Exploiting the hopping parameter expansion in the hybrid Monte Carlo simulation of lattice QCD with two degenerate flavors of Wilson fermions

Martin Hasenbusch[*]

Institut für Physik, Humboldt-Universität zu Berlin, Newtonstr. 15, 12489 Berlin, Germany

We show how the hopping parameter expansion at order κ^2 and κ^4 can be exploited in the simulation of lattice QCD with two flavors of degenerate Wilson fermions. A natural extension of this idea is a "UV filtering" by using rooted polynomials. These approaches can be easily combined with, for example, mass preconditioning. First numerical tests are performed for the Wilson gauge action at $\beta = 5.6$ and $\kappa = 0.156$ and 0.1575.

I. INTRODUCTION

Lattice QCD simulations are our primary tool to obtain nonperturbative results from QCD. To this end, a fair share of CPU time on the largest supercomputers that are available today is used. Still we would not mind getting more accurate results from such simulations. Hence any algorithmic progress is highly desirable. Here I shall address the generation of the gauge field.

In order to fix the notation, let us briefly recall the definition of lattice QCD. It is defined on a four-dimensional hypercubic lattice. On the links of the lattice there are matrices $U_{x,\mu} \in SU(3)$, representing the gluon field, where x denotes a site of the lattice and $\mu \in \{0, 1, 2, 3\}$ labels the directions. The fields that represent the fermions live on the sites. These fields assume Grassmanian values. The interactions are encoded by the Euclidian action,

$$S[U, \psi, \bar{\psi}] = S_G[U] + S_F[U, \psi, \bar{\psi}]. \quad (1)$$

The fermion fields ψ, $\bar{\psi}$ appear in bilinear form, and therefore can be integrated out exactly in the partition function Z. It remains an integral over the gauge field only:

$$Z = \int D[U] \exp(-S_G[U]) \prod_{f=1}^{n} \det M_f[U], \quad (2)$$

where $M_f[U]$ is the fermion matrix and the product runs over the flavors of the quarks. In the literature different types of fermion actions are discussed. In the following we shall consider two degenerate flavors of Wilson fermions. The fermion matrix is given by

$$M = \mathbb{1} - \kappa H, \quad (3)$$

where

$$H_{xy} = \sum_{\mu} \{(1 - \gamma_\mu) U_{x,\mu} \delta_{x+\hat{\mu},y} + (1 + \gamma_\mu) U^{\dagger}_{x-\hat{\mu},\mu} \delta_{x-\hat{\mu},y}\} \quad (4)$$

is the hopping matrix and the hopping parameter κ is a real number. The Wilson plaquette action is given by

$$S_G[U] = -\frac{\beta}{3} \sum_{x} \sum_{\mu > \nu} \text{ReTr}(U_{x,\mu} U_{x+\hat{\mu},\nu} U^{\dagger}_{x+\hat{\nu},\mu} U^{\dagger}_{x,\nu}), \quad (5)$$

where $\hat{\mu}$ is a unit vector in the μ-direction. For a more detailed discussion, see for example the textbooks and review articles [1–4].

For lattice sizes that are needed to extract continuum physics from the simulation, it is by far too expensive to evaluate the determinant of the fermion matrix exactly. Therefore, following the proposal of Weingarten and Petcher [5], in the case of two degenerate flavors, one introduces auxiliary degrees of freedom, so-called pseudofermions:

$$\det M^{\dagger} M \propto \int D[\phi] D[\phi^{\dagger}] \exp(-S_{PF}), \quad (6)$$

where

$$S_{PF} = |M^{-1}\phi|^2, \quad (7)$$

where ϕ is a vector with complex components. Hence the action, as a function of the gauge field and the pseudofermion fields, is given by

---

[*]Martin.Hasenbusch@physik.hu-berlin.de

$$S(U, \phi) = S_G(U) + S_{PF}(U, \phi). \tag{8}$$

Still the pseudofermion action is nonlocal and the evaluation requires the solution of a system of linear equations. The nonlocality is in particular a problem for local algorithms that are used to simulate the pure gauge action. The hybrid Monte Carlo (HMC) algorithm [6] is better adapted to this situation, since all gauge degrees of freedom evolve simultaneously. To this end, an artificial Hamiltonian is introduced:

$$H = \frac{1}{2}(\Pi, \Pi) + S(U, \phi), \tag{9}$$

where the anti-Hermitian momenta $\Pi_{x,\mu}$ are conjugate to the gauge field $U_{x,\mu}$. They are auxiliary variables that are solely introduced for algorithmic reason. Their scalar product is defined as

$$(\Pi, \Pi) = -2 \sum_{x,\mu} \mathrm{Tr} \Pi_{x,\mu}^2. \tag{10}$$

Here we follow the convention of, for example, Ref. [7]. Note that in the literature often the factor 2 is omitted in the definition of the scalar product, see e.g., Ref. [8]. Note that this leads to a relative factor $\sqrt{2}$ in the fictitious Monte Carlo time τ that is introduced below. A discussion of this point is given in Ref. [9], below Eq. (3.2). The momenta and the gauge field evolve according to the equations of motion,

$$\frac{\mathrm{d}}{\mathrm{d}\tau}\Pi_{x,\mu} = -\mathcal{F}_{x,\mu} \quad \text{and} \quad \frac{\mathrm{d}}{\mathrm{d}\tau}U_{x,\mu} = \Pi_{x,\mu}U_{x,\mu}, \tag{11}$$

where τ is the fictitious Monte Carlo time and the force $\mathcal{F}$ fulfills $(\omega, \mathcal{F}) = \delta_\omega S(U)$ for infinitesimal variations of the gauge field $\delta_\omega U_{x,\mu} = \omega_{x,\mu}U_{x,\mu}$. Here we consider the so-called ϕ-algorithm [10], where the pseudofermions stay fixed during the evolution of the gauge field and the momenta.

The equations of motion (11) cannot be integrated exactly. Therefore a numerical integration scheme with a finite step-size is used. This leads to an integration error. The idea of the HMC algorithm [6] is that this error can be corrected for by a Metropolis accept/reject step.

One update cycle (or trajectory) of the HMC is composed of the following three steps:

(i) Perform a heat bath for both the conjugate momenta Π and the pseudofermion field ϕ. In the case of the pseudofermion field one generates a field η with a Gaussian distribution $P(\eta) \propto \exp(-|\eta|^2)$ and then

$$\phi = M\eta. \tag{12}$$

Evaluate the Hamiltonian

$$H(U, \phi, \Pi) = S_G(U) + |\eta|^2 + \frac{1}{2}(\Pi, \Pi) \tag{13}$$

and save the initial gauge configuration U.

(ii) Keeping ϕ fixed, we evolve the gauge field U and the conjugate momenta Π according to the classical equations of motion for the fictitious time τ. Since this cannot be done exactly, a numerical integration scheme with the finite step-size $\delta\tau$ is used. At the end of the integration we have the fields U', Π', and $\phi' = \phi$. For a detailed discussion of the integration scheme see below.

(iii) Accept U' as the new gauge field with the probability

$$P_{\text{acc}} = \min[1, \exp(-\Delta H)], \tag{14}$$

where

$$\Delta H = H(U', \phi, \Pi') - H(U, \phi, \Pi) \tag{15}$$

else we keep U.

In order to fulfill detailed balance, the numerical integration scheme has to be area preserving and reversible. Reversible means that changing the sign of the momenta at the end of the integration time, we run back exactly to the initial gauge field U. Such integration schemes are called symplectic integrators. Let us introduce a shorthand for finite update steps by $\delta\tau$:

$$P(\delta\tau): \Pi_{x,\mu} \to \Pi'_{x,\mu} = \Pi_{x,\mu} + \delta\tau \mathcal{F}_{x,\mu}, \tag{16}$$

$$T(\delta\tau): U_{x,\mu} \to U'_{x,\mu} = \exp(i\delta\tau \Pi_{x,\mu})U_{x,\mu}. \tag{17}$$

Integrators are built from these basic steps. Here we consider the second order Omelyan integrator [11],

$$T_O = P(\lambda\delta\tau)T(\delta\tau/2)P([1 - 2\lambda]\delta\tau)T(\delta\tau/2)P(\lambda\delta\tau), \tag{18}$$

where we get for $\lambda = 1/6$ the scheme proposed in Ref. [12], which is also discussed for example in Ref. [13]. A trajectory of length τ is given by T_O^m with $\tau = m\delta\tau$. Taking $\lambda = 1/2$ the expression (18) simplifies to the well-known leapfrog scheme:

$$T_L = P(\delta\tau/2)T(\delta\tau)P(\delta\tau/2). \tag{19}$$

In our simulations, we use both the leapfrog and the Omelyan scheme with $\lambda = 1/6$. Sexton and Weingarten proposed a multilevel integration scheme [12]. Each level is associated with a term in the action. For example, in Eq. (8), we can associate the gauge action with level $i = 0$ and the pseudofermion action with $i = 1$. For each level a time step $\delta\tau_i = 2m_{i-1}\delta\tau_{i-1}$ is defined. The scheme can be iteratively defined:

$$T_{SW,i} = P_i(\lambda\delta\tau_i)[T_{SW,i-1}]^{m_{i-1}} P_i([1-2\lambda]\delta\tau_i)[T_{SW,i-1}]^{m_{i-1}}$$
$$\times P_i(\lambda\delta\tau_i) \qquad (20)$$

and

$$T_{SW,0} = P_0(\lambda\delta\tau_0)T(\delta\tau_0/2)P_0([1-2\lambda]\delta\tau_0)T(\delta\tau_0/2)$$
$$\times P_0(\lambda\delta\tau_0). \qquad (21)$$

Note that for the leapfrog scheme, we use the convention $\delta\tau_i = m_{i-1}\delta\tau_{i-1}$, which is more natural in this case. For a nice discussion of this scheme see for example Sec. 2.2 of Ref. [14]. The scheme can be generalized even further. The parameter λ might depend on the level i. Or me might use a fourth order scheme at low levels and a second order scheme at higher levels. An important property of symplectic integrators is that they preserve a so-called shadow Hamiltonian. Here we will not delve into this discussion but refer the reader to Refs. [13,15] and references therein.

Applying the HMC algorithm to the pseudofermion action (7), two problems are encountered: Going to lighter quark masses, sending κ to κ_c, the condition number of the fermion matrix increases. As a result, for iterative solvers like the conjugate gradient (CG) or the biconjugate gradient stabilized method (BiCGstab) [16,17], the number of iterations needed to solve the system of linear equations is increasing. The second problem is less obvious. It turns out that, in order to keep the acceptance rate fixed, the step-size of the integration scheme has to be reduced with decreasing quark mass. At the Lattice 2001 conference, which took place in Berlin, the situation was referred to as "Berlin wall." At the time, it seemed impossible to reach sufficiently small masses, to reliably extrapolate, by using chiral perturbation theory, to the physical mass of the pion.

The situation considerably improved by the advent of better solvers, for example [18,19], and by replacing the pseudofermion action (7) by better alternatives. Note that the representation of the fermion determinant by pseudofermions is not unique. In [20] a large number of pseudofermion fields were introduced, allowing to express the fermion determinant in terms of a local pseudofermion action. This approach did not outperform the HMC algorithm in the end. It turned out that the large number of fields implicate that only small steps can be performed in the update. An alternative approach to local updating, which also did not outperform the HMC, is discussed in Ref. [21]. See also [22] and references therein.

Based on this experience, alternatives to Eq. (7), to be used in HMC simulations, were proposed. These are primarily mass preconditioning [23,24], domain decomposition [7,25], and rooting [26]. The basic idea behind these approaches is to split the fermion matrix M into (several) factors, and introduce a separate pseudofermion field for each of the factors. By using a suitable factorization, the stochastic estimate of the fermion determinant

becomes less noisy, allowing for a larger step-size in the integration scheme. A second potential advantage is that different parts of the pseudofermion action can be put on different timescales of the integration scheme [12]. In the ideal case, the numerically most expensive parts can be put on large timescales.

In the case of a finite step updating scheme [21], the multiboson (MB) algorithm [20] and the polynomial hybrid Monte Carlo (PHMC) algorithm [27–29], it has been shown that the updating scheme becomes more efficient by incorporating the hopping parameter expansion [30–32]. The hopping parameter expansion, taken at a low order, is used as UV filter for the pseudofermion action. Here we demonstrate how this can efficiently be done for the HMC algorithm applied to two degenerate flavors. Compared with the simulation using the pseudofermion action (7) we get a speed-up of a factor of 2 or 3, depending on the order of the hopping parameter expansion. In large scale simulations, this idea can be combined with mass preconditioning [23] and might allow to reduce the number of auxiliary fermion matrices by one. Furthermore, we give a preliminary discussion of UV filtering by using rooted polynomials. The motivation is similar to Ref. [26] and could also be seen as a natural extension of the UV filtering by using the hopping parameter expansion.

The outline of the paper is the following. In the next section we discuss in detail how the hopping parameter expansion is used as UV filter. Then we discuss how this idea can be naturally extended by using polynomial approximations of the rooted inverse of the fermion matrix. We briefly summarize results on the acceptance rate, the variance of ΔH and the forces that are given in the literature. Then in Sec. IV we discuss our numerical results. First we study the effect of UV filtering by using the hopping parameter expansion up to the orders κ^2 and κ^4. Then we present our still very preliminary results on rooted polynomials. Finally we give a summary and an outlook.

II. INCORPORATING THE HOPPING PARAMETER EXPANSION INTO THE HYBRID MONTE CARLO SIMULATION

In the case of two degenerate flavors, the fermion determinant can be expressed as

$$\det M^\dagger M = \exp(\text{Tr}\ln M^\dagger + \text{Tr}\ln M), \qquad (22)$$

where one expands

$$\ln M = \ln(1-\kappa H) = -\sum_{n=1}^{\infty}\frac{1}{n}\kappa^n H^n. \qquad (23)$$

For small values of n, $\text{Tr}H^n$ can be evaluated analytically. In the case of Wilson fermions, terms with odd values of n do not contribute. Furthermore $n = 2$ also does not

contribute. The leading nonvanishing contribution $\mathrm{Tr}H^4$ amounts to a plaquette term. This can be written as a shift of the parameter β. In the case of two degenerate Wilson fermions one gets $\Delta\beta = 96\kappa^4$. For $n = 6$ we get contributions from three different Wilson loops. With increasing n, the number of Wilson loops that contribute rapidly increases and things become intractable. For a more detailed discussion see Sec. III of Ref. [33]. In the case of clover-improved Wilson fermions the situation is worse. There is already a nonvanishing contribution for $n = 2$, see Eq. (2.8) of Ref. [32]. Already $n = 4$ was not considered in Ref. [32], since it is too involved.

In the simulation we consider a modified gauge action,

$$\tilde{S}_G = S_G + 2\sum_{n=1}^{k}\frac{1}{n}\kappa^n\mathrm{Tr}H^n, \qquad (24)$$

where k is the order, up to which $\mathrm{Tr}H^n$ in terms of Wilson loops is tractable in the simulation.

In Ref. [21] we discussed preconditioning by using the hopping parameter expansion in the context of a finite step updating scheme. To this end the value of the pseudofermion action has to be evaluated. Following Eq. (8) of Ref. [21] a modified fermion matrix is introduced by

$$\tilde{M} = M\exp\left(\sum_{n=1}^{k}\frac{1}{n}\kappa^n H^n\right), \qquad (25)$$

and correspondingly

$$\tilde{S}_{PF} = |\tilde{M}^{-1}\phi|^2. \qquad (26)$$

The idea is that $\tilde{S}_{PF}$ fluctuates less than S_{PF} and hence allows for a larger step-size in the HMC simulation. In Ref. [21] we evaluated $\tilde{M}^{-1}\phi$ by using the series expansion of $\tilde{M}^{-1}$ in κH. Also in the case of the MB algorithm [30,31] and the PHMC algorithm [32] it is natural to represent $\tilde{M}$ by using a polynomial in M.

Here we discuss an alternative representation that is more suitable for the HMC algorithm applied to two degenerate fermion flavors. In particular, we express $\tilde{M}^{-1}$ essentially in terms of M^{-1} to make use of iterative solvers to compute $\tilde{M}^{-1}\phi$. For simplicity, let us first discuss the case $k = 1$. The series expansion of the inverse of $\tilde{M}$ in κH is given by

$$\tilde{M}^{-1} = \exp(-\kappa H)(1-\kappa H)^{-1} = \sum_{n=0}^{\infty}a_n\kappa^n H^n. \qquad (27)$$

Since all coefficients of the expansion of M^{-1} are equal to one, we can easily evaluate the coefficients

$$a_n = \sum_{l=0}^{n}(-1)^l\frac{1}{l!}, \qquad \lim_{n\to\infty}a_n = \exp(-1). \qquad (28)$$

Hence we can write

$$\tilde{M}^{-1} = \sum_{n=0}^{\infty}b_n\kappa^n H^n + \alpha M^{-1}, \qquad (29)$$

where $\alpha = \exp(-1)$ and

$$b_n = -\sum_{l=n+1}^{\infty}(-1)^l\frac{1}{l!}. \qquad (30)$$

Since b_n rapidly converges to 0, the sum

$$\sum_{n=0}^{\infty}b_n\kappa^n H^n \qquad (31)$$

can be truncated at a low order $n_{\max}$. For larger values of k we get a similar result, where $\alpha = \exp(-\sum_{n=1}^{k}1/n)$. The coefficients b_n can be evaluated by using an algebra program like MAPLE or MATHEMATICA. Note that the coefficients in Eq. (24) are tunable parameters of the algorithm. Previous experience [21,32] however shows that taking the values given by the hopping parameter expansion is a good choice. Here we will not further discuss this question.

Now let us discuss how the HMC algorithm can be implemented for $\tilde{M}^{-1}$. The crucial question is how the forces can be computed. Here we have to put together the results obtained for the HMC algorithm and the PHMC algorithm. The variation of the pseudofermion action S_{PF} with respect to the gauge field can be computed as

$$\delta S_{PF} = -X^\dagger\delta M Y + \mathrm{H.c.}, \qquad (32)$$

where

$$X = (MM^\dagger)^{-1}\phi, \qquad Y = M^{-1}\phi. \qquad (33)$$

The variation of the polynomial has been worked out in Refs. [28,29]. We follow the implementation of Refs. [29,32] using Horner's scheme. Here we need the variation of $\tilde{S}_{PF} = |\tilde{M}^{-1}\phi|^2$ with

$$\tilde{M}^{-1} = \alpha M^{-1} + \sum_{n=0}^{n_t}b_n\kappa^n H^n. \qquad (34)$$

Note that we are free to take $n_t < n_{\max}$, since the truncation error introduced is corrected for in the accept/reject step, where the summation is performed up to $n_{\max}$. We get

$$\delta\tilde{S}_{PF} = \phi^\dagger\tilde{M}^{-1\dagger}\delta\tilde{M}^{-1}\phi + \mathrm{H.c.}, \qquad (35)$$

where

$$\delta\tilde{M}^{-1}\phi = \left[-\alpha M^{-1}\delta M M^{-1} + \sum_{n=1}^{n_t} b_n \kappa^n \delta(H^n)\right]\phi, \quad (36)$$

where

$$\delta(H^n) = \sum_{l=1}^{n} H^{l-1}\delta H H^{n-l}. \quad (37)$$

In order to compute the variation for the polynomial efficiently, n_t vectors have to be precomputed, following Horner's scheme:

$$Y_{n_t} = b_{n_t}\phi \quad (38)$$

and then recursively

$$Y_{l-1} = b_{l-1} + \kappa H Y_l \quad (39)$$

down to

$$Y_0 = \left[\sum_{l=0}^{n_t} b_l \kappa^l H^l\right]\phi. \quad (40)$$

Then we compute recursively

$$X_1 = Y_0 + \tilde{Y}, \quad \text{where } \tilde{Y} = \alpha M^{-1}\phi \quad (41)$$

and

$$X_l = \kappa H X_{l-1}. \quad (42)$$

The variation of the pseudofermion action can be written as

$$\delta\tilde{S}_{PF} = \kappa\tilde{X}^\dagger \delta H \tilde{Y} + \kappa\sum_{l=1}^{N} X_l^\dagger \delta H Y_l + \text{H.c.}, \quad (43)$$

where $\tilde{X} = M^{-1}X_1$.

In the following we refer to exploiting the hopping parameter expansion up to order κ^k as κ^k-filtering.

A. Rooted polynomials

In our simulations we make use of the hopping parameter expansion up to κ^4. It is practically impossible to push the hopping parameter expansion to higher order. Therefore, with a similar motivation as Ref. [26], where the rational HMC is considered, we propose to use polynomials that approximate $M^{-1/N}$ as UV filters. Introducing a pseudo-fermion field for each factor $M^{-1/N}$, the pseudofermion action, Eq. (7), is replaced by

$$S_{PF,N} = \sum_{l=1}^{N} |M^{-1/N}\phi_l|^2. \quad (44)$$

In the limit $N \to \infty$, as discussed in Sec. II. B of Ref. [21], $S_{PF,N}$ converges, up to an additive constant, to $\text{Tr}\ln M^\dagger + \text{Tr}\ln M$. This means that for sufficiently large N, we can approximate the hopping parameter expansion by using low order polynomials that approximate $M^{-1/N}$. Let us define $M_1 = \tilde{M}$, Eq. (25), and then recursively

$$M_{j+1} = W_j^{-N_j} M_j \quad (45)$$

up to some maximal $j_{\max}$, where

$$W_j^{-1} = \sum_{l=0}^{n_j} a_{j,l}\kappa^l H^l = M_j^{-1/N_j} + \mathrm{O}(\kappa^{n_j+1}), \quad (46)$$

where $n_j > n_{j-1}$. The remainder can be written as

$$M_{j_{\max}+1}^{-1} = \sum_{n=0}^{\infty} b_n \kappa^n H^n + \alpha M^{-1}, \quad (47)$$

where b_n and α are computed by using an algebra program. The corresponding pseudofermion action is

$$S_{PF,\text{Polynom}} = \sum_{j=1}^{j_{\max}} S_{PF,j} + |M_{j_{\max}+1}^{-1}\phi_{j_{\max}+1}|^2, \quad (48)$$

where

$$S_{PF,j} = \sum_{l=1}^{N_j} |W_j^{-1}\phi_{j,l}|^2. \quad (49)$$

The construction proposed here contains both the noise reduction by rooting as proposed in [21,26] as well as a hierarchical splitting similar to mass preconditioning. Note that a hierarchical splitting, in the framework of the PHMC, was already discussed in Refs. [34–36].

In particular, aiming at the application to a single flavor, one would like to investigate how well $M_{j_{\max}+1}^{-1/N}$ can be approximated by a rational approximation. Also it might be feasible to compute $\det M_{j_{\max}+1}$, without using a noisy estimator, since likely only a few smallest eigenvalues of M contribute. In this case, it might be sufficient to compute $\det M_{j_{\max}+1}$ in the accept/reject step only.

In our numerical tests we have used $j_{\max} = 2$ and $N_1 = N_2 = N$ for simplicity. The general framework contains a large number of free parameters that are hard to tune without having a theoretical understanding of the dependence of the acceptance rate on these parameters. Reference [13] and possible extensions might be helpful to this end.

In the case of the pseudofermion action (7) it is simple to perform a heat-bath update, Eq. (12), of the pseudofermions at the beginning of the trajectory. The fermion matrix M has to be applied to a vector with a Gaussian distribution.

In the case of the rooted polynomials the numerical costs are considerably larger, since W_j has to be represented by a high order polynomial in M or equivalently H. In our preliminary study, we implemented the heat-bath update of the pseudofermions associated with W_j in the straightforward way. A more efficient solution is provided by Ref. [37], where only a good approximation of W_j is needed to update the pseudofermions.

B. Even/odd preconditioning

In all our numerical tests, we started from the even/odd preconditioned fermion matrix,

$$M_{oo} = \mathbb{1}_{oo} - \kappa^2 H_{oe} H_{eo}, \qquad (50)$$

where e and o denote the collection of even and odd sites, respectively. Note that $\det M_{oo} = \det M$ and the condition number of M_{oo} is reduced compared with M. In the discussion of the algorithm above, essentially κH has to be replaced by $\kappa^2 H_{oe} H_{eo}$. Note that indices in Sec. IV below refer to powers of $\kappa^2 H_{oe} H_{eo}$. Note that in Ref. [29] it is explicitly spelled out, how the PHMC algorithm can be implemented for even/odd preconditioned clover-improved Wilson fermions.

III. THE ACCEPTANCE RATE AND FORCES

Typically the step-size of the HMC is tuned such that the acceptance rate $0.8 \lesssim P_{\text{acc}} \lesssim 0.9$. The optimal value depends on the integration scheme that is used. Also the occurrence of spikes might require to decrease the step-size $\delta\tau$. Spikes mean that occasionally $\Delta H \gg 1$ appears in the simulation. Here we have encountered this phenomenon when using the second order Omelyan integrator.

The acceptance rate can be determined by simply counting the accepted configuration. The statistical error is reduced by sampling $\min[1, \exp(-\Delta H)]$ instead. Detailed balance implies

$$\langle \exp(-\Delta H) \rangle = 1. \qquad (51)$$

It is a useful check for the correctness of the program to sample $\exp(-\Delta H)$ and check whether the average is consistent with one. Based on Eq. (51) one can derive for high acceptance rates

$$P_{\text{acc}} = \text{erfc}\left(\sqrt{\text{Var}(\Delta H)/8}\right). \qquad (52)$$

See Eq. (3.1) of Ref. [13] and references therein. In our simulations, as long as no spikes occur, Eq. (52) turned out to be valid to good precision.

The HMC simulation using improved pseudofermion actions [7,24,26,34–36] requires to tune a number of parameters. Therefore it is highly desirable to know how the acceptance rate, or equivalently $\text{Var}(\Delta H)$, depends on

these parameters. A step in this direction is taken by Ref. [13], where the variances of the forces associated with the different parts of the action are related to $\text{Var}(\Delta H)$. For the second order Omelyan scheme with $\lambda = 1/6$ the authors of Ref. [13] find, see their Eq. (3.4),

$$\text{Var}(\Delta H) = \frac{2\delta\tau^4}{72^2}\left[\text{Var}(|\mathcal{F}_{i_{\max}}|^2) + \frac{\text{Var}(|\mathcal{F}_{i_{\max}-1}|^2)}{(4m^2_{i_{\max}-1})^2} + \cdots\right]. \qquad (53)$$

Note that for $\lambda \neq 1/6$ also other terms than the forces appear at the order $\delta\tau^4$. For a more general result see Ref. [38]. A main ingredient in the derivation of Eq. (53) is the fact that a symplectic integrator conserves a shadow Hamiltonian. The deviation of the shadow Hamiltonian from the true Hamiltonian can be computed as a power series in the step-size $\delta\tau$. Furthermore, it is assumed that the forces due to different pieces of the action are not correlated.

IV. NUMERICAL RESULTS

The study is performed on three servers with two CPUs with ten cores each, which were immediately available to us. For programming convenience no highly optimized code was used. As solver, we have used the BiCG-stab [16,17] algorithm. Here we did not experiment much with the stopping criterion, but did run the solver essentially up to machine precision. We simulate comparatively small lattices at $\beta = 5.6$. In particular we have tested κ^2- and κ^4-filtering extensively by simulating a $12^3 \times 24$ lattice at $\kappa = 0.156$. To consolidate the result, two simulations of a $16^3 \times 32$ lattice at $\kappa = 0.1575$ are performed. Our preliminary study of the performance of the HMC using rooted polynomials are also performed for a $16^3 \times 32$ lattice at $\kappa = 0.1575$. The linear lattice sizes are measured in units of the lattice spacing a. We use periodic boundary conditions in the spatial direction. In the case of the temporal direction, periodic boundary conditions are employed for the gauge action and antiperiodic ones for the fermion action.

A rather detailed study at this value of β is presented in Ref. [39]. Based on the Sommer scale r_0 [40], the authors of Ref. [39] find that for $\beta = 5.6$, $\kappa = 0.156$ on a $16^3 \times 32$ lattice $a = 0.09796(56)$ fm. For the same parameters they find $m_{PS} = 0.9002(69)$ GeV for the mass of the lightest pseudoscalar particle. For $\beta = 5.6$, $\kappa = 0.1575$ on a $16^3 \times 32$ lattice they obtain $a = 0.0839(11)$ fm and $m_{PS} = 0.6524(86)$ GeV. This means that the masses are still quite large compared with the mass of the pion $m_{\pi^0} \approx 135$ MeV. Note that a number of algorithmic studies were performed at $\beta = 5.6$, the values of κ and lattice sizes that were studied in Ref. [39]. See for example [7,14,34].

A. Exploiting the hopping parameter expansion

In this set of simulations, we tested the efficiency of κ^k-filtering. To this end, we simulated the system with the pseudofermion action (7) and the modified pseudofermion action (26) up to κ^2 and κ^4. We simulated by using the leapfrog as well as the second order Omelyan integrator at $\lambda = 1/6$. In both cases, we used two timescales. On the coarse time step we put the pseudofermion action and on the fine one the gauge action. The time step of the gauge action was chosen to be such that further decreasing it virtually does not increase the acceptance rate. Next we have to decide how to truncate Eq. (31). In the extended runs that we performed first, we set *ad hoc* $n_t = 7$ and $n_{max} = 19$ for κ^2. Note that $b_7 = 1.98\ldots \times 10^{-4}$ and $b_{19} = 8.22\ldots \times 10^{-18}$. Instead, for κ^4 we took $n_t = 15$ and $n_{max} = 29$, where $b_{15} = 7.91\ldots \times 10^{-6}$ and $b_{29} = 1.15\ldots \times 10^{-14}$. Later we carefully checked the dependence of the acceptance rate on n_t. Furthermore we demonstrate that the value of n_t has no influence on the reversibility.

1. Extended runs

We performed a few extended runs. This way we checked for spikes in ΔH and tried to estimate autocorrelation times. Throughout we used trajectories of length $\tau = \sqrt{2}$, corresponding to $\tau = 1$ in the convention of, for example, Ref. [8].

A first set of runs was performed by using the leapfrog integration scheme. We performed preliminary simulations to find the step-size $\delta\tau$ that gives $P_{acc} \approx 0.8$. In Table I we summarize the results of our extended runs. The plaquette value is $P = \frac{1}{3N_p}\sum_p \mathrm{ReTr}U_p$, where the sum runs over all plaquettes on the lattice and U_p denotes the ordered product of the gauge variables around the plaquette p and N_p is the number of plaquettes. Since the effort required for the evaluation of the polynomial (31) is small compared with that for the iterative solver, the performance gain achieved by the κ^k-filtering is essentially given by the ratio of the step numbers m. This means that even in the

TABLE I. Extended runs using the leapfrog algorithm for a $12^3 \times 24$ lattice at $\beta = 5.6$ and $\kappa = 0.156$. We study the effect of κ^k-filtering. stat gives the number of trajectories, the number of coarse time steps m, and the expectation value of the plaquette $\langle P \rangle$. The acceptance rate is given by $P_{acc} = \langle \min[1, \exp(-\Delta H)] \rangle$. In all three cases, the estimate of P_{acc} obtained from the variance $\mathrm{Var}(\Delta H)$ by using Eq. (52) is consistent with the result given in column 5.

k	m	stat	$\langle P \rangle$	P_{acc}	$\mathrm{Var}(\Delta H)$
0	42	2770	0.56982(7)	0.8006(43)	0.2673(54)
2	21	7050	0.56991(6)	0.7981(26)	0.2643(43)
4	16	7610	0.56995(4)	0.8106(24)	0.2264(40)

case of κ^2-filtering that is still achievable in the case of clover improvement [32], we see a gain of a factor of 2. Next we redid the exercise by using the second order Omelyan integrator at $\lambda = 1/6$. In order to get an acceptance rate of $\approx 80\%$, we find from preliminary simulations that $m = 18$ and 8 for the order 0 and 2 are needed, respectively. Hence the performance gain is even a bit larger than in the case of the leapfrog integrator. Performing longer runs, spikes in ΔH appeared. Therefore we do not further discuss these runs. It is known that the second order Omelyan integrator at $\lambda = 1/6$ is more susceptible to this problem than the leapfrog. The problem can be cured by reducing the step-size. In the case of κ^4-filtering we could not find an m that gives an acceptance rate of $\approx 80\%$. For $m = 6$, the acceptance rate is considerably larger and for $m = 5$ it is smaller. We decided to perform a longer run for $m = 6$. From 24540 trajectories we get $\langle P \rangle = 0.56991(2)$, $P_{acc} = 0.8830(15)$, and $\mathrm{Var}(\Delta H) = 0.0886(15)$. In this run no spikes appear. We find that the direct determination of P_{acc} and the result obtained from Eq. (52) are consistent. From this run we get the estimates $\tau_{\mathrm{int},P} = \sqrt{2} \times 9.3(1.7)$ and $\tau_{\mathrm{int,iter}} = \sqrt{2} \times 24.8(4.5)$ for the integrated autocorrelation times of the plaquette and the iteration number of the solver, respectively. Given the relatively low accuracy of the autocorrelation time, we are not able to decide whether the UV filtering has an influence on the autocorrelation time.

2. The forces

As it is argued in Ref. [13], the acceptance rate can be inferred from the variance of the forces $\mathrm{Var}(|\mathcal{F}|^2)$. Computing $\mathrm{Var}(|\mathcal{F}|^2)$ for κ^4-filtering, we get essentially consistent results from the run with the leapfrog and the second order Omelyan integrator. We conclude $\mathrm{Var}(|\mathcal{F}_{PF}|^2) = 57500(1000)$, where the error is only a rough estimate. In the case of the runs without filtering and κ^2-filtering, using the leapfrog integration scheme, we get $\mathrm{Var}(|\mathcal{F}_{PF}|^2) = 11400000(200000)$ and $344000(4000)$, respectively. The runs with the second order Omelyan scheme contain spikes in ΔH. These spikes can also be seen in $\mathcal{F}_{PF}$. As a result, $\mathrm{Var}(|\mathcal{F}_{PF}|^2)$ is by far larger than for the runs with the leapfrog. Excluding the spikes by hand, $\mathrm{Var}(|\mathcal{F}_{PF}|^2)$ is much reduced, and very roughly consistent with what we find in the runs with the leapfrog integrator. Following Eq. (53), keeping $\mathrm{Var}(|\mathcal{F}_{PF}|^2)\delta\tau^4$ fixed, should result in a fixed acceptance rate. Indeed, $(11400000/344000)^{1/4} \approx 2.4$ and $(11400000/58000)^{1/4} \approx 3.7$ are roughly consistent with the speed-ups that we have observed directly.

For the gauge action, we get from the runs with the leapfrog and the second order Omelyan scheme for both κ^2-filtering and no filtering consistent results that can be summarized as $\mathrm{Var}(|\mathcal{F}_G|^2) = 28800000(400000)$. In the case of κ^4-filtering, due to the larger value of β in $\tilde{S}_G$, we get

the larger value $\text{Var}(|\mathcal{F}_G|^2) = 30000000(400000)$. We checked that, also according to Eq. (53), our choices of $\delta\tau_G$ are small enough, not to influence the acceptance rate markedly.

3. Truncation of the series and reversibility

In principle we can relax the accuracy of the calculation of the force to the point, where the acceptance rate is markedly affected. However it turned out that, using iterative solvers, the reversibility of the integration is increasingly violated with decreasing accuracy of the solution. With exact numerics, reversibility would be given at any precision of the solver. However we work with double precision numbers, and rounding errors occur. Furthermore, iterative solvers approach the solution in a chaotic way. Hence, if we stop the solver at a moderate precision, deviations caused by rounding errors are blown up. This phenomenon does not occur when we evaluate a series with fixed coefficients. Therefore the truncation at the order $n_t < n_{\max}$ of the sum (31) can be chosen such that the acceptance rate is reduced by little compared with larger values of n_t. We checked this reasoning for κ^4-filtering and the second order Omelyan scheme at $\lambda = 1/6$. To this end, we selected ten configurations, which were separated by 400 trajectories each from our extended run. For each of these configurations, we started a trajectory using the same parameters as for our extended run. At the end of the trajectory the momenta are reversed and the trajectory is run backwards, resulting in the configuration U'. We compute

$$\Delta = \sum_{x,\mu}|U_{x,\mu} - U'_{x,\mu}|^2. \tag{54}$$

For $n_t = 5, 6,$ and 15 and running the solver essentially up to machine precision, we get $\Delta \approx 6.3 \times 10^{-21}$ for all three choices. Instead, keeping $n_t = 15$ fixed and relaxing the stopping criterion of the BiCG-stab, Δ is clearly increasing.

4. The acceptance rate as a function of n_t

For both κ^2- and κ^4-filtering, we performed runs with different values of n_t. We used the second order Omelyan scheme with $\lambda = 1/6$ throughout. In all cases the trajectories have the length $\tau = \sqrt{2}$. As expected, we find that with increasing n_t the acceptance rate rapidly reaches a plateau value.

For κ^2-filtering, we first performed runs with $m = 8$. Similar to the extended run, spikes appeared. Therefore we redid the runs with $m = 10$, where we did not encounter this problem for $n_t > 3$. The results are summarized in Table II. The acceptance rate as well as $\text{Var}(\Delta H)$ rapidly approach a plateau, which is reached at the level of our numerical precision for $n_t = 5$. For $n_t > 3$, the estimate of P_{acc} obtained from the variance $\text{Var}(\Delta H)$, by using Eq. (52) is consistent with the direct measurement.

TABLE II. Numerical results for κ^2-filtering. Simulations are performed with the second order Omelyan scheme at $\lambda = 1/6$. We give the acceptance rate and $\text{Var}(\Delta H)$ as a function of the maximal summation index n_t. Throughout we use $m = 10$ and the length of the trajectory is $\tau = \sqrt{2}$. In the run for $n_t = 3$ spikes occurred. After removing them by hand we get $\text{Var}(\Delta H) = 0.88(5)$.

n_t	stat	P_{acc}	$\text{Var}(\Delta H)$
3	1000	0.6420(11)	
4	2500	0.9187(21)	0.0424(14)
5	2000	0.9361(21)	0.0268(9)
6	1950	0.9412(20)	0.0249(8)
7	2000	0.9406(20)	0.0242(9)

Our results for κ^4-filtering are summarized in Table III. Also here the acceptance rate as well as $\text{Var}(\Delta H)$ rapidly reach a plateau value. At the level of our accuracy this happens for $n_t = 7$. As expected, this value is larger than for κ^2-filtering.

We conclude that the choice of n_t is uncritical. Using a few short runs we can locate the point, where the acceptance rate as a function of n_t levels off. In the production run we then add a small safety margin.

5. Scaling with the lattice size and κ

To get an idea of how the performance scales with the hopping parameter κ, we performed two short runs at $\kappa = 0.1575$ on a $16^3 \times 32$ lattice by using κ^4-filtering. In both cases the length of a trajectory is $\tau = \sqrt{2}$. We started the simulations with a configuration taken from the runs discussed in the section below. In the first simulation we used the leapfrog algorithm with $m = 32$. From 500 trajectories we get $P_{\text{acc}} = 0.874(8)$. Note that from Table I of Ref. [39] we read off that without UV filtering, $m = 100$ results in $P_{\text{acc}} = 0.78$. Hence we see a speed-up by roughly a factor of 3 as it is the case for $\kappa = 0.156$ and the $12^3 \times 24$ lattice. In the second simulation we used the second order Omelyan scheme with $\lambda = 1/6$ and $m = 16$. Performing 500 trajectories we find $P_{\text{acc}} = 0.926(6)$.

TABLE III. Same as Table II but for κ^4-filtering. Here we use $m = 6$ throughout.

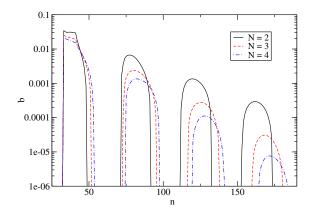| n_t	stat	P_{acc}	$\text{Var}(\Delta H)$
3	200	0.22(3)	4.98(59)
4	1030	0.177(8)	8.12(24)
5	6400	0.8631(22)	0.1180(31)
6	2200	0.8506(45)	0.1512(63)
7	2200	0.8868(32)	0.0920(36)
8	2000	0.8845(31)	0.0848(29)
9	2200	0.8851(36)	0.0904(33)
15	24500	0.8830(15)	0.0886(15)

FIG. 1. We give the coefficients b_n, Eq. (47), as a function of the index n for $N = 2$, 3 and 4. Only results for $b_n \geq 10^{-6}$ are shown.

B. Runs with rooted polynomials

We performed a few runs with the rooted polynomial action. We simulated a $16^3 \times 32$ lattice at $\beta = 5.6$ and $\kappa = 0.1575$. All runs are characterized by $j_{\max} = 2$. In all cases we use for simplicity the leapfrog scheme with different timescales. Throughout we use the trajectory length $\tau = \sqrt{2}$.

1. Without hopping parameter expansion

In this first set of runs we simulated without making use of the hopping parameter expansion. This means that we start the recursion, Eq. (45), with $M_1 = M$. The polynomials are characterized by $n_1 = 8$ and $n_2 = 32$ and rooting with $N = 2$, 3, 4, 6, 8, and 16. In Fig. 1 we show the coefficient b_n of Eq. (47) for $N = 2$, 3, and 4. For $n > n_2 = 32$, b_n is oscillating, with a decreasing amplitude. As it can be seen from the figure, the decay is exponential in n. The decay becomes faster with increasing N. With increasing N, the decay rate converges to a finite limit. In Table IV we summarize the basic parameters of the simulations and give the acceptance rate P_{acc} and

$\mathrm{Var}(\Delta H)$. We have taken m such that $P_{\mathrm{acc}} \approx 0.8$. The parameters m_2, m_1, and m_0 are chosen *ad hoc* and are likely larger than the optimal values. Note that error bars might be underestimated, since the lengths of the runs are relatively short. It is reassuring that our estimates of $\langle P \rangle$ are consistent with the result given in Table I of [39].

In Table V we summarize the results obtained for the variances of the forces. As one might expect, $\mathrm{Var}(|\mathcal{F}_G|^2)$ does not depend on N. Furthermore, comparing with the runs for the $12^3 \times 24$ lattice of the previous section, we see that $\mathrm{Var}(|\mathcal{F}_G|^2)$ is roughly proportional to the volume of the lattice. In the case of the rooted pseudofermion action we find that $\mathrm{Var}(|\mathcal{F}_{PF,1}|^2)$ and $\mathrm{Var}(|\mathcal{F}_{PF,2}|^2)$ are decreasing with increasing N. In the limit $N \to \infty$, a finite value, corresponding to the hopping parameter expansion should be reached. Here, it seems that we are still far away from this limit. Going from $N = 8$ to 16, $\mathrm{Var}(|\mathcal{F}_{PF,1}|^2)$ and $\mathrm{Var}(|\mathcal{F}_{PF,2}|^2)$ are reduced by roughly a factor of 4. Following Eq. (53), this should allow to increase the corresponding step-size by a factor of $\sqrt{2}$. Since the numerical effort for evaluating S_{PF} increases by a factor of 2, the algorithm becomes less efficient. In order to compare the numerical costs, we define the cost index $c = n_j N \, \mathrm{Var}(|\mathcal{F}_{PF,1}|^2)^{1/4}$, where the exponent $1/4$ is motivated by Eq. (53). Our results are summarized in Table VI. In the case of $S_{PF,1}$ we see a small increase from $N = 2$ to 4. For $S_{PF,2}$ the cost index is very similar for $N = 2$, 3, and 4. On the other hand, $\mathrm{Var}(|\mathcal{F}_{PF,3}|^2)$ is clearly decreasing going from $N = 2$ to 4. The costs related with $S_{PF,3}$ depend on the solver that is used. Here we made no effort to find the optimal solver. Therefore we refrain from quoting a performance index for $S_{PF,3}$. Anyway, it seems likely that the optimal overall performance is reached for $N > 2$.

2. Employing κ^4-filtering

UV filtering by using the hopping parameter expansion can by easily implemented in the PHMC algorithm [32]. Here we perform a preliminary study, employing κ^4-filtering.

TABLE IV. Basic properties of our runs with rooted polynomials. We employ κ^k-filtering. The polynomials approximate the Nth root of M^{-1}. n_1 and n_2 give the maximal order of the first and second level approximation. stat gives the number of trajectories, m_0, m_1, m_2 and m are the number of steps that characterize the multilevel integration scheme, and $\langle P \rangle$ is the expectation value of the plaquette. In the 10th column we give α of Eq. (47). Note that in the limit $N \to \infty$ we get $\alpha = 0.017275\ldots$ and $\alpha = 0.013210\ldots$ for $n_2 = 32$ and 42, respectively. Finally, in the 11th and 12th columns we give the acceptance rate P_{acc} and the variance $\mathrm{Var}(\Delta H)$, respectively.

k	N	n_1	n_2	stat	m_0	m_1	m_2	m	α	$\langle P \rangle$	P_{acc}	$\mathrm{Var}(\Delta H)$
0	2	8	32	290	6	6	4	8	$0.022110\ldots$	0.57279(6)	0.870(11)	0.110(9)
0	2	8	32	510	4	4	2	8	$0.022110\ldots$	0.57257(5)	0.836(12)	0.191(16)
0	3	8	32	500	10	6	3	6	$0.020504\ldots$	0.57255(6)	0.788(9)	0.253(22)
0	4	8	32	910	6	5	4	5	$0.019687\ldots$	0.57258(3)	0.793(9)	0.299(16)
0	6	8	32	600	10	5	2	5	$0.018872\ldots$	0.57256(5)	0.773(11)	0.309(18)
0	8	8	32	500	6	5	2	5	$0.018467\ldots$	0.57254(5)	0.792(12)	0.311(19)
0	16	8	32	200	6	5	2	5	$0.017866\ldots$	0.57259(6)	0.806(17)	0.241(25)
4	8	16	42	500	40	3	2	4	$0.013902\ldots$	0.57265(5)	0.790(10)	0.249(22)

TABLE V.   Results for the variance of the forces $\text{Var}(|\mathcal{F}_G|^2)$, $\text{Var}(|\mathcal{F}_{PF,1}|^2)$, $\text{Var}(|\mathcal{F}_{PF,2}|^2)$, and $\text{Var}(|\mathcal{F}_{PF,3}|^2)$ from our simulations with rooted polynomials listed in Table IV. We employ κ^k-filtering. The fermion matrix is taken to the Nth root.

k	N	G	$PF,1$	$PF,2$	$PF,3$
0	2	85000000(4500000)	1110000(60000)	11000(900)	1300(120)
0	3	82000000(4700000)	290000(15000)	2020(180)	540(60)
0	4	84000000(3500000)	114000(4000)	710(100)	360(60)
0	6	77000000(4000000)	42400(2000)	197(17)	156(15)
0	8	79000000(4000000)	20200(1000)	81(8)	123(14)
0	16	83000000(6000000)	4860(400)	16.5(3.0)	156(40)
4	8	90000000(5000000)	2370(160)	23.8(2.5)	42(5)

TABLE VI.   Cost index related to the terms $S_{PF,1}$ and $S_{PF,2}$ of the pseudofermion action. The estimates of $\text{Var}(|\mathcal{F}|^2)$ are taken from Table V.

N	$8N\ \text{Var}(|\mathcal{F}_{PF,1}|^2)^{1/4}$	$32N\ \text{Var}(|\mathcal{F}_{PF,2}|^2)^{1/4}$
2	519(7)	655(13)
3	557(7)	644(14)
4	588(5)	661(22)
6	689(8)	719(15)
8	763(9)	768(18)
16	1069(21)	1032(44)

The basic parameters of our run are summarized in the last row of Table IV. Our results for the variance of the forces is given in the last row of Table V. These results can be compared with those for $N = 8$ without κ^k-filtering. $\text{Var}(|\mathcal{F}_{PF,1}|^2)$ is considerably reduced by using κ^4-filtering. The reduction in the case of $\text{Var}(|\mathcal{F}_{PF,2}|^2)$ and $\text{Var}(|\mathcal{F}_{PF,3}|^2)$ is due to the increased values of n_1 and n_2, respectively.

V. CONCLUSION AND OUTLOOK

We discuss how the hopping parameter expansion can be used as an efficient UV filter in the HMC simulation of lattice QCD with two degenerate fermion flavors. We have carefully tested the idea for the Wilson gauge action and Wilson fermions at $\beta = 5.6$ and $\kappa = 0.156$ and the relatively small lattice size $12^3 \times 24$. Compared with the pseudofermion action (7) we find a speed-up of a factor of 2 and 3, using κ^2- and κ^4-filtering, respectively. The latter result is confirmed by short runs performed for a $16^3 \times 32$ lattice and $\kappa = 0.1575$.

In large scale simulations, going to small fermion masses, the idea can be combined with mass preconditioning or domain decompositioning. In the case of mass preconditioning one might be able to skip the term in the action that corresponds to the most heavy mass. In the case of domain decompositioning one applies the idea to the fermion matrix that is restricted to the domains.

A natural extension of applying the hopping parameter expansion as UV filter is the use of rooted polynomials. This idea is related with the rooting proposed in Ref. [26] as well as the idea of hierarchically factorized polynomials [35,36]. Here our results are still preliminary, and both a better theoretical understanding as well as further numerical experiments are needed.

[1] H. J. Rothe, *Lattice Gauge Theories: An Introduction*, World Sci. Lect. Notes Phys. Vol. 82 (World Scientific, Germany, 2012).

[2] I. Montvay and G. Münster, *Quantum fields on a lattice* (Cambridge University Press, Cambridge, England, 1994).

[3] C. Gattringer and C. B. Lang, *Quantum Chromodynamics on the Lattice: An Introductory Presentation*, Lect. Notes Phys. Vol. 788 (Springer, Berlin Heidelberg, 2010)

[4] R. Gupta, Introduction to lattice QCD, at the LXVIII Les Houches Summer School "Probing the Standard Model of Particle Interactions", arXiv:hep-lat/9807028.

[5] D. Weingarten and D. Petcher, Monte Carlo integration for lattice gauge theories with fermions, Phys. Lett. B **99**, 333 (1981).

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

[7] M. Lüscher, Schwarz-preconditioned HMC algorithm for two-flavor lattice QCD, Comput. Phys. Commun. **165**, 199 (2005).

[8] R. Gupta, A. Patel, C. F. Baillie, G. Guralnik, G. W. Kilcup, and S. R. Sharpe, QCD with dynamical Wilson fermions, Phys. Rev. D **40**, 2072 (1989).

[9] S. Schaefer, R. Sommer, and F. Virotta, Critical slowing down and error analysis in lattice QCD, Nucl. Phys. **B845,** 93 (2011).

[10] S. A. Gottlieb, W. Liu, D. Toussaint, R. L. Renken, and R. L. Sugar, Hybrid molecular dynamics algorithms for the numerical simulation of quantum chromodynamics, Phys. Rev. D **35,** 2531 (1987).

[11] I. P. Omelyan, I. M. Mryglod, and R. Folk, Symplectic analytically integrable decomposition algorithms: Classification, derivation, and application to molecular dynamics, quantum and celestial mechanics simulations, Comput. Phys. Commun. **151,** 272 (2003).

[12] J. C. Sexton and D. H. Weingarten, Hamiltonian evolution for the hybrid Monte Carlo algorithm, Nucl. Phys. **B380,** 665 (1992).

[13] A. Bussone, M. Della Morte, V. Drach, and C. Pica, Tuning the hybrid Monte Carlo algorithm using molecular dynamics forces' variances, arXiv:1801.06412.

[14] C. Urbach, K. Jansen, A. Shindler, and U. Wenger, HMC algorithm with multiple timescale integration and mass preconditioning, Comput. Phys. Commun. **174,** 87 (2006).

[15] A. D. Kennedy, P. J. Silva, and M. A. Clark, Shadow Hamiltonians, Poisson brackets, and gauge theories, Phys. Rev. D **87,** 034511 (2013).

[16] H. A. Van der Vorst, Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems, SIAM J. Sci. Stat. Comput. **13,** 631 (1992).

[17] M. H. Gutknecht, Variants of BICGSTAB for matrices with complex spectrum, SIAM J. Sci. Comput. **14,** 1020 (1993).

[18] M. Lüscher, Local coherence and deflation of the low quark modes in lattice QCD, J. High Energy Phys. 07 (2007) 081.

[19] R. Babich, J. Brannick, R. C. Brower, M. A. Clark, T. A. Manteuffel, S. F. McCormick, J. C. Osborn, and C. Rebbi, Adaptive Multigrid Algorithm for the Lattice Wilson-Dirac Operator, Phys. Rev. Lett. **105,** 201602 (2010).

[20] M. Lüscher, A new approach to the problem of dynamical quarks in numerical simulations of lattice QCD, Nucl. Phys. **B418,** 637 (1994).

[21] M. Hasenbusch, Speeding up finite step-size updating of full QCD on the lattice, Phys. Rev. D **59,** 054505 (1999).

[22] J. Finkenrath, F. Knechtli, and B. Leder, Fermions as global correction: The QCD case, Comput. Phys. Commun. **184,** 1522 (2013).

[23] M. Hasenbusch, Speeding up the hybrid Monte Carlo algorithm for dynamical fermions, Phys. Lett. B **519,** 177 (2001).

[24] M. Hasenbusch and K. Jansen, Speeding up lattice QCD simulations with clover-improved Wilson fermions, Nucl. Phys. **B659,** 299 (2003).

[25] M. Lüscher, Lattice QCD and the Schwarz alternating procedure, J. High Energy Phys. 05 (2003) 052.

[26] M. A. Clark and A. D. Kennedy, Accelerating Dynamical Fermion Computations using the Rational Hybrid Monte Carlo (RHMC) Algorithm with Multiple Pseudofermion Fields, Phys. Rev. Lett. **98,** 051601 (2007).

[27] Ph. de Forcrand and T. Takaishi, Fast fermion Monte Carlo, Nucl. Phys. B, Proc. Suppl. **53,** 968 (1997).

[28] R. Frezzotti and K. Jansen, A polynomial hybrid Monte Carlo algorithm, Phys. Lett. B **402,** 328 (1997).

[29] S. Aoki *et al.*, Polynomial hybrid Monte Carlo algorithm for lattice QCD with an odd number of flavors, Phys. Rev. D **65,** 094507 (2002).

[30] Ph. de Forcrand, UV filtered fermionic Monte Carlo, Nucl. Phys. B, Proc. Suppl. **73,** 822 (1999).

[31] C. Alexandrou, Ph. de Forcrand, M. D'Elia, and H. Panagopoulos, Efficiency of the UV-filtered multiboson algorithm, Phys. Rev. D **61,** 074503 (2000).

[32] K.-I. Ishikawa *et al.* (PACS-CS Collaboration), An application of the UV-filtering preconditioner to the polynomial hybrid Monte Carlo algorithm, Proc. Sci., LAT2006 (**2006**) 027.

[33] C. Thron, S. J. Dong, K. F. Liu, and H. P. Ying, Padé-Z_2 estimator of determinants, Phys. Rev. D **57,** 1642 (1998).

[34] W. Kamleh and M. Peardon, Polynomial filtered HMC: An algorithm for lattice QCD with dynamical quarks, Comput. Phys. Commun. **183,** 1993 (2012).

[35] T. Haar, W. Kamleh, J. Zanotti, and Y. Nakamura, Improving polynomial-filtered hybrid Monte Carlo with Hasenbusch, Proc. Sci., INPC2016 (**2017**) 319.

[36] W. Kamleh, T. Haar, Y. Nakamura, and J. Zanotti, Single flavor filtering for RHMC in BQCD, EPJ Web Conf. **175,** 09004 (2018).

[37] Ph. de Forcrand, Monte Carlo quasi-heat-bath by approximate inversion, Phys. Rev. E **59,** 3698 (1999).

[38] B. Straßberger, Evaluation of fermion determinant splitting in flavor lattice QCD simulations, Master thesis, Humboldt-Universität zu Berlin, 2017.

[39] B. Orth, T. Lippert, and K. Schilling, Finite-size effects in lattice QCD with dynamical Wilson fermions, Phys. Rev. D **72,** 014503 (2005).

[40] R. Sommer, A new way to set the energy scale in lattice gauge theories and its application to the static force and α_s in SU(2) Yang-Mills theory, Nucl. Phys. **B411,** 839 (1994).