Speeding up finite step-size updating of full QCD on the lattice

M. Hasenbusch*
Humboldt Universität zu Berlin, Institut für Physik, Invalidenstrasse 110, D-10115 Berlin, Germany
(Received 30 July 1998; published 3 February 1999)

We propose various improvements of finite step-size updating for full QCD on the lattice that might turn finite step-size updating into a viable alternative to the hybrid Monte Carlo algorithm. These improvements are noise reduction of the noisy estimator of the fermion determinant, unbiased inclusion of the hopping parameter expansion and a multi-level Metropolis scheme. First numerical tests are performed for the 2 dimensional Schwinger model with two flavors of Wilson fermions and for QCD with two flavors of Wilson fermions and Schrödinger functional boundary conditions. [S0556-2821(99)04105-3]

PACS number(s): 12.38.Gc, 02.70.Lq, 11.15.Ha

I. INTRODUCTION

The incorporation of fermionic degrees of freedom in the simulation of lattice QCD is a long standing problem. At present the hybrid Monte Carlo algorithm [1] is the state of the art algorithm for the simulation of full QCD on the lattice. Most of QCD simulations up to now were performed in the so-called quenched approximation where the fermion determinant is approximated by a constant factor. Simulations of 2-flavor QCD on lattices of a size that might just allow a physically meaningful interpretation were performed just recently [2–5].

Extremely large autocorrelation times of the topological charge have been observed in hybrid Monte Carlo simulations of QCD with staggered fermions [6]. One might ask whether the small step size of the hybrid Monte Carlo creates particular problems in switching the topological sector. The authors of Ref. [7] found that the topological charge is indeed the slowest mode in the hybrid Monte Carlo simulation of QCD with Wilson fermions. However, the integrated autocorrelation time of the topological charge is only larger by a small factor than that of other quantities. Nevertheless, it seems desirable to have a finite step-size updating algorithm as a complement of the hybrid Monte Carlo algorithm.

Formally the multi-boson approach of Lüscher [8] allows for a finite-step-size updating of the gauge field. However, in the chiral limit the number of bosonic fields has to be increased. These fields amount to a large ''force'' on the gauge field and allow only small changes in a single update step. See Refs. [9–11].

In fact the first proposal for a practical QCD algorithm by Weingarten and Petcher [12] in 1981 is a finite step-size algorithm. Since the update of a single link (or a fixed small number of links) requires the evaluation of the inverse of the fermion matrix applied to a vector, the CPU time required for a full sweep over the lattice increases with $Volume^2$ even at fixed β and κ.

For that reason this algorithm and variants of it [13–16] (and many more references) were abandoned when the hybrid Monte Carlo algorithm [1] was introduced in 1987 which has a volume dependence like $Volume^{5/4}$ for fixed β and κ.

In this paper we will demonstrate how finite step-size algorithms can be speeded up by a large (order 100) factor. Still progress is needed to overcome the $Volume^2$ increase of the CPU time, such that the algorithm becomes an alternative to the hybrid Monte Carlo algorithm in present day simulations.

The paper is organized as follows. In Sec. II we will discuss the action to be simulated. Here we shall explain how the hopping parameter expansion can be incorporated into the simulation in an unbiased form. Next we show how the variance of the noisy estimator of the fermion determinant can be reduced. The major novelty of the simulation is the use of a sequence of approximations of the exact action in a multi-level Metropolis scheme (Sec. III). In Sec. IV we will present first numerical tests of the methods proposed. These tests are performed with the two flavor 2D Schwinger model with Wilson action and two-flavor QCD with Wilson action and Schrödinger functional boundary conditions. In Sec. V we compare our method with related approaches. Finally we give a short outlook on possible improvements of the methods discussed.

II. THE ACTION TO BE SIMULATED

In order to perform numerical simulations the Grassmann variables in the path-integral formulation of QCD are in general integrated out. What remains is a Boltzmann factor that only depends on the gauge degrees of freedom. For example in the case of two flavors of mass-degenerate fermions we obtain

$$Z = \int \mathrm{D}[U] \exp(-S_G[U]) \det M^\dagger M, \tag{1}$$

where $S_G[U]$ is the gauge action and M the fermion matrix in its non-Hermitian form.

For reasonably large lattice sizes the problem still remains intractable in this form since the evaluation of the determinant requires of the order $Volume^3$ operations.

A way out of the problem was proposed by Weingarten and Petcher who use the identity

*Email address: hasenbus@physik.hu-berlin.de

$$\det M^\dagger M \propto \int \mathrm{D}[\eta]\mathrm{D}[\eta^\dagger]\exp(-|M^{-1}\eta|^2) \tag{2}$$

to introduce auxiliary bosonic degrees of freedom:

$$Z=\int \mathrm{D}[U]\mathrm{D}[\eta]\mathrm{D}[\eta^\dagger]\exp(-S_G[U]-|M^{-1}\eta|^2). \tag{3}$$

This reduces the calculation of the action to a problem which takes of the order *Volume* operations at the price of a noisy estimate of the fermion matrix. Note that the hybrid Monte Carlo algorithm is also based on this action.

In the following we will use the hopping parameter expansion to evaluate part of the fermion determinant non-noisy while the remainder is treated in a similar fashion as Weingarten and Petcher treat the full fermion matrix.

A. Making use of the hopping parameter expansion

Let us start the discussion with the effective weight for two flavors of degenerate fermions

$$\det M^\dagger M = \det M^\dagger \det M = \exp(\mathrm{tr}\ln M^\dagger + \mathrm{tr}\ln M). \tag{4}$$

M can be written as

$$M = 1-\kappa H \tag{5}$$

(in the red-black preconditioned case we have to replace κ by κ^2). $\ln M$ can now be written as a Taylor series in κ:

$$\ln M = -\kappa H - \frac{1}{2}\kappa^2 H^2 - \frac{1}{3}\kappa^3 H^3 \cdots . \tag{6}$$

In the following we use the first k terms explicitly as action for our updates of the link variables while the remaining part is dealt with stochastically.

We define

$$\ln \widetilde{M} = \ln M + \sum_{n=1}^{k} \frac{1}{n}\kappa^n H^n \tag{7}$$

or equivalently

$$\widetilde{M} = M \exp\left(\sum_{n=1}^{k} \frac{1}{n}\kappa^n H^n \right). \tag{8}$$

So we arrive at the action

$$S[U,\eta] = S_G[U] + \mathrm{tr}\sum_{n=1}^{k} \frac{1}{n}\kappa^n H[U]^n + \mathrm{H.c.} + |\widetilde{M}[U]^{-1}\eta|^2. \tag{9}$$

In the following section we will give a noise reduced replacement of $|\widetilde{M}^{-1}\eta|^2$ which will lead to the action which is simulated at the end.

B. Reducing noise by using roots of the fermion matrix

Here we consider as noise the fluctuations of

$$|M(U)^{-1}\eta|^2 + \mathrm{tr}\ln M(U) + \mathrm{H.c.} \tag{10}$$

with varying U for a fixed noise vector η. The motivation for reducing the noise of the estimator of the fermion determinant is to obtain larger Metropolis acceptance rates for a given proposal or to allow for a larger change of the gauge field within one proposal at a sustained acceptance rate. A natural idea to reduce the noise is to use the average obtained from several noise vectors η_i rather than using a single noise vector. Below we will discuss how this can be achieved in a consistent way.

The fermionic determinant can be rewritten as

$$\begin{aligned}\det M^\dagger M &= \det M^\dagger \det M = (\det M^{1/r\dagger})^r (\det M^{1/r})^r \\ &= (\det M^{1/r\dagger} M^{1/r})^r\end{aligned} \tag{11}$$

which allows us to rewrite each of the factors separately as an integral over auxiliary bosonic variables:

$$\begin{aligned}(\det M^{1/r\dagger} M^{1/r})^r \propto \int \mathrm{D}[\eta_1]\mathrm{D}[\eta_1^\dagger]\cdots\mathrm{D}[\eta_r]\mathrm{D}[\eta_r^\dagger] \\ \times \exp\left(-\sum_{i=1}^{r} |M^{-1/r}\eta_i|^2 \right).\end{aligned} \tag{12}$$

The $M^{1/r}$ and $M^{-1/r}$ are computed as Taylor series in κH.

In the limit $r\to\infty$ the noise is completely eliminated. In this limit the integrand gives, up to a factor that does not depend on the gauge field, the fermion determinant.

Note that for complex x in a unit circle around 1 we have

$$\lim_{r\to\infty} r[(1-x)^{1/r}-1] = -\lim_{r\to\infty} r[(1-x)^{-1/r}-1] = \ln(1-x). \tag{13}$$

This can be easily shown: With $\exp(\widetilde{x}) := 1-x$ we get

$$r[(1-x)^{1/r}-1] = r[\exp(\widetilde{x}/r)-1] = \widetilde{x} + O(1/r). \tag{14}$$

Equilibrium η_i for a given gauge field are given by

$$\eta_i = M'^{1/r}\chi_i = \chi_i + \frac{1}{r}\ln M' \chi_i + O(1/r^2) \tag{15}$$

where the χ_i have a Gaussian distribution. Here M' is the fermion matrix of a fixed reference gauge field U'. Note that in the Monte Carlo simulation updates of the gauge fields and the noise vectors alternate. Hence one might think of U' as the gauge field for which the η_i were updated before.

We obtain, for the integrand of Eq. (12),

$$\exp\left(-\sum_{i=1}^{r}|M^{-1/r}\eta_i|^2\right)$$
$$=\exp\left[-\sum_{i=1}^{r}\left|\left(1-\frac{1}{r}\ln M\right)\eta_i+O(1/r^2)\right|^2\right]$$
$$=\exp\left[-\sum_{i=1}^{r}\left|\left(1-\frac{1}{r}\ln M\right)\chi_i+\frac{1}{r}\ln M'\chi_i+O(1/r^2)\right|^2\right]. \tag{16}$$

With

$$\lim_{r\to\infty}\frac{1}{r}\sum_{i=1}^{r}\chi_i^\dagger \ln M\chi_i=\mathrm{tr}\ln M, \tag{17}$$

we arrive at the result

$$\lim_{r\to\infty}\exp\left(-\sum_{i=1}^{r}|M^{-1/r}\eta_i|^2\right)=\mathrm{const}(M')\times\det M^\dagger M. \tag{18}$$

Finally we arrived at the action (up to red-black preconditioning) which is used for the simulations:

$$S=S_G+\sum_{n=1}^{k}\frac{1}{n}\kappa^n(\mathrm{tr}H^n+\mathrm{H.c.})+\sum_{i=1}^{r}|\tilde{M}^{-1/r}\eta_i|^2. \tag{19}$$

The parameters that characterize the action are the order of the hopping parameter expansion k and the root r of the modified fermion matrix $\tilde{M}$.

The price to pay for the noise reduction is to deal with a larger number of noise vectors η_i. How the benefit compares with the extra cost will be studied numerically in Sec. IV A 1.

III. HIERARCHY OF ACCEPTANCE STEPS

The novel feature of our updating scheme is that a sequence of approximations

$$S_1,S_2,\ldots,S_l=S \tag{20}$$

of the full action is used. This sequence of actions is organized such that the actions become better approximations of the full action while at the same time the computational effort to compute them increases.

For the action above the sequence of approximations is realized in a rather trivial way:

The first approximation is given by the gauge action plus the ''hopping part'' of the fermion action

$$S_1=S_G+\sum_{n=1}^{k}\frac{1}{n}\kappa^n(\mathrm{tr}H^n+\mathrm{H.c.}), \tag{21}$$

while better approximations are given by the truncation of the Taylor series of $\tilde{M}^{-1/r}\eta_i$ at a finite order t.

Hence the sequence of actions is characterized by a sequence of truncation orders

$$t_2<t_3<\cdots<t_l=\text{``}\infty\text{''} \tag{22}$$

where ''∞'' means that the series is truncated at an order such that the sum of the remaining terms is below a given (small) bound.

We should note that the auxiliary field η can be updated in a global heat-bath step

$$\eta_j=\tilde{M}^{1/r}\chi_j \tag{23}$$

where χ_j has a Gaussian distribution. Therefore we will update the η_j just once for every update cycle. In the following discussion of the update of the gauge field U we therefore assume a fixed η_j.

A. Simplest case: $l=2$

Let us first discuss the algorithm for the simplest case which is given by $l=2$. Such two-step decompositions of the action are actually in common use.

For the simplicity of discussion we assume $r=1$ and $k=0$. First a number of link updates are performed with Cabibbo-Marinari [17] updating or micro-canonical overrelaxation [18,19] such that the whole update sequence respects detailed balance with respect to the pure gauge action. (For a detailed discussion see below.) This way a proposal U' for the full action is generated. The proposal U' is accepted with the probability given by

$$A(U',U)=\min[1,\exp(-s_2[U']+s_2[U])] \tag{24}$$

with $s_2=S_2-S_1$ and

$$-s_2[U']+s_2[U]=-|M(U')^{-1}\eta|^2+|M(U)^{-1}\eta|^2. \tag{25}$$

It is easy to see that this algorithm satisfies detailed balance with respect to the action S_2:

For $U\neq U'$ we get the following:

Case 1: $s_2[U']\geq s_2[U]$. For the update of U to U' we obtain

$$P_2(U',U)=P_1(U',U)\exp(-s_2[U']+s_2[U]), \tag{26}$$

where $P_2(U',U)$ and $P_1(U',U)$ are the probabilities to update from U to U' at level 2 and level 1 respectively. For the update of U' to U we get

$$P_2(U,U')=P_1(U,U'). \tag{27}$$

Taking the ratio and using the fact that P_1 satisfies detailed balance with respect to S_1 we get

$$\frac{P_2(U',U)}{P_2(U,U')}=\exp(-S_1[U']+S_1[U])\times\exp(-s_2[U']+s_2[U])$$
$$=\exp(-S_2[U']+S_2[U]). \tag{28}$$

Case 2. $s_2[U']<s_2[U]$ works just analogously.

B. Generalization to $l>2$

The generalization to $l>2$ is done recursively. Given an update algorithm UP_{i-1} that satisfies detailed balance with respect to an action S_{i-1} we construct an algorithm UP_i that satisfies detailed balance with respect to the action S_i. This process is iterated until we reach the exact action of the model.

We use the algorithm UP_{i-1} that satisfies detailed balance with respect to the action S_{i-1} in order to construct a proposal. Starting from a configuration U we apply the elementary update step of $UP_{i-1}m_{i-1}$ times to obtain the proposal U'. The composition of the elementary update steps has to be done in such a way that detailed balance is maintained for the whole sequence. (A discussion of this point is given below.) The proposal U' is then accepted with the probability

$$A_i(U',U)=\min[1,\exp(-s_i[U']+s_i[U])] \tag{29}$$

with $s_i=S_i-S_{i-1}$.

Let us try to give an intuitive argument why such a sequence of levels could be of advantage. Changes of the gauge field are done on level 1 of the algorithm. The ''only'' thing that happens at subsequent levels is that part of these updates at level 1 are rejected. One might imagine the levels $i>1$ as a set of filters that are used subsequently. The potential advantage of the set of filters compared to a single filter is that the expensive filters (i large) have to be applied less frequently because the cheaper filters of the low levels already give a quite good preselection.

C. Composing updates

Next we have to discuss how the updates with the action S_1 should look in detail. The basic building blocks in the case of QCD will be the well known Cabibbo-Marinari heat-bath update [17] and the micro-canonical over-relaxation update [18,19]. Both algorithms satisfy detailed balance when applied to a single subgroup of a given link variable.

However, one should note that a sequence of (different) updating steps which individually satisfy detailed balance in general does not satisfies detailed balance as a whole. This statement in particular applies to sweeping through the lattice with a Cabibbo-Marinari or over-relaxation update in a given order.

In simulations of the pure gauge theory this does not pose a problem since the sequence still satisfies the weaker and sufficient condition of stability:

$$\exp[-S(U')]=\int \mathrm{D}U P(U',U)\exp[-S(U)]. \tag{30}$$

However, the basic building blocks of our algorithm have to satisfy detailed balance. The simplest composition that does satisfy detailed balance is to select the link and the sub-group randomly for each link update.

For performance reasons however it is desirable to stay with a regular pattern of sweeping through the lattice. A simple way to achieve this is to choose with probability 1/2 either a given sequence of elementary update steps or with equal probability its exact reverse (see for example Ref. [11]). This symmetrization restores detailed balance. Let as prove the statement for a sequence of two updates.

The update probability of the symmetrized composite of the two updates p_1 and p_2 is given by

$$P(U'',U)=\frac{1}{2}\int dU'[p_1(U'',U')p_2(U',U) + p_2(U'',U')p_1(U',U)]. \tag{31}$$

Now for any intermediate configuration U' we have

$$\begin{aligned}&\frac{p_1(U'',U')p_2(U',U)+p_2(U'',U')p_1(U',U)}{p_1(U,U')p_2(U',U'')+p_2(U,U')p_1(U',U'')}\\ &\quad=\exp[-S(U'')+S(U')]\exp[-S(U')+S(U)]\\ &\quad=\exp[-S(U'')+S(U)],\end{aligned} \tag{32}$$

where we have used that p_1 and p_2 satisfy detailed balance. Since that ratio is identical for any intermediate configuration U', the ratio of the integral over the U' takes the same value as for each of the individual U'. Hence detailed balance is satisfied for the whole sequence.

IV. NUMERICAL RESULTS

As first tests of the algorithm proposed above we simulated the (1+1)-dimensional 2-flavor Schwinger model with Wilson fermions and (3+1)-dimensional 2-flavor QCD with Wilson fermions and with Schrödinger-functional boundary conditions. In both cases we performed most of the simulations at one set of parameters. The sets of parameters were chosen such that we could compare our results with the literature [20] in the case of the Schwinger model and with results obtained within the ALPHA Collaboration [21,22] in the case of QCD. The aim of this numerical study is to obtain a first impression of the effectiveness of our new proposals compared to the hybrid Monte Carlo and multi-boson algorithms. Since it is very likely that further substantial improvements of the algorithm can be found, we think that it is not the time yet to systematically study the dependence of the performance of the algorithm on all parameters of the theory.

A. Schwinger model in 2 dimensions

We simulate the two-flavor two dimensional lattice Schwinger model with Wilson fermions. The gauge part of the action is given by

$$S_G=-\beta\sum_x \mathrm{Re}\, U_{plaq,x}, \tag{33}$$

where

$$U_{plaq,x}=U_{x,1}U_{x+(1,0),2}\overline{U}_{x+(0,1),1}\overline{U}_{x,2}, \tag{34}$$

where the link variables $U_{x,\mu}$ are elements of $U(1)$. The fermion matrix can be written as

$$M = 1 - \kappa H. \tag{35}$$

The hopping part of the fermion matrix is given by

$$H = \sum_{\mu} \left[\delta_{x-\hat{\mu},y}(1+\gamma_\mu) U_{x-\hat{\mu},\mu} + \delta_{x+\hat{\mu},y}(1-\gamma_\mu) \overline{U}_{x,\mu} \right], \tag{36}$$

where in two dimensions we choose the γ-matrices as

$$\gamma_1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \gamma_2 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{37}$$

As for other algorithms preconditioning improves the performance of our algorithm considerably. A simple version of preconditioning is the so-called red-black preconditioning. The sites of the lattice are decomposed in even and odd sites. Then the fermion matrix can be written in the from

$$M = \begin{pmatrix} 1_{ee} & -\kappa H_{eo} \\ -\kappa H_{oe} & 1_{oo} \end{pmatrix}. \tag{38}$$

For the fermion determinant the identity

$$\det M = \det(1_{ee} - \kappa^2 H_{eo} H_{oe}) \tag{39}$$

holds. Hence the original problem is reduced by half in the dimension (of the fermion determinant). The bosonic field η which is used for the stochastic estimate of the fermion determinant resides only on even sites. The red-black preconditioned fermion matrix is given by

$$M_{ee} = 1_{ee} - \kappa^2 H_{eo} H_{oe}. \tag{40}$$

Most of our tests are done for the single parameter set $\beta = 2.5$ and $\kappa = 0.26$. This set was chosen to compare our results for Wilson loops of various sizes with those recently given by Irving and Sexton [20]. One simulation was performed at $\beta = 2.5$ and $\kappa = 0.266$ to see the effects of going closer to κ_c.

As elementary updates at level 1 we took updates of randomly selected link variables. These link variables are updated by a heat bath which was implemented by a multi-hit Metropolis update. As the criterion to stop the Taylor series of $M^{-1/r}\eta_i$ we used

$$\frac{|c_t H^t \eta_i|^2}{|M^{-1/r}\eta_i|^2} < 10^{-8}, \tag{41}$$

where c_t is the Taylor coefficient of the order t and $M^{-1/r}\eta_i$ is evaluated up to order t.

1. Testing noise reduction

In a first set of numerical experiments we studied the effect of preconditioning and noise reduction on acceptance rates. In order to keep things (conceptually) simple we used only a two-step decomposition ($l=2$) for these studies and made no use of the hopping parameter expansion.

In order to obtain the acceptance rate as a function of the number of link variables that are updated in a single proposal we performed in a single simulation between 200 and 1000 sequences of update cycles, where in each sequence the number of updated links for one proposal runs from 1 to some maximal number. The acceptance rates are collected during the run.

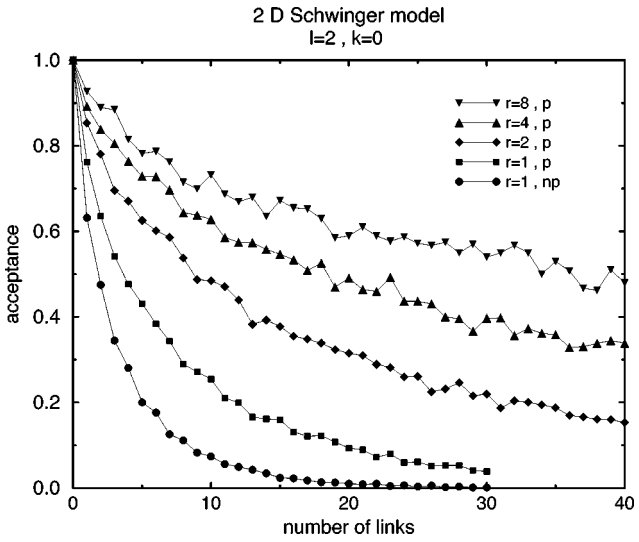

FIG. 1. Acceptance rates as function of the number of link variables that are updated in for the proposal. For details see the text.

The simulations are all performed on a 16×16 lattice. We tested the cases of no preconditioning and $r=1$, and red-black preconditioning in combination with $r=1$, 2, 4 and 8. Our results are given in Fig. 1. We made no effort to compute error bars. The errors should be roughly of the size of the fluctuations of the curves.

Fifty percent acceptance is reached for no preconditioning with $r=1$ at about 2 links updated, while for preconditioning, with $r=1$, 50% acceptance is obtained for about 4–5 links updated. We should note that this amounts to a performance advantage of preconditioning by a factor of 4–5 since the computation of the action requires only half of the operations that are needed in the non-preconditioned case.

With preconditioning and $r=2$, 4, 8 we obtain 50% acceptance with about 9, 20 and 40 links updated. Since the numerical effort of computing the action grows linearly in r, there is no direct performance gain by using the roots of the fermion matrix.

However, as we will see later it is quite useful that a larger number of links can be updated in one update proposal. The roots might also allow for a simple version of parallelization: The application of H could be done independently by one processor for each of the η_i.

We should note of course that the acceptance rate as a function of r is bounded by the acceptance rate that is obtained with the exactly evaluated fermion determinant. Reference [23] however suggests that for lattices of the size that we consider here even for full sweeps over the lattice reasonable acceptance rates are obtained when the fermion determinant is evaluated exactly. We produced almost independent configurations by updating 6400 links (which is several times the number of links of the lattice) in one proposal. We obtained an acceptance rate of 0.39(2).

2. Exploiting the hopping parameter expansion

In a second set of numerical experiments we studied the inclusion of the hopping parameter expansion to order $k=2$ and $k=4$. Again only a two-step decomposition ($l=2$) of the action is used. We performed the test with $r=1$ and $r=4$.

For $r=1$ on the 16×16 lattice and preconditioning we find 50% acceptance for $k=4$ at about 30 links updated, for $k=2$ at about 13 links updated. For $k=0$ we found already above 4–5 links. This means that the performance is increased by a factor more than 6 by using the hopping parameter expansion up to the order $k=4$.

For $r=4$ on the 16×16 lattice and preconditioning we find 50% acceptance for $k=4$ at about 95 links updated, for $k=2$ at about 50 links updated. For $k=0$ we found already above 20 links.

Hence the performance of the algorithm is more then doubled when $k=2$ is used instead of $k=0$ and becomes four-fold for $k=4$.

We did not investigate the inclusion of higher orders of the hopping parameter expansion into the algorithm. The order $k=4$ is accomplished by a shift of the β to $\beta'=\beta+16\kappa^4$.

Since higher orders of the hopping parameter expansion require new terms in the gauge action, there will be a trade-off between the evaluation of these terms and a larger number of links that can be updated with action S_1 at a given acceptance rate.

3. Using a sequence of actions with l>2

Finally for $r=4$, $k=4$ fixed and red-black preconditioning we studied the performance of update cycles with $l>2$.

The first question that arises is how many levels one should choose and how one should optimize the parameters t_i [truncation order of Eq. (22)] and m_i of the cycle, where m_i is the number of updates with the update UP_i at level i to generate the proposal for the accept/reject step of level $i+1$ (see Sec. III B).

The most direct criterion to judge the quality of an algorithm is the product of the square of the statistical error of the observable that is measured multiplied with the CPU time needed to obtain the result. However, this criterion requires one to perform full simulations for each parameter set to be tested. Therefore one would like to have a more practical method to estimate the performance of a given cycle. It seems very natural to assume that the autocorrelation times are proportional to the number of links that have been updated within one cycle. On the other hand the major part of the CPU time is spent with the multiplication of the off-diagonal part of the fermion matrix H on a vector. Both these numbers can be determined with reasonable accuracy from rather short runs (order 10 cycles). Therefore we take the ratio of accepted link updates divided by the number of H times vector applications as performance index (PI) of a cycle.

We made a first attempt to perform the optimization of the cycle parameters automatically. We start from a guess for the parameters t_i and m_i. The PI is computed by averaging over 10 or 20 cycles. In a step of the optimization we propose small random changes of the t_i and m_i to obtain a new proposal t_i' and m_i'. The PI of the t_i' and m_i' is computed by averaging over 10 or 20 cycles. If the PI of the new parameters is larger than the old one, t_i and m_i are replaced by t_i' and m_i'. Typically we performed 200 steps in this procedure. Fortunately it turns out that the performance does not depend very sharply on the parameters of the cycle.

TABLE I. Cycle parameters for the simulations of the 2D Schwinger model at $\beta=2.5$, $\kappa=0.26$ on a 16×16 lattice. For details see the text.

Run	Level	t_i	m_i	Accepted
1	1		120	120
	2	120.5(7)	1	58.2(5)
2	1		220	2640
	2	14	12	993.6(5.7)
	3	124.68(39)	1	782.7(6.9)
3	1		120	4200
	2	6	5	2192.7(5.1)
	3	25	7	1333.6(6.6)
	4	125.35(35)	1	1249.8(7.3)
4	1		230	4600
	2	3	1	2775.2(7.6)
	3	18	20	1318.2(7.3)
	4	125.71(40)	1	1119.2(8.6)

In a first set of experiments we checked the dependence of the performance on the number of levels. We simulated a 16×16 lattice at $\beta=2.5$ and $\kappa=0.26$. The truncation orders t_i and the number of applications m_i are summarized in Table I.

The t_i refer to the expansion of the red-black preconditioned fermion matrix. Hence the corresponding order in the hopping parameter expansion is $2t_i$. The truncation order of the last level is determined by the truncation criterion, Eq. (41). In addition in the last column we give the total number of link updates per cycle that are accepted at a given level. At the first level this is just the total number of link updates per cycle. The number given for the last level is the number of link updates that eventually is accepted in one cycle.

The statistics of the first run ($l=2$) was 20000 cycles, where the first 2000 are discarded in the analysis. The statistics of the other 3 runs ($l>2$) was 10000 cycles each, and the first 500 cycles are discarded.

In Table II we summarize the autocorrelation times of the Wilson loops of sizes 1×1 up to 5×5. In addition we give in the second column the performance index of the run. And in the last column we give the autocorrelation time of the order t_{stop} at which the Taylor series of $M^{-1/r}\eta_i$ is truncated. This number should be strongly correlated with the small eigenvalues of $MM^\dagger$. Comparing the four runs we observe that a larger number of accepted link updates per cycle indeed corresponds to a smaller autocorrelation time. However, comparing runs 1 and 2 we see that the number of accepted link updates increases by a factor of 13.4 but autocorrelation times only decrease by a factor of about 7.

TABLE II. Performance index (PI) and autocorrelation times from simulations of the 2D Schwinger model at $\beta=2.5$ and $\kappa=0.26$ on a 16×16 lattice.

Run	PI	$\tau_{1\times1}$	$\tau_{2\times2}$	$\tau_{3\times3}$	$\tau_{4\times4}$	$\tau_{5\times5}$	τ_{stop}
1	0.0643(6)	7.6(6)	9.6(8)	10.5(9)	8.3(7)	5.6(5)	7.1(5)
2	0.481(5)	1.27(7)	1.41(8)	1.38(7)	1.26(8)	1.01(5)	
3	0.523(3)	0.73(3)	0.81(4)	0.81(4)	0.86(4)	0.72(4)	0.77(5)
4	0.576(5)	1.00(5)	1.16(6)	1.14(6)	0.98(5)	0.90(5)	0.92(10)

Comparing the performance index we see an improvement by a factor of 7.5 from the two-level scheme to the three-level scheme. Going further to four levels one gains about 20% in performance. Here one should note that the imperfection of the optimization of the cycle parameters is a source of uncertainty.

In Table III we give the results for Wilson loops of the size 1×1 up to 5×5 summed over the lattice. In the first line we give for comparison the corresponding results of Ref. [20] which were obtained with the hybrid Monte Carlo algorithm. The results are consistent with each other.

Next we tested the dependence of the performance on the lattice size. Therefore we simulated the 32×32 lattice at $\beta=2.5$ and $\kappa=0.26$ with $r=4$. We tested two different cycles with $l=5$. The parameters of the runs are summarized in Table IV. In Table V we give the PI and the autocorrelation times of the measured observables. The performance index is slightly better than for $L=16$. However, we should note that we still have to take into account the $Volume$ dependence of the cost to apply H on a vector. Therefore we find essentially the expected $Volume^2$ dependence of the cost at fixed β and κ of the algorithm.

The results for Wilson loops of size 1×1 up to 5×5 are summarized in Table VI. The results of our runs are consistent. However, there is some mismatch with the data of Ref. [20]. In particular the value for the 2×2 Wilson loop is by 4.7 standard deviations smaller than the combined result from our simulations. Note that our result for the 32×32 lattice is consistent with the results obtained for the 16×16 lattice.

Finally we performed one run on a 32×32 lattice at $\beta=2.5$ and $\kappa=0.266$ in order to check the dependence of the performance on κ. Note that for $\beta=2.5\kappa_c\approx0.272$ from interpolating the results given in Table 1 of Ref. [24]. The simulation consists of 9000 cycles. The first 500 cycles were discarded from the data analysis. The simulation (which was the most expensive discussed in this section) took about 7 days of CPU on a 200 MHz Pentium Pro PC. The parameters of the cycle are given in Table VII. We see that the number of terms to compute $M^{-1/r}\eta_i$ increases by a factor of 2.28 compared with $\kappa=0.26$.

In Table VIII we give the autocorrelation times of Wilson loops and the autocorrelation time of the truncation order of the Taylor series at the final level. In the first column we give the performance index. The performance index degrades by a factor of 2.77 compared with our best cycle for $\kappa=0.26$ and $L=32$. This means that in addition to the larger costs for evaluating $M^{-1/r}\eta_i$ there is a small degradation of the performance due to reduced acceptance rates.

In Table IX we give the results for Wilson loops of size 1×1 up to 5×5. In particular the values of the large loops are considerably larger than for $\kappa=0.26$.

TABLE III. Results for Wilson loops of size 1×1 up to 5×5 for $\beta=2.5$, $\kappa=0.26$ and $L=16$. The results are summed over the lattice.

Run	$W_{1\times1}$	$W_{2\times2}$	$W_{3\times3}$	$W_{4\times4}$	$W_{5\times5}$
[20]	201.5(2)	105.2(6)	40.5(7)	12.9(6)	3.6(4)
1	201.61(14)	105.48(45)	41.20(61)	13.63(50)	4.34(36)
2	201.57(8)	104.99(24)	40.61(30)	13.34(27)	3.97(21)
3	201.51(6)	105.32(19)	41.08(23)	13.63(22)	4.19(18)
4	201.60(7)	105.35(23)	41.29(28)	13.74(24)	4.29(20)

B. Two flavor QCD with Schrödinger functional boundary conditions

We simulated the standard Wilson gauge action with two flavors of mass-degenerate Wilson fermions. We performed runs at $\beta=8.3$ and $\kappa=0.1386\approx\kappa_c$ on a 8^4 lattice. We applied Schrödinger functional boundary conditions [25–28]. The gauge fields at the boundaries are chosen as specified in Ref. [26] with $c_t=1.0$. The boundary conditions for the fermions are taken as specified in Ref. [28] with $\theta=\pi/5$.

The Schrödinger functional boundary conditions and the particular set of parameters were chosen in order to compare the performance of the algorithm with that of the hybrid Monte Carlo and non-Hermitian versions of the multi-boson algorithm which are benchmarked by the ALPHA Collabo-

TABLE IV. Cycle parameters for the simulations of the 2D Schwinger model at $\beta=2.5$, $\kappa=0.26$ on a 32×32 lattice.

Run	Level	t_i	m_i	Accepted
1	1		200	12000
	2	5	1	5061(12)
	3	20	10	3081(10)
	4	45	6	2778(10)
	5	124.48(19)	1	2749(11)
2	1		150	32400
	2	7	6	14369(21)
	3	20	6	8260(20)
	4	45	6	7184(23)
	5	124.28(14)	1	7104(24)

TABLE V. Performance index (PI) and autocorrelation times from the simulations at $\beta=2.5$ and $\kappa=0.26$ on a 32×32 lattice.

Run	PI	$\tau_{1\times1}$	$\tau_{2\times2}$	$\tau_{3\times3}$	$\tau_{4\times4}$	$\tau_{5\times5}$	τ_{stop}
1	0.604(3)	0.89(6)	1.05(7)	1.13(8)	1.02(6)	0.69(5)	0.97(6)
2	0.703(3)	0.60(2)	0.63(3)	0.63(3)	0.58(3)	0.52(2)	0.59(2)

ration at these parameters. This benchmark is a preparation of a study of the running coupling in the presence of dynamical fermions. Note that $\beta=8.3$ is considerably larger than the β values typically used for spectroscopy.

As observables we have implemented the plaquette, the inverse of the running coupling $\overline{g}^{-2}$ and $\overline{v}$ (for the definitions of these quantities see Ref. [26]). In addition we computed the autocorrelation time of the number of iterations needed for convergence of the Taylor series. This number should be closely related to the smallest eigenvalues of $MM^{\dagger}$.

Biased by the results of the previous section we used red-black preconditioning and the hopping parameter expansion to order $k=4$ for our simulations. To this order the hopping parameter expansion leads to a shift in β:

$$\Delta\beta=96\kappa^4. \tag{42}$$

For our QCD simulations we tried to do better than randomly selecting the links to be updated at level 1 of the update cycle. We sweep through sub-blocks of a certain size in an ordered way. Here one should note that for fixed auxiliary bosonic fields η the action S_l is not gauge invariant. As a consequence even proposals that consist of pure gauge transformations would have an acceptance rate smaller than one. Therefore we fixed the gauge for the elementary (i.e. level 1) updates of the gauge field.

As elementary link updates we used Cabibbo-Marinari heat-bath updating and micro-canonical over-relaxation. In the case of a Cabibbo-Marinari update we performed a sequence of 5 $SU(2)$-subgroup heat-bath updates where the subgroups are given by the (1,2), (2,3), (1,3), (2,3), (1,2) components of the $SU(3)$ matrix. Also in the case of over-relaxation we updated in a sequence of $SU(2)$ subgroups. The sequence is given by the (1,2), (2,3), (1,3) components of the $SU(3)$ matrix.

We tested two different update schemes for the update of the action S_1.

In our first update scheme, which will in the following be referred to as ''point-update,'' we choose with uniform probability a lattice point with $0<t<T$. Then 7 of the links attached to that point are updated as explained below. The 8th link is kept fixed in order to avoid updating of gauge degrees of freedom. We sweep 5 times over the 7 links, where the single link variable is updated by a Cabibbo-Marinari heat-bath update. The order of sweeping through the 7 links is again symmetrized.

Since the links in time direction at the boundaries $t=0$ and $t=T$ do not couple to the fermions, updates of these links come almost for free. Therefore we performed a lexicographic sweep with over-relaxation updating over all links in time direction at the boundaries after half of the point updates are performed. Also here the sequence of the updating is symmetrized. Note that most of the specifications of the details of the update are taken *ad hoc.*

The second scheme that we tested, which will in the following be referred to as ''block update,'' is characterized as follows. First a sub-block of size 4^4 is selected. The position in the spatial directions is chosen randomly with an uniform distribution. In temporal direction the block is either attached to the $t=0$ or the $t=T$ boundary; i.e., the block runs either from $t=1$ to $t=4$ or from $t=4$ to $t=7$ (with equal probability for the two cases).

In order to avoid updating gauge degrees of freedom, only spatial links and temporal links at the boundaries $t=0$ and $t=T$ are updated.

The update sequence for a given sub-block is the following: First a sweep in lexicographic order with a Cabibbo-Marinari update through the spatial links of the sub-block is performed. Then there are 8 over-relaxation sweeps over the spatial links of the sub-block and the temporal links of the sub-block at the boundary. Finally there is a heat-bath sweep over all temporal boundary links. With probability 1/2 the exact reverse of this sequence is performed in order to satisfy detailed balance. Note that in addition to the order of the links also the order of the subgroups taken for the overrelaxation updates has to be reversed.

In addition to the runs with dynamical fermions we performed simulations with the pure gauge action using the two updating schemes discussed above. The idea is that the

TABLE VI. Results for Wilson loops of size 1×1 up to 5×5 at $\beta=2.5$, $\kappa=0.26$ on a 32×32 lattice.

Run	$W_{1\times1}$	$W_{2\times2}$	$W_{3\times3}$	$W_{4\times4}$	$W_{5\times5}$
[20]	805.1(3)	415.7(9)	158.9(11)	51.2(10)	15.1(10)
1	805.79(16)	419.62(52)	162.12(68)	52.79(55)	16.12(40)
2	805.95(14)	420.52(42)	163.83(54)	54.44(48)	17.11(35)

TABLE VII. Cycle parameters for the simulations of the 2D Schwinger model at $\beta=2.5$ and $\kappa=0.266$ on a 32×32 lattice.

Level	t_i	m_i	Accepted
1		150	32400
2	6	1	11654(23)
3	16	6	7355(19)
4	40	6	5073(16)
5	80	6	4377(19)
6	283.5(1.5)	1	4270(22)

TABLE VIII. Performance index (PI) and autocorrelation times from the simulation at $\beta=2.5$ and $\kappa=0.266$ on a 32×32 lattice.

PI	$\tau_{1\times1}$	$\tau_{2\times2}$	$\tau_{3\times3}$	$\tau_{4\times4}$	$\tau_{5\times5}$	τ_{stop}
0.254(13)	0.80(4)	0.85(5)	0.86(5)	0.80(5)	0.59(3)	0.90(7)

TABLE IX. Results for Wilson loops of size 1×1 up to 5×5 at $\beta=2.5$ and $\kappa=0.266$ on a 32×32 lattice. The results are summed over the lattice.

$W_{1\times1}$	$W_{2\times2}$	$W_{3\times3}$	$W_{4\times4}$	$W_{5\times5}$
810.21(14)	435.28(40)	182.76(51)	68.59(47)	25.09(35)

Monte Carlo dynamics of the quenched simulation is very similar to that of the simulation (with our algorithm) of the full theory. The motivation is that the elementary updates are governed by the pure gauge action, while the fermions only enter in the accept/reject steps.

Therefore we could obtain an estimate of the performance of the full algorithm by combining autocorrelation times of the quenched simulation with acceptance rates of the dynamical fermion simulation. Acceptance rates can be obtained reasonably well from rather short runs, while reliable estimates of auto-correlation times require rather good statistics.

$$\tau_{fermions} \approx \frac{n_{acc}}{n_{quenched}} \tau_{quenched}, \tag{43}$$

where n_{acc} is the number of point updates or block updates accepted in an update cycle of the dynamical fermion simulation and $n_{quenched}$ is number of point updates or block updates which is performed per measurement in the quenched simulation.

1. Quenched simulations

First we performed quenched simulations with the two updating schemes discussed above.

For the point update we performed 20 000 measurements. Per measurement 12×360 point updates are performed. For the block update we performed 30 000 measurements. Per measurement we updated 16 4^4 sub-blocks.

In both cases the first 1000 measurements were discarded for the evaluation of the observables and integrated autocorrelation times which are summarized in Table X. We observe that the integrated autocorrelation time of $\overline{v}$ is the largest of the measured autocorrelation times.

2. Simulations with dynamical fermions

First we tested the ''point-updating'' scheme. After some preliminary testing we decided to use $l=6$ for the approximation sequence, the root $r=6$ and the order of the hopping parameter expansion $k=4$. The truncation order and the multiplicities for each level is given in Table XI. In Table XI we give in addition the average total number of point updates that are accepted at a given level during an update cycle. At level 1 this is just the total number of ''point updates'' in one cycle

TABLE XI. Cycle parameters of the dynamical fermion simulation with the ''point-update'' scheme.

Level	t_i	m_i	Accepted
1	Gauge + hopping	360	21600
2	4	1	8844(27)
3	7	1	6178(25)
4	10	10	5386(24)
5	24	6	4675(24)
6	''∞''	1	4548(24)

$$21600 = 360\times1\times1\times10\times6\times1. \tag{44}$$

In order to obtain a reasonably large statistics we used trivial parallelization. We performed 6 independent runs, where the first 400 cycles of each run are discarded for the data analysis. All runs were started from an ordered configuration. Then a small number of update sweeps with the pure gauge action plus the $\Delta\beta$ shift was performed in order to avoid convergence problems of the Taylor series. In total we generated 8450 measurements that were used for the averaging. The simulation took about 135 days of Pentium Pro 200 MHz time, where the generation of the discarded configurations is not taken into account.

The CPU costs of an update cycle are essentially given by the number of applications of the off-diagonal part of the fermion matrix to a vector. For our run on average 4075(3) applications of H per cycle were performed.

The average truncation order of the Taylor series is 98.5(1). The autocorrelation time of the (over the 6 replicas averaged) truncation order is $\tau_{stop}=2.4(2)$. Results for the observables are given in Table XII.

First we note that the results for the observables are consistent with those obtained by Wolff with the multi-boson algorithm [21]. As for the quenched simulation the autocorrelation time of the plaquette is the smallest while that of $\overline{v}$ is the largest. It is even considerably larger than τ_{stop}. Next we checked our hypothesis that the autocorrelation times can be predicted from autocorrelation times of the quenched run in combination with acceptance rates of the dynamical fermion simulation. The numbers of Table X have to be multiplied by 4320/4548. Then we obtain the estimates $\tau_{p,est}=0.67(3)$, $\tau_{g,est}=2.49(12)$ and $\tau_{v,est}=7.8(9)$ which are in quite good agreement with the directly measured results given in Table

TABLE X. Results for the average plaquette (Plaq), the running coupling $\overline{g}^{-2}$, $\overline{v}$ and the corresponding autocorrelation times of the quenched runs. For details see the text.

Update	Plaq	τ_p	$\overline{g}^{-2}$	τ_g	$\overline{v}$	τ_v
Point	0.73963(1)	0.71(3)	0.6814(51)	2.62(13)	0.075(11)	8.2(9)
Block	0.73963(1)	1.36(4)	0.6742(31)	1.52(4)	0.062(7)	3.8(2)

TABLE XII. Results for the average plaquette (Plaq), the running coupling $\overline{g}^{-2}$, $\overline{v}$ and the corresponding autocorrelation times of the dynamical fermion simulation with the ''point-update'' scheme. For details see the text.

Plaq	τ_p	$\overline{g}^{-2}$	τ_g	$\overline{v}$	τ_v
0.742746(18)	0.83(4)	0.7157(73)	2.47(20)	0.096(16)	6.9(9)

XII. The ALPHA Collaboration intends to compute the running coupling of QCD with dynamical fermions. Therefore the comparison of the algorithms which are candidates for this project is based on the statistical error of $\overline{g}^2$. Within the ALPHA Collaboration we agreed on the following cost definition:

$$D_{cost} := (\text{total number of applications of } H) \times \text{error}[\overline{g}^{-2}]^2. \tag{45}$$

For the run with the ''point update'' we obtain

$$D_{cost} = 1835(150) + 181(15), \tag{46}$$

where the error of D_{cost} is derived from the statistical error of τ_g. The second contribution comes from converting the updating costs at level 1. The cost of the Cabibbo-Marinari updates at level 1 have been converted to units of H application and are given by the second number.

In a similar way as for the autocorrelation times we tried to estimate the D_{cost} from the statistical error of $\overline{g}^{-2}$ in the quenched simulation and the total acceptance in the dynamical fermion simulation. We obtain

$$D_{cost,est} = 1913(95) + 189(9), \tag{47}$$

which is again in good agreement with the directly obtained result.

Next we studied the ''block-update'' scheme. Here we also used $l=6$, $r=6$ and $k=4$. The truncation orders and the multiplicities are summarized in Table XIII. We performed only a run of 100 cycles that was started from an equilibrium configuration. The total acceptances are also given in Table XIII. On average a cycle of the update requires 4836(16) applications of the off-diagonal part of the fermion matrix. The 100 measurements are not sufficient to produce reliable estimates of the autocorrelation times and of the statistical error of $\overline{g}^{-2}$. Therefore we only quote the estimated D_{cost}:

TABLE XIII. Cycle parameters of the dynamical fermion simulation with the ''block-update'' scheme.

Level	t_i	m_i	Accepted
1	Gauge + hopping	1	60
2	4	1	34.3(5)
3	8	3	25.7(4)
4	14	4	22.2(4)
5	25	5	20.1(5)
6	''∞''	1	19.9(6)

$$D_{cost,est} = 1084(50) + 62(3). \tag{48}$$

We conclude that the ''block update'' is more efficient than the ''point update.''

Our D_{cost} results can be compared with the other two algorithms benchmarked by the ALPHA Collaboration:

$$D_{cost} \approx 900(30) \tag{49}$$

was obtained as preliminary result by Wolff with the non-Hermitian version of the multi-boson algorithm with reweighting [21] and

$$D_{cost} \approx 460 \tag{50}$$

was obtained by Jansen [22] with the polynomial hybrid Monte Carlo algorithm [29]. In the last case only H applications have been counted. Additional costs have not been converted into D_{cost}.

We conclude that for a 8^4 system we already have reached similar efficiency as the multi-boson and polynomial hybrid Monte Carlo algorithms. However, at least with the present implementation, we have to expect a worse scaling of the efficiency with the lattice size as for the multi-boson and the hybrid Monte Carlo algorithms. Therefore further progress is needed to obtain a competitive algorithm for lattice sizes that are needed for the calculation for the running coupling or light hadron spectroscopy.

V. COMPARISON WITH RELATED APPROACHES

Irving and Sexton [20] use approximations of the fermion determinant to simulate the 2D Schwinger model and QCD. They try to approximate the fermion action by adding Wilson loops of different sizes and shapes to the gauge action. Among other things they propose to construct an exact algorithm by generating a proposal by using the approximate action. They have in mind a two-level Metropolis scheme. However, also in their context one could think of a sequence of approximations by incorporating more and more Wilson loops.

In a recent paper Duncan, Eichten, and Thacker [30] propose to split the fermion determinant into two parts. One part is given by the product of the smallest eigenvalues. They suggest that the remaining part can be expressed in terms of small Wilson loops. In their simulation they approximated the fermion determinant just by the product of the smallest eigenvalues. They argue that the remaining part of the fermion determinant effectively amounts to a β-shift.

In the same spirit one could simulate with a $\tilde{M}$ that corresponds to a moderate order of the hopping-parameter expansion and ignore the small orders of the hopping parameter expansion.

Thron *et al.* [31] estimate the fermion determinant stochastically. Their method to reduce the noise of their estimator incorporates, as $\tilde{M}$ in this paper, the hopping parameter expansion.

VI. OUTLOOK

There are several directions in which progress could be made.

First one could try to use higher orders of the hopping-parameter expansion. In the present paper we only considered the hopping parameter expansion up to order $k=4$. This leads only to a shift in β. At order $k=6$, in 4 dimensions, Wilson loops of three different shapes contribute. These were implemented for example in Ref. [31]. Going to even higher orders, the number of Wilson loops needed grows exponentially in the order. Therefore one has to look for a more efficient method to compute $\mathrm{tr}H^k$ than expressing it in terms of Wilson loops. See Ref. [32].

One might find noise reduced unbiased estimators of the fermion determinant that are less CPU intensive than those discussed in Sec. II B of this paper.

Finally one might look at approximation schemes different from that discussed in this paper. In this paper the Taylor series of the fermion matrix in κ is used as the basis of the approximation scheme. One might construct approximations of the action by truncating M^{-1} in real space, i.e. allowing only non-zero matrix elements $(M^{-1})_{xy}$ with $|x-y|<d$. The quality of the approximation is then controlled by d.

ACKNOWLEDGMENTS

I would like to thank the ALPAH Collaboration for providing FORTRAN90 code for pure QCD with Schrödinger functional boundary conditions (Stefan Sint, Stefano Capitani), performance data for the multi-boson and hybrid Monte Carlo algorithms (Ulli Wolff, Karl Jansen) and finally for discussions with all participants of the tuesday seminar in our group at Humboldt University, Klaus Pinn and Stefano Vinti.

[1] S. Duane, A. D. Kennedy, B. J. Pendleton, and D. Roweth, Phys. Lett. B **195**, 216 (1987).
[2] S. Aoki, G. Boyd, R. Burkhalter, S. Hashimoto, N. Ishizuka, Y. Iwasaki, K. Kanaya, T. Kaneko, Y. Kuramashi, M. Okawa, A. Ukawa, and T. Yosie, Nucl. Phys. B (Proc. Suppl.) **63A–C**, 221 (1998).
[3] Th. Lippert, G. Bali, N. Eicker, L. Giusti, U. Glässner, S. Güsken, H. Hoeber, G. Martinelli, F. Rapuano, G. Ritzenhöfer, K. Schilling, A. Spitz, and J. Viehoff, Nucl. Phys. B (Proc. Suppl.) **63A–C**, 946 (1998); N. Eicker, U. Glässner, S. Güsken, H. Hoeber, P. Lacock, Th. Lippert, K. Schilling, A. Spitz, Th. Struckmann, P. Ueberholz, and J. Viehoff, Phys. Rev. D **59**, 014509 (1998).
[4] M. Talevi, Nucl. Phys. B (Proc. Suppl.) **63A–C**, 227 (1998); Z. Sroczynski, S. M. Pickles, and S. P. Booth, *ibid.* **63A–C**, 949 (1998).
[5] R. D. Mawhinney, Nucl. Phys. B (Proc. Suppl.) **63A–C**, 212 (1998).
[6] B. Alles, G. Boyd, M. D'Elia, A. Di Giacomo, and E. Vicari, Phys. Lett. B **389**, 107 (1996).
[7] B. Alles, G. Bali, M. D'Elia, A. Di Giacomo, N. Eicker, S. Guesken, H. Hoeber, Th. Lippert, K. Schilling, A. Spitz, T. Struckmann, P. Ueberholz, and J. Viehoff, Phys. Rev. D **58**, 071503 (1998).
[8] M. Lüscher, Nucl. Phys. **B418**, 637 (1994).
[9] B. Bunk, K. Jansen, B. Jegerlehner, M. Lüscher, H. Simma, and R. Sommer, Nucl. Phys. B (Proc. Suppl.) **42**, 49 (1995).
[10] C. Alexandrou, A. Borrelli, P. de Forcrand, A. Galli, and F. Jegerlehner, Nucl. Phys. **B456**, 296 (1995).
[11] A. Borrelli, P. de Forcrand, and A. Galli, Nucl. Phys. **B477**, 809 (1996).
[12] D. Weingarten and D. Petcher, Phys. Lett. **99B**, 333 (1981).
[13] F. Fucito, E. Marinari, G. Parisi, and C. Rebbi, Nucl. Phys. **B180**, 369 (1981).
[14] A. D. Kennedy and J. Kuti, Phys. Rev. Lett. **54**, 2473 (1985).
[15] A. D. Kennedy, J. Kuti, S. Meyer, and B. J. Pendleton, Phys. Rev. D **38**, 627 (1988).
[16] R. Horsley, A. D. Kennedy, S. Meyer, and B. J. Pendleton, Z. Phys. C **44**, 445 (1989).
[17] N. Cabibbo and E. Marinari, Phys. Lett. **119B**, 387 (1982).
[18] S. L. Adler, Phys. Rev. D **23**, 2901 (1981).
[19] R. Petronzio and E. Vicari, Phys. Lett. B **245**, 581 (1990).
[20] A. C. Irving and J. C. Sexton, Phys. Rev. D **55**, 5456 (1997).
[21] U. Wolff (private communication).
[22] K. Jansen (private communication).
[23] C. B. Lang and T. K. Pany, Nucl. Phys. **B513**, 645 (1998).
[24] C. R. Gattringer, I. Hip, and C. B. Lang, Nucl. Phys. **B508**, 329 (1997).
[25] M. Lüscher, R. Narayanan, P. Weisz, and U. Wolff, Nucl. Phys. **B384**, 168 (1992).
[26] M. Lüscher, R. Sommer, P. Weisz, and U. Wolff, Nucl. Phys. **B413**, 481 (1994).
[27] S. Sint, Nucl. Phys. **B421**, 135 (1994); **B451**, 416 (1995).
[28] Martin Lüscher, Stefan Sint, Rainer Sommer, Peter Weisz, and Ulli Wolff, Nucl. Phys. **B491**, 323 (1997).
[29] R. Frezzotti and K. Jansen, Phys. Lett. B **402**, 328 (1997).
[30] A. Duncan, E. Eichten, and H. Thacker, Phys. Rev. D **59**, 014504 (1998).
[31] C. Thron, S. J. Dong, K. F. Liu, and H. P. Ying, Phys. Rev. D **57**, 1642 (1998).
[32] I. Montvay, Phys. Lett. **139B**, 70 (1984).