The Kentucky noisy Monte Carlo algorithm for Wilson dynamical fermions

B. Joó
School of Physics, University of Edinburgh, Edinburgh, EH9 3JZ, Scotland, United Kingdom

I. Horváth and K. F. Liu
Department of Physics and Astronomy, University of Kentucky, Lexington, Kentucky 40506

(Received 31 December 2001; revised manuscript received 31 December 2002; published 29 April 2003)

We develop an implementation for a recently proposed noisy Monte Carlo approach to the simulation of lattice QCD with dynamical fermions by incorporating the full fermion determinant directly. Our algorithm uses a quenched gauge field update with a shifted gauge coupling to minimize fluctuations in the trace of the logarithm of the Wilson-Dirac matrix. The details of tuning the gauge coupling shift as well as results for the distribution of noisy estimators in our implementation are given. We present data for some basic observables from the noisy method, as well as the acceptance rate information and discuss potential autocorrelation and sign violation effects. Both the results and the efficiency of the algorithm are compared against those of the hybrid Monte Carlo algorithm.

DOI: 10.1103/PhysRevD.67.074505 PACS number(s): 12.38.Gc, 02.70.Uu, 11.15.Ha

I. INTRODUCTION

Monte Carlo (MC) calculations in lattice QCD with dynamical fermions are notoriously time consuming. These simulations generally proceed through a numerical realization of an ergodic Markov process having the desired lattice QCD probability distribution as its fixed point. In direct approaches, the major stumbling block is the evaluation of the fermion determinant which is typically needed somewhere in the process. For interesting volumes V, the fermion matrix is extremely high dimensional and the time to compute the determinant exactly scales as V^3. Hence computing the fermion determinant exactly is not a feasible option.

The current standard workhorse for dynamical lattice QCD computations is the hybrid Monte Carlo (HMC) algorithm [1]. In this case the problem of evaluating the fermion determinant is sidestepped by expressing the determinant as an integral over bosonic (pseudofermion) fields which become full-fledged dynamical fields in the Markov process. One criticism of the HMC method is its supposed inability to deal with an odd number of fermion flavors. Indeed, the natural settings for HMC are even-flavor theories where the pseudofermion heatbath is straightforward and the bosonized action is manifestly positive. However, this limitation is not fundamental and can be addressed within the framework of molecular dynamics algorithms [2,3]. This is a topic of current research.

Even though the direct simulation of the fermion determinant is infamous for being nearly impossible to implement, it promises distinct advantages over the pseudofermion method employed in HMC. In addition to being able to accommodate any number of flavors, it has the potential of being a viable finite density algorithm in the canonical ensemble approach. The usual finite chemical potential algorithm in the grand canonical ensemble has the well-known sign problem and the imaginary chemical potential approach has the overlap problem [4,5]. Considering the canonical ensemble instead, one can project out a definite baryon number from the fermion determinant before the acceptance test, to stay in a given baryon number sector so that the overlap problem can be avoided [5]. In this case, it is essential to have an algorithm which accommodates the determinant directly.

An interesting proposal for simulating the determinant directly has been put forward recently in Ref. [6]. In that approach the idea was to split the determinant into infrared and ultraviolet parts and to treat the infrared part exactly and the ultraviolet part approximately. This can in principle be turned into an exact algorithm [7], but it is not yet clear how well the systematic error of the splitting of the determinant was under control, particularly for small quark masses and large lattices.

The approach that will be followed here has several roots. One important ingredient is an efficient evaluation of the determinant based on Padé-Z_2 stochastic estimators of the trace of logarithm of the fermion matrix [8]. For example, using the unbiased subtraction, one can reduce the error on the trace of the logarithm of the Wilson fermion matrix on an $8^3 \times 12$ lattice at $\beta = 5.6$ by a factor of 25–40 relative to an unsubtracted one with negligible overhead. At $\kappa = 0.154$ with 400 Z_2 noise vectors, the absolute error on the trace of the logarithm M is about 0.29, which translates into the same relative error for the determinant.

Nevertheless, this would still not be good enough if one intended to develop a Metropolis-like [9] algorithm, because the acceptance probability has to be evaluated exactly. To address this problem, Kennedy and Kuti (KK) proposed an algorithm in which the nonlinear Metropolis acceptance step was replaced with a linear one [10]. This opened up the possibility of using unbiased noisy estimators for the required probability ratios instead of having to evaluate them exactly. Indeed, the required unbiased estimators can be developed based on the idea of stochastic series summation [11]. However, the quantity used as the KK linear acceptance probability can in principle be negative or greater than 1 [12] when the noisy estimate comes from the outlying tails of the underlying distribution. This introduces a bias in the results but the authors of Refs. [10,11] argue that in practice it is possible to tune both the expression for the linear acceptance probability and the estimators so that the bias is substantially smaller than the statistical errors.

The above discussion motivates the second root of our approach which amounts to choosing stochastic variables so that they provide unbiased estimators for the determinant itself (rather than acceptance probability), which eliminates the need for the linear acceptance step, and allows these variables to be treated as full-fledged fields in the Markov process. This has been accomplished in Ref. [13] and resulted in a procedure without probability bound violation problems. We will refer to algorithms based on this approach as Kentucky noisy Monte Carlo (KNMC) algorithms. In Ref. [13] this idea was applied to a simple five state model where the amount of noise in the estimators could be precisely tuned. Although the statistical errors in the results of the KNMC method grew with increasing levels of noise, the result did remain unbiased while the bias in the KK procedure was substantially greater than the KNMC errors.

Applying this approach to QCD requires not only a satisfactory way of estimating the determinant, but also an efficient way of proposing new configurations in the Markov process. Indeed, one can easily construct a useless algorithm when proposed configurations are almost always rejected. It is well known that changes of the gauge field constructed from sweeps guided solely by a pure gauge action can lead to a widely fluctuating determinant and an essentially vanishing acceptance probability for small quark masses. To address this issue we adopt the idea of splitting the short-distance part of the determinant by the loop action and incorporating it into the pure gauge action [7,14,15]. This is the third ingredient of our approach. As a matter of fact, one of our points is that while we have only split the determinant with the simplest plaquette action, we nevertheless obtain a working algorithm at least for relatively heavy quark masses. We view the inclusion of optimized higher loop actions for the split as being the most promising way of improving our algorithm further.

In what follows, we present the results from applying the KNMC algorithm with the above specifics to two flavors of Wilson dynamical fermions. Even though the number of flavors is a mere parameter in our approach, we use the two-flavor setting to be able to compare to HMC easily. The remainder of this paper is organized as follows. We begin by outlining the main ideas on which our algorithm is built in Sec. II. We then discuss the concrete application of the algorithm to Wilson fermions in Sec. III where we discuss some of the numerical techniques used in our implementation as well as some work estimates. After presenting some computational details in Sec. IV we discuss our assorted numerical results in Secs. V, VI and VII. We summarize and discuss these results in Secs. VIII and IX and present our conclusions in Sec. X.

II. THE ALGORITHM

We start by describing the basic ideas on which our algorithm is built. Our goal is to simulate a distribution given by the gauge invariant function of lattice link variables of the form

$$P^{\mathrm{QCD}}(U) \propto e^{-S_g(U)} \prod_{f=1}^{N_f} \det M_f(U)$$

$$= e^{-S_g(U) + \sum_{f=1}^{N_f} \mathrm{Tr} \ln M_f(U)} \tag{1}$$

where $S_g(U)$ is the gauge action and $M_f(U)$ is the fermion matrix $[\det M_f(U) > 0]$ [16] for a given flavor of dynamical quark. The indices f run over the number of flavors one wishes to simulate. For clarity of the discussion and notation below, we shall describe our algorithm using just a single flavor of fermion and drop the subscript f for now, with the understanding that the generalization to many flavors is straightforward.

It will be assumed that there is a suitable approximation $R_M(U)$ of $\ln M(U)$ that is easy to evaluate, and whose accuracy can be controlled so that the corresponding distribution

$$P(U) \propto e^{-S_g(U) + \mathrm{Tr}\, R_M(U)} \tag{2}$$

is arbitrarily close to $P^{\mathrm{QCD}}(U)$.

We will construct an exact algorithm for $P(U)$ of Eq. (2) based on the following considerations:

(1) As pointed out in the Introduction, the exact computation of $\mathrm{Tr} R_M(U)$ is not feasible. For this reason one would like to use noisy estimators of this quantity. Let us consider

$$x \equiv E[\mathrm{Tr} R_M(U), \eta] = \eta^{\dagger} R_M(U) \eta \tag{3}$$

where η is a vector in the linear space of $M(U)$ whose elements are random numbers drawn from a distribution $P^{\eta}(\eta)$ satisfying the property that

$$\langle \eta_i^{\dagger} \eta_j \rangle_{P^{\eta}(\eta)} = \delta_{ij}. \tag{4}$$

In Eq. (4) the subscripts on the angle brackets imply that the expectation value is to be taken in the measure defined by $P^{\eta}(\eta)$. In Eq. (3) the notation $E[\mathrm{Tr} R_M(U), \eta]$ is also introduced, which may be used throughout this paper to indicate that a given quantity is an unbiased estimator for the first argument in the square brackets depending on the subsequent arguments. In this case, for example, x is an estimator of $\mathrm{Tr} R_M(U)$, depending on the noise vector η.

From Eqs. (3) and (4) it is straightforward to show that x is indeed an estimator for $\mathrm{Tr} R_M(U)$. However for simulating the measure defined by Eq. (2) estimates of the quantity e^x are needed. If a sequence of estimates $x_i = \eta_i^{\dagger} R_M(U) \eta_i$, with $i = 1,2 \ldots$, for $x = \mathrm{Tr} R_M(U)$ is available to us, where the subscripts on η now refer to the position of η_i in the sequence rather than its elements, we can construct an estimator for $e^{\mathrm{Tr}\, R_M(U)}$ by evaluating the function [11]

$$f(U,\{\eta_i\},\{\rho_k\},c)=1+\Bigg\{x_1+\theta\left(\frac{c}{2}-\rho_2\right)\Bigg\{\frac{x_2}{c}+\theta\left(\frac{1}{3}c-\rho_3\right)\times\Bigg\{\frac{x_3}{c}+\ldots+\theta\left(\frac{c}{n}-\rho_n\right)\times\Bigg\{\frac{x_n}{c}+\ldots\Bigg\}\Bigg\}\Bigg\}\Bigg\}\Bigg\}, \tag{5}$$

where $c>0$ is a tunable constant, $\theta(x)$ is the Heavyside step function and the ρ_k is the random number uniformly distributed in the range $0\leqslant\rho_k\leqslant 1$ [in other words, ρ_k has distribution $P^{\rho}(\rho_k)=\theta(\rho_k)-\theta(\rho_k-1)$.] One can easily verify that

$$\langle f(U,\{\eta_i\},\{\rho_k\},c)\rangle\prod_{i=1}^{\infty}P^{\eta}(\eta_i)\prod_{k=2}^{\infty}P^{\rho}(\rho_k)=e^{\mathrm{Tr}\,R_M(U)}. \tag{6}$$

(2) Motivated by the discussion above, and by the form of Eq. (2), we extend the variable space and write the corresponding partition function in the form

$$Z=\int dU e^{-S_g(U)}\int\prod_{i=1}^{\infty}d\eta_i\;\; P^{\eta}(\eta_i)\times\prod_{k=2}^{\infty}d\rho_k\;\; P^{\rho}(\rho_k)\;\; f(U,\eta,\rho), \tag{7}$$

where we have introduced the shorthand $f(U,\eta,\rho)$ for $f(U,\{\eta_i\},\{\rho_k\},c)$. We have thus introduced an infinite number of auxiliary variables. How can one deal with them in a practical simulation? The point is that given the nature of terms in Eq. (5) only a finite number of them will be used in any particular evaluation of $f(U,\eta,\rho)$, since the series terminates stochastically. The average number of terms can be tuned by appropriate choice of the constant c, and if the typical values of x_k can be kept reasonably small during the simulation then a practical scheme with effectively finite number of noise fields present can be developed.

(3) The basic problem with partition function (7) is that $f(U,\eta,\rho)$ is not positive definite, causing the well-known difficulties to standard simulation techniques. We will assume (and demonstrate later) that things can be arranged so that the occurrence of negative $f(U,\eta,\rho)$ in typical equilibrium configurations (U,η,ρ) is very small. In that case one can cure this problem by absorbing the sign into the observables in the usual way, i.e.

$$\langle\mathcal{O}\rangle_P=\frac{\langle\;\mathcal{O}\,\mathrm{sgn}(P)\;\rangle_{|P|}}{\langle\;\mathrm{sgn}(P)\;\rangle_{|P|}}. \tag{8}$$

Our goal then is to find a suitable Markov process for generating the probability distribution

$$P(U,\eta,\rho)\propto e^{-S_g(U)}\;\;|f(U,\eta,\rho)|\prod_{i=1}^{\infty}P^{\eta}(\eta_i)\prod_{k=2}^{\infty}P^{\rho}(\rho_k). \tag{9}$$

(4) One may attempt to simulate the distribution (9) in several possible ways. To explain the approach adopted here, let us introduce the collective notation $\xi\equiv(\eta,\rho)$, $f(U,\xi)\equiv f(U,\eta,\rho)$, $P(U,\xi)\equiv P(U,\eta,\rho)$. We can then write schematically $P(U,\xi)\propto P_1(U)P_2(U,\xi)P_3(\xi)$ with

$$P_1(U)\propto e^{-S_g(U)}$$
$$P_2(U,\xi)\propto|f(U,\xi)| \tag{10}$$
$$P_3(\xi)\propto\prod_{i=1}^{\infty}P^{\eta}(\eta_i)\;\prod_{k=2}^{\infty}P^{\rho}(\rho_k).$$

We will use two steps based on the following two statements that can be verified directly:

(a) Let $T_1(U,U')$ be the ergodic Markov matrix satisfying detailed balance with respect to P_1; in other words, $P_1(U)T_1(U,U')dU=P_1(U')T_1(U',U)dU'$. Then the transition matrix

$$T_{12}(U,U')=T_1(U,U')\;\min\left[\;1,\frac{P_2(U',\xi)}{P_2(U,\xi)}\;\right] \tag{11}$$

satisfies detailed balance with respect to the $P_1(U)P_2(U,\xi)$ (with ξ fixed).

(b) The transition matrix

$$T_{23}(\xi,\xi')=P_3(\xi')\min\left[\;1,\frac{P_2(U,\xi')}{P_2(U,\xi)}\;\right] \tag{12}$$

satisfies detailed balance with respect to $P_2(U,\xi)P_3(\xi)$ (with U fixed).

From (a), (b) it follows that T_{12} and T_{23} keep the original distribution $P(U,\xi)$ invariant and interleaving them will lead to an ergodic Markov process with the desired fixed point.

We note that there is a lot of freedom in choosing the pure gauge process $T_1(U,U')$. If local updates are used, then it is necessary to ensure that a given sequence of such updates satisfies detailed balance with respect to $P_1(U)$. This can be achieved, for example, by updating the sites at random or selecting the order of updated variables appropriately. We adopt the procedure wherein only links corresponding to chosen even/odd part of the lattice and chosen direction are updated. One can easily check that such a ‘‘subsweep’’ satisfies detailed balance for the Wilson pure gauge action if the elementary local updates also do so. Further, we note that in step (b) use is made of the fact that the probability distribution $P_3(\xi)$ for the noise can be generated directly from a heatbath.

Finally it should be emphasized that in Eqs. (11) and (12) one needs to compute a ratio of the form

$$\frac{P_2(U',\xi)}{P_2(U,\xi)}=\frac{|f(U',\xi)|}{|f(U,\xi)|} \tag{13}$$

where $f(U,\xi)$ in Eq. (5) is an estimator for e^x. Since the quantity x is an estimator for the quantity $\mathrm{Tr}R_M(U)$, it can be very large, as $R_M(U)$ is an extensive quantity. Looking at Eq. (5) it can be seen that $f(U,\eta,\rho)$ can indeed give a very poor estimate of the exponential, if the x_k are large, and only a few terms are taken.

Ideally one would like to be in a situation where $-1 < x_k < O(1)$. Certainly when $x_k < -1$, one faces the problem that $f(U,\eta,\rho)$ can become negative depending on the number of terms taken. If this happens only occasionally the effects can be taken into account by folding the sign of $f(U,\eta,\rho)$ into the observable as in Eq. (8). However, if it happens often, it can cause a large effective reduction in statistics.

While no firm upper limit has been placed on x_k we do note that the exponential function diverges rapidly for increasing $x>0$. Given an infinite amount of statistics, the stochastic exponentiation technique will still give an unbiased estimator for e^x. However, when $x>1$ the terms in Eq. (5) have increasing absolute value, thus causing the variance of the estimators to become very large. Furthermore, in a Markov process such as the one described above, the evolution can potentially get stuck in a region of configuration space with a given number of terms (noise fields ρ) being used to estimate $f(U,\eta,\rho)$. This is because although having a large number of terms is unlikely, once reached with $x_k > 1$, then $f(U,\eta,\rho)$ will have a higher numerical value than it would with fewer terms (corresponding to a potential new noise field configuration ρ') in which case the new field is likely to be rejected. For this reason it is prudent in a simulation to arrange matters so that x_k is of $O(1)$.

The above discussion suggests that, while the approach described above theoretically leads to simulating the distribution (2), additional steps need to be taken to turn it into a practical scheme. We now discuss some ways that can be employed to deal with the issue of typical magnitudes and variances of x_n below.

A. Shifting the action by a constant

Motivated by the fact that a ratio of exponentials can be written as

$$\frac{e^{x'}}{e^{x}} = \frac{e^{(x'-x_0)}}{e^{(x-x_0)}}, \tag{14}$$

one notices that the fermionic action can be shifted by a constant through making the replacement:

$$x(U,\eta) = \eta^\dagger R_M(U)\,\eta \rightarrow x(U,\eta,x_0) = \eta^\dagger R_M(U)\,\eta - x_0\,. \tag{15}$$

Such a shift can move the mean of the distribution of the values of x to an arbitrary real number without affecting the simulation in any way. With this in mind, our main goal is to minimize the variance of x.

B. Splitting with the loop action

It is well known that a significant portion of $\mathrm{Tr}\ln M(U)$ can be typically taken into account by a short-distance loop action $\Delta S_g(U)$ [14,15], especially at larger quark masses, and this is expected to remain true for $\mathrm{Tr}R_M(U)$ also. This fact can be used to reduce the magnitude of the fluctuations in x by splitting this part of $\mathrm{Tr}R_M(U)$ into the gauge action when setting up the Markov process (see, e.g. [7]). To recall the argument let us write

$$\begin{aligned} P(U) &\propto e^{-S_g(U)}\; e^{\mathrm{Tr}\,R_M(U)} \\ &= e^{-S_g(U)+\Delta S_g(U)}\; e^{\mathrm{Tr}\,R_M(U)-\Delta S_g(U)}. \end{aligned} \tag{16}$$

We can thus replace

$$\begin{aligned} S_g(U) &\rightarrow S_g(U) - \Delta S_g(U) \\ \mathrm{Tr}\,R_M(U) &\rightarrow \mathrm{Tr}\,R_M(U) - \Delta S_g(U) \end{aligned} \tag{17}$$

in our Monte Carlo procedure. Then the gauge updates are performed with the new local action, and evaluation of $f(U,\eta,\rho)$ involves the variables x_n estimating $\mathrm{Tr}R_M(U) - \Delta S_g(U)$. The specifics of how to do this will be discussed in Sec. III C.

C. Explicit splitting

Utilizing the fact that $e^x = (e^{x/N})^N$, one can also split $\mathrm{Tr}R_M(U)$ directly by writing $\mathrm{Tr}R_M(U) = \Sigma_{i=1}^N (1/N)\mathrm{Tr}\,R_M(U)$, and use separate noise fields for every $(1/N)\mathrm{Tr}R_M(U)$. Since N divides $\mathrm{Tr}R_M(U)$ into N pieces, each carrying $1/N$ flavor, we shall refer to it as the number of fractional flavors. Indeed, the corresponding modification of Markov process is straightforward. To see this, consider for simplicity the case $N=2$. Originally, the simulated probability distribution was written schematically as $P(U,\xi) \propto P_1(U)P_2(U,\xi)P_3(\xi)$, while now we have

$$P(U,\xi_1,\xi_2) \propto P_1(U)P_2^s(U,\xi_1)P_2^s(U,\xi_2)P_3(\xi_1)P_3(\xi_2),$$

where the P_2^s is P_2 of Eq. (10) with x from Eq. (5) replaced by x/N.

In the step (a) of the MC procedure we thus have $P_2 \rightarrow P_2^s(U,\xi_1)P_2^s(U,\xi_2)$ with ξ_1,ξ_2 fixed. There is an arbitrariness in selecting the process (b). For example, if one chooses to update a single set of noise at a time, step (b) does not change at all, and one can choose, for example, the sequence (a), $(\mathrm{b})_{\xi_1}$, (a), $(\mathrm{b})_{\xi_2}$ as an elementary Markov step. The only requirement here is the overall ergodicity.

The main effect of explicit splitting is to scale the width of the distribution of x by the number of fractional flavors N. The optimal mixture of action shifting, splitting by the loop action and explicit splitting is a matter to be explored.

D. Reducing the variance from noise

While splitting the loop action as described above reduces the fluctuations in x arising from the fluctuation of the gauge configurations, the variance of x also receives a contribution from the noise fields η since $\eta^\dagger R_M(U)\,\eta$ is used in the construction of x. Further variance reduction techniques can be applied to reduce this contribution. The particular technique depends on the kind of noise used. In the specific case when Z_2 noise is used, it has been shown [17] that all the contributions to the variance of $\mathrm{Tr}R_M(U)$ come from off-diagonal elements of $R_M(U)$ in which case the unbiased subtraction noise reduction technique of [8] is highly effective. We will present details of this method in Sec. III B.

III. APPLICATION TO LATTICE QCD WITH DYNAMICAL WILSON FERMIONS

To demonstrate that the ideas described in the previous section can lead to a working algorithm, we now describe the details of the implementation of the algorithm that we used to perform simulations with two flavors of degenerate Wilson quarks. Although in principle both the algorithm and the implementation can handle an arbitrary number of flavors, the case of two degenerate flavors is convenient from the point of view that its results can be checked against HMC simulations. Further, we can also carry out some tuning using these reference simulations as we shall detail in Secs. III C and VI A.

We simulate the theory with the standard Wilson gauge action

$$S_g(U) = -\frac{\beta}{3}\,\mathrm{Re}\,\mathrm{Tr}\,U_{\Box} \tag{18}$$

where β is the gauge coupling parameter. The quantity $U_{\Box}$ is obtained as usual by evaluating the product of link matrices around each elementary plaquette and summing the results over the whole lattice. After integrating out the Grassmann numbers, the effective fermion action is

$$S_f(U) = -\sum_{f=1}^{N_f} \mathrm{Tr}\ln M(U,\kappa_f) \tag{19}$$

where the sum is over all desired flavors, $M(U,\kappa_f)$ is the Wilson fermion matrix

$$M(U,\kappa_f) = 1 - \kappa_f D(U), \tag{20}$$

$D(U)$ is the usual Wilson hopping matrix and κ_f is the hopping parameter for the flavor with index *f*. In our simulations we used an approximation $R_M(U)$ to $\ln M(U)$ given by a Padé approximation, which we will discuss in more detail in Sec. III B.

A. Local gauge update

In order to update the gauge fields, we use the quasi heatbath method [18] amended as described previously. We split the lattice into even and odd sites such that even sites have only odd neighbors and vice versa. This is a common technique known as checkerboarding. We randomly pick one-half or our lattice sites corresponding to either even or odd sites, and within that sublattice we choose—also randomly—one of the 4 space-time dimensions. This identifies a particular subset of lattice links with the chosen checkerboard value and which connect sites in the chosen direction. We update all the links in this subset simultaneously. Each such subsweep allows us to update $\frac{1}{8}$ of our lattice. As outlined earlier, one is free to perform any number of such updates before updating the noise fields. In fact, this remains a free parameter (N_s) in our code. However, for the results presented here we have always used $N_s = 1$.

B. Estimating Tr$\boldsymbol{R_M(U)}$

In order to estimate Tr$R_M(U,\kappa)$ we turn to the technology described in [8]. The logarithm is approximated using a Padé approximation, which after a partial fraction expansion, has the form:

$$\ln M(U,\kappa) \approx R_M(U) \equiv b_0 I - \sum_{i=1}^{N_P} b_i [M(U,\kappa) + c_i I]^{-1} \tag{21}$$

where N_P is the order of the Padé approximation, and the constants b_i and c_i are the Padé coefficients. In our implementation we have used an 11th order approximation whose coefficients are tabulated in [8].

The traces are then estimated by evaluating bilinears of the form $\eta^\dagger R_M(U)\eta$. If the components of η are chosen from the Z_2 group, then the contributions to the variance of these bilinears come only from off-diagonal elements of $R_M(U)$ as discussed previously. In this case [19] an effective method reducing the variance is to subtract off a linear combination of traceless operators from $R_M(U)$ and to consider

$$E[\mathrm{Tr}\,R_M(U),\eta] = \eta^\dagger [R_M(U) - \omega_i \mathcal{O}_i]\eta. \tag{22}$$

Here the $\mathcal{O}_i$ are operators with $\mathrm{Tr}\,\mathcal{O}_i = 0$. Clearly since the $\mathcal{O}_i$ are traceless they do not bias our estimators in any way. The ω_i are constants that can be tuned *a priori* to minimize the fluctuations in $E[\mathrm{Tr}R_M(U),\eta]$.

In practice the $\mathcal{O}_i$ are constructed by taking traceless terms from the hopping parameter expansion for $M^{-1}(U)$. These reduce the noise coming from the terms $[M(U) + c_i]^{-1}$ in Eq. (21). The terms D, D^2, D^3 and further odd powers of D are explicitly traceless and terms which have even powers such as D^4 have known traces given in terms of various loops. For example

$$\mathrm{Tr}\,D^4(U) = -64\,\mathrm{Tr}\,U_{\Box} \tag{23}$$

and hence $\mathcal{O}_4 = D^4(U) + 64\,\mathrm{Tr}\,U_{\Box}$ is traceless. Details of finding the traces of even powers of D can be found, for example, in [20]. In our computations we have subtracted observables involving D, D^2, D^3, D^4, D^5 and D^7.

Although the parameters ω_i are tunable in principle, the hopping parameter expansion for $M^{-1}(U)$ is sufficiently good for heavier quark masses, so that for such masses, the ω_i are numerically close to unity. Hence in our simulations we have always used $\omega_i = 1$ for all *i*.

Since in our implementation we need the sum of $R_M(U)$ for all flavors, we can estimate the whole sum using a single noise field η. This allows us to compute all the $[M(U,\kappa_f) + c_i I]^{-1}\eta$ for all c_i and all flavors κ_f, for a given η using a single multiple shift inversion [21,22]. In practice we employ the M^3R [22,23] algorithm as it is the most memory efficient, and memory was a bottleneck on our target computers.

C. Loop splitting specifics

We now turn to the details of splitting the loop action. The fermionic action for a single flavor can be written as

$$S_f = -[\mathrm{Tr} R_M(U,\kappa_f) - \lambda^f \ \mathrm{Re}\,\mathrm{Tr} U_\Box] - \lambda^f \mathrm{Re}\,\mathrm{Tr} U_\Box, \tag{24}$$

where the λ^f is a tunable parameter for that particular flavor. One can then shift the fermion action for each flavor as follows:

$$S_f(U) \to -[\mathrm{Tr} R_M(U,\kappa_f) - \lambda^f \ \mathrm{Re}\,\mathrm{Tr} U_\Box]. \tag{25}$$

At this point it becomes convenient to introduce the shorthand $T(U,\lambda^f)$ for the quantity

$$T(U,\lambda^f) \equiv \mathrm{Tr}\, R_M(U,\kappa_f) - \lambda^f \ \mathrm{Re}\,\mathrm{Tr} U_\Box \tag{26}$$

and to write

$$S_f(U) \to -T(U,\lambda^f). \tag{27}$$

In order to absorb this change, the gauge action needs to be correspondingly shifted as

$$S_g(U) \to -\frac{\beta}{3}\mathrm{Re}\,\mathrm{Tr} U_\Box - \lambda^f \ \mathrm{Re}\,\mathrm{Tr} U_\Box$$

$$= -\frac{(\beta + 3\lambda^f)}{3}\mathrm{Re}\,\mathrm{Tr} U_\Box \tag{28}$$

with an extra shifted term for each flavor of fermion. The end result is that the gauge action becomes

$$S_g(U) = -\frac{\beta'}{3}\mathrm{Re}\,\mathrm{Tr} U_\Box \quad \text{with} \quad \beta' = \beta + 3\sum_{f=1}^{N_f} \lambda^f. \tag{29}$$

The λ^f need to be tuned to minimize the variance of $T(U,\lambda^f)$. The tuning procedure is given by the action matching technology of Sexton, Irving and Weingarten [15,24]. In fact, finding $\lambda^f_{\min}$, the values of λ^f for which the fluctuations of $T(U,\lambda^f)$ are minimized, corresponds exactly to tuning a quenched simulation to a dynamical fermion one in an action matching sense. The quantity $\lambda^f_{\min}$ is given (see [24]) by the formula

$$\lambda^f_{\min} = -\frac{\mathrm{Cov}[\mathrm{Tr} R_M(U,\kappa_f), \mathrm{Re}\,\mathrm{Tr} U_\Box]}{\sigma^2(\mathrm{Re}\,\mathrm{Tr} U_\Box)} \tag{30}$$

where $\sigma^2(\mathrm{Re}\,\mathrm{Tr} U_\Box)$ is the variance of the plaquette and the quantity in the numerator is the standard covariance between the $\mathrm{Tr} R_M(U,\kappa_f)$ and the plaquette. We note the action matching technology of [24] is not limited to simply tuning the Wilson plaquette action, but is fairly generic. In particular, it can be used to tune the splitting of the determinant by actions which are linear combinations of higher order loops.

When a preliminary reference simulation is available at the desired parameters, one can measure the required covariances and correlations on this data set. Otherwise, since the tuning of [24] can be carried out in any measure, one can perform a quenched simulation, and employ a self-consistent procedure to find $\lambda^f_{\min}$.

TABLE I. Summary of implementation parameters.

Parameter	Description
β	Gauge coupling
κ_f	Fermion hopping parameter (1 per flavor)
N_η	Number of noise vectors per estimator of $E[R_M(U)]$ (we use $N_\eta = 1$)
ω_i	Parameters for reducing the noise in $E[\mathrm{Tr}\, R_M(U),\eta]$ (we use $\omega_i = 1$ for all i)
r	Target fractional residual in the multiple mass inverter (we use $r = 10^{-6}$ for the lightest shifted mass)
$\lambda_f^{\min}$	Loop action splitting parameters (1 per flavor). The shifted gauge coupling is $\beta' = \beta + 3\Sigma_f \lambda^f_{\min}$
N	Number of fractional flavors (explicit splitting terms)
x_0^f	Action shifting constants (1 per flavor)
x_1	Action shift fine tuning factor (we use $x_1 = 2$)
c	Variance control parameter for Eq. (5) (we use $c = 1.5$)
N_s	The number of (checkerboard, direction) subsweeps in the gauge update algorithm (we used $N_s = 1$)

Once $\lambda^f_{\min}$ are determined, one can immediately compute $\langle T(U,\lambda^f_{\min})\rangle$ which are good first estimates for the action shift parameters x_0^f, which will ensure the quantities $x^f = E[T(U,\lambda^f_{\min})] - x_0^f$ have means of 0. These may not be the optimal shift factors x_0^f, since it may be desirable to have $\langle x^f \rangle > 0$, to minimize the number of negative sign violations.

One can then shift the x^f even further so that practically all the values of x^f are greater than 0. This can be achieved by defining

$$x_0^f = \langle T(U,\lambda^f_{\min})\rangle - \frac{1}{N_f}x_1, \tag{31}$$

where x_1 is some factor of $\sigma(E[T(U,\lambda^f_{\min})])$.

The final value x that we use in Eq. (5) is then

$$x = \frac{1}{N}\sum_f \ E[T(U,\lambda^f)] - x_0^f \tag{32}$$

with x_0^f as defined in Eq. (31).

For later reference, the values and definitions of all the parameters in our implementation are summarized in Table I.

D. Work estimates

The cost C of the present implementation of KNMC for each accepted update of the gauge field and noise fields is

$$C \sim \frac{N_\eta \;\; N N_{\mathrm{exp}} C_{\mathrm{M}} + C_{\mathrm{G}}}{P^U_{\mathrm{acc}}} + \frac{N_\eta \;\; N_{\mathrm{exp}} C_{\mathrm{M}}}{P^\xi_{\mathrm{acc}}}. \tag{33}$$

In Eq. (33), the first term represents the computational cost of updating the gauge field, and the second corresponds to the contribution from updating a single noise field (out of the N). Here N_{exp} is the average number of terms in the stochastic expansion of the exponential function in Eq. (5) which is e for the case $c=1$. C_{M} is the cost of estimating $\mathrm{Tr} R_M(U)$ for all flavors but for only one noise field, C_{G} is the cost of updating the gauge configuration U. The quantities P^U_{acc} and P^ξ_{acc} are the acceptance rates for the gauge and noise updates, respectively. The cost C_G is negligible in comparison with C_{M} which is dominated by the time to perform the multiple mass solution of the system $[M(\kappa)+c_i]X=\eta$ for all κ and c_i.

E. Volume scaling

The cost for creating a single estimator for x is dominated by the cost of the multiple mass solve. This should scale linearly with the volume. The quantity x itself is expected to scale with the square root of the volume, since evaluating the bilinear involves a sum of random numbers over the volume which can be positive or negative with equal likelihood. Hence one would expect the variance $\sigma^2(x)$ of x to scale linearly with V and so $\sigma(x)$ should scale as $\sqrt{V}$. In this case the number of fractional flavors needed to keep $\sigma(x)$ to be $O(1)$ must also increase as $\sqrt{V}$. Hence the total cost of the algorithm must scale at least as $O(V^{3/2})$.

F. Comparison to HMC

Let us compare our work estimate to that of a typical HMC accepted configuration. The work involved in generating this configuration grows as

$$C_{\mathrm{HMC}} \sim \frac{N_{\mathrm{MD}} C_{\mathrm{F}} + 2 C_{\mathrm{H}}}{P^U_{\mathrm{acc}}} \tag{34}$$

where N_{MD} is the number of time steps one takes while integrating the Hamiltonian equations of motion for one trajectory. The predominant contribution to the cost of carrying out such a time step is the cost C_{F} of the computation of the molecular dynamics force for the time step, which for fermionic systems involves solving the system of equations: $(M^\dagger M)x=\phi$, where ϕ are the pseudo-fermion fields. The cost C_{H} is the cost of calculating the energy which also requires the solution of a system similar to that of the force computation. The energy calculations are done at the start and end of the trajectory. While in principle one can carry out the inversions for the energy using a different stopping criterion from the one used for the force computation, it is convenient now to consider a case where this is not done and $C_{\mathrm{F}}=C_{\mathrm{H}}$.

Since the predominant cost for our KNMC algorithm (assuming that $P^U_{\mathrm{acc}} \leqslant P^\xi_{\mathrm{acc}}$) comes from the accept/reject step following the gauge field update; when the determinant has to be estimated for all the fractional flavors [cf. the first term of Eq. (33)] we will neglect the cost of updating a single noise field [where the determinant only has to be estimated for a single fractional flavor—cf. the second term of Eq. (33)]. Also, as C_{G}, the cost of performing the gauge update sweep, is negligible in the current implementation in comparison to C_{M}, the cost of performing a multiple mass inversion, the cost of the noisy algorithm is approximately

$$C_{\mathrm{KNMC}} \sim \frac{N_\eta \;\; N N_{\mathrm{exp}} C_{\mathrm{M}}}{P^U_{\mathrm{acc}}}. \tag{35}$$

Comparing Eqs. (35) and (34) and assuming that $C_{\mathrm{M}} \sim C_{\mathrm{F}}$ since they both involve a solution of a similar set of linear equations we note that the two algorithms are comparable when

$$\frac{N_\eta \;\; N N_{\mathrm{exp}}}{P^{\mathrm{KNMC}}_{\mathrm{acc}}} \sim \frac{N_{\mathrm{MD}}}{P^{\mathrm{HMC}}_{\mathrm{acc}}} \tag{36}$$

where $P^{\mathrm{KNMC}}_{\mathrm{acc}}$ refers to the gauge acceptance rate of the KNMC algorithm and $P^{\mathrm{HMC}}_{\mathrm{acc}}$ refers to the HMC acceptance rate. In a typical application, $N_{\mathrm{MD}} \sim O(100)$ and $P^{\mathrm{HMC}}_{\mathrm{acc}} \sim 0.8$. As we shall see later on, our simulations using the KNMC algorithm managed to achieve $P^{\mathrm{KNMC}}_{\mathrm{acc}} \sim 0.3$, with $N_{\mathrm{exp}} \sim 3$ and $N \approx 20$, which makes our current simulations somewhat more expensive than their HMC counterparts.

IV. COMPUTATIONAL DETAILS

We now briefly describe our numerical computations. In all we have performed three sets of numerical studies. Firstly, we have performed a brief study of the stochastic exponentiation technique which we describe in Sec. V. Thereafter we performed two sets of lattice QCD simulations:

The first set of simulations was carried out using a volume of $V=8^4$ lattice sites, and we used the results from these simulations to study the tuning of the algorithm. This set consisted of a reference HMC simulation (hereafter referred to as $H1$), and three KNMC simulations denoted $K1$, $K2$ and $K3$, respectively. We used the reference simulation $H1$ to tune the parameter λ^f_{min} for the KNMC simulations. Between simulations $K1$, $K2$ and $K3$ we varied the number of fractional flavors, and examined the effect of these on the distribution of the estimators x, which are used to make estimators of $\mathrm{Tr}\, R_M$. We also looked for consistency between observables from the KNMC simulations and the reference simulation $H1$. A detailed description for the results from these simulations is given in Sec. VI.

While observables from the first set of simulations seemed unbiased compared to their reference HMC counterparts, we found the statistics inadequate to make a strong statement about the autocorrelations in our ensemble. For this reason we carried out a second set of simulations, this time using a smaller volume of $V=4^4$ sites. This reduction in volume allowed us to generate sufficient statistics to be able to discuss the issues of autocorrelations meaningfully, and

TABLE II. Summary of simulations and their main parameters. All the simulations were carried out using $N_f = 2$ flavors of dynamical fermions, except for simulation Q which was performed in the quenched approximation.

Simulation	Type	V	β	κ	$\lambda^f_{\min}(\times 10^{-2})$	x_0	x_1	N
$H1$	HMC	8^4	5.5	0.1550	–	–	–	–
$K1$	KNMC	8^4	5.5	0.1550	3.27	−679.6	2	10
$K2$	KNMC	8^4	5.5	0.1550	3.27	−679.6	2	15
$K3$	KNMC	8^4	5.5	0.1550	3.27	−679.6	2	20
$H2$	HMC	4^4	5.5	0.1550	–	–	–	–
$K4$	KNMC	4^4	5.5	0.1550	3.25	−38.92	0.7	15
Q	HMC	4^4	5.695	–	–	–	–	–

indeed to demonstrate that the autocorrelation times are finite. The second set of simulations consists of a reference HMC simulation, a single KNMC simulation and a quenched simulation to which we shall refer to as $H2$, $K4$ and Q, respectively. The results for this simulation are discussed in Sec. VII. The labels, types and main parameters of our lattice simulations are collected and summarized in Table II.

Our implementation of the KNMC algorithm was coded for the QCDSP [25] supercomputer, and simulations $K1$, $K2$ and $K3$ were performed on 1, 2 and 4 motherboard QCDSP computers located at Columbia University and at the Brookhaven National Laboratory. Our code was written in C++ utilizing the Columbia Physics Software System (also known as CPS) which was made available to us by the RIKEN-BNL-Columbia (RBC) Collaboration. Our simulation $K4$ was run on a Linux workstation. Porting the code between the two platforms was assisted by the efforts of the Edinburgh Parallel Computing Center (EPCC).

Our reference hybrid Monte Carlo (HMC) simulation $H1$ was carried out at the T3E facility at NERSC, using the GHMC code [26] made available to us by the UKQCD Collaboration. Simulations $H2$ and Q were carried out on a Linux workstation using the publicly available SZIN software system, currently maintained at the Thomas Jefferson National Laboratory (JLAB) under the SciDAC program [27].

Our analysis program, as well as our investigation of the stochastic exponentiation, was carried out on workstations.

V. STOCHASTIC EXPONENTIATION STUDY

Before we describe our simulation results, we will make a detour and experiment with the technique of stochastic exponentiation. A question of interest is this: How good an estimator $E[e^x]$ of e^x can one obtain by applying Eq. (5) to estimators $E[x]$ of x? In this section we attempt to give a partial answer to this question in a situation where both x and its fluctuations, as characterized by its standard deviation $\sigma(x)$, are under explicit control.

In this study, noisy estimates $E[x]$ were made for several values of x by adding Gaussian noise of known variance $\sigma^2(x)$ to the actual values of x. Equation (5) was then applied to these values of $E[x]$ to make estimators $E[e^x]$ of e^x.

The results of this study are shown in Fig. 1 where the bias in the results of the stochastic exponentiation is plotted against the number of samples of $E[e^x]$. To be more precise, a number of samples of $E[e^x]$ were averaged to obtain a measurement of $\langle e^x \rangle$ and this was subtracted from the true value of e^x. It can be seen from Fig. 1 that the technique works quite well for $x=3$, $c=1$ and about 1000 samples. Increasing c to $c=1.5$ allows one to get unbiased estimates for $x=5$ for the same number of samples and it is even possible to get unbiased estimates for $x=6$ for such a sample size if $c=2.0$. However, we note that as x is increased the fluctuations increase enormously, too, as can be seen when one compares the scales on the vertical axes of Fig. 1. The data shown in Fig. 1 confirm our earlier reasoning about the distribution of x in our earlier discussions, namely that it is preferable for the value x in Eq. (5) to be small.

VI. RESULTS FROM THE LARGE VOLUME SIMULATION: $H1$, $K1$, $K2$ AND $K3$

A. Reference HMC simulation $H1$

In order to carry out the required tuning, and to have some benchmark results for our noisy simulations we have performed a reference HMC simulation with two flavors of Wil-

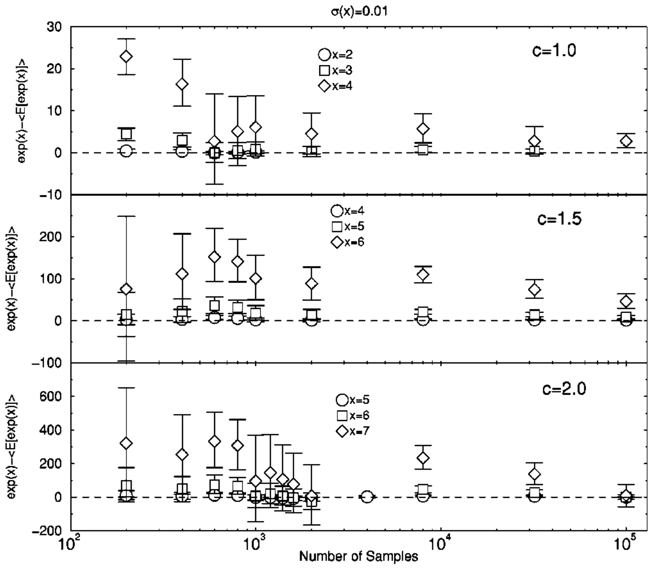

FIG. 1. Bias in the stochastic estimation as a function of statistics.

TABLE III. Results from simulation $H1$. The first set of errors are the naive bootstrap errors. The second set shows the effects of autocorrelation estimated by blocking the data.

Observable	Value
$\langle$Plaquette$\rangle$	0.5476(1)(4)
$\langle \mathrm{Tr} R_M(U) \rangle$	640.9(6)(20)

son dynamical fermions using the desired physical simulation parameters listed in Table II. We generated 1280 HMC trajectories, of which the first 625 were discarded for equilibration. Of the remaining 655 trajectories we stored every fifth one to measure $\mathrm{Tr}R_M(U)$ giving us a total of 132 configurations to work with. On these configurations we have estimated $\mathrm{Tr}R_M(U)$ using 100 noise vectors per configuration. When the noise fields per configuration were averaged, the measurement of $\langle \mathrm{Tr}R_M(U)\rangle_\eta$ was accurate to a relative error of less than 1% per configuration.

B. HMC observables

The values of $\mathrm{Tr}R_M(U)$ and the plaquette (normalized by the volume and the number of planes) measured in our HMC computations are shown in Table III. In the case of the plaquette, we used the values of the observable on all 655 trajectories. The statistical errors were first estimated using a simple bootstrap technique with 500 bootstrap samples. A blocking technique was then used to estimate the effects of autocorrelation on the observables. This technique consisted of averaging successive values of the observable in the time series into a single observable of a new data set (with less statistics than the original). The naive variance was then measured on the resulting new data set. This procedure was repeated until we ran out of statistics, or observed a plateau in the variance. Unfortunately, these data are rather noisy and hence estimating the plateaus is somewhat subjective. We believe we have been conservative in Table III.

C. Tuning λ^f_{min}

We now describe the results of performing the tuning for the λ^f_{min}. Since, both the HMC and the KNMC simulations were done using degenerate flavors of fermions, we will drop the flavor index f on this quantity from now on.

We used the estimators of $\mathrm{Tr}R_M(U)$ to estimate λ_{min} using the tuning formula of Eq. (30). Before outlining the results we note that there are two ways of computing the variances and covariances in Eq. (30), the choice of which has a bearing on the resulting standard deviation, $\sigma(T(U,\lambda_{\mathrm{min}}))$ of $T(U,\lambda_{\mathrm{min}})$.

(1) *Method 1.* In this method, all the estimators $E[\mathrm{Tr}R_M(U),\eta]$ are first averaged over all the noise fields η for a given configuration. This gives an estimate of $\langle \mathrm{Tr}R_M\rangle_\eta$ per configuration with some small error. These estimates can then be used (neglecting the small errors) to perform averages with respect to the gauge fields as usual when computing variances, covariances and correlations. The results for $\sigma(T(U,\lambda))$ as a function of λ for this method are plotted in Fig. 2.

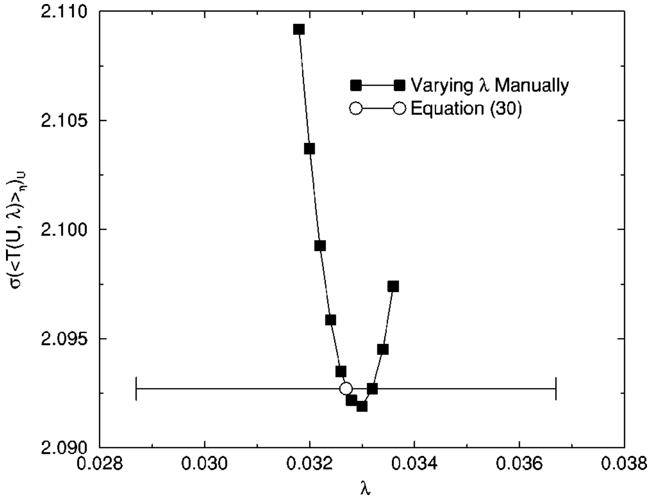

FIG. 2. Tuning for λ_{min} using Method 1. The circle gives the result from Eq. (30). The squares are the results of explicitly varying λ around this minimum.

(2) *Method 2.* In this method one does not first average over the η fields. Instead the averaging is performed over all the noise fields and gauge fields simultaneously when evaluating variances, covariances and correlations. The results for $\sigma(T(U,\lambda))$ are plotted in Fig. 3 for this method.

While method 1 is perhaps the preferred method from the point of view of action and observable matching, the numbers from it may be misleading from the point of view of a noisy algorithm since it neglects the effects of noise in the estimation of $\mathrm{Tr}R_M(U)$. However, one would expect the two methods to both give the same λ_{min} because they are both equivalent to carrying out the same path integral. In method 2, since more statistics are available, one may expect to get

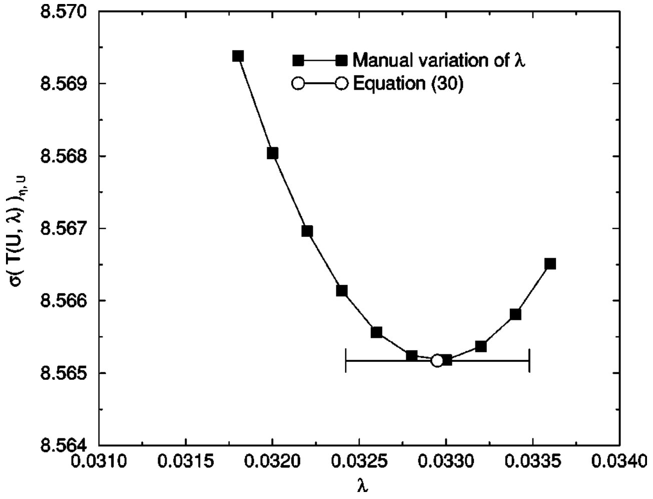

FIG. 3. Tuning for λ_{min} using Method 2. The circle gives the result from Eq. (30). The squares are the results of explicitly varying λ around this minimum.

TABLE IV. HMC tuning results.

Statistic	Method 1	Method 2
$\sigma(\mathrm{Tr}\, R_M(U))$	7.99	11.51
$\mathrm{Corr}(\mathrm{Tr}\, R_M(U), \mathrm{Tr}\, U_\Box)$	0.96(12)	0.66(1)
$\lambda_{\min}$ $(\times 10^{-2})$	3.27(38)	3.29(5)
$\sigma(T(U,\lambda_{\min}))$	2.09	8.56
$\langle T(U,\lambda_{\min})\rangle$	$-679.6(1)$	$-689.60(7)$

smaller errors on $\lambda_{\min}$. Finally, comparing the results of methods 1 and 2, one can get a rough idea of how much of the variance in our $\mathrm{Tr}\, R_M(U)$ comes from the noise fields η and how much comes from fluctuations from gauge configuration to gauge configuration.

We note in passing that methods 1 and 2 can be thought of as opposite extremes of carrying out KNMC simulations with various values of N_η. Method 1 corresponds to the situation where N_η is large, and many conventional estimators $E[\mathrm{Tr} R_M(U);\eta]$ are averaged, to get a better estimator, whereas method 2 corresponds to the situation where $N_\eta = 1$.

Looking at Table IV it can be seen that the two methods do in fact give similar results for $\lambda_{\min}$. Method 2 appears more accurate, presumably because of the larger number of estimators available. By examining Figs. 2 and 3 the increase in statistics is clearly visible from the size of the horizontal error bar on the tuned point. It can also be seen that the minima are quite shallow in terms of λ. The error bar on the point obtained with method 1 is quite large, despite the fact that the point itself lies near the minimum. With method 2 the error bar is smaller and the point is better placed. Our recommendation from these results would be to always check that the minimum is found, by performing some manual tuning around the value of $\lambda_{\min}$ given by Eq. (30).

We note that we carried out the measurements of method 2 after our KNMC simulations as an afterthought. Hence our simulations all used values determined by method 1.

D. KNMC simulations $K1$, $K2$ and $K3$

We now turn to the discussion of our KNMC simulations $K1$, $K2$ and $K3$. In all three simulations we have used the same physical parameters as in $H1$. We used the loop splitting factor $\lambda_{\min} = 3.27\times 10^{-2}$ for both flavors as obtained by Method 1 of the tuning (see Table IV). For our value of x_0^f we used $\langle T(U,\lambda_{\min})\rangle = -679.6$ (Table IV) with an additional fine tuning factor of $x_1 = 2$ as per Eq. (31).

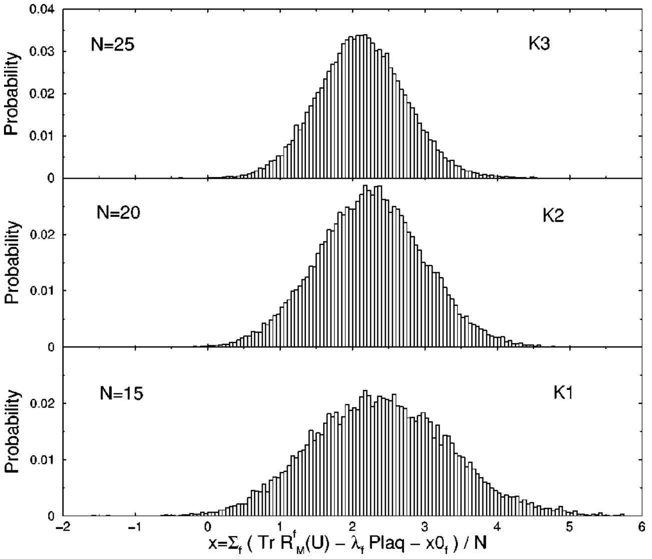

FIG. 4. Distributions of x for the three noisy simulations.

Using this value of $\lambda_{\min}$ resulted in a gauge coupling shift of $\Delta\beta = 3N_f \times \lambda^f_{\min} = 0.1962$ giving a value of $\beta' = 5.6962$ to use in the quenched gauge updating algorithm (instead of the $\beta = 5.5$ of the HMC computations).

The only difference between the three large volume KNMC simulations was the value of the number of fractional flavors N which took the values $N = 15$, $N = 20$ and $N = 25$ for simulations $K1$, $K2$ and $K3$, respectively. This choice was based on the values of $\sigma(T(U,\lambda_{\min}))$ measured in the preliminary HMC simulation using Method 1. These parameters are summarized in Table II. The value N was chosen so that $\sigma(x)$ with x defined in Eq. (32), which should be close to $\sigma(T(U,\lambda_{\min}))N_f/N$, is of the order unity.

We show some basic statistics for the simulations in Table V. In particular, we give the number of negative signs for $f(U,\eta,\rho)$ that we counted along each simulation and the width of the distribution of x as characterized by its standard deviation $\sigma(x)$ which are indeed close to $\sigma(T(U,\lambda_{\min}))N_f/N$.

E. Distribution of x

In Fig. 4 we plot the distributions of the quantity x as measured in the three simulations. The distributions appear to be Gaussian as one would expect from the Central Limit Theorem.

It can clearly be seen, that simulation $K1$ is quite near the limits prescribed upon the values of the quantity x by the stochastic exponentiation study, namely that the values of x

TABLE V. Summary of statistics for the noisy simulations.

Simulation	N	# Gauge updates	Negative signs of $f(U,\eta,\rho)$ (Number, %)	$\sigma(x)$
K1	15	2400	(70, 2.9)	0.944
K2	20	4229	(1, 0.023)	0.734
K3	25	4050	(0, 0)	0.6

TABLE VI. Acceptance rates for the noisy simulation.

Simulation	N	Gauge update acceptance (%)	Noise update acceptance (%)
$K1$	15	32(1)	49(1)
$K2$	20	33(1)	53(1)
$K3$	25	33(1)	55.7(7)

are getting near the upper limit of $x=4$, $x=5$ where the stochastic exponentiation technique begins to break down for our limited statistics. Also for simulation $K1$, it can clearly be seen that the lower tail of the distribution stretches well beyond 0. This manifests itself in that about 2.9% of the estimators for $E[e^x]$ were negative, which has a noticeable effect on the statistical errors for observables as will be demonstrated shortly.

Simulation $K2$ seems to be more or less where one would expect this noisy method to behave well. A few of the estimates for x are larger than $x=5$ and although the tail of the distribution stretches into the negative region, in practice this results in very few sign violations of $f(U,\eta,\rho)$ (only 1 out of the total number of statistics equating to 0.02%). The trick of folding the sign of f into the observable may be a practical proposition in this case.

Finally $K3$ is the best behaved of the simulations, with few values of $x>4$ and no sign of violations in $f(U,\eta,\rho)$. The results of this simulation can be analyzed with conventional techniques.

F. Acceptance rates

The acceptance rates of the three KNMC simulations are shown in Table VI. One can see that the gauge acceptance rate seems not to depend on the number of fractional flavors used (N), whereas there is a marked increase in the noise update acceptance rate when N is increased. We believe that being able to achieve a gauge acceptance rate of around 33% by performing quenched updates at a shifted β is a great success of the action matching technology; however, for the algorithm to be practical it is somewhat low. Such a low acceptance rate, combined with updating only one-eighth of the lattice gauge fields with every update, can result in long autocorrelation times.

G. Observables

In Fig. 5 we show our measurements of the plaquette and $\mathrm{Tr}R_M(U)$ for the KNMC simulations as well as the result of the reference HMC calculation for comparison. The error estimates for the noisy simulations do not include the effects of autocorrelations so as not to obscure the effects incurred by the sign violations in $f(U,\eta,\rho)$.

We note with gratification that the results for simulation $K1$ appear unbiased, even with 2.9% of the estimates of $f(U,\eta,\rho)$ having negative signs. However, the statistical errors on this value are massive when compared to those of the other simulations.

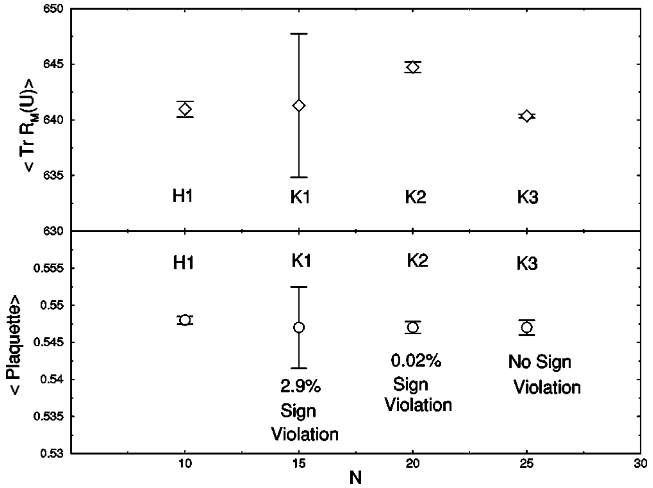

FIG. 5. Observables from the KNMC simulations. The plaquette is shown on the bottom graph, and $\langle \mathrm{Tr}R_M(U)\rangle$ is shown on the top. The values plotted at $N=10$ are the HMC results for comparison.

We also note that the autocorrelation times for the KNMC simulations are probably very long (see Sec. VI H for the detailed discussion), which may further increase the error estimates. Due to this fact, we do not wish to attach any more significance to the consistency between the HMC and KNMC results than to make the claim that the presence of sign violations does not appear to bias our results.

In Table VII we show the bootstrap errors on the numerator and denominator of Eq. (8) used in evaluating the expectation value of the plaquette in the presence of sign violations (for $K1$ since $K3$ is free of sign violations and there is only one single violation in $K2$). In the third line we tabulate the relative error in the plaquette measurements when the sign is not folded in—although it must be borne in mind that doing the analysis this way would give a biased value for the plaquette.

In the case of simulation $K1$ it can clearly be seen that the magnitude of the relative errors, when the sign is folded in, is about two orders of magnitude greater than when it is not, and that the relative errors in the numerator and denominator (first two lines in Table VII) are approximately the same. This clearly suggests that the errors are entirely dominated by the error in the sign.

H. Autocorrelations

Let us now turn to the question of autocorrelations. It is not entirely clear how to best estimate autocorrelation effects

TABLE VII. The relative errors in the numerator and denominator of the quantity needed to estimate the expectation value of the plaquette for $K1$. The third line shows the relative error on the plaquette without folding in the sign.

Simulation	Sign violation	Observable	Relative bootstrap error
$K1$	2.9%	$\langle$Plaquette sgn(f)$\rangle$	0.709%
		$\langle$sgn(f)$\rangle$	0.708%
		$\langle$Plaquette$\rangle$	0.0037%

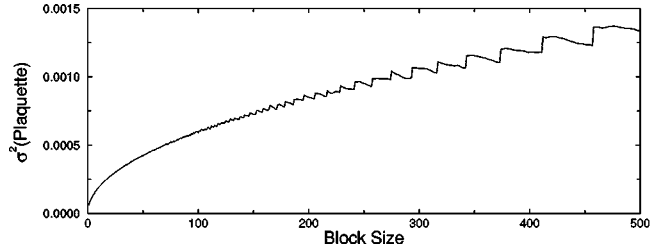

FIG. 6. Variance of the plaquette measurement of $K3$ as a function of block size.

in the presence of sign violations. When a substantial amount of sign violations are present, one would expect these to be the dominant contributors to the statistical error in any case. However, we did attempt to make an investigation into autocorrelation effects in simulation $K3$ where no negative signs are present in $f(U,\eta,\rho)$.

Once again, we used the blocking procedure that was outlined earlier for our HMC simulation (Sec. VI A). The growth of the variance of the plaquette is plotted as a function of block size in Fig. 6. It can be seen that the errors do not plateau as a function of block size, indicating that the integrated autocorrelation time is very long. We expect this is due in part to the fact that only one-eighth of the lattice is updated with every gauge update, and in part because the rate of acceptance for the gauge updates is quite low—about 33%.

We note that in our experience with the quasi-heatbath method in quenched simulations, the plaquette usually decorrelates in about 20–40 sweeps (at around this level of gauge coupling and on similar volumes). However, in this implementation of KNMC only one-eighth of the lattice is updated with any one sweep. This in itself could be expected to increase the autocorrelation time to about 160–320 accepted sweeps. If this is the case, the amount of statistics available to us (around 4000 sweeps) is quite inadequate to measure the autocorrelation time accurately [28] and the lack of the plateau in Fig. 6 should come as no surprise to us. The only statement we wish to make here is that the autocorrelation time is sufficiently long for us not to be able to measure it accurately.

While autocorrelation times of this large magnitude would not present a problem in a conventional quenched simulation where gauge field updates are cheap and no sweeps are rejected (they are after all from a heatbath). However, when one couples a potentially expensive noisy accept/reject step after each one-eighth update the computational cost increases significantly, so that one cannot hope to achieve the level of statistics in quenched simulations. In this case the low acceptance rate of the simulations becomes a problem. Clearly for the KNMC approach to be practicable, a better gauge update algorithm is needed than the one used here, with a higher acceptance rate.

I. Rechecking the tuning

Before we proceed to discuss our small volume simulations, we would like to discuss the quality of the tuning for the three KNMC simulations. On physical grounds, one would expect that the parameter $\lambda_{\rm min}$ which minimizes the variance of the fermion action, is a universal quantity, and should depend largely on the physical parameters which define the expectation value of $\mathrm{Tr}\ln M(U)$. The amount of variance reduction thus achieved is expected to depend on the gauge generation algorithm to some degree, but certainly, one would expect some self-consistency when carrying out the tuning on the HMC and the KNMC data sets.

TABLE VIII. KNMC Tuning results for one flavor for simulation $K3$.

Statistic	Method 1	Method 2
$\sigma(\mathrm{Tr}\,R_M(U))$	10.38	12.73
$\mathrm{Corr}(\mathrm{Tr}\,R_M(U),\mathrm{Tr}\,U_\Box)$	0.98	0.81
$\lambda_{\rm min}$ $(\times 10^{-2})$	3.51(4)	3.526(7)
$\sigma(T(U,\lambda_{\rm min}))$	1.87	7.45
$\sigma(T(U,\lambda_{H1}))$	2.0	7.49

To this end we repeated the procedure for tuning $\lambda_{\rm min}$ using both methods 1 and 2 on estimates of $\mathrm{Tr}R_M(U)$ produced during simulation $K3$ which is not affected by sign violations. The main difference here is that the number of estimates of $\mathrm{Tr}R_M(U)$ varied slightly from configuration to configuration since the number of noisy estimates for x differs for each update. However, with $N=25$ and a value of $c=1.5$ the average number of terms used in evaluating $f(U,\eta,\rho)$ was about 65 terms per gauge update. With over 4000 updates, these statistics should prove adequate.

The retuning results are shown in Table VIII. We note that the value of $\lambda_{\rm min}$ has now increased a little with respect to the HMC results (Table IV); however, this is not a very large change. Indeed, it is less than 10% of the HMC value in Table IV.

We also show in Table VIII the corresponding value of $\sigma(T(U,\lambda_{\rm min}))$. For the purpose of comparison we also redisplay the value of $\sigma(T(U,\lambda))$ using λ obtained from the original tuning on the ensemble of $H1$. We denote this latter quantity as $\sigma(T(U,\lambda_{H1}))$ in Table VIII. It can be seen that the change in the value of $\lambda_{\rm min}$ from that of the HMC result does not reduce the σ value by a great deal, probably because of the very flat minimum of σ as a function of λ.

A similar trend can be seen when switching to method 1 from method 2, as was visible when tuning in the ensemble of $H1$. When averaging the noise fields and effectively measuring $\langle\mathrm{Tr}R_M(U)\rangle$ on each configuration, the subtraction of the loop action from the plaquette is much more effective than when using method 2—(a reduction from $\sigma\approx 8$ to $\sigma\approx 2$ in the former case against a reduction from $\sigma\approx 12$ to $\sigma\approx 7.5$ in the latter).

VII. SMALL VOLUME SIMULATIONS

A. Reference HMC simulation $H2$

Simulation $H2$ was performed by generating 10000 HMC trajectories, of which the first 250 were dropped for equilibration. We measured $\mathrm{Tr}R_M(U)$ on every trajectory with 20 noise vectors per configuration. Since we left the physical

TABLE IX. Results from simulation $H2$. The first set of errors are the naive bootstrap errors. The second set shows the effects of autocorrelation estimated by blocking the data.

Observable	Value
$\langle$Plaquette$\rangle$	0.5653(1)(2)
$\langle \mathrm{Tr}\, R_M(U) \rangle$	45.73(2)(4)
$\langle \overline{\psi}\psi \rangle$	0.92105(3)(7)

parameters unchanged with respect to $H1$ but decreased our lattice volume substantially—see Table II—we also computed the chiral condensate $\langle \overline{\psi}\psi \rangle$ on this ensemble to make sure that decreasing the lattice spacing has not pushed the simulation into the deconfined sector. This quantity was defined as

$$\langle \overline{\psi}\psi \rangle = \left\langle \frac{1}{12V} \mathrm{Tr}\, M(\kappa;U)^{-1} \right\rangle_U . \tag{37}$$

In practice the trace $\mathrm{Tr}\, M(\kappa;U)^{-1}$ was estimated by averaging over $N_\eta = 20$ noisy estimators $\eta^\dagger M^{-1}(\kappa;U)\eta$. The required inverse could be obtained with minimal overhead, from the procedure we used to estimate $\mathrm{Tr} R_M(U)$, with an added shift term of zero in our multiple shift solver.

The results of observables are shown in Table IX. An important thing to note is that the value of $\langle \overline{\psi}\psi \rangle = 0.92105$ indicates that the simulation is still in the confined region.

B. Tuning λ_{min} for simulation $K4$

We used the results from simulation $H2$ to tune λ_{min} for simulation $K4$. Since the lattice volume is quite small, and because of the amount of statistics available, we simply employed method 1 without any additional manual fine tuning. The tuning results are shown in Table X.

C. KNMC simulation $K4$

We performed simulation $K4$ with the same physical parameters as all the others. From the tuning exercise above, we identified our actions splitting parameters that we show in Table II. We maintained the same update scheme as for the other simulations, namely, that our updates consisted of updating one-eighth of the gauge field, followed by an update of the noise fields corresponding to one fractional flavor. Overall we performed 15000 such compound updates, having started from an equilibrated configuration from simulation $H2$.

TABLE X. Tuning results from H2.

Statistic	Value
$\sigma(\mathrm{Tr} R_M(U))$	1.58
$\mathrm{Corr}(\mathrm{Tr} R_M(U), \mathrm{Tr}\, U_\square)$	0.92(7)
λ_{min} $(\times 10^{-2})$	3.25(2)
$\sigma(T(U,\lambda_{\mathrm{HMC}}))$	0.67
$\langle T(U,\lambda_{\mathrm{min}}) \rangle$	$-38.920(7)$

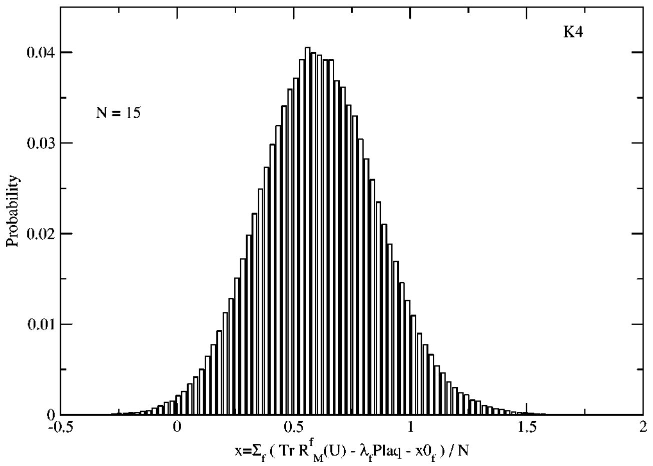

FIG. 7. Distribution of quantity x for simulation $K4$.

D. Distribution of x for simulation $K4$

We show the distribution of the estimators x in Fig. 7; this shows the same information as Fig. 4. We can see that because of the smaller volume, and the comparatively large number of fractional flavors, the distribution of x for $K4$ is close to optimal, in the sense that there appear to be no values of x such that $x < -1$ which implies that we are completely free from sign violations, and that the x have a mean of around 0.6, with no values of $x > 2$. The stochastic exponentiation technique should be working perfectly in this regime.

E. Acceptance rates for $K4$

The acceptance rates for $K4$ were 79% for gauge update steps and 87% for the noise update steps, respectively. This again seems to imply that the algorithm is behaving optimally.

F. Observables

We show the mean values of the plaquette and $\mathrm{Tr}\, R_M(U)$, along with the corresponding values from the reference $H2$ run in Table XI. We can see from Table XI that there is a systematic difference in the mean plaquette and $\mathrm{Tr} R_M$ between the KNMC simulation $K4$ and the corresponding HMC simulation $H2$. However, the difference is rather small, less than 1% in the case of the plaquette and about 3% in the case of the $\mathrm{Tr} R_M$.

In order to ensure that the systematic discrepancy is not due to updating the gauge fields with a quenched action at a shifted value of the coupling β, we carried out a short quenched simulation, using $\beta = 5.695$. This was the value of β used in our KNMC simulations when we took into account the shift from splitting the determinant. We display the values of the observables from this quenched simulation in Table XI labelling it as simulation Q. It is clear that the observables from $K4$ are substantially different from the

TABLE XI. Observables from simulation $K4$ and $H2$. The simulation labeled Q is from a quenched simulation carried out at $\beta = 5.695$. Autocorrelations have been taken into account in the errors.

Simulation	Observable	Value
$K4$	⟨Plaquette⟩	0.5603(8)
$H2$	⟨Plaquette⟩	0.5653(2)
Q	⟨Plaquette⟩	0.5585(5)
$K4$	$\langle \mathrm{Tr} R_M(U) \rangle$	44.20(15)
$H2$	$\langle \mathrm{Tr} R_M(U) \rangle$	45.73(4)
Q	$\langle \mathrm{Tr} R_M(U) \rangle$	41.92(15)

quenched simulation, and lie much closer to the values from $H4$. We are therefore confident that the difference between $K4$ and $H2$ is not due to updating the gauge fields at the shifted value of $\beta = 5.695$. Rather we believe it comes from the fact that our KNMC simulations use $R_M(U)$ in the fermion action which is an approximation to $\mathrm{Tr} \ln M(U)$. In effect, the HMC and KNMC simulations are simulating slightly different actions in which case the small systematic discrepancy is perhaps unsurprising.

G. Autocorrelations in *K*4

Let us now turn to the question of autocorrelations in simulation $K4$. We show in Fig. 8 the results of our blocking procedure, both for the plaquette and also for our estimates of $\mathrm{Tr} R_M(U)$. The graphs show that the variance of both observables stops increasing at a block size of about 250 samples, indicating that the data is free from autocorrelations at that point, or put in a different way, that the integrated autocorrelation time is around 125 compound updates. This is in contrast to Fig. 6 showing the same kind of measurement on the larger volume simulations, where the variance does not seem to plateau. Hence, our small volume simulations show that the Monte Carlo procedure will in fact converge.

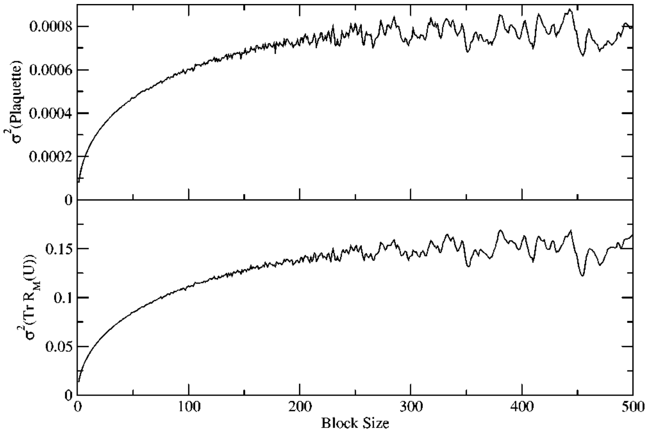

FIG. 8. Variances of the plaquette and $\mathrm{Tr} R_M(U)$ as a function of block size after the blocking procedure of Sec. VI H.

VIII. SUMMARY OF NUMERICAL RESULTS

A. Stochastic exponentiation technique

The stochastic exponentiation technique works well when the argument x to be exponentiated is small and positive. When $x > 1$ successive terms in the expression for $f(U,\eta,\rho)$ have greater and greater numerical value although the probability of reaching these terms still drops factorially. This implies that the variance of the estimates is likely to be large when x is large, and also that the estimates for the exponential are likely to be poor when only a few terms are taken. If x is negative, one risks getting negative values of $f(U,\eta,\rho)$ which can result in large statistical errors.

B. Tuning λ_{min}

Our main result here is that the tuning can be done in two ways (methods 1 and 2) to carry out the minimization of $\sigma(T(U,\lambda))$. We found that a much larger degree of noise reduction can be achieved by subtracting the loop action using method 1 rather than method 2. Further, the minima thus found is very flat with respect to λ (see Figs. 2 and 3), implying that not much gain may be made by dynamically tuning the λ parameter.

These results seem to imply that a greater improvement may be achieved in the acceptance rates of the noisy algorithm using the loop-splitting technique if more noise vectors were used in the noisy estimators of $\mathrm{Tr} R_M(U)$ instead of the current one vector per estimator (i.e. if N_η was increased.) However, this would also imply more numerical work as computing each noisy estimator involves a multi-mass inversion. On the other hand, it may be possible to reduce the number of fractional flavors (N) in return. Further investigation is required to establish when the trade-off becomes worthwhile. Finally, it was found that switching to method 2 from method 1 on the noisy data sets showed a behavior pattern very similar to switching between methods 1 and 2 on the HMC data, even if the actual values were somewhat different.

C. Observables and sign violations

While the expectation values of observables appear to be unbiased in our large volume simulations and show a small systematic discrepancy in the small volume high statistics simulation, it appears that even a small number of negative signs in $f(U,\eta,\rho)$—such as 2.9% of the total number of estimates—can completely dominate the statistical errors. In this situation the effort of creating more and more configurations goes into reducing the error in the estimate of $\langle \mathrm{sgn}(f(U,\eta,\rho)) \rangle$ a more difficult problem than the usual $1/\sqrt{N}$ problem of reducing the errors in the bare observables. While in the KK linear accept/reject approach these sign violations manifest themselves as an explicit bias in the result, in KNMC this bias is traded for a larger statistical error.

D. Autocorrelations

We have shown in our small volume simulations that the autocorrelation times for the plaquette and $\mathrm{Tr} R_M(U)$ are long but finite, in other words, that eventually the algorithm would converge to the correct equilibrium distribution. We do accept, however, that the autocorrelation times are rather long. This coupled with the drop in the acceptance rate, from around 80% to about 30% as the volume was increased from 4^4 to 8^4 sites, indicates that this method of updating will not scale very well to larger volumes. This issue needs to be addressed if the algorithm is to be competitive with say HMC.

IX. ISSUES NOT ADDRESSED IN THIS STUDY

This study was the initial foray into the study of KNMC algorithms. There are several issues which have not been addressed which are also relevant to the algorithm. We outline two of these here.

A. Equilibration

In our study we have always started our simulations from an equilibrated configuration produced by our preliminary HMC study. One may very well ask the question: ''How would we equilibrate our algorithm and tune the necessary parameters if the reference simulation was not present?'' We point to the idea outlined in [24]. The idea presented there is that one can carry out an initial quenched simulation, which can be used to carry out a preliminary tuning. This will provide amongst other things a shifted β value. One can then carry out a second quenched computation with the shifted β value, thus bringing the quenched configuration distribution as close to the intended dynamical one as possible. At this point, one can start to carry out simulations with the noisy algorithm, retuning β and the other parameters along the way until a self-consistency is achieved. This is possible because the tuning in [24] can be carried out in any measure.

B. The question of an infinite number of noise fields

One may be concerned that since technically an infinite number of dynamical noise fields are present in Eq. (5) it is not possible to update them all. In particular, very high order terms in Eq. (5) may never be reached. Thus some of these noise fields will have infinitely long autocorrelation times. Another way of saying this is that the KNMC algorithm may not be ergodic in its infinite variable state space.

While this is a problem in principle, we do not expect it to be a problem in practice, since the probability of reaching the higher order terms is factorially suppressed. Because of this suppression, we expect that these fields can have little effect on our partition function and that any bias in our results from such fields is expected to be very much smaller than statistical errors. It may be possible to construct operators that probe these high order terms explicitly, where the effect of long autocorrelations should be clearly visible. Perhaps a more relevant potential setback comes from the visibly long autocorrelations of our gauge update procedure. We note that in the KK approach, this problem does not arise, since in that case the noise is not part of the state space. On the other hand, algorithms adopting the KK accept/reject step have the in-principle problem of probability bound violations which can introduce a bias into the answers. Hence in choosing between the two approaches, one has a choice of which in-principle problem one wishes to accept as the challenge.

X. CONCLUSIONS

We have developed a QCD implementation for the Kentucky noisy Monte Carlo approach and performed an initial numerical study in the context of two flavors of dynamical Wilson fermions. This study was a success in several ways, most notably since we have managed to assemble all the necessary numerical technology required for incorporating the fermion determinant directly for the first time. The method produced results that are consistent with reference hybrid Monte Carlo simulations, barring small systematic effects.

We have gained valuable insight into the necessary tuning methodology, and have learned what essentially drives the algorithm, notably the stochastic properties of the quantity x, which needs to be distributed so that it is of $O(1)$ and has a small variance. A large variance leads to many excessively large estimates in the tail of the distribution, causing the stochastic exponentiation technique to be inefficient. Also, on the other side of the distribution, one could get many sign violations which, while not introducing bias, can lead to large statistical errors. Even though the distribution can be made arbitrarily narrow by employing more noise fields, by using more loops for splitting the determinant, and by using a larger number of fractional flavors, all these come at the price of an increase in computational cost.

Unfortunately, in our current implementation, the algorithm is not particularly efficient. It suffers from the problem of long autocorrelations and rather low acceptance rates, on even fairly small lattices such as those having $V=8^4$ sites. One possible way for addressing this issue could be the use of molecular dynamics for updating the gauge field. Once again, however, this improvement would come at a potentially high computational cost as in HMC.

In addition, several other improvements have been suggested for making the algorithm more efficient [29], notably using the technique of eigenvalue deflation [30] and the use of additional noisy techniques [31,32], both to improve the convergence of matrix inversions.

Despite the relative inefficiency of the current implementation, we believe that our approach holds great future promise with its capability to handle an arbitrary number of fermion flavors. More importantly, in combination with the projection of the definite baryon number from the determinant, it is a viable candidate to be used in a finite density algorithm at zero temperature [5].

Since the submission of this paper for publication, we became aware of a recent algorithm which also uses local updates for the gauge links and noisy estimators for the ratio of the fermion determinant—the partial global stodastic metropolis (PGSM) algorithm [33,34]. This approach is similar to ours, and in particular employs the same fractional flavor idea to split the determinant.

ACKNOWLEDGMENT

We would like to acknowledge DOE grant DE-FG05-84ER0154 and the Center for Computational Sciences at the

University of Kentucky for financial support. We are extremely grateful to N. H. Christ, R. D. Mawhinney, Columbia University and the RBC Collaboration for providing us with access to the QCDSP hardware and the application code to ease our development, and for providing a platform for simulation and their hospitality while B. J. was at Columbia University. We would like to thank the UKQCD Collaboration for allowing us to use their GHMC code, optimized for the T3E. B. J. would like to thank the Department of Physics at Columbia University for providing support for travel between Columbia University and the University of Kentucky under the SciDAC addition to their DOE grant DE-FG02-92ER40699, and PPARC for financial support by way of employment through grant PPA/G/O/1998/00621 through the later stages of this work. B. J. would also like to thank A. D. Kennedy for many useful discussions on the subject.

[1] S. Duane, A.D. Kennedy, B.J. Pendleton, and D. Roweth, Phys. Lett. B **195**, 216 (1987).

[2] A.D. Kennedy, I. Horváth, and S. Sint, Nucl. Phys. B (Proc. Suppl.) **73**, 834 (1999).

[3] T. Takaishi and P. de Forcrand, Int. J. Mod. Phys. C **13**, 343 (2002).

[4] M.G. Alford, Nucl. Phys. B (Proc. Suppl.) **83**, 345 (2000).

[5] K.F. Liu, Int. J. Mod. Phys. B **16**, 2017 (2002).

[6] A. Duncan, E. Eichten, and H. Thacker, Phys. Rev. D **59**, 014505 (1999).

[7] I. Horváth, Nucl. Phys. B (Proc. Suppl.) **83**, 804 (2000).

[8] C. Thron, S.J. Dong, K.F. Liu, and H.P. Ying, Phys. Rev. D **57**, 1642 (1998).

[9] N. Metropolis *et al.*, J. Chem. Phys. **21**, 1087 (1953).

[10] A.D. Kennedy and J. Kuti, Phys. Rev. Lett. **54**, 2473 (1985).

[11] G. Bhanot and A.D. Kennedy, Phys. Lett. **157B**, 70 (1985).

[12] Naturally, when this is the case the quantity in question fails to be a probability. We refer to the problem of the acceptance probabilities being negative or greater than one as *low and high probability bound violations*, respectively.

[13] L. Lin, K.F. Liu, and J.H. Sloan, Phys. Rev. D **61**, 074505 (2000).

[14] A. Hasenfratz and T.A. DeGrand, Phys. Rev. D **49**, 466 (1994).

[15] J.C. Sexton and D.H. Weingarten, Nucl. Phys. B (Proc. Suppl.) **42**, 361 (1995).

[16] In our general discussion we assume that the determinant is positive for arbitrary gauge background. For Wilson fermions (unlike overlap fermions) this is not strictly satisfied but the rare encounter of a negative sign can be monitored, in principle, and taken care of by including the sign into the observables. In our case this standard strategy will be adopted for other reasons anyway.

[17] S. Bernardson, P. McCarty, and C. Thron, Comput. Phys. Commun. **78**, 256 (1993).

[18] A.D. Kennedy, J. Kuti, S. Meyer, and B.J. Pendleton, J. Comput. Phys. **64**, 133 (1986).

[19] In this sense Z_2 noise is optimal. With other types of noise such as Gaussian noise, the variance receives contributions from diagonal terms which one cannot subtract off. In this case the unbiased subtraction scheme described here is ineffective.

[20] I. Montvay and G. Munster, *Quantum Fields on a Lattice* (Cambridge University Press, Cambridge, England, 1994), p. 491.

[21] Also known as multimass.

[22] B. Jegerlehner, hep-lat/9612014.

[23] U. Glassner *et al.*, hep-lat/9605008.

[24] A.C. Irving and J.C. Sexton, Phys. Rev. D **55**, 5456 (1997).

[25] D. Chen *et al.*, Nucl. Phys. B (Proc. Suppl.) **60A**, 241 (1998).

[26] Z. Sroczynski, S.M. Pickles, and S.P. Booth, Nucl. Phys. B (Proc. Suppl.) **63**, 949 (1998).

[27] R.G.E. *et al.*, http://www.jlab.org/∼edwards/szin.html.

[28] Folklore holds that to measure a particular autocorrelation time τ with some good accuracy, roughly 100τ to 1000τ statistics are needed. In our case, that could be as much as 320,000 sweeps as opposed to the 4000 we have available to us.

[29] W. Wilcox (private communication).

[30] R.B. Morgan and W. Wilcox, Nucl. Phys. B (Proc. Suppl.) **106**, 1067 (2002).

[31] W. Wilcox, Nucl. Phys. B (Proc. Suppl.) **106**, 1064 (2002).

[32] P. de Forcrand, Phys. Rev. E **59**, 3698 (1998).

[33] A. Hasenfratz and F. Knechtli, Comput. Phys. Commun. **148**, 81 (2002).

[34] A. Alexandru and A. Hasenfratz, Phys. Rev. D **66**, 094502 (2002).