquires a more significant role in characterizing an approximation. Whether the step size is sufficiently small can in principle be determined by comparing results for several values and doing an extrapolation to zero. Unfortunately, the amount of computer time necessary is sufficiently large that this check is rarely made.

Another source of error appears if the pseudofermionic fields are not calculated with the appropriate distribution. This would happen with insufficient equilibration time during the Monte Carlo simulation from which they are obtained. This error can in principle be eliminated by a trick if M is the square of a simple operator, say $M = DD^\dagger$. In this case consider first generating a random vector χ with a Gaussian distribution

$$P(\chi) \propto e^{-\chi^\dagger \chi} \tag{30}$$

As all components of χ are uncorrelated, it can be rather quickly generated. Then a simple change of variables gives a properly distributed pseudofermionic field

$$\chi = D^\dagger \xi \tag{31}$$

This equation can in principle be solved by some iterative algorithm such as the conjugate gradient method. This trick replaces the convergence of a Monte Carlo updating of the pseudofermionic fields with a potentially tedious inversion. I mention it here because this use of Gaussian random numbers is potentially quite useful with other fermionic algorithms as well.

The finite number of configurations of pseudofermionic fields used to estimate the expectation value in Eq. (28) introduces a random error into the estimate of the inverse of M. These errors, however, average out in the final extrapolation of observables to zero step size. This is a point I return to later when I discuss interpolation between pseudofermions and the Langevin approach.

5. Langevin, microcanonical, and hybrid schemes

Both the Langevin and microcanonical algorithms for lattice gauge theory are formulated as differential equations for evolution in a fictitious "time" τ. While these approaches are also applicable for the pure gauge theory, their main interest appears with fermionic simulations, where a differential evolution permits time consuming conjugate gradient inversions to be done only once per sweep of the lattice variables.

Rather than in terms of field theory, I frame this discussion in the context of a single degree of freedom, the coordinate x of a particle of mass m moving in one dimension. I do this because the basic ideas of these algorithms are nothing more than generalizations of Newton's equation. I will also treat the Langevin and microcanonical approaches together as limits of a more general hybrid formalism. I will be using a second order version of the Langevin equation, similar to that advocated by Horowitz. [13]

I begin by considering a particle moving in a potential $V(x)$. Newton's equation for the particle motion is

$$m\frac{d^2x}{d\tau^2} = -\frac{\partial V}{\partial x}. \tag{32}$$

I now doctor this motion by adding two terms. First I add a drag slowing the particle down with a force proportional to its velocity. This will tend to damp out any motion until the particle lies at a minimum of the potential. To keep things moving, I then add a random noise to the system. Thus consider the equation

$$m\frac{d^2x}{d\tau^2} = -\frac{\partial V}{\partial x} - \alpha\frac{dx}{d\tau} + (\frac{2\alpha}{\beta})^{1/2}\ \eta(\tau). \tag{33}$$

where α and β are parameters. Here the noise $\eta(\tau)$ formally satisfies

$$\langle\eta(\tau)\eta(\tau')\rangle = \delta(\tau-\tau'). \tag{34}$$

How the $\eta(\tau)$ is actually defined will become clearer momentarily when I make the evolution discrete. I have written the coefficient of the noise as $(2\alpha/\beta)^{1/2}$ with hindsight. It is convenient to introduce the momentum p of the particle and rewrite this second order equation as two first order equations

$$\begin{aligned} {dp\over d\tau} &= -{\partial V\over\partial x} - {\alpha p\over m} + ({2\alpha\over\beta})^{1/2}\ \eta(\tau), \\ {dx\over d\tau} &= {p\over m}. \end{aligned} \tag{35}$$

For simulation purposes the fictitious time is discreetly made discrete. Thus consider taking steps of size ϵ in τ. In one such step, p and x at time τ will become p' and x' at time $\tau+\epsilon$. Eq. 6.4 becomes

$$p' = p + \epsilon\ (-{\partial V\over\partial x} - {\alpha p\over m} + ({2\alpha\over\beta})^{1/2}\ \eta), \tag{36}$$

$$x' = x + {\epsilon p'\over m}. \tag{37}$$

Note that I have written the updating of x such as to use the new value p' of the momentum. This amounts to alternately updating the coordinates and the momenta of the system. Such a "leap frog" procedure effectively treats these variables at interleaved times. In the deterministic limit this advantageous technique serves to eliminate $\mathcal{O}(\epsilon^2)$ errors in the evolution. This transformation also preserves phase space volumes and is inverted by changing the sign of ϵ. These properties will be particularly helpful later when I return to exact algorithms.

The quantity η is obtained from a random number generator with probability distribution $\rho(\eta)$. The properties required of $\rho(\eta)$ are simply specified in terms of its moments

$$\int d\eta\ \rho(\eta)\ \eta^j = \begin{cases} = 1, & j=0 \\ = 0, & j=1 \\ = 1/\epsilon, & j=2 \\ \le \mathcal{O}(\epsilon^{-j/2}), & j\ge 3. \end{cases} \tag{38}$$

In this equation the first part indicates that the probability distribution is normalized, the second balances positive and negative noise, the third normalizes the delta function in Eq. (34), and the fourth eliminates nasty tails from the distribution. In many discussions the noise is considered as Gaussian, but this is not generally necessary.

To proceed, consider ensembles of particle coordinates and momenta. A necessary condition for any simulation algorithm is that it leave the equilibrium ensemble unchanged. Indeed, this condition is also sufficient if the algorithm is ergodic. Thus I am interested in finding ensembles of (x,p) pairs invariant under the evolution of Eq. (35), or its discretization in Eqs. (36-37).

Consider an ensemble with a probability density $P(x,p)$ of finding a state with given coordinate x and momentum p. Updating the states gives a new ensemble with probability distribution

$$
\begin{aligned}
P'(x',p') &= \int dx\ dp\ P(x,p)P(x,p \to x',p') \\
&= \int dx\ dp\ d\eta\ \rho(\eta)\ P(x,p) \\
&\quad \times \delta(p'-p-\epsilon\ (-\frac{\partial V}{\partial x} - \frac{\alpha p}{m} + (\frac{2\alpha}{\beta})^{1/2}\ \eta)) \\
&\quad \times \delta(x'-x-\frac{\epsilon p'}{m})
\end{aligned}
\tag{40}
$$

A little algebra gives the result

$$
\begin{aligned}
&P'(x,p) = P(x,p) \\
&\quad +\epsilon\ [(\frac{\partial H}{\partial x}\frac{\partial P}{\partial p} - \frac{\partial H}{\partial p}\frac{\partial P}{\partial x}) + \alpha(\frac{1}{\beta}\frac{\partial^2 P}{\partial p^2} + \frac{p}{m}\frac{\partial P}{\partial p} + \frac{1}{m}P)] + \mathcal{O}(\epsilon^2)
\end{aligned}
\tag{41}
$$

Here I have defined the Hamiltonian corresponding to the original Newton's equation of Eq. (32)

$$H = \frac{p^2}{2m} + V(x) \tag{42}$$

In deriving Eq. (41) it is necessary to keep terms of order η^2 because of the $1/\epsilon$ in the third part of Eq. (38).

Eq. (41) is a Fokker-Planck equation for the evolution of the probability density $P(x,p)$. It is now easily verified to order ϵ that a stationary distribution for this evolution is the simple Boltzmann weight

$$P(x,p) = \exp\{-\beta H(p,x)\}. \tag{43}$$

When α is non-zero, the algorithm is ergodic, and the solution is unique. Note that this distribution factors into a function of p times a function of x. Thus the equilibrium distributions of p and x are independent.

We see that repeated updating of an ensemble with the stochastic differential equation of Eq. (33) will eventually give thermal equilibrium at inverse temperature β. The source term can be thought of as a thermal bath coupled to the system. To apply these ideas to the gauge theory problem and obtain a distribution of fields as in Eq. (14), I merely generalize, replacing the variable x with the fields A and ϕ and replacing the potential $\beta V(x)$ with the action $S_G + \phi^* M^{-1}\phi$. Note that calculating the term involving $\frac{\partial V}{\partial x}$ will require, among other things, the evaluation of

$$\frac{\partial}{\partial A}\phi^* M^{-1}\phi = -\phi^* M^{-1}\frac{\partial M}{\partial A}M^{-1}\phi. \tag{44}$$

This involves $M^{-1}\phi$, which requires a conjugate gradient or equivalent inversion every time step.

It is interesting to consider various limits of this stochastic evolution. First, suppose that I had not included the drag term in Eq. (33). This situation follows from taking α to zero with β varying proportionally to keep the stochastic term. Thus noise without drag gives infinite temperature; that is, the random force will, on the average, increase the system energy without bound. Alternatively, if I include the drag but not the noise, the system will drop into a minimum energy state at effectively zero temperature. A finite temperature simulation requires the presence of both the drag and noise terms. The relation between the dissipative term proportional to α and the fluctuations of strength $(2\alpha/\beta)^{1/2}$ is the essence of the fluctuation dissipation theorem.

Note that the distribution in Eq. (43) is independent of the parameter α. Thus there is a class of algorithms. One of these corresponds to taking parameter m to zero. This can be effected by simultaneously adjusting α and rescaling the units of time. In this case Eq. (33) becomes first order and is the usual Langevin equation as used in Refs. [6] and [7].

$$\frac{dx}{d\tau} = -\frac{\partial V}{\partial x} + (\frac{2}{\beta})^{1/2}\ \eta(\tau). \tag{45}$$

For later reference, I write this in the discrete form for evolving x to x' in one time step of length ϵ

$$x' = x + \epsilon \times (-\frac{\partial V}{\partial x} + (\frac{2}{\beta})^{1/2}\ \eta). \tag{46}$$

Another interesting limit corresponds to taking α to zero while holding β constant. This case removes both the drag and noise terms, and returns simply to Newton's equation. This is the microcanonical approach, first advocated for pure gauge theories by Callaway and Rahman [14] and proposed for fermionic simulations in Ref. [11]. In this case the algorithm has no explicit dependence on β. Indeed, microcanonical algorithms require the temperature to be determined after the fact by some sort of thermometer. A convenient monitor is the average kinetic energy $\frac{1}{2}kT = \langle\frac{p^2}{2m}\rangle$. To change the temperature, one should start with a different total initial energy, which remains constant during the evolution.

Intermediate values of α give hybrid algorithms interpolating between the Langevin and microcanonical approaches. An alternative hybrid approach was proposed by Duane and Kogut, [15], who advocate updating with a microcanonical scheme for some number of iterations and then doing a step where all the momenta touch a heat bath. Eq. (43) shows that the momenta in equilibrium are Gaussianly distributed; thus, the latter step consists of replacing them all with new Gaussian random numbers.

This mixing of a molecular dynamics simulation with a randomizing noise on the momenta is is philosophically similar to keeping a small α in the above discussion. This would represent a small continuous refreshing of the momenta rather than a large refreshing of all momenta at the end of a "trajectory." Such an approach has been advocated in Ref. 13.

When the microcanonical trajectory length is short, the hybrid algorithm in fact becomes the first order Langevin equation. To see this, consider first replacing p with a Gaussianly distributed random number and then update the system microcanonically for a short time δ. This will take the coordinate x into

$$x' = x + \frac{p\delta}{m} - \frac{\delta^2}{2m}\frac{\partial V}{\partial x} + \mathcal{O}(\delta^3). \tag{47}$$

If the microcanonical updating time δ is small enough that the $\mathcal{O}(\delta^3)$ effects are negligible, then this evolution is identical to that in Eq. (46) where $\frac{\delta^2}{2m}$ plays the role of ϵ and $(\frac{\beta}{m\epsilon})^{\frac{1}{2}}p$ represents the noise η. Because the Langevin method is a special case of the hybrid approach, it will not in general represent the optimum choice of hybrid parameters.

In these hybrid approaches, one should adjust together both the step size ϵ and either the parameter α or the refreshing frequency in such a manner as to hold the finite step errors in observables at an acceptable size. The optimal algorithm minimizes the number of steps required to decorrelate lattices at a given error.

6. A Langevin-pseudofermion hybrid

In this section I discuss a fermion algorithm presented by Gavai and myself. [16] The approach has similarities with the Langevin evolution but is based on a small step-size limit of the Metropolis *et al.* [7] scheme. The interest in this approach is that a simple modification permits an interpolation to the pseudofermionic algorithm, thus clarifying the connection between them.

The goal is to generate an ensemble of configurations of fields A and ϕ distributed as in Eq. (14). To explicitly insure the positivity of the fermionic matrix M, assume that it is a square

$$M = DD^\dagger \tag{48}$$

This effectively doubles the number of fermionic species, one interacting with A via $D(A)$ and the other via $D^\dagger(A)$. I will later mention a possible way to remove this doubling. With this form for M, the desired probability distribution for A and ϕ is

$$P(A,\phi) \propto \exp(-S_G - \phi^*(D^\dagger)^{-1}D^{-1}\phi). \tag{49}$$

The algorithm consists of alternate sweeps through the ϕ and A fields. The ϕ updating is particularly simple, and represents a variation on eqs. (30) and (31). First generate a random vector χ with Gaussian weight

$$P(\chi) \propto e^{-\chi^\dagger\chi} \tag{50}$$

I now change variables and construct

$$\phi = D\chi. \tag{51}$$

This will be distributed with the desired probability

$$P_\phi \propto \exp(-\phi^*(D^\dagger)^{-1}D^{-1}\phi). \tag{52}$$

The Jacobian factor associated with the change of variables in Eq. (51) is irrelevant as the fields A are being held fixed during this step.

This construction is computationally fast because the individual components of χ are independent and because the matrix D is assumed to be local. Thus I can rapidly obtain a new ϕ field independent of its old value. This trick for updating ϕ is also used in the implementation of the Langevin algorithm in Ref. 9. Actually, the remainder of the algorithm does not explicitly need ϕ. Although I could eliminate this field and consider only χ, the discussion is simpler in terms of the coupled probability in Eq. (49).

In addition to the field χ, the updating of the gauge fields will require another quantity

$$\xi = (D^\dagger)^{-1}\chi = M^{-1}\phi. \tag{53}$$

This, unfortunately, is not so trivial to obtain, requiring a conjugate gradient inversion. Such a step is in common with the Langevin and microcanonical approaches.

I now come to the updating of the gauge field. What would be most desirable would be something like a Metropolis *et al.* [7] procedure where the acceptance of trial changes is governed by changes in the action

$$S(A,\phi) = S_G + \phi^*(D^\dagger)^{-1}D^{-1}\phi. \tag{54}$$

However, this is impractical because every time A is changed, D changes and its inverse on ϕ would have to be recalculated. To avoid this slow procedure, consider making only small changes in A. The changes in the action are then related to the first derivative with respect to A

$$\begin{aligned}\frac{\partial S}{\partial A}|_\phi &= \frac{\partial S_G}{\partial A} - 2\ \mathrm{Re}(\phi^*(D^\dagger)^{-1}D^{-1}\frac{\partial D}{\partial A}D^{-1}\phi)\\ &= \frac{\partial S_G}{\partial A} - \xi^*\frac{\partial D}{\partial A}\chi - \chi^*\frac{\partial D^\dagger}{\partial A}\xi.\end{aligned} \tag{55}$$

where ξ is defined in Eq. (53).

Now consider the quantity

$$S_T(A,\chi,\xi) = S_G - \xi^* D\chi - \chi^* D^\dagger\xi. \tag{56}$$

Eq. (55) implies

$$\frac{\partial S}{\partial A}|_\phi = \frac{\partial S_T}{\partial A}|_{\chi,\xi}. \tag{57}$$

If I consider small changes in A, first order changes of the action S at constant ϕ equal the changes in S_T calculated at constant χ and ξ. As only changes in the action enter into the Metropolis *et al.* [7] algorithm, updating the A fields using S_T is equivalent in lowest order to using the exact action S. This is the proposal of Ref. 16, and is easily implemented because S_T is local.

As with the pseudofermion, Langevin, and microcanonical methods, this algorithm makes a small-step-size approximation. To have confidence that the errors induced by a finite step are small, one should study a desired measurable for a few values of this step size and extrapolate to the infinitesimal limit. Ref. 9 argued that with for the Langevin algorithm a finite step represents a simulation with an effective action which differs from the initial one by terms vanishing with the step size. If this new action has the same continuum limit, then these finite step simulations should give the same numerical results for physical observables. Nevertheless, an extrapolation to vanishing step is still necessary to compare results of different algorithms with a given set of parameters at a finite lattice spacing.

The solid points in figure 1, taken from Ref. 13, show the average plaquette $P = \langle \frac{1}{3}\mathrm{Re\ Tr}\ U_p \rangle$ measured with this algorithm for the $SU(3)$ theory at $\beta = 4.5$. This is plotted versus the acceptance per hit, a simple measure of the step size. This simulation was done using the action from Ref. 14 with eight flavors and a fermion mass of 0.1 in lattice units. The zero step limit, which follows from extrapolation to unit acceptance, represents the correct plaquette value with the inclusion of the dynamical fermions. The crosses in this figure were obtained in a standard pseudofermionic run.

The Metropolis *et al.* [7] algorithm in the limit of small step size is quite close to the Langevin approach. Both cases involve small random changes in the field variables. A standard Metropolis *et al.* [7] program first tries unbiased changes about the old field, and then, to maintain the desired peaking of the distribution towards lower action, rejects a fraction of those changes which go towards larger action. In contrast, the Langevin approach always accepts the changes, but makes them in a direction biased towards lower action. This bias is determined by the same first derivative of the action with respect to A used above to construct S_T. The similarity of the approaches suggests that the finite step errors should be comparable. To directly make such a comparison, one should use a common definition of step size. One such measure would be to use the number of iterations needed to decorrelate lattices. I conjecture that the behavior of the solid points in figure (1) will mimic that of a Langevin simulation when plotted versus the decorrelation time.

We can now show the close connections of this algorithm with the pseudofermion method. To see this note that the field ξ has a probability distribution precisely the same as the pseudofermionic one in Eq. (29). Indeed, the present algorithm is equivalent to using but a single pseudofermionic field for the expectation value used in Eq. (28) to estimate M^{-1}. As mentioned earlier, the systematic errors

Fig. 1: The average plaquette as a function of the acceptance probability per Metropolis *et al.* [7] hit. The parameter N_ϕ is discussed in the text. Note the interpolation between the simple algorithm of this section as shown by the solid points and the pseudofermionic simulation shown by the crosses.

from using a finite number of pseudofermionic configurations average out after the extrapolation to zero step size.

Clearly the present algorithm represents an extreme case. One could interpolate between this and the pseudofermionic algorithm by averaging over some fixed number N_ϕ of ξ fields. This may also be thought of as considering N_ϕ species of fermions, each with its own ξ field, but then letting each species contribute only $1/N_\phi$ in the updating of the A field. As N_ϕ increases, I approach the pseudofermion algorithm. The remaining points in figure 1 exhibit this interpolation.

The allowing of each species to contribute only fractionally to the updating of the A field may provide a scheme to reduce the effective number of fermion species overall. Naively, this can remove the extra doubling introduced in Eq. (48) as well as any inherent doubling in the basic formulation of the fermions. Such a possibility has been frequently mentioned in the context all the approximate algorithms. There may, however, be some danger in this procedure because chiral symmetry breaking and anomalies suggest nonanalytic behavior as the number of fermionic species varies. This is an important issue which warrants further analytic study.

7. Exact global algorithms

These small-step fermionic algorithms, including pseudofermions, Langevin, microcanonical, and that of the previous section, all involve an extrapolation in a step size parameter. This is unfortunate in that lattice gauge calculations already involve tenuous extrapolations to zero lattice spacing and infinite volume, and this just gives us another thing to worry about.

The difficulty with the exact approaches discussed earlier is that a time consuming inversion must be done to test every trial change in the gauge field. The approximate schemes all work to reduce the frequency of such inversions to one per sweep. One could imagine making trial changes of all lattice variables simultaneously, and then accepting or rejecting the entire new configuration using the exact action. The problem with this approach is that a global random change in the gauge fields will generally increase the action by an amount proportional to the lattice volume, and thus the final acceptance rate will fall exponentially with the volume. The acceptance rate could in principle be increased by decreasing the step size of the trial changes, but then the step size would have to decrease with the volume. Exploration of a reasonable region of phase space would thus require a number of steps growing as the lattice volume. The net result is again an exact algorithm which requires computer time growing as volume squared.

So far this discussion has assumed that the trial changes are made in a random manner. If, however, one can properly bias these variations, it might be possible to reduce the volume squared behavior. An algorithm of this type was proposed in [17] For example, one could do either a Langevin or Metropolis *et al.* [7] sweep using the action S_T of the last section. By keeping track of all the probabilities for accepting changes along the way, one could in principle calculate the inverse probability for taking the new lattice back to the original in a similar sweep. Then one can construct a generalized acceptance for the entire lattice which will exactly restore detailed balance. If the changes in S_T are a good approximation to the changes in the true action, the factors in the acceptance criterion should tend to cancel, giving a reasonably large final acceptance rate. Attempts to use this approach in [18] were moderately successful, although those authors felt that standard hybrid techniques were superior. A variation on this idea where one does a global accept/reject step on the entire lattice after a microcanonical trajectory was presented in Ref. [19]. These algorithms have an interesting theoretical volume dependence that I now discuss. [20-22]

In some sense, the difficulties with fermions stem from the time consuming evaluation of $M^{-1}\phi$ appearing in the action of Eq. (13). For simplicity, let me not write ϕ explicitly and assume that I have some action which is particularly difficult to calculate; so, I want to evaluate it as rarely as possible. To further simplify the notation, I write the following equations in term of a single variable A.

To begin, consider a possible trial change of this variable A to

$$A' = A + p\delta + F(A)\delta^2. \tag{58}$$

Here δ is an adjustable step size parameter introduced for bookkeeping purposes. The "momentum" variable p represents a random noise, which for convenience I take to be Gaussianly distributed

$$P(p) \propto e^{-p^2/2}. \tag{59}$$

The function $F(A)$ represents a driving force or bias in the trial selection procedure and is for the moment arbitrary. It will soon correspond to the force term in the hybrid approach.

The Metropolis *et al.* [7] scheme accepts trial changes with a conditional probability chosen to maintain detailed balance when applied to an equilibrium ensemble. With an unbiased trial change this acceptance is determined entirely by the exponentiated change in the action. Here, however, the force term in the selection procedure must be corrected for in the acceptance condition. I can fully restore detailed balance by accepting the new value A' with probability

$$P_{\text{acc}} = \min[1, \exp(H(p, A) - H(p', A'))]. \tag{60}$$

Here H is a classical "Hamiltonian" analagous to that in Eq. (42)

$$H(p, A) = p^2/2 + S(A). \tag{61}$$

In Eq. 60 I introduce p' as the reverse noise, *i.e.* the noise which would be required for the selection of A as the trial had A' been the initial value

$$p' = -p - (F(A) + F(A'))\delta. \tag{62}$$

After the accept/reject step, the momenta should be refreshed; thus, they should be modified in a Monte Carlo or other fashion that the preserves the distribution in Eq. 59. Note that H is precisely the Hamiltonian used in the microcanonical algorithm [10] to describe evolution in "simulation time." Because of this analogy, I refer to H as the classical energy.

Note that with the second order terms in δ, the mapping defined by Eq. 58 and Eq. 62 exactly preserves areas in phase space

$$dA\ dp = dA'\ dp'. \tag{63}$$

Were this not so, the acceptance criterion would also need to depend on a ratio of measures.

Because of this preservation of areas, it is easy to show that the average change of energy is positive. In particular, I have

$$\begin{aligned} Z &= \int dAdp\ \exp(-H(A,p)) \\ &= \int dA'dp'\ \exp(H(A',p') - H(A,p))\ exp(H(A,p)) \\ &= Z\langle \exp(H(A',p') - H(A,p))\rangle \end{aligned} \tag{64}$$

where the expectation is with the weighting $e^{-H(p,a)}$. Using Jensen's inequality $\langle e^f\rangle \geq e^{\langle f\rangle}$, we find

$$\langle H(A',p') - H(A,p)\rangle \geq 0. \tag{65}$$

Thus regardless of the biasing force, on average the trial change will tend to increase the energy. A useful relation comes from expanding Eq. (64) in powers of the energy change. This gives immediately

$$\langle H(A',p') - H(A,p)\rangle = \frac{1}{2}\langle (H(A',p') - H(A,p))^2\rangle + \mathcal{O}((H'-H)^3) \tag{66}$$

To proceed I expand the energy change in the parameter δ. It is readily verified that

$$H' - H = (p\delta + \frac{1}{2}(p^2\frac{\partial}{\partial A} + 2F(A))\delta^2)(\frac{\partial S(A)}{\partial A} + 2F(A)) + \mathcal{O}(\delta^3). \tag{67}$$

This implies that the choice

$$F_L(A) = -\frac{1}{2}\frac{\partial S}{\partial A} \tag{68}$$

leads to an energy change

$$H' - H = \mathcal{O}(\delta^3) \tag{69}$$

Indeed, making this choice and ignoring the possibility of rejecting the trial change gives the usual Langevin algorithm [9,10], where the parameter δ is the square root of the step size used for discretization. In particular, compare Eq. (58) with (46), where ϵ corresponds to δ^2.

Let me first consider not making the Langevin choice for the driving force. In this case I use Eq. (66) to obtain

$$\langle H' - H\rangle_{A,p} = \frac{\delta^2}{2}\langle (\frac{\partial S(A)}{\partial A} + 2F(A))^2\rangle + \mathcal{O}(\delta^4). \tag{70}$$

Note that terms with odd powers of δ in the energy change expansion all involve odd powers of p and thus vanish on averaging. If I now consider updating some large number V of variables together, the positive $\mathcal{O}(\delta^2)$ quantities will coherently add and I expect to find a total energy change increasing linearly with V. By the central limit theorem, the fluctuations about this growth will become gaussian. Thus for large volumes I expect to find

$$H' - H \simeq C\delta^2 V + B\rho\delta V^{1/2} \tag{71}$$

where C and B are constants and ρ is a gaussian random variable which I normalize such that its probability distribution is

$$P(\rho) \sim e^{-\rho^2/2}. \tag{72}$$

If Eq. 71 were exact, then Eq. (64) would relate C and B

$$C = B^2/2. \tag{73}$$

With this explicit form for the energy change, I can obtain the expected acceptance in the large V limit

$$\langle P_{\mathrm{acc}} \rangle = \langle \min[1, e^{H-H'}] \rangle = \frac{2}{\sqrt{\pi C V \delta^2}} e^{-CV\delta^2/4} \times (1 + \mathcal{O}(\frac{1}{CV\delta^2})). \tag{74}$$

The calculations required to derive Eq. (73) and Eq. (74), however, depend strongly on the tails of the distribution of the energy change and thus cannot be regarded as completely rigorous. For a further discussion on this point see [23]. For the following I will only assume that the expected acceptance is exponentially suppressed when $V\delta^2$ is large.

To avoid this exponential suppression and have a reasonable acceptance requires $\delta \sim V^{-1/2}$. However a small value for the step size raises the issue that the lattice will evolve only slowly from its original configuration. More precisely, consider taking N sweeps over the lattice. As the motion of A has both random and driven terms, the overall change in any given variable should go as

$$\Delta A = \mathcal{O}(\delta\sqrt{N}) + \mathcal{O}(\delta^2 N) = \mathcal{O}(\sqrt{N/V}) + \mathcal{O}(N/V). \tag{75}$$

The final result is that the number of sweeps required to obtain a substantially new configuration should grow as V. If V is proportional to the system volume, then the overall algorithm requires time growing as volume squared, one factor of volume from the number of sweeps, and the other from the fact that each sweep takes time proportional to the volume.

For a bosonic simulations this growth would be a disaster. The standard algorithms only grow as the system volume, and thus should be preferred over updating many variables simultaneously. On the other hand, for fermions this is the same overall growth observed above for the exact Monte Carlo inside a Monte Carlo approaches. That this is the same, however, indicates that there is no obvious additional penalty in going to global updates. Indeed, it might be possible to gain something by a judicious choice of F which will reduce the coefficient of this growth. This is the basis of the algorithms discussed below.

I now return to the Langevin choice of Eq. (68) for the driving force. Consider again updating only a single variable. At first glance one might think that since the exact action is so difficult to calculate, the requisite derivative for this force would be yet more intractable. Note, however, that this derivative was also needed to linearize the action for the approximate algorithms.

To proceed I slightly generalize this force and take

$$F(A) = -\frac{1}{2}\frac{\partial S}{\partial A} + g(A)\delta^2. \tag{76}$$

The $g\delta^2$ piece is included for the purpose of discussing possible higher order improvements. Using this, I calculate the next term in the expansion for the energy change

$$H' - H = -\frac{\delta^3}{12}(S_3\ p^3 - 3S_1\ S_2\ p - 24g\ p) + \mathcal{O}(\delta^4). \qquad 77$$

Here I use the notation

$$S_n = \frac{\partial^n S}{\partial A^n}. \qquad 78$$

Note that if S_3 is non-vanishing, *i.e.* if the theory is not harmonic, then no choice of $g(A)$ can make the $\mathcal{O}(\delta^3)$ term in this equation vanish for all p.

I now consider applying this procedure to a group of V variables simultaneously, as in the earlier discussion of unbiased changes. I am interested in any coherent addition of changes which could give an exponential suppression of the final acceptance. Because the $\mathcal{O}(\delta^3)$ term in Eq. 77 contains only odd powers of p, it will vanish on the average. The expectation for the energy change can again be most easily found using Eq. (66). With a little algebra and explicitly doing the average over p, I find

$$\langle H' - H\rangle = \frac{\delta^6}{96}\langle 2S_3^2 + 3(8g - S_3 + S_1\ S_2)^2\rangle + \mathcal{O}(\delta^8). \tag{79}$$

This can be written in many forms; this expression as the sum of two squares emphasizes positivity.

I now return to updating a large number V of independent variables simultaneously. The positive contributions indicated in Eq. (79) will add coherently. Similar arguments to those leading to Eq. (74) now give an expected acceptance falling as

$$P_{\text{acc}} \sim e^{-CV\delta^6} \tag{80}$$

To have a reasonable acceptance requires only $\delta \sim V^{-1/6}$. This changes Eq. (75) to

$$\Delta A = \mathcal{O}(\delta\sqrt{N}) + \mathcal{O}(\delta^2 N) = \mathcal{O}(\sqrt{N/V^{1/3}}) + \mathcal{O}(N/V^{1/3}). \tag{81}$$

Thus, the number of sweeps for an independent lattice grows as $V^{1/3}$ and the overall computer time for decorrelation increases as

$$T \sim V^{4/3}, \tag{82}$$

This behavior is only slightly worse than the linear growth of the pure bosonic theory.

This algorithm was proposed in Ref. 17 and tested further with somewhat discouraging results in Ref. 18. Ref. [19] presents a quite promising variation, generally referred to as the "hybrid Monte Carlo" algorithm, which I now discuss. Recapitulating on the above treatment of biased updatings, I constructed both the trial new A and the noise needed to return

$$A' = A + p\delta + F(A)\delta^2 \tag{83a}$$

$$p' = -p - (F(A) + F(A'))\delta. \tag{83b}$$

This is an area preserving map of the (A, p) plane onto itself. The scheme proposed in Ref. 19 is to iterate the combination of this mapping with an inversion $p' \to -p'$ several times before making the accept/reject decision. This iterated map remains reversible and area preserving. The second order terms in this equation make it equivalent to the leap frog procedure with an initial half step as used in Ref. 19. The procedure thus generates a microcanonical trajectory.

The important point is that after each step the momentum remains exactly the negative of that which would be required to reverse the entire trajectory and return to the initial variables. If at some point on the trajectory I were to reverse all the momenta, the system would exactly reverse itself and return throughtr the same set of states from whence it came. Thus a final acceptance with the probability of Eq. (60) still makes the overall procedure exact. This slight modification of the hybrid algorithm of Ref. 15 makes it exact, just as the procedure with a single step removes the systematic errors of Langevin evolution. After each accept/reject step, the momenta p are refreshed, their values being replaced by new Gaussian random numbers. The fields ϕ could also be refreshed at this time, or less often, as turns out to be appropriate. The goal of the procedure is to use the microcanonical evolution as a way to restrict changes in the action so that the final acceptance will remain high for reasonable step sizes.

This procedure contains several parameters which can be adjusted for optimization. First is N_{mic}, the number of microcanonical iterations taken before the global accept/reject step and refreshing of the momenta p. Then there is the step size δ, which presumably should be set to give a reasonable acceptance. Finally, one can also vary the frequency with which the auxiliary scalar fields ϕ are updated.

The arguments for following a microcanonical trajectory for some distance before refreshing the momenta have been stressed in Ref. [15]. Refs. [21] and [22] show that this approach gives an algorithm where the computer time grows as $V^{5/4}$. I now review that argument.

The goal of the approach is to speed flow through phase space by replacing a random walk of the A field with a coherent motion in the dynamaical direction determined by the conjugate momenta. As long as the total microcanonical time for a trajectory is smaller than some characteristic time for the system, the net change in A will grow linearly with both N_{mic} and δ; thus Eq. 3.24 is replaced by

$$\Delta A \sim N_{mic}\delta. \tag{84}$$

which should be valid as long as

$$N_{mic}\delta < \mathcal{O}(1). \tag{85}$$

With large N_{mic}, the change in the classical energy will also grow. In any given microcanonical step the energy changes by an amount of order δ^3. For N_{mic} of order δ^{-1}, the total energy change will then be of order δ^2. Because the evolution preserves areas in phase space, Eq. (66) still applies to the overall evolution and I have for the expected energy change

$$\langle H' - H \rangle = \frac{1}{2}\langle (H' - H)^2 \rangle + \mathcal{O}((H' - H)^3) = \mathcal{O}(\delta^4). \tag{86}$$

Now if I update V independent variables together, these positive contributions can coherently add and earlier arguments give an overall acceptance falling as

$$P_{\rm acc} \sim \exp(-CV\delta^4). \tag{87}$$

This means that δ should be taken to decrease with volume as $V^{-1/4}$. Correspondingly, N_{mic} should grow as $V^{1/4}$, the maximum allowed by Eq. (85). The final result is that the total time required to obtain a substantially changed lattice grows as

$$T \sim V^{5/4} \tag{88}$$

This may be only an asymptotic statement, valid for systems much larger than the correlation length. The main uncertainty lies in the unknown characteristic time scales that determine the $\mathcal{O}(1)$ right hand side of Eq. (85).

A variation on the hybrid Monte Carlo scheme in Ref. 13 essentially corresponds to keeping the parameter α of the earlier second order Lanbgevin discussion, and doing an acceptance after each step. Presumably this gives a similar time dependence to the above.

It has been argued [24] that the hybrid Monte Carlo algorithm will perform better if one does not always use a constant trajectory length. This will certainly be the case if the trajectory is some multiple of a fundamental frequency of the dynamical system being considered. Having variable trajectory lengths, in either a random of systematic manner, is a straightforward and worth while addition to the approach.

8. Higher order schemes

While the above theoretical volume dependence is quite promising, it is interesting to ask whether one can do better. Campostrini and Rossi [25] presented such a higher order scheme, while Ref. [26], generalized the scheme to reduce the errors to an arbitrarily small power of delta. This gives a higher order hybrid Monte Carlo scheme whose volume growth is arbitrarily close to that for bosonic simulations. On the other hand, the coefficient of this growth seems to increase with order, and in practice the above scheme appears to work adequately. I now review this higher order approach.

The change of variables in Eq. (83) is an area preserving map on (A, p) that conserves energy to order δ^3. It is often written in a "leapfrog" manner as a combination of the transformation

$$T_A(\delta) : (A, p) \longrightarrow (A + p\delta,\ p) \tag{89}$$

and

$$T_p(\delta) : (A, p) \longrightarrow (a,\ p - S'(A)\delta) \tag{90}$$

Eq. (83) is equivalent to the application of $T_p(\delta/2)T_A(\delta)T_p(\delta/2)$. I refer to this combintation as $T_2(\delta)$ because it preserves the energy through order δ^2).

I now wish to generalize this transformation giving a $T_n(\delta)$ which preserves the energy through order δ^n while maintaining the property of preserving areas in phase space. To apply the Metropolis *et al.* [7] algorithm one also needs the property of reversability

$$T^{-1}(\delta) = T(-\delta) \tag{91}$$

Then one can use this transformation to replace T_2 in the generation trial changes for the hybrid Monte Carlo algorithm discussed above, and presumably improve on the asymptotic volume dependence..

A simple construction of a suitable T_n is recursive; I consider combinations of lower order T_n which cancel out the errors at order δ^n. Since I have above an explicit T_2, I can recursively generate any higher order transformation required.

To proceed, consider the Hamiltonian as the generator of translations in time. In particular, for any function $F(t)$ which depends on the phase-space variables at time t, I write

$$e^{H\delta} : F(t) \longrightarrow F(t + \delta) \tag{92}$$

where $F(T + \delta)$ is the value of F after evolving along the exact classical trajectory for time δ. Suppose I now have an area preserving transformation T_n which gives the same evolution accuate through order δ^n. Thus I assume

$$e^{H\delta} = T_n(\delta) + \Delta\delta^{n+1} + (\text{higher order terms}) \tag{93}$$

Now consider two such transformations of different distances in time

$$T_n(\delta_2)T_n(\delta_1) = e^{(\delta_1+\delta_2)H} + \Delta e^{H\delta_1}\delta_2^{n+1} + e^{H\delta_2}\Delta\delta_1^{n+1} + \ldots \tag{94}$$

To order $n+1$ in the step sizes, this reduces to

$$T_n(\delta_2)T_n(\delta_1) = e^{(\delta_1+\delta_2)H} + (\delta_2^{n+1} + \delta_1^{n+1})\Delta + \ldots \tag{95}$$

Note that with the reversibility property of Eq. (91), $T(-\delta)T(\delta) = 1$ is exact. This implies that the operator Δ in the above equation must vanish when n is odd. Indeed, this is a simple way to understand why the simple leapfrog algorithm has errors that do not start until the third order in the step size. Indeed, only the elimination of odd powers of δ is a problem in going to higher orders. Thus I consider n even in the following.

For the inductive step, I now pick an arbitrary integer i. Consider taking i steps of size δ forward with our transformation T_n, and then one step backward with a different size $s\delta$, and finally i more forward steps of size δ. This combination results in a net motion of distance $(2i-s)\delta$, with the errors adding non-linearly

$$T_n(\delta)^i\ T_n(-s\delta)\ T_n(\delta)^i = e^{H(2i-s)\delta} + (2i - s^{n+1})\delta^{n+1}\Delta + \mathcal{O}(\delta^{n+3}).$$

If I pick

$$s = (2i)^{1/(n+1)},$$

the order δ^{n+1} terms cancel and I can write

$$T_{n+2}((2i-s)\delta) = T_n(\delta)^i\ T_n(-s\delta)\ T_n(\delta)^i \tag{96}$$

This algorithm is not unique, in particular there is the parameter i giving the number of forward steps before a backward one. In [26] it was argued that one should adjust i to minimize the largest step in a microcanonical trajectory of fixed length. Tests on simple models, however, showed that the simple leapfrog algorithm was quite adequate in most practical cases. One should bear in mind, however, in going to large systems that higher order schemes may eventually become useful.

9. Concluding remarks

So with all these algorithms, which is best? It appears that the local algorithms are all too time consuming for practical use. As the simple Langevin approach is a special case of the hybrid algorithm, one should certainly include the more general possibilities included in the latter. Adding the accept reject step to make the algorithm exact has the additional advantage that one need not worry about systematic biases beyond the lattice cutoff. These arguments are responsible for the current popularity of the hybrid Monte Carlo approach.

This method does, however, require the fermion matrix to be a square, requiring at least two species for Wilson fermions and eight for the Kogut Susskind case. Users of the hybrid algorithm without the global accept-reject step have argued for adjusting the number of fermion species by inserting a factor proportional to the number of flavors in front of the pseudofermionic term when the gauge fields are updated. Such a possibility was mentioned above in the discussion of the interpolation between Langevin and pseudofermionic methods. This modification is simple to make, but is not completely theoretically understood. Indeed, it may introduce spurious behavior if the physics is not smooth in the number of flavors. Nevertheless, this flexability of the hybrid algorithm has made it quite popular, although when the global accept condition can be applied, it is probably worthwhile for the extra confidence it supplies.

Despite the successes of these fermion algorithms, the overall procedure remains somewhat awkward, particularly when compared with the ease of setting up a pure bosonic simulation. This appears to be due to the non-local actions resulting from integrating out the fermions. Indeed, had one integrated out a set of bosons coupled quadratically to the gauge field, one would again have a non-local effective action, indicating that this analytic integration was not a good idea. Perhaps we should step back and explore algorithms before integrating out the fermions. One such attempt is the world line Monte Carlo method of [27]. Here one explicitly follows the paths of the fermions through the lattice and the Monte Carlo procedure involves random changes in these paths. The method has been shown to work well in two space-time dimensions, although sign problems have severely limited work in higher dimensionality. A recent attempt to treat these signs exactly using a recursive enumeration of paths appears to work well, although memory issues restrict the approach to extremely small systems. [28]

As Monte Carlo methods are particularly difficult with fermions, perhaps one should look for other numerical methods not based on stochastic processes. Here I include such possibilities as direct diagonalization of the Hamiltonian. So far these methods have been very constrained on possible system size, suggesting that one should look for new approximations to discard irrelevant information as the systems grow in volume.

An extremely difficult unsolved question is the simulation of fermionic systems when the corresponding determinant is not always positive. This situation is of considerable interest because it arises in the study of quark-gluon thermodynamics when a chemical potential is present. This issue is extensively discussed and referenced in Gavai's chapter of this book. All known approaches to this problem are extremely demanding on computer resources. One can move the phase of the determinant into the observables, but then one must divide out the average value of this sign. [29] This is a number which is expected to go to zero exponentially with the lattice volume;, thus, such an algorithm will require computer time growing exponentially with the system size. Another approach is to do an expansion about zero baryon density, but again to get to large chemical potential will require rapidly growing resources. New techniques are badly needed to avoid this growth; hopefully this will be a particularly fertile area for future algorithm development.

10. References

1. F.A. Berezin, The method of second quantization (Academic Press, NY, 1966).
2. P.T. Matthews and A. Salam, Nuovo Cimento 12 (1954) 563.
3. D. Weingarten and D. Petcher, Phys. Lett. 99B (1981) 333.
4. D. Weingarten and D. Petcher, Phys. Lett. 99B (1981) 333; H. Hamber and G. Parisi, Phys. Rev. Lett. 47 (1981) 1792.
5. A.N. Burkitt and A.C. Irving, Comp. Phys. Comm. 59 (1990) 447.
6. G. Bhanot, U.M. Heller, and I.O. Stamatescu, Phys. Lett. 129B (1983) 440.
7. N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller, J. Chem. Phys. 21 (1953) 1087.
8. M. Grady, Phys. Rev. D32 (1985) 1496.
9. G.G. Batrouni, G.R. Katz, A.S. Kronfeld, G.P. Lepage, B. Svetitsky and K.G. Wilson, Phys. Rev. D32 (1985) 2736.
10. A. Ukawa and M. Fukugita, Phys. Rev. Lett. 55 (1985) 1854.
11. J. Polonyi and H.W. Wyld, Phys. Rev. Lett. 51 (1983) 2257; J. Kogut, J. Polonyi, H.W. Wyld, and D.K. Sinclair, Phys. Rev. Lett. 54 (1983) 1475.
12. F. Fucito, E. Marinari, G. Parisi, and C. Rebbi, Nucl. Phys. B180[FS2] (1981) 369.
13. A.M. Horowitz, Physics Letters 156B (1985) 89; Nucl. Phys. B280[FS18] (1987) 510; preprint Cern-Th-6172/91 (1991).
14. D. Callaway and A. Rahman, Phys. Rev. D28 (1983) 1506.
15. S. Duane and J. Kogut, Phys. Rev. Lett. 55 (1985) 2774; Nucl. Phys. B275 (1986) 398.

16. M. Creutz and R. Gavai, Nucl. Phys. B280 (1987) 181.

17. R. T. Scalettar, D.J. Scalapino and R.L. Sugar, Phys. Rev. B34 (1986) 7911.

18. S. Gottlieb, W. Liu, D. Toussaint and R.L. Sugar, Phys. Rev. D35 (1986) 2611.

19. S. Duane, A. D. Kennedy, B.J. Pendleton and D. Roweth, Phys. Lett., 195 (1987) 2.

20. H. Gausterer and S. Sanielevici, Phys. Rev.D38 (1988) 1220.

21. R. Gupta, G. Kilcup, and S. Sharpe, Phys. Rev.D38 (1988) 1278 (1988).

22. M. Creutz, Phys. Rev. D38 (1988) 1228.

23. A.D. Kennedy and B. Pendleton, Nucl. Phys. B (Proc. Suppl.) 20 (1991) 118.

24. P.B. Mackenzie, Phys. Lett. B226 (1989) 369; S. Gupta, preprint CERN-TH.6178/91.

25. M. Campostrini and P. Rossi, Nucl. Phys. B329 (1990) 753.

26. M. Creutz and A. Gocksch, Phys. Rev. Lett. 63 (1989) 9.

27. J.E. Hirsch, R.L. Sugar, and D.J. Scalapino, Phys. Rev. B26 (1982) 5033.

28. M. Creutz, preprint BNL-46782 (1991).

29. A. Gocksch, Phys. Rev. Letters 61 (1988) 2054.