Polynomial hybrid Monte Carlo algorithm for lattice QCD with an odd number of flavors

S. Aoki,[1] R. Burkhalter,[2] M. Fukugita,[3] S. Hashimoto,[4] K-I. Ishikawa,[2] N. Ishizuka,[1,2] Y. Iwasaki,[1,2] K. Kanaya,[1,2] T. Kaneko,[4] Y. Kuramashi,[4] M. Okawa,[4,*] T. Onogi,[5] S. Tominaga,[2] N. Tsutsui,[4] A. Ukawa,[1,2] N. Yamada,[4] and T. Yoshié[1,2]

(JLQCD Collaboration)

[1]*Institute of Physics, University of Tsukuba, Tsukuba, Ibaraki 305-8571, Japan*
[2]*Center for Computational Physics, University of Tsukuba, Tsukuba, Ibaraki 305-8577, Japan*
[3]*Institute for Cosmic Ray Research, University of Tokyo, Kashiwa, Chiba 277-8582, Japan*
[4]*High Energy Accelerator Research Organization(KEK), Tsukuba, Ibaraki 305-0801, Japan*
[5]*Yukawa Institute for Theoretical Physics, Kyoto University, Kyoto 606-8502, Japan*

(Received 8 January 2002; published 24 April 2002)

We present a polynomial hybrid Monte Carlo (PHMC) algorithm for lattice QCD with odd numbers of flavors of $O(a)$-improved Wilson quark action. The algorithm makes use of the non-Hermitian Chebyshev polynomial to approximate the inverse square root of the fermion matrix required for an odd number of flavors. The systematic error from the polynomial approximation is removed by a noisy Metropolis test for which a new method is developed. Investigating the property of our PHMC algorithm in the $N_f=2$ QCD case, we find that it is as efficient as the conventional HMC algorithm for a moderately large lattice size ($16^3\times 48$) with intermediate quark masses ($m_{PS}/m_V\sim 0.7$–0.8). We test our odd-flavor algorithm through extensive simulations of two-flavor QCD treated as an $N_f=1+1$ system, and comparing the results with those of the established algorithms for $N_f=2$ QCD. These tests establish that our PHMC algorithm works on a moderately large lattice size with intermediate quark masses ($16^3\times 48, m_{PS}/m_V\sim 0.7$–$0.8$). Finally we experiment with the $(2+1)$-flavor QCD simulation on small lattices ($4^3\times 8$ and $8^3\times 16$), and confirm the agreement of our results with those obtained with the R algorithm and extrapolated to a zero molecular dynamics step size.

DOI: 10.1103/PhysRevD.65.094507 PACS number(s): 12.38.Gc, 02.60.Cb, 02.60.Gf, 75.40.Mg

I. INTRODUCTION

An essential step toward realistic lattice simulations of quantum chromodynamics (QCD) is to develop efficient algorithms to incorporate the dynamical sea quark effects of up, down, and strange quarks. Most of the recent dynamical QCD simulations have been, however, limited to two-flavor QCD where up and down quarks are treated dynamically while the loop effect of the strange quark is still neglected. This is mainly due to the lack of efficient algorithms to treat an odd number of dynamical quark flavors. The R algorithm [1] is a possible candidate for this purpose, but its serious drawback is the systematic error of $O(dt^2)$ stemming from a finite step size dt in the molecular dynamics evolution. To control this systematic error, one has to keep dt small enough and to monitor the size of the error by performing simulations at various values of dt, which requires much computational effort. Therefore, an exact algorithm such as the hybrid Monte Carlo (HMC) algorithm [2], which is widely used for simulations with an even number of flavors, is clearly desirable.

Recently, Takaishi and de Forcrand proposed an algorithm for an odd number of dynamical flavors [3]. They use the polynomial hybrid Monte Carlo (PHMC) algorithm [4–6] with a non-Hermitian Chebyshev polynomial, with which one can approximate the inverse square root of the fermion matrix needed for the simulation of an odd number of flavors [7,8]. They introduced a method to calculate the correction factor required to compensate for the truncation error due to the finite order of the polynomial, and hence the algorithm is exact. The algorithm was tested on a small lattice for 1, $1+1$, and $2+1$ flavors of Wilson fermions.

Clearly, the next step toward realistic simulations of QCD is to investigate the practical feasibility of their algorithm for two light (up and down) quarks and one relatively heavy (strange) quark on large physical volumes. In this case, up and down quarks are treated with the usual pseudofermion method, while the strange quark is incorporated with the polynomial approximation. It is known that the multiboson algorithms, which also rely on the polynomial approximation for the inverse of fermion matrix, fail for light quarks [9]. Therefore, we need to examine whether the algorithm with the polynomial approximation works for intermediate quark masses (around the strange quark). An implementation of the algorithm for the $O(a)$-improved Wilson (clover) quark action [10] is also important to carry out simulations with reduced systematic errors due to finite lattice spacing.

In this work we present a modified algorithm for $(2+1)$-flavor QCD with the $O(a)$-improved Wilson quark action. Our algorithm is a variant of PHMC with the non-Hermitian Chebyshev polynomial as that of Takaishi and de Forcrand [3], while the treatment of the correction factor is different. We test our algorithm for two different systems. One is two-flavor QCD treated as a system with $1+1$ flavors, and the simulation results are compared with those of the conventional HMC for two flavors. The other is $(2+1)$-flavor QCD, where our algorithm is compared with the R algorithm [1] after extrapolating to zero step size $dt\rightarrow 0$.

*Present address: Department of Physics, Hiroshima University, Higashi-Hiroshima, Hiroshima 739-8526, Japan.

We also perform two systematic numerical tests of the HMC and PHMC algorithms in two-flavor QCD in order to provide a basis to find the best method and parameter choices for an extension to realistic simulation with $2+1$ flavors.

As a first step, we test the even-odd preconditioning for the $O(a)$-improved Wilson fermion action, which was first proposed by Luo [11] and Jansen and Liu [12]. They introduced symmetrical and asymmetrical preconditioning, and mainly considered the asymmetric version. In our practical tests we found significant improvement for both versions over the simulation without preconditioning. The improvement is more pronounced for the symmetric case and the computer time can be reduced almost by a factor two from that without preconditioning.

Second, we investigate the efficiency of the PHMC algorithm depending on the quark mass and on the degree of the polynomial. We found that the PHMC is as effective as the conventional HMC algorithm for two different quark masses corresponding to $m_{PS}/m_V=0.8$ and 0.7 on a reasonably large lattice. This observation is encouraging, as it suggests that the polynomial approximation is useful for future simulations of $(2+1)$-flavor QCD.

The rest of the paper is organized as follows. In Sec. II we outline the algorithms we consider in this paper. the polynomial hybrid Monte Carlo (PHMC) algorithm and its generalization to an odd number of flavors is described. In Sec. III we test the efficiency of the even-odd preconditioning for the $O(a)$-improved Wilson fermion action using the usual HMC algorithm with two-flavor of quarks. We then investigate the efficiency of the PHMC algorithm for two-flavor QCD in Sec. IV. Section V describes details of our algorithm for an odd number of flavors, and presents some numerical tests with which the consistency and the applicability is investigated. Our conclusion is given in Sec. VI. Our algorithm and simulation code have already been used for a study of the phase structure of three-flavor QCD with the Wilson-type fermion actions [13].

II. OUTLINE OF THE ALGORITHM

We first present the outline of our algorithm for $(N_{f_1}+N_{f_2})$-flavor QCD, where N_{f_1} is an even number while N_{f_2} is odd. The details of the algorithm will be explained separately in later sections.

In this section we consider the Wilson gauge and fermion actions, but the algorithm can be applied to more complicated lattice actions arising in the Symanzik improvement program [14]. In particular, the algorithm is suitable for the $O(a)$-improved Wilson action [10] which has a clover-leaf-type operator to remove the discretization error of $O(a)$.

A. Pseudofermion representation for even number of flavors

Let D_1 and D_2 be the Dirac operators for two different fermion masses corresponding to N_{f_1} and N_{f_2} flavors, respectively. The partition function of this fermion system is given by

$$\mathcal{Z}=\int \mathcal{D}U(\det[D_1])^{N_{f_1}}(\det[D_2])^{N_{f_2}}e^{-S_g[U]}, \qquad (1)$$

where $S_g[U]$ represents the gauge action.

Since N_{f_1} is an even number, the fermion determinant $(\det[D_1])^{N_{f_1}}$ can be expressed in terms of the usual pseudofermion integral

$$(\det[D_1])^{N_{f_1}}=\int \mathcal{D}\phi_1^\dagger \mathcal{D}\phi_1 \exp[-|D_1^{-N_{f_1}/2}\phi_1|^2], \qquad (2)$$

where we have used the relation $D_1^\dagger=\gamma_5 D_1 \gamma_5$. We use a short-hand notation for the norm of a vector X as $|X|^2 \equiv \Sigma_{n,\alpha,a}|X_\alpha^a(n)|^2$ with n the site index, α the spinor index, and a the color index.

In the usual HMC algorithm one uses some iterative solver to calculate the inverse of the fermion matrix D_1. In the PHMC algorithm [5,6], on the other hand, one introduces a polynomial $P_{N_{poly}}[z]$ of order N_{poly} that converges $1/z$ as $N_{poly}\to\infty$. The non-Hermitian Chebyshev polynomial

$$P_{N_{poly}}[z]=\sum_{i=0}^{N_{poly}} c_i(1-z)^i, \qquad (3)$$

with $c_i=(-1)^i$ is an example of such a polynomial, when $|1-z|<1$. Supposing that all eigenvalues of D_1 fall inside the complex domain $|1-z|<1$, we have

$$\begin{aligned}(\det[D_1])^{N_{f_1}}&=\left[\frac{\det[D_1 P_{N_{poly}}[D_1]]}{\det[P_{N_{poly}}[D_1]]}\right]^{N_{f_1}}\\ &=(\det[D_1 P_{N_{poly}}[D_1]])^{N_{f_1}}\int \mathcal{D}\phi_1^\dagger\mathcal{D}\phi_1\\ &\quad\times\exp[-|(P_{N_{poly}}[D_1])^{N_{f_1}/2}\phi_1|^2]. \qquad (4)\end{aligned}$$

We notice that the inversion of the fermion matrix $D_1^{-N_{f_1}/2}$ is replaced by a calculation of the polynomial $(P_{N_{poly}}[D_1])^{N_{f_1}/2}$.

Following the original proposal of the multiboson algorithm by Lüscher [15], Frezzotti and Jansen [5,6] considered a Hermitian operator $Q=c_M\gamma_5 D_1$ with c_M a normalization factor and used a polynomial approximation of $\det[D_1]^2=\det[Q]^2$ rather than the non-Hermitian $\det[D_1]$, using the γ_5 Hermiticity property $D_1^\dagger=\gamma_5 D_1\gamma_5$ of the Wilson-type lattice fermions. In this work, however, we consider the non-Hermitian relation Eq. (4), as it is suitable for the extension to an odd number of flavors.

Since the polynomial approximation introduces a truncation error, one has to evaluate the correction factor $\{\det(D_1 P_{N_{poly}}[D_1])\}^{N_{f_1}}$ in order to make the algorithm exact. As the correction factor is close to unity when the polynomial is a good approximation of the inverse, a stochastic technique can be used to incorporate the correction factor. The reweighting method [15] and the global Metropolis test [7,16] have been proposed and used in the multiboson algorithm. For the PHMC algorithm, the reweighting method is

applied in Refs. [5,6], and the global Metropolis test in Ref. [3]. We use the global Metropolis test developed for a multi-boson algorithm [7,16]. The details of the global Metropolis test in the case of $N_{f_1}=2$ will be given in Sec. IV C.

B. Pseudofermion representation for an odd number of flavors

For an odd number of flavors N_{f_2}, we use the method developed by Alexandrou *et al.* [8] to take a ''square root'' of the polynomial as described below.

We consider a polynomial $P_{N_{poly}}[z]$ with an even degree N_{poly} and rewrite it as a product of monomials

$$P_{N_{poly}}[z]=\sum_{i=0}^{N_{poly}} c_i(z-1)^i=c_{N_{poly}}\prod_{k=1}^{N_{poly}}(z-z_k), \qquad (5)$$

which approaches $1/z$ as N_{poly} increases. At this point the convergence radius is assumed to cover all eigenvalues of the Wilson-Dirac operator, which will be confirmed in Sec. V numerically. Since z_k appears with its complex conjugate, we may rewrite Eq. (5) as

$$P_{N_{poly}}[z]=c_{N_{poly}}\prod_{j=1}^{N_{poly}/2}(z-z^*_{k'(j)})(z-z_{k(j)}), \qquad (6)$$

where $k(j)$ and $k'(j)$ are the arbitrary reordering indices defined to satisfy the relation $z^*_{k(j)}=z^*_{k'(j)}$ with $j=1\ldots N_{poly}/2$. Using the property $D_2^\dagger=\gamma_5 D_2\gamma_5$ one can show that $\det[D_2-z^*_{k'(j)}]=\det[D_2-z_{k(j)}]^\dagger$ and

$$\begin{aligned}\det[P_{N_{poly}}[D_2]]&=c_{N_{poly}}\prod_{j=1}^{N_{poly}/2}\det[D_2-z_{k(j)}]^\dagger\\&\quad\times\det[D_2-z_{k(j)}]\\&=\det[T^\dagger_{N_{poly}}[D_2]T_{N_{poly}}[D_2]],\end{aligned} \qquad (7)$$

where $T_{N_{poly}}[z]\equiv\sqrt{c_{N_{poly}}}\Pi_{j=1}^{N_{poly}/2}(z-z_{k(j)})$. Then we obtain a pseudofermion representation for an odd number of flavors

$$\begin{aligned}(\det[D_2])^{N_{f_2}}&=\left[\frac{\det[D_2P_{N_{poly}}[D_2]]}{\det[P_{N_{poly}}[D_2]]}\right]^{N_{f_2}}\\&=[\det[D_2P_{N_{poly}}[D_2]]]^{N_{f_2}}\int\mathcal{D}\phi_2^\dagger\mathcal{D}\phi_2\\&\quad\times\exp[-|(T_{N_{poly}}[D_2])^{N_{f_2}}\phi_2|^2].\end{aligned} \qquad (8)$$

As in the case of an even number of flavors, the correction factor $\{\det(D_2P_{N_{poly}}[D_2])\}^{N_{f_2}}$ has to be kept to construct an exact algorithm. We describe the calculation of the correction factor for the $N_{f_2}=1$ case in Sec. V B.

We note that in this construction, the positivity of $\det[D_2]$ is assumed. Since the Wilson-type lattice fermions do not have chiral symmetry, the Wilson-Dirac operator D_2 may develop a real and negative eigenvalue, which could make $\det[D_2]$ negative. In actual simulations, we do not expect that this happens for the following reason. Under a continuous change of gauge configuration, as in the molecular dynamics evolution, the eigenvalues also change continuously. To change the sign of a real eigenvalue it has to cross zero, for which the determinant $\det[D_2]$ vanishes which is suppressed. In addition, since the single flavor part is to be identified with strange quark in realistic applications, we expect that the intermediate mass of strange quark behaves as an infrared cutoff obstructing the appearance of negative eigenvalues.

In our implementation, we use the fact that the correction factor $(\det[D_2P_{N_{poly}}[D_2]])^{N_{f_2}}$ is close to unity. If this does not hold, the calculation will fail to converge. We should, therefore, be aware of the appearance of a negative determinant. Our algorithm fails if this happens, but a negative determinant should be considered as a problem of the formulation of the lattice fermion rather than the problem of the algorithm, since it is related to the lack of chiral symmetry.

C. Hybrid Monte Carlo algorithm

Once we write an effective action for the fermion determinant using pseudofermions as in Eqs. (2), (4), and (8), it is straightforward to apply the hybrid Monte Carlo algorithm [2] to obtain an ensemble of gauge configurations including the effect of the approximated fermion determinant.

Introducing a fictitious momentum P conjugate to the link variable U (we suppress the site, direction, and color indices), the partition function Eq. (1) is written as

$$\mathcal{Z}=\int\mathcal{D}U\,\mathcal{D}P\,\mathcal{D}\phi_1^\dagger\mathcal{D}\phi_1\mathcal{D}\phi_2^\dagger\mathcal{D}\phi_2\det[W]e^{-H}. \qquad (9)$$

If we use the usual form Eq. (2) for an even number of flavors, and the polynomial representation Eq. (8) for the rest of the fermions, the effective Hamiltonian H and the correction factor $\det[W]$ take the form

$$\begin{aligned}H&=\frac{1}{2}P^2+S_g[U]+|D_1^{-N_{f_1}/2}\phi_1|^2\\&\quad+|(T_{N_{poly}}[D_2])^{N_{f_2}}\phi_2|^2,\\\det[W]&=(\det[D_2P_{N_{poly}}[D_2]])^{N_{f_2}}.\end{aligned} \qquad (10)$$

The HMC algorithm consists of the following four steps, for a given gauge configuration U.

(1) Generate momenta P and pseudo-fermion fields ϕ_1 and ϕ_2 from a Gaussian distribution with unit variance and zero mean.

(2) Integrate link variables U according to the discretized molecular dynamics evolution equation derived from the equation of motion

$$\dot{U}_\mu(n) = iP_\mu(n)U_\mu(n),$$

$$\dot{P}_\mu(n) = -i[U_\mu(n)F_\mu(n)]_{\text{T.A.}}, \tag{11}$$

where $\dot{X}$ is the derivative of a field X with respect to the fictitious time t and $[\cdots]_{\text{T.A.}}$ means the traceless anti-Hermitian part of the matrix in the bracket. The force $F_\mu(n)$ is defined through a variation of the effective Hamiltonian under an infinitesimal change $\delta U_\mu(n)$ of the gauge link variable

$$\delta H = \sum_{n,\mu} \text{Tr}[\{\delta U_\mu(n) F_\mu(n)\} + \text{H.c.}]. \tag{12}$$

The length in the fictitious time t is arbitrary, which we set equal to unity throughout this paper.

(3) Make a Metropolis test with respect to the energy difference dH between the initial configuration $U(0)$ and the trial configuration $U(t)$. The acceptance probability is $P_{acc}[(U(0),P(0)) \to (U(t),P(t))] = \min[1, e^{-dH}]$. If the test is accepted go to the next step (4), or else the new configuration is set to $(U(0),P(0))$ and go back to step (1).

(4) Make a Metropolis test with respect to the correction factor $\det[W]$. If the test is accepted $(U(t),P(t))$ is taken as the new configuration, or else the new configuration is $(U(0),P(0))$. Then return to step (1). The details to obtain the acceptance probability is described in Sec. V B.

III. EVEN-ODD PRECONDITIONING FOR THE $O(a)$-IMPROVED WILSON FERMION ACTION

Before going to the PHMC algorithm we discuss the even-odd preconditioning of the fermion determinant. The even-odd preconditioning is a widely used technique to accelerate the fermion matrix inversion [17], but it can also be used to reformulate the fermion determinant so that the pseudofermion field lives only on odd sites [18,19]. For the unimproved Wilson fermion action, no extra computational cost is required by the reformulation, while the HMC simulation becomes faster, since the phase space to be covered is reduced by a factor of two. Luo [11] and Jansen and Liu [12] introduced the even-odd preconditioning for the $O(a)$-improved Wilson fermion which includes the clover-leaf-type operator. In this section we review their formulation and describe our extensive numerical test to see how it improves the efficiency of the HMC algorithm.

A. Description of the preconditioning

The determinant of the $O(a)$-improved Wilson fermion operator D is written as

$$\det[D] = \det\begin{pmatrix} 1+T_{ee} & M_{eo} \\ M_{oe} & 1+T_{oo} \end{pmatrix}, \tag{13}$$

when the site index n is numbered such that even sites come earlier than any odd site. Here, the site is even (odd), if $n_x + n_y + n_z + n_t$ is an even (odd) number. The hopping term M (M_{eo} or M_{oe}) represents the usual Wilson fermion matrix

$$M_{n,n'} = -\kappa \sum_{\mu=1}^{4} \{(1-\gamma_\mu)U_\mu(n)\delta_{n+\hat{\mu},n'} + (1+\gamma_\mu)U_\mu^\dagger(n-\hat{\mu})\delta_{n-\hat{\mu},n'}\}, \tag{14}$$

while T (T_{ee} or T_{oo}) describes the $O(a)$-improvement term (or SW term)

$$T_{n,n'} = -\frac{1}{2} c_{\text{sw}} \kappa \sigma_{\mu\nu} \mathcal{F}_{\mu\nu}(n) \delta_{n,n'}, \tag{15}$$

with the clover-leaf-type field strength $\mathcal{F}_{\mu\nu}$ given by

$$\begin{aligned}\mathcal{F}_{\mu\nu}(n) = \frac{1}{8i}[&\{U_\mu(n)U_\nu(n+\hat{\mu})U_\mu^\dagger(n+\hat{\nu})U_\nu^\dagger(n) \\ &+ U_\nu(n)U_\mu^\dagger(n+\hat{\nu}-\hat{\mu})U_\nu^\dagger(n-\hat{\mu})U_\mu(n-\hat{\mu}) \\ &+ U_\mu^\dagger(n-\hat{\mu})U_\nu^\dagger(n-\hat{\mu}-\hat{\nu})U_\mu(n-\hat{\mu}-\hat{\nu}) \\ &\times U_\nu(n-\hat{\nu}) + U_\nu^\dagger(n-\hat{\nu})U_\mu(n-\hat{\nu}) \\ &\times U_\nu(n-\hat{\nu}+\hat{\mu})U_\mu^\dagger(n)\} - \text{H.c.}],\end{aligned} \tag{16}$$

where H.c. denotes the Hermitian conjugate of the preceding bracket. The Dirac matrix γ_μ is defined such that it is Hermitian, and $\sigma_{\mu\nu} = (i/2)[\gamma_\mu, \gamma_\nu]$.

Factoring out the even-even component $(1+T_{ee})$ from the determinant Eq. (13), we have

$$\det[D] = \det[1+T_{ee}]\det[\hat{D}^A_{oo}], \tag{17}$$

where

$$\hat{D}^A_{oo} = (1+T)_{oo} - M_{oe}(1+T)^{-1}_{ee} M_{eo}. \tag{18}$$

It is also possible to factor out both the even-even and odd-odd components as

$$\det[D] = \det[1+T_{ee}]\det[1+T_{oo}]\det[\hat{D}^S_{oo}], \tag{19}$$

where

$$\hat{D}^S_{oo} = 1 - (1+T)^{-1}_{oo} M_{oe}(1+T)^{-1}_{ee} M_{eo}. \tag{20}$$

In the following, we refer to Eqs. (17) and (19) as asymmetric and symmetric preconditioning, respectively. To our knowledge, previous simulations in the literature have exclusively been made with the asymmetric even-odd preconditioning.

Using Eqs. (17) and (18), the asymmetrically preconditioned partition function for two flavor QCD can be written as

$$\mathcal{Z}_{\text{A-HMC}} = \int \mathcal{D}U\, \mathcal{D}P\, \mathcal{D}\phi_o^\dagger\, \mathcal{D}\phi_o\, e^{-H_{\text{A-HMC}}[P,U,\phi_o]}, \tag{21}$$

$$H_{\text{A-HMC}}[P,U,\phi_o] = \frac{1}{2}P^2 + S_g[U] + S_q^A[U,\phi_o] + S_{det}^A[U],$$

$$S_q^A[U,\phi_o]=|(\hat{D}_{oo}^A)^{-1}\psi_o|^2,$$

$$S_{det}^A[U]=-2\log\det[1+T_{ee}].$$

The pseudofermion field ϕ_o lives on odd sites, whereas the determinant $\det[1+T_{ee}]$ of the local SW term is calculated on even sites.

For the symmetrically preconditioned partition function, from Eqs. (19) and (20) we have

$$\mathcal{Z}_{\text{S-HMC}}=\int \mathcal{D}U\,\mathcal{D}P\,\mathcal{D}\phi_o^\dagger\,\mathcal{D}\phi_o e^{-H_{\text{S-HMC}}[P,U,\phi_o]},$$

$$H_{\text{S-HMC}}[P,U,\phi_o]=\frac{1}{2}P^2+S_g[U]+S_q^S[U,\phi_o]+S_{det}^S[U],$$

$$S_q^S[U,\phi_o]=|(\hat{D}_{oo}^S)^{-1}\phi_o|^2, \tag{22}$$

$$S_{det}^S[U]=-2(\log\det[1+T_{ee}]+\log\det[1+T_{oo}]).$$

In this case the determinant of the local SW term is calculated both on even and odd sites.

The calculation of the force defined in Eq. (12) can be divided into several parts corresponding to the contribution from the pure gauge action, the pseudofermion part, and the determinant of the local SW term. We write down the contribution from the quark part in the Appendix for both preconditioning methods.

B. Efficiency of the even-odd preconditioning

The even-odd reformulation of the fermion determinant reduces the phase space to be covered by the HMC simulation. Another important effect of the preconditioning is that it lifts the lowest eigenvalue of the fermion matrix and thus the condition number is reduced. The strength of the force coming from the pseudofermionic part $S_q[U,\phi]$ of the effective Hamiltonian becomes smaller [4], since it is proportional to the inverse of the lowest eigenvalue of the Dirac matrix. Therefore, the error dH accumulating in the molecular dynamics evolution is also expected to become smaller, resulting in a better acceptance rate in the HMC algorithm. To what extent the condition number is reduced depends on the particulars of preconditioning. We expect the symmetric one to work better, since in the hopping parameter expansion $\hat{D}_{oo}^S$ behaves as $1-O(\kappa^2)$ while $\hat{D}_{oo}^A$ contains a term proportional to κ coming from T_{oo}.

In the following we describe a systematic test of the effect of the preconditioning of both types. The test is performed on three lattices: (i) a small lattice of size $8^3\times16$ with a heavy quark mass, which we call the ''small heavy'' lattice, (ii) a large lattice of size $16^3\times48$ with a heavy quark mass called ''large heavy,'' and (iii) a large lattice of size $16^3\times48$ with a light quark mass called ''large light.'' Here,

TABLE I. Lattice parameters.

	Small heavy	Large heavy	Large light
Size	$8^3\times16$	$16^3\times48$	$16^3\times48$
β	5.0	5.2	5.2
κ	0.1415	0.1340	0.1350
c_{sw}	1.855	2.02	2.02
m_{PS}/m_V	~0.8[a]	~0.8	~0.7

[a]This number is measured on a $12^3\times32$ lattice.

TABLE II. Parameters on the small heavy lattice. MD step size dt satisfies $dt\times N_{\text{MD}}=1$.

	HMC	A-HMC	S-HMC	C-PHMC	A-PHMC
N_{MD}	100, 50, 40, 30	100, 50, 40, 30, 25, 20	100, 50, 40, 32, 25, 20	50, 40, 32, 25, 20	32
N_{poly}	-	-	-	18, 20, 22, 24, 28, 30, (for $N_{MD}=32$) 26, (for all N_{MD})	18, 20, 22
Stopping condition force	10^{-2}	10^{-12}	10^{-12}	-	-
Stopping condition Hamiltonian	10^{-14}	10^{-14}	10^{-14}	10^{-14}[a]	10^{-14}[a]

[a]This is used to generate a pseudofermion field and global Metropolis test for the correction factor.

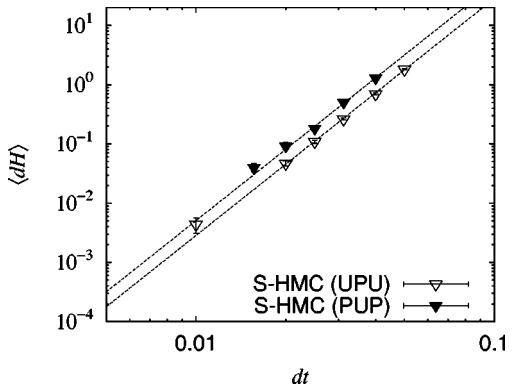

FIG. 1. MD step size dependence of $\langle dH \rangle$ for two integration methods (UPU and PUP) in the MD evolution. The lines show the fit with $\langle dH \rangle = \pi (a \cdot dt)^4$.

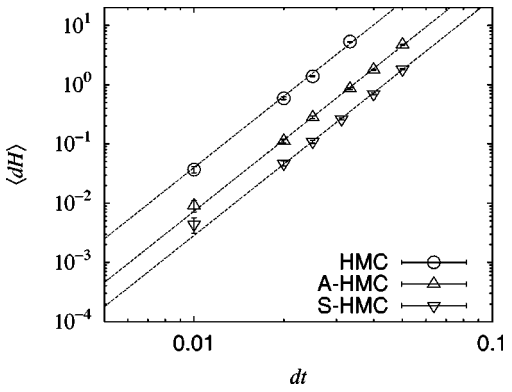

FIG. 3. MD step size dependence of $\langle dH \rangle$ for preconditioned and unpreconditioned effective actions. The lines show the fit with $\langle dH \rangle = \pi (a \cdot dt)^4$.

heavy and light quarks roughly correspond to $m_{PS}/m_V = 0.8$ and 0.7, respectively. The lattices (ii) and (iii) are reasonably large to study the light hadron spectrum. They are actually used in our production run [20]. Details of the lattice parameters are listed in Table I.

C. Extensive test on a small lattice

On the small heavy lattice, we investigate the molecular dynamics (MD) step size dt dependence of the acceptance rate P_{acc} for each algorithm: "HMC" denotes the HMC algorithm without the preconditioning, "A-HMC" and "S-HMC" are used for the asymmetrically or symmetrically preconditioned HMC algorithm.

We employ the BiCGStab algorithm [21] to calculate the inverse of the Dirac matrix D (or $\hat{D}^A_{oo}$, $\hat{D}^S_{oo}$). The symmetrical even-odd preconditioning is applied in the solver to accelerate the convergence of inversion. The stopping condition is defined so that the solver iterates until the residual defined by $r \equiv \sqrt{|Dx-b|^2/|b|^2}$ becomes smaller than a certain value, where b is a source vector and x is the solution vector. On the small heavy lattice, we use a rather strict stopping condition to avoid systematic errors coming from the matrix inversion. All numerical calculations are made with the double precision (64 bit) arithmetic. In Table II, we show the number of the molecular dynamics (MD) steps $N_{\rm MD}$ ($dt = 1/N_{\rm MD}$) and the stopping condition for the BiCGStab solver in force and Hamiltonian calculations.

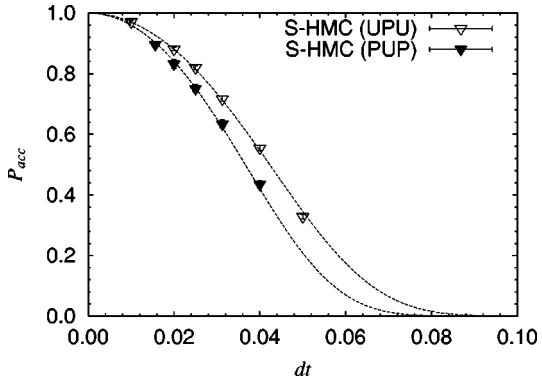

FIG. 2. MD step size dependence of the acceptance for two integration methods (UPU and PUP) in the MD evolution. The lines show the function erfc$[\sqrt{\pi}(a \cdot dt)^2/2]$ with a obtained from Fig. 1.

For the MD evolution of the kinematical variables U and P, the simplest integration scheme to satisfy the reversibility and measure preservation is the leapfrog algorithm. In this work we first consider two options of the leapfrog algorithm, i.e., UPU and PUP integrators. In the UPU integrator, the link variable U is updated at the first half step and then the integration of P with a unit step size dt follows. Thus the link variable U is assigned at $(n+1/2) \cdot dt$ with an integer n, while P is assigned at $n \cdot dt$. The integration is performed in the reverse order in the PUP integrator.

The acceptance rate in the HMC algorithm is governed by a change of the effective Hamiltonian during the MD evolution $\langle dH \rangle$ as $P_{acc} = \mathrm{erfc}(\langle dH \rangle^{1/2}/2)$. With the leapfrog integrator the change of effective Hamiltonian behaves as $\langle dH \rangle \sim dt^4$ for small dt [22–24].

In Fig. 1 we show the MD step size dt dependence of $\langle dH \rangle$ for both UPU and PUP integrators. The dotted lines represent a fit with a form $\langle dH \rangle = \pi (a \cdot dt)^4$. The Metropolis acceptance rate is plotted in Fig. 2 as a function of dt. The expected behavior erfc$[\sqrt{\pi}(a \cdot dt)^2/2]$ is also shown by dotted curves. We observe that the data is described by the expected functional form. We also find that the UPU integrator gives better acceptance at a fixed dt than the PUP integrator, which has been known for a long time for the staggered fermion action [23]. The computational cost with the UPU integrator is lower by a factor $N_{\rm MD}/(N_{\rm MD}+1)$ than the PUP integrator since the computer time in dynamical QCD simulations is dominated by the force calculation that involves the fermion matrix inversion. Therefore the advantage of the UPU integrator is very clear. We then use the UPU integrator in the rest of this work.

Let us now discuss the effect of preconditioning. Figures 3 and 4 show the MD step size dependence of $\langle dH \rangle$ and P_{acc} for the HMC, A-HMC and S-HMC algorithms. The dt dependence for each algorithm is described very well by the relation $\langle dH \rangle \propto dt^4$ as shown in Fig. 3, and the value of $\langle dH \rangle$ for A-HMC (S-HMC) at a fixed dt is about a factor 5 (13) smaller than the unpreconditioned HMC. As a result, the

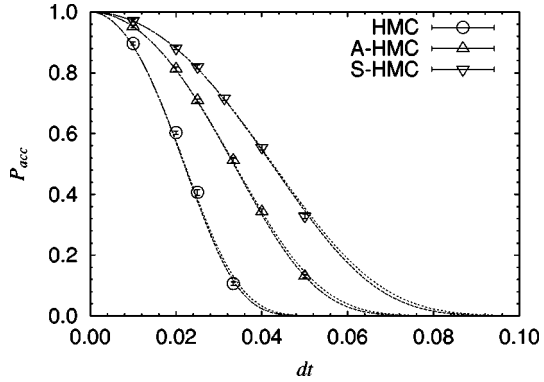

FIG. 4. MD step size dependence of the acceptance for preconditioned and unpreconditioned effective actions. The dashed lines show the function $\mathrm{erfc}[\sqrt{\pi}(a\cdot dt)^2/2]$ with a obtained from Fig. 3. The dotted lines are approximations $\exp[-(a\cdot dt)^2-(a\cdot dt)^4/2]$.

acceptance is greatly improved as shown in Fig. 4. For instance, at $dt=0.02$ P_{acc} is 81% (88%) for A-HMC (S-HMC) compared to 60% for the unpreconditioned case.

The efficiency of the algorithm may be defined as $P_{acc}\cdot dt$ following Ref. [25]. In order to plot the efficiency $P_{acc}\cdot dt$ as a function of P_{acc}, we make use of an approximation of P_{acc}:

$$P_{acc}=\exp\left(-(a\cdot dt)^2-\frac{1}{2}(a\cdot dt)^4\right). \tag{23}$$

This approximation is valid for small dt [up to $O(dt^6)$] and $\langle dH\rangle=\pi(a\cdot dt)^4$. The validity can be ascertained in Fig. 4, where the approximation Eq. (23) is plotted (dotted curve) as well as the exact one $\mathrm{erfc}[\sqrt{\pi}(a\cdot dt)^2/2]$ (dashed curve). Solving Eq. (23) for dt, we obtain the explicit functional form for the efficiency $P_{acc}\cdot dt$ as

$$P_{acc}\cdot dt=\frac{P_{acc}}{a}\sqrt{\sqrt{1-2\log(P_{acc})}-1}, \tag{24}$$

where the only parameter is a defined through $\langle dH\rangle=\pi(a\cdot dt)^4$. In Fig. 5 we plot the efficiency $P_{acc}\cdot dt$ as a function of P_{acc}, and Eq. (24) is plotted as a dotted line. It is remarkable that the optimal efficiency is reached when $P_{acc}\simeq 0.65$ irrespective of details of the algorithm as far as we use the simplest leapfrog integrator for the MD evolution [25].[1] The efficiency of the algorithm can be measured by the parameter a. We therefore conclude that the efficiency of the A-HMC is a factor 1.5 better than the unpreconditioned HMC on the ''small heavy'' lattice, and that of S-HMC by a factor of 1.9 which is even better.

[1]In Ref. [25] the maximum efficiency is reached at $P_{acc}\simeq 0.61$ rather than 0.65. This difference comes from the expansion Eq. (23) of the erfc function: the author of Ref. [25] considered the lowest order only, while we include the second order.

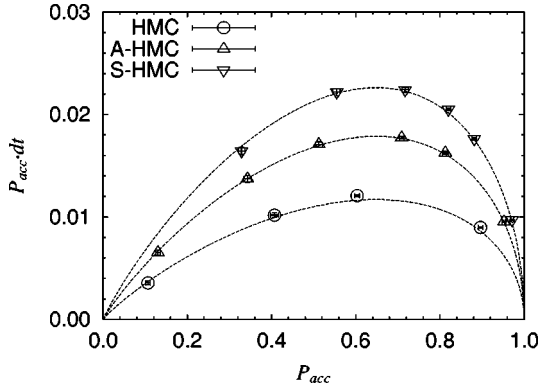

FIG. 5. Efficiency $P_{acc}\cdot dt$. The lines show the function $P_{acc}\sqrt{\sqrt{1-2\log P_{acc}}-1}/a$ with a obtained in Fig. 3.

D. Reversibility

Before we extend the comparison of the preconditioning to the large $(16^3\times 48)$ lattice, we describe our choice of the stopping condition for the Wilson-Dirac operator inverter on the large lattice, since it is computationally not realistic to keep the very strict conditions of Sec. III C for the large lattice size. The stopping condition in the calculation of the force may be relaxed as far as the reversibility condition is maintained, which is tested in the following. In this section we employ the ''S-HMC'' preconditioning to investigate the reversibility.

As a measure of how far one may loosen the stopping condition, we use the violation of the reversibility condition for the effective Hamiltonian defined by

$$|\Delta H|=|H(t_r-t_r)-H(0)|, \tag{25}$$

where $H(t_r-t_r)$ means the effective Hamiltonian calculated for the reversed configuration which is obtained from the initial configuration at $t=0$ by integrating the equation of motion to $t=t_r$ and then integrating back to $t=0$. The length of trajectory is $t_r=1$. For the S-HMC effective action, we measure $|\Delta H|$ for several values of the stopping condition on 20 thermalized configurations separated by 10 trajectories. Figures 6 and 7 show $\langle|\Delta H|/H\rangle$ measured on the ''large heavy'' and ''large light'' lattices, respectively. While the

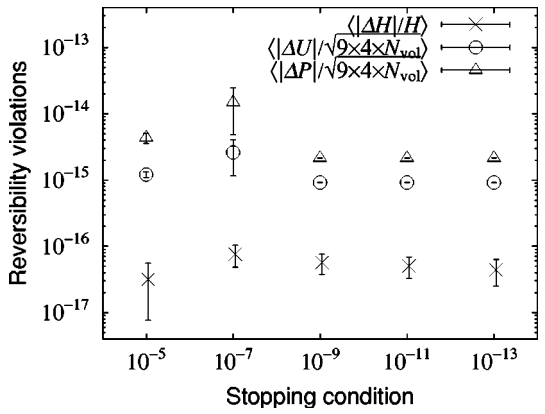

FIG. 6. The violation of the reversibility as a function of the stopping condition on the large heavy lattice.

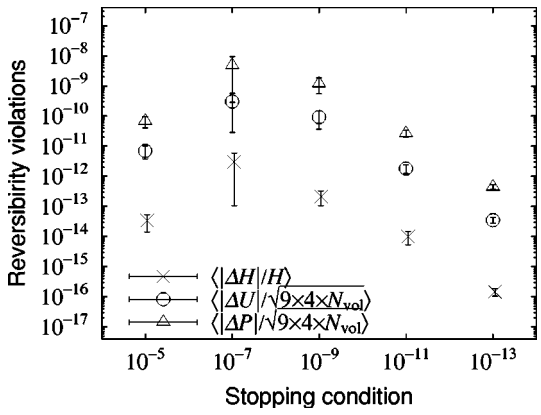

FIG. 7. Same as Fig. 6 but for the large light lattice.

violation stays around the limit of the double precision arithmetic for the heavy dynamical quark (Fig. 6), it depends on the stopping condition for the light dynamical quark (Fig. 7).

The behavior for the light quark mass can be understood as follows. If the initial vector in the BiCGStab solver is reversible ($x=b$ is adopted in this work), the only source of the reversibility violation is the round-off error in the numerical computation. Therefore, the error accumulates as the BiCGStab solver iterates and thus the violation increases as the stopping condition is tightened. This can be seen in Fig. 7 from $r=10^{-5}$ to 10^{-7}. As we further decrease the stopping condition, the BiCGStab solver gives a solution vector with better accuracy, and the value of $\langle|\Delta H|/H\rangle$ is governed by the accuracy of the solution vector. It decreases as we tighten the stopping condition from $r=10^{-7}$ to 10^{-13}.

As criteria to choose the stopping condition, we demand that the solver iterates to the region where the $\langle|\Delta H|/H\rangle$ is governed by the accuracy of the solution vector and that the variation of the Hamiltonian over the trajectory dH is not distorted by the error of the solution vector. These criteria are satisfied for $r\leqslant 10^{-7}$, and we choose 10^{-8} in the following simulations in this work.

For completeness, we also calculate the violation of reversibility in the link variables U and the conjugate momenta P

$$|\Delta U| = \sqrt{\sum_{n,\mu,a,b} |(U_\mu)_{a,b}(n)(t_r-t_r) - (U_\mu)_{a,b}(n)(0)|^2},$$

$$|\Delta P| = \sqrt{\sum_{n,\mu,a,b} |(P_\mu)_{a,b}(n)(t_r-t_r) - (P_\mu)_{a,b}(n)(0)|^2}, \tag{26}$$

where the sum runs over all sites n, color indices a,b, and vector index μ. The results for $|\Delta U|$ and $|\Delta P|$ normalized by $\sqrt{9\times4\times N_{\rm vol}}$ with $N_{\rm vol}$ the total number of lattice site are also plotted in Figs. 6 and 7, where we observe the same pattern of the stopping condition dependence as that of $\langle|\Delta H|/H\rangle$.

Since the MD evolution is chaotic, the violation of reversibility due to the rounding error may grow exponentially [26,27]. The UKQCD Collaboration studied the reversibility for the same lattice action as ours (but with the asymmetric preconditioning) with similar lattice parameters. They confirmed the exponential instability when the stopping condition is too loose [28]. The stopping condition we adopt $r<10^{-8}$ is strict enough and no such problem emerges in our case. We also note that in Ref. [28] most of the numerical calculation is made with the single precision (32 bits) arithmetic, while we use double precision throughout this work.

TABLE III. Simulation with the HMC algorithm on the large heavy lattice.

	HMC	A-HMC	S-HMC
$N_{\rm MD}$	160	100	80
Stopping condition (force)	10^{-18} [a]	10^{-8}	10^{-8}
Stopping condition (Hamiltonian)	10^{-20} [a]	10^{-14}	10^{-14}
Trajectories	3000	1200	1200
$\langle dH\rangle$	0.144(15)	0.182(17)	0.187(28)
HMC acceptance	0.799(9)	0.764(12)	0.759(23)
Plaquette	0.52801(10)	0.52803(9)	0.52827(13)

[a]The residual is defnined by $|Ax-b|$ in the HMC case.

For the stopping condition in the Hamiltonian calculation, we keep a strict condition $r<10^{-14}$, since there is a large cancellation in the difference $dH=H(t_r)-H(0)$, and the accuracy of dH is essential for the Metropolis test to be correct.

E. Efficiency on large lattices

We list the simulation parameters used for A-HMC and S-HMC algorithms in Tables III ("large heavy") and IV ("large light"). HMC means without preconditioning. We observe that the number of MD steps is much reduced for the preconditioned HMC algorithm compared to the unpreconditioned one at the almost same acceptance rate. More precisely, using the relation $\langle dH\rangle=\pi(adt)^4$, we can compare the best efficiency of the algorithm which depends only on a as in Eq. (24). The gain is 1.5 from HMC to A-HMC (1.9 from HMC to S-HMC) on the large heavy lattice. For the

TABLE IV. Same as in Table III but for the large light lattice.

	HMC	A-HMC	S-HMC
$N_{\rm MD}$	200	125	100
Stopping condition (force)	10^{-18} [a]	10^{-8}	10^{-8}
Stopping condition (Hamiltonian)	10^{-20} [a]	10^{-14}	10^{-14}
Trajectories	3000	1200	850
$\langle dH\rangle$	0.313(23)	0.182(17)	0.218(22)
HMC acceptance	0.702(11)	0.724(13)	0.761(16)
Plaquette	0.53413(5)	0.53404(9)	0.53393(11)

[a]The residual is defnined by $|Ax-b|$ in the HMC case.

large light lattice it is 1.8 from HMC to A-HMC (2.2 from HMC to S-HMC). The effect of even-odd preconditioning becomes more significant for lighter quark masses (note that for infinite quark mass there is no choice for even-odd preconditioning and is no improvement). From our tests we conclude that the symmetrically even-odd preconditioning is again the best choice within the simple even-odd preconditioning. Further improvement of the HMC algorithm may be achieved by preconditioning the partition function with incomplete LU factorization type preconditioning [4,19,29,30]. Hereafter we employ the symmetrically even-odd preconditioned form for the $O(a)$-improved Wilson-Dirac operator and the QCD partition function to develop the PHMC algorithm.

IV. PHMC ALGORITHM FOR TWO-FLAVOR QCD

The use of the polynomial hybrid Monte Carlo (PHMC) algorithm is essential for the construction of an exact algorithm for odd number of flavors. Before going to the PHMC algorithm with odd number of quarks, we investigate and develop the PHMC algorithm with two-flavors of quarks. The PHMC algorithm in two-flavor QCD was first proposed by Frezzotti and Jansen [5,6]. Some numerical tests of its performance were made [6,31] on small lattices and used for the determination of $c_{\rm sw}$ [32] or the running coupling constant [33] with the Schrödinger functional method. In this section we perform further tests to explore the most effective choice of the polynomial and its degree with the PHMC algorithm in two-flavor QCD. The numerical simulation and its comparison to the HMC algorithm are carried out with the same lattice parameters employed in Sec. III.

A. Partition function

The partition function of two-flavor QCD with the symmetrical preconditioning in the PHMC algorithm is given by

$$\mathcal{Z}_{\rm PHMC}=\int \mathcal{D}U\,\mathcal{D}P\,\mathcal{D}\phi_o^\dagger\,\mathcal{D}\phi_o(\det[W_{oo}])^2 \times e^{-H_{\rm PHMC}[P,U,\psi_o]},$$

$$W_{oo}=\hat{D}^S_{oo}P_{N_{poly}}[\hat{D}^S_{oo}],$$

$$H_{\rm PHMC}[P,U,\phi_o]=\frac{1}{2}P^2+S_g[U]+S^S_{poly}[U,\phi_o]+S^S_{det}[U],$$

$$S^S_{poly}[U,\phi_o]=|P_{N_{poly}}[\hat{D}^S_{oo}]\phi_o|^2, \quad (27)$$

and $S^S_{det}[U]$ is the same as in Eq. (22). The difference from Eq. (22) is in the pseudo-fermion action $S^S_{poly}[U,\phi]$ and in the insertion of the correction factor $(\det[W_{oo}])^2$.

The polynomial $P_{N_{poly}}[z]=\Sigma_{i=0}^{N_{poly}}c_i(z-1)^i$ approximates $1/z$ for a complex z placed in the convergence region as $N_{poly}\to\infty$, and the coefficients c_i are chosen so as to make the correction factor $(\det[W_{oo}])^2$ as close to unity as possible for small N_{poly}. For this purpose, several polynomials have been investigated in the literature. They include Chebyshev [7], least-squares [34], and adopted (with or without the UV filtering) [35,36] polynomials. We consider the Chebyshev and adopted polynomials in this work.

The Chebyshev polynomial with unit circle convergence domain in the complex plane is defined by $c_i=(-1)^i$. This is the same as the Taylor expansion with respect to the hopping matrix. We call the PHMC algorithm with the Chebyshev polynomial as C-PHMC. In this case, the accuracy of the polynomial is characterized by $|zP_{N_{poly}}[z]-1|=(z-1)^{N_{poly}+1}$.

The coefficients c_i for the adapted polynomial are determined so as to minimize $|\hat{D}^S_{oo}P_{N_{poly}}[\hat{D}^S_{oo}]\eta_o-\eta_o|^2$ with a Gaussian noise vector η_o with unit variance on a thermalized background gauge configuration [35]. The coefficients thus obtained do neither show a strong dependence on the background gauge configuration nor on the noise vector. We call this choice as A-PHMC. We note that the adapted polynomial with the UV filtering is simple and proven to be more efficient for the unimproved Wilson fermion in the multiboson algorithm [35,36]. For the $O(a)$-improved Wilson fermion, on the other hand, the UV filtering requires an additional term in the effective Hamiltonian, and to our knowledge its efficiency has not been tested yet. We leave it as a future subject to study the efficiency of the UV filtering for the PHMC algorithm with the $O(a)$-improved Wilson fermion action.

B. Force calculation

The molecular dynamics step in the PHMC algorithm requires a calculation of the force $\delta S^S_{poly}[U,\phi_o]/\delta U_\mu(n)$. Frezzotti and Jansen [5,6] proposed to use a product representation of the polynomial

$$P_{N_{poly}}[z]=\sum_{i=0}^{N_{poly}}c_i(z-1)^i=c_{N_{poly}}\prod_{k=1}^{N_{poly}}(z-z_k), \quad (28)$$

with z_k the roots of $P_{N_{poly}}[z]=0$. The computational cost can be reduced by holding the intermediate vectors obtained by multiplying monomials on the pseudofermion field.

In the product representation with a naive ordering of monomials, however, there is a problem of numerical instability and accumulation of round-off errors due to the fact that the partial product $\Pi^l_{k=1}(z-z_k)$ fluctuates by several orders of magnitude for intermediate l [37]. The problem becomes more severe for small z and for large order of the polynomial. Bunk *et al.* proposed some ordering schemes to minimize the problem [37]. In this work, instead of the product representation, we consider the Clenshaw's recursive relation [38]

$$P_{N_{poly}}[z]=c_0\Bigg[1+\frac{c_1}{c_0}(z-1)\Bigg[1+\frac{c_2}{c_1}(z-1)\times\Bigg[\cdots\Bigg[1+\frac{c_{N_{poly}}}{c_{N_{poly}-1}}(z-1)\Bigg]\cdots\Bigg]\Bigg]\Bigg], \quad (29)$$

for which the numerical instability is suppressed, because we add from the term giving the smallest contribution to the terms with larger contributions. We assume that the polynomial is converging and the higher order terms are smaller; otherwise the algorithm does not work efficiently.

Adopting this representation, we expect that the calculation of the pseudofermion action S^S_{poly} in Eq. (27) before and after the MD evolution become stable numerically. The force from $\delta S^S_{poly}[U,\phi_o]$ is given by

$$\delta S^S_{poly} = \left\{ \sum_{j=1}^{N_{poly}} [X^{P(j)\dagger} \delta M Y^{P(j)} + X^{P(j)\dagger} \delta T Z^{P(j)}] \right\} + \text{H.c.}, \tag{30}$$

where

$$X^{P(j)} = \begin{pmatrix} -(1+T)^{-1}_{ee} M^{\dagger}_{oe} \hat{X}^{P(j)}_o \\ \hat{X}^{P(j)}_o \end{pmatrix}, \tag{31}$$

$$Y^{P(j)} = \begin{pmatrix} -(1+T)^{-1}_{ee} M_{eo} \hat{Y}^{P(j)}_o \\ \hat{Y}^{P(j)}_o \end{pmatrix}, \tag{32}$$

$$Z^{P(j)} = \begin{pmatrix} -(1+T)^{-1}_{ee} M_{eo} \hat{Y}^{P(j)}_o \\ (1+T)^{-1}_{oo} M_{oe} (1+T)^{-1}_{ee} M_{eo} \hat{Y}^{P(j)}_o \end{pmatrix}, \tag{33}$$

$$\hat{X}^{P(j)}_o = (1+T)^{-1}_{oo} [(\hat{D}^S_{oo} - 1)^{\dagger}]^{j-1} \hat{Y}^{P(0)}_o, \tag{34}$$

$$\hat{Y}^{P(j)}_o = c_j \left[1 + \frac{c_{j+1}}{c_j} (\hat{D}^S_{oo} - 1) \left[1 + \frac{c_{j+2}}{c_{j+1}} (\hat{D}^S_{oo} - 1) \times \left[1 + \frac{c_{j+3}}{c_{j+2}} (\hat{D}^S_{oo} - 1) \times \left[\cdots \times \left[1 + \frac{c_{N_{poly}}}{c_{N_{poly}-1}} (\hat{D}^S_{oo} - 1) \right] \cdots \right] \right] \right] \right] \phi_o. \tag{35}$$

In our implementation of the simulation code, we first calculate $\hat{Y}^{P(j)}_o$ from $j = N_{poly}$ to 0 and store them on memory. We then calculate $\hat{X}^{P(j)}_o$ and the force from $j=1$ to N_{poly} using the stored $\hat{Y}^{P(j)}_o$. We do not need to store $\hat{X}^{P(j)}_o$. The requirement for memory is therefore the same as in the product representation used in Refs. [5,6].

A potential source of the round-off errors in the calculation of the force is the sum over j in Eq. (30), because the sum runs from the highest order to the lowest order in κ, the jth term being of order $\kappa^{2(j-1)}$. The numerical problem in the calculation of the force may be checked by looking at the violation of reversibility. We expect that the reversibility violation is small compared to HMC because the MD evolution involves no iterative processes and is completely deterministic. Numerical stability of the summation representation of polynomial and the reversibility of the molecular dynamics will be discussed in Sec. IV E.

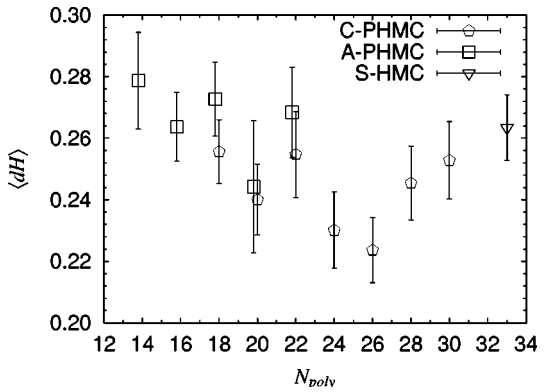

FIG. 8. N_{poly} dependence of $\langle dH \rangle$ on the small heavy lattice. $\langle dH \rangle$ with the S-HMC algorithm is also plotted on the most right side in the figure for comparison.

The pseudofermion field ϕ_o is generated from the Gaussian noise vector η_o at the beginning of the molecular dynamics step through

$$\phi_o = P_{N_{poly}} [\hat{D}^S_{oo}]^{-1} \eta_o = \hat{D}^S_{oo} W^{-1}_{oo} \eta_o. \tag{36}$$

Since W_{oo} is a matrix close to the identity matrix, the inversion of W_{oo} is easily performed by any iterative solver within a few iterations. We use the BiCGStab solver in our implementation.

C. Noisy Metropolis test for the correction factor

In order to construct an exact algorithm, we have to take account of the correction factor $(\det[W_{oo}])^2$. We use the noisy Metropolis test method of Kennedy and Kuti [39], which was previously applied to make the multiboson algorithm exact in Refs. [7,16].

After a trial configuration U' is accepted by the HMC Metropolis test, we make another Metropolis test for the correction factor. Generating a Gaussian vector χ_o with unit variance and zero mean, the probability $P_{corr}[U \to U']$ to accept the trial configuration is given by

$$P_{corr}[U \to U'] = \min[1, e^{-dS}], \tag{37}$$

where

$$dS = |(W_{oo}[U'])^{-1} W_{oo}[U] \chi_o|^2 - |\chi_o|^2. \tag{38}$$

The inverse $(W_{oo}[U'])^{-1}$ is calculated using the BiCGStab solver as in the generation of the pseudofermion field ϕ_o.

D. Numerical test of the efficiency

The total acceptance rate P_{total} of the PHMC algorithm is a product of that of the HMC Metropolis test P_{acc} and of the noisy Metropolis test P_{corr}. In this section we present several numerical tests on how P_{acc} and P_{corr} depend on parameters of the algorithm, such as the order of polynomial, and the MD step size. The simulations are made on the small heavy lattice, whose parameters are summarized in Table I.

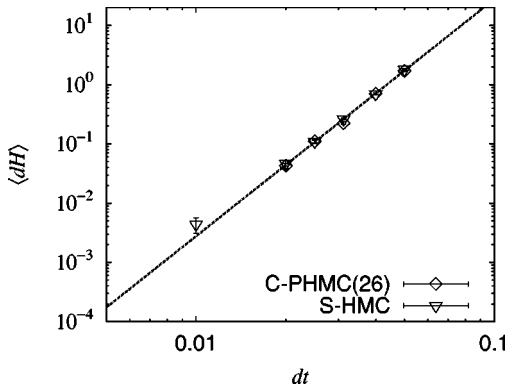

FIG. 9. $\langle dH \rangle$ versus dt with C-PHMC(26) and S-HMC algorithms.

For the choice of polynomial type, we test the (non-Hermitian) Chebyshev polynomial (C-PHMC) and the adapted polynomial [35,36] (A-PHMC). The order of the polynomial tested is listed in Table II.

We first study the N_{poly} dependence of P_{acc}. In Fig. 8 we show $\langle dH \rangle$ as a function of N_{poly} on the small heavy lattice at a fixed $dt=1/32$ ($N_{poly}=32$) for both C-PHMC and A-PHMC algorithms. We find that $\langle dH \rangle$ is almost independent of N_{poly} and agrees with the same quantity for the S-HMC algorithm. This is expected if the effective action of the PHMC algorithm approximates the original action well, because the PHMC replaces $[\hat{D}^S_{oo}]^{-1}$ by a polynomial $P_{N_{poly}}[\hat{D}^S_{oo}]$ and the two are equivalent if the polynomial is a good approximation of the inverse. The MD step size dependence of $\langle dH \rangle$ is plotted in Fig. 9 for the usual S-HMC and for the C-PHMC with $N_{poly}=26$ [which we call C-PHMC(26)], where we find good agreement among different algorithms. This means that the acceptance P_{acc} in PHMC is almost the same as that in the usual HMC.

In contrast, P_{corr} is expected to be sensitive to N_{poly}. Since the acceptance rate P_{corr} is directly related to the expectation value of dS as defined in Eq. (38), we measure the dependence of $\langle dS \rangle$ on N_{poly}. Figure 10 shows the plot for C-PHMC and A-PHMC at a fixed $dt=1/32$ ($N_{poly}=32$). The dotted lines represent a fit with an exponential form [16]

$$\langle dS \rangle = \pi a^2 \exp(-2bN_{poly}). \tag{39}$$

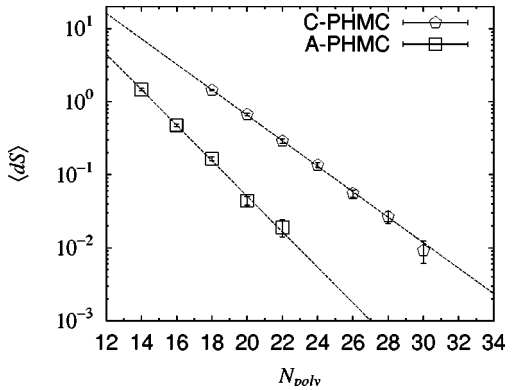

FIG. 10. $\langle dS \rangle$ versus N_{poly} in the PHMC algorithm. The lines show a fit function $\langle dS \rangle = \pi a^2 \exp(-2bN_{poly})$.

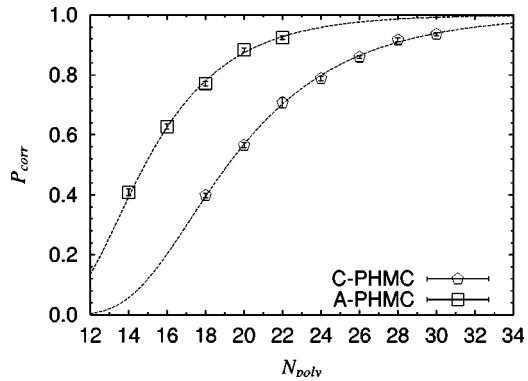

FIG. 11. P_{corr} as a function of N_{poly}. The lines represent $P_{corr}=\mathrm{erfc}[\sqrt{\pi}a \exp(-bN_{poly})/2]$ with a and b are obtained from a fit in Fig. 10.

The exponential form is expected because the error of the polynomial approximation behaves as $|zP_{N_{poly}}[z]-1|=(z-1)^{N_{poly}+1}$ in the Chebyshev polynomial case. We find that the data are well described by the exponential form and that $\langle dS \rangle$ is much smaller for the adapted polynomial A-PHMC than that for the Chebyshev polynomial C-PHMC, which demonstrates the efficiency of the adapted polynomial.

The acceptance rate in the noisy Metropolis step P_{corr} is related to $\langle dS \rangle$ as $P_{corr}=\mathrm{erfc}(\langle dS \rangle^{1/2}/2)$. We then obtain a plot of P_{corr} as a function of N_{poly} in Fig. 11. The dotted curves represent

$$P_{corr}=\mathrm{erfc}\left(\frac{\sqrt{\pi}a}{2}\exp(-bN_{poly})\right), \tag{40}$$

with a and b the parameters in Eq. (39). We clearly see that A-PHMC requires a smaller polynomial order N_{poly} than C-PHMC to achieve the same acceptance rate. For instance, to obtain $P_{corr}\simeq 0.8$ we need $N_{poly}=24$ for C-PHMC while A-PHMC requires only $N_{poly}=18$.

The efficiency of the PHMC algorithm for the noisy Metropolis test step can be quantified by P_{corr}/N_{poly}, because the number of arithmetic operations is roughly proportional to N_{poly}. In Fig. 12 we plot P_{corr}/N_{poly} against P_{corr}, and find that the A-PHMC is about 30% more efficient than

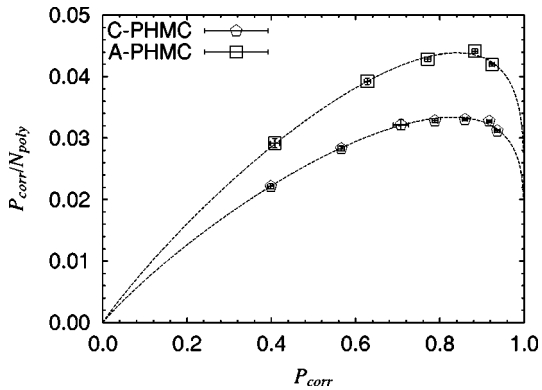

FIG. 12. Efficiency of the noisy Metropolis test P_{corr}/N_{poly}. See the text for details.

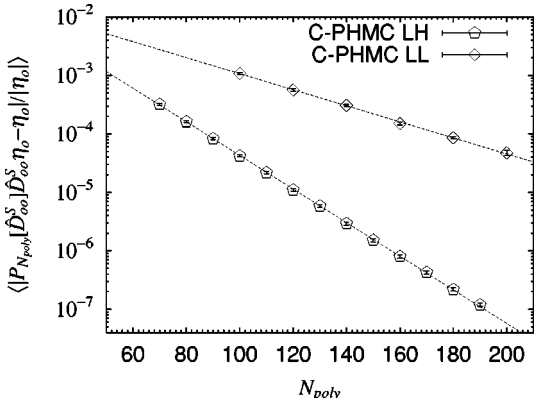

FIG. 13. N_{poly} dependence of the residual $\langle |P_{N_{poly}}[\hat{D}^S_{oo}]\hat{D}^S_{oo}\eta_o - \eta_o|/|\eta_o| \rangle$.

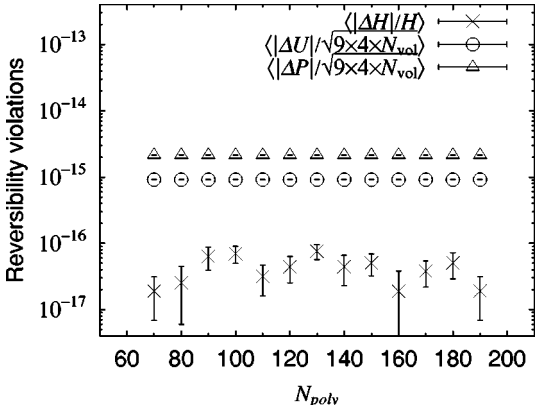

FIG. 14. N_{poly} dependence of the reversibility violation on the large heavy lattice.

C-PHMC. We also find that the efficiency peaks around $P_{corr}=0.85$. From Eq. (40) we obtain

$$\frac{P_{corr}}{N_{poly}} = \frac{1}{b}\frac{P_{corr}}{-\log\left\{\frac{1}{a}[\sqrt{1-2\log(P_{corr})}-1]\right\}} \tag{41}$$

using an expansion $\mathrm{erfc}(x)=\exp\{-(2/\sqrt{\pi})[x+(x^2/\sqrt{\pi})]\}+O(x^3)$. The efficiency is proportional to $1/b$, which controls the exponential fall off of $\langle dS \rangle$ as in Eq. (39), and the position of the maximum efficiency depends on a. When $a \sim O(10)$, as we observe in these tests, the maximum appears around $P_{corr}=0.85$ and it moves to larger values of P_{corr} as a becomes larger. Since a is expected to scale as $V^{1/2}(\kappa/\kappa_c)$ [16], the maximum efficiency is obtained for $P_{corr}>0.85$ when the lattice volume becomes larger or when the sea quark becomes lighter.

E. PHMC on large lattices

The PHMC algorithm works well with a reasonable order of the polynomial on the small heavy lattice. It is not trivial, however, whether it really works on larger lattices, because we expect that a polynomial with much larger order is needed.

For a numerical test on the ''large heavy'' and ''large light'' lattices (Table I) we consider the Chebyshev polynomial (C-PHMC) only, since we were not able to obtain an optimized polynomial for the A-PHMC. The reason is that the minimization of $|\hat{D}^S_{oo}P_{N_{poly}}[\hat{D}^S_{oo}]\eta_o - \eta_o|^2$ with respect to the coefficients of polynomial failed to converge for polynomials of a large ($\sim$100) order which are needed for these large lattices. This is likely a problem of the steepest descent algorithm used in the minimization, and not a fundamental difficulty of the adapted polynomial. We leave a resolution of this problem to future studies.

We first consider the question of how the polynomial approximation of $[\hat{D}^S_{oo}]^{-1}$ works for reasonably large lattices. To investigate this we define a residual

$$\frac{|P_{N_{poly}}[\hat{D}^S_{oo}]\hat{D}^S_{oo}\eta_o - \eta_o|}{|\eta_o|}, \tag{42}$$

with a Gaussian noise vector η_o. We expect that the residual becomes exponentially smaller as N_{poly} increases, if the polynomial provides a good approximation of the inverse Dirac matrix. We measure this quantity on 20 thermalized configurations of large heavy and large light lattices and plot it as a function of N_{poly} in Fig. 13. For both heavy and light dynamical quarks, we find a clear exponential decrease, while the slope significantly depends on the sea quark mass. We also note that the polynomial approximation is not distorted by the round-off error even for $N_{poly}\sim$100–200.

When the order of polynomial is large, another important test is the check of the reversibility in the MD steps. As we mentioned in Sec. IV B, our implementation of the force calculation may cause round-off errors. As in Sec. III D we investigate the violation of reversibility in $\langle |\Delta H|/H \rangle$, $\langle |\Delta U| \rangle$, and $\langle |\Delta P| \rangle$ by measuring these quantities on the same 20 configurations. The results are plotted in Figs. 14 and 15, for large heavy and large light, as a function of N_{poly}. We observe no dependence on N_{poly} for both lattices and the violation of reversibility remains close to the limit of the double precision arithmetic. This implies that the Clenshaw-type representation of the polynomial Eq. (29) adopted in our implementation of the PHMC algorithm does not accumulate round-off errors even for large N_{poly}. We

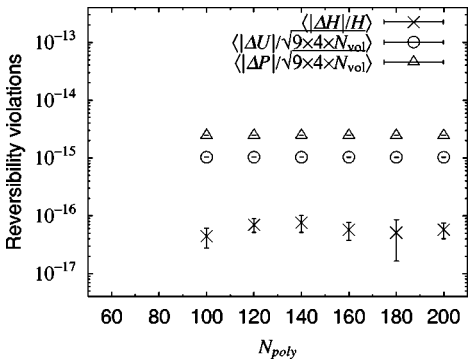

FIG. 15. Same as Fig. 14 but for the large light lattice.

TABLE V. Simulation with the C-PHMC algorithm on the large heavy lattice.

	C-PHMC(70)	C-PHMC(80)	C-PHMC(90)
N_{MD}	80	80	80
Stopping condition[a]	10^{-14}	10^{-14}	10^{-14}
Trajectories	1300	1000	1000
$\langle dH \rangle$	0.151(21)	0.187(22)	0.154(31)
HMC acceptance	0.775(17)	0.763(23)	0.787(21)
$\langle dS \rangle$	0.244(32)	0.069(24)	0.013(7)
Correction acceptance	0.731(20)	0.851(21)	0.930(13)
Total acceptance	0.568(17)	0.658(27)	0.732(30)
Plaquette	0.52803(11)	0.52809(10)	0.52809(10)

[a]This is used for the generation of the pseudofermion field and the calculation of the correction factor.

also emphasize that the violation is much smaller than in the usual HMC plotted in Figs. 6 and 7. In the HMC algorithm the number of arithmetic operations can be different between forward and backward steps, because the convergence of the BiCGStab solver is controlled by the condition that the residual is smaller than a certain value. We suspect that the reversibility becomes better if the number of iterations (thus the number of arithmetic operations) is fixed in the solver. Even if this is the case, the numerical stability is not optimized in the BiCGStab solver, and the PHMC is still expected to perform better regarding the reversibility.

We then measure the actual efficiency on large lattices. The simulation parameters and some results are summarized in Tables V and VI for heavy and light dynamical quarks. We plot $\langle dS \rangle$ and $P_{corr} = \mathrm{erfc}(\langle dS \rangle^{1/2}/2)$ as functions of N_{poly} in Figs. 16 and 17. Compared to the small lattice, substantially larger N_{poly} are needed to keep the acceptance rate at reasonably large values.

Furthermore, $\langle dS \rangle$ and the acceptance depends substantially on the sea quark mass. As discussed in Ref. [16] the parameter b, which parametrizes the slope of $\langle dS \rangle$, is expected to be proportional to the quark mass. This expectation is confirmed in our simulations: the ratio of the quark masses in the two simulations is 2.04(6), while the ratio of b is 2.15(15).

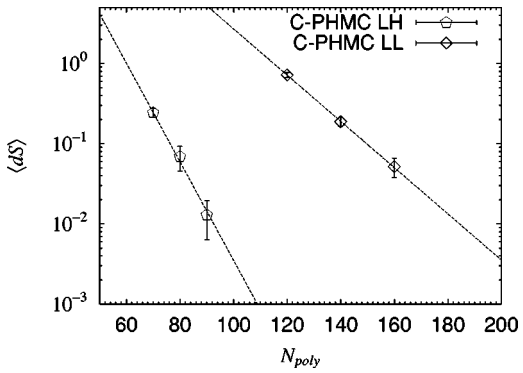

FIG. 16. $\langle dS \rangle$ versus N_{poly} for the large heavy (pentagons) and large light (diamonds) lattices.

The efficiency of the noisy Metropolis step P_{corr}/N_{poly} is plotted in Fig. 18. The maximum efficiency is achieved around $P_{corr} = 0.9$, and the height at the maximum is lower for the lighter quark mass than that for the heavier one by about a factor of two, as we expected from the ratio of b [and from Eq. (41)].

Finally, we compare the total efficiency of the PHMC algorithm with that of the usual HMC. The efficiency is parametrized as $P_{total}/[N_{\mathrm{Mult}}/\mathrm{traj}]$, which is plotted in Fig. 19 against the total acceptance ratio P_{total}. The total acceptance ratio P_{total} of the PHMC algorithm is defined by $P_{total} = P_{acc} P_{corr}$; for the HMC algorithm it is $P_{total} = P_{acc}$. The number of hopping matrix multiplications to cover a unit trajectory, $[N_{\mathrm{Mult}}/\mathrm{traj}]$, is counted in the program. The efficiency of PHMC is slightly better than the usual HMC for both heavy and light dynamical quarks. We note that the efficiency of HMC depends substantially on the stopping condition imposed. As we discussed in Sec. III D, we carefully chose the stopping condition for HMC, but the remaining violation of the reversibility is still large compared to the PHMC. Therefore, in order to guarantee the exactness of the algorithm strictly, a strict stopping condition is required and then the efficiency of HMC becomes much lower.

V. PHMC ALGORITHM FOR AN ODD NUMBER OF FLAVORS

In this section we describe an extension of the PHMC algorithm to the case of odd number of flavors. As we al-

TABLE VI. Same as Table V but for the large light lattice.

	C-PHMC(120)	C-PHMC(140)	C-PHMC(160)
N_{MD}	100	100	100
Stopping condition[a]	10^{-14}	10^{-14}	10^{-14}
Trajectories	1600	1200	1100
$\langle dH \rangle$	0.197(18)	0.243(43)	0.194(20)
HMC acceptance	0.750(12)	0.768(13)	0.765(14)
$\langle dS \rangle$	0.719(48)	0.188(17)	0.052(14)
Correction acceptance	0.563(18)	0.770(12)	0.886(19)
Total acceptance	0.422(14)	0.597(15)	0.678(17)
Plaquette	0.53417(11)	0.53396(18)	0.53411(15)

[a]This is used for the generation of the pseudofermion field and the calculation of the correction factor.

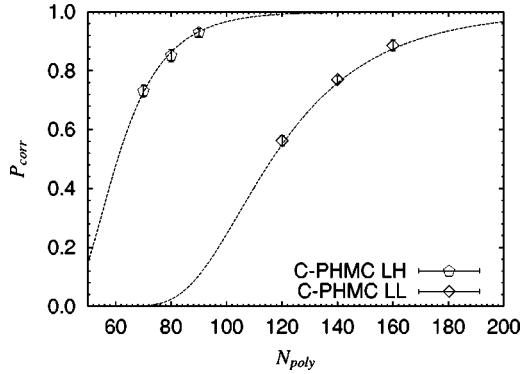

FIG. 17. P_{corr} versus N_{poly} for the large heavy (pentagons) and large light (diamonds) lattices.

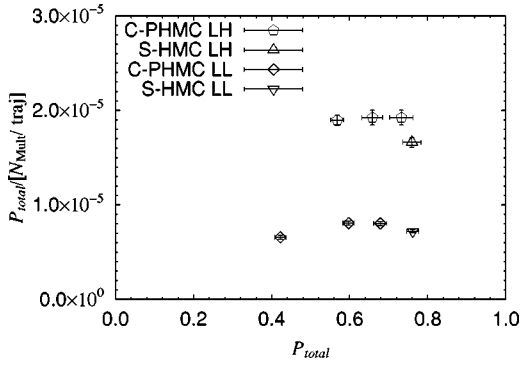

FIG. 19. Total efficiency $P_{total}/[N_{\rm Mult}/\text{traj}]$ as a function of P_{total}.

ready outlined in Sec. II, the algorithm is almost the same as that for even number of flavors, except for the polynomial in the evaluation of the correction factor. We make a numerical check of the algorithm by comparing the simulation of $1+1$-flavor QCD with the two-flavor case simulated with the HMC and PHMC algorithms. In addition we carry out a simulation of $2+1$-flavor QCD and compare the results with that obtained by the *R* algorithm.

A. PHMC for one-flavor QCD

In order to construct a real and positive definite effective action for one-flavor of dynamical quark, we use the trick proposed by Boriçi and de Forcrand [7] and Alexandrou *et al.* [8], which was already described in Sec. II B.

A polynomial of even degree $P_{N_{poly}}[z]$ can be split into the product of two polynomials $T_{N_{poly}}[z]$ and $\overline{T}_{N_{poly}}[z]$ as

$$P_{N_{poly}}[z]=T_{N_{poly}}[z]\overline{T}_{N_{poly}}[z], \tag{43}$$

$$T_{N_{poly}}[z]\equiv\sum_{i=0}^{N_{poly}/2} d_i(z-1)^i, \tag{44}$$

$$\overline{T}_{N_{poly}}[z]\equiv\sum_{i=0}^{N_{poly}/2} d_i^*(z-1)^i. \tag{45}$$

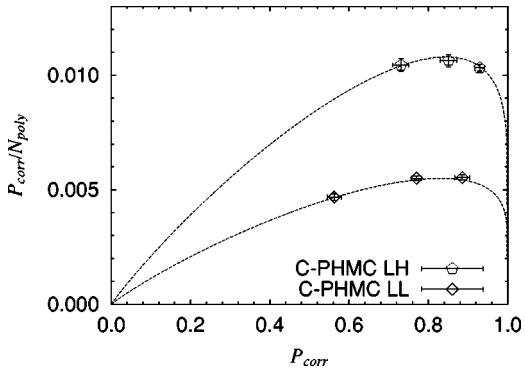

FIG. 18. Efficiency P_{corr}/N_{poly} versus P_{corr} on the large heavy (pentagons) and large light (diamonds) lattices.

Note that here we use the summation representation for $T_{N_{poly}}[z]$ ($\overline{T}_{N_{poly}}[z]$) instead of the product representation as in Eq. (7). The coefficients d_i in $T_{N_{poly}}[z]$ are determined as follows. First, we consider the product representation of $P_{N_{poly}}[z]$ as $P_{N_{poly}}[z]=c_{N_{poly}}\Pi_{k=1}^{N_{poly}}(z-z_k)$. The ordering of the monomials is defined so that $\arg(z_k-1)$ increases monotonically with increasing k. Since the roots z_k appear with their complex conjugate, we find $z_k=z^*_{N_{poly}+1-k}$ ($k=1\ldots N_{poly}/2$). We then split the polynomial into the product of two polynomial as $P_{N_{poly}}[z]=c_{N_{poly}}\Pi_{j=1}^{N_{poly}/2}(z-z_{k(j)})(z-z^*_{k(j)})$, where the reordering index $k(j)$ is defined by $k(j)=2j-1$. Then we obtain a "square root" of the polynomial as $T_{N_{poly}}[z]=\sqrt{c_{N_{poly}}}\Pi_{j=1}^{N_{poly}/2}(z-z_{k(j)})$, from which we arrive at the polynomial representation Eq. (44) by expanding the product representation. Since we do not use the product representation of $T_{N_{poly}}[z]$ in the numerical simulation, the problem of the ordering of monomials is irrelevant as long as one uses long enough decimal precision or computer algebra systems to obtain the coefficients d_i.

We note that $T_{N_{poly}}[z]^\dagger\neq\overline{T}_{N_{poly}}[z]$ for complex z, but for the determinant of the Wilson-Dirac operator D one can prove the relation

$$\det[T_{N_{poly}}[D]]^*=\det[\overline{T}_{N_{poly}}[D]], \tag{46}$$

using the γ_5 hermiticity property $D^\dagger=\gamma_5 D\gamma_5$. It follows that

$$\begin{aligned}\det[P_{N_{poly}}[D]]&=\det[\overline{T}_{N_{poly}}[D]]\cdot\det[T_{N_{poly}}[D]]\\&=|\det[T_{N_{poly}}[D]]|^2.\end{aligned} \tag{47}$$

For the preconditioned case, the Hermiticity is modified to $\hat{D}^{S}_{oo}{}^\dagger=\gamma_5(1+T)_{oo}\hat{D}^S_{oo}(1+T)^{-1}_{oo}\gamma_5$, for which Eq. (46) holds as well.

The partition function for one-flavor QCD can be written as

$$\mathcal{Z}=\int \mathcal{D}U\,\mathcal{D}P\,\mathcal{D}\phi_o^\dagger\,\mathcal{D}\phi_o \det[W_{oo}] \times e^{-H_{\rm PHMC}[P,U,\phi_o]},$$

$$H_{\rm PHMC}[P,U,\phi_o]=\frac{1}{2}P^2+S_g[U]+S^S_{poly}[\phi_o]+S^S_{det}[U], \tag{48}$$

$$S^S_{poly}[\phi_o]=|T_{N_{poly}}[\hat{D}^S_{oo}]\phi_o]|^2,$$

$$S^S_{det}[U]=-(\log\det[1+T_{ee}]+\log\det[1+T_{oo}]).$$

The polynomial $P_{N_{poly}}[\hat{D}^S_{oo}]$ in the two-flavor case Eq. (27) is replaced by $T_{N_{poly}}[\hat{D}^S_{oo}]$. The correction factor $\det[W_{oo}]$ is the same as that defined in Eq. (27), but the exponent is 1.

Every step of the HMC part of the simulation is the same as the corresponding step in the two-flavor case, except that we use the polynomial $T_{N_{poly}}$ rather than $P_{N_{poly}}$. The pseudofermion field is similarly generated by

$$\phi_o=T_{N_{poly}}[\hat{D}^S_{oo}]^{-1}\eta_o=\overline{T}_{N_{poly}}[\hat{D}^S_{oo}]\hat{D}^S_{oo}W^{-1}_{oo}\,\eta_o, \tag{49}$$

with a Gaussian noise vector η_o at the beginning of each MD step. On the other hand, the noisy Metropolis step to incorporate the correction factor requires a special treatment, because the correction factor is not $(\det[W_{oo}])^2$ but $\det[W_{oo}]$.

B. Noisy Metropolis test for the one-flavor case

If the fermion determinant $\det[\hat{D}^S_{oo}]$ is positive, $\det[W_{oo}]$ is also positive and its square root is well defined. We calculate the square root of the matrix W_{oo} by solving the equation $A^2_{oo}=W_{oo}$ using the Taylor expansion

$$A_{oo}=1+\sum_k^\infty \frac{(2k-3)!!}{(2k)!!}\delta^k_{oo} =1+\frac{1}{2}\delta_{oo}-\frac{1}{8}\delta^2_{oo}+\frac{1}{16}\delta^3_{oo}\cdots \tag{50}$$

with $\delta_{oo}\equiv W_{oo}-1$, because we expect that W_{oo} is close to the identity matrix when the polynomial $P_{N_{poly}}[\hat{D}^S_{oo}]$ is a good approximation of $(\hat{D}^S_{oo})^{-1}$. We obtain

$$\det[W_{oo}]=|\det[A_{oo}]|^2, \tag{51}$$

using the (preconditioned) γ_5 Hermiticity property $A^\dagger_{oo}=\gamma_5(1+T)_{oo}A_{oo}(1+T)^{-1}_{oo}\gamma_5$.

Once we obtain the matrix A_{oo}, we can perform the noisy Metropolis test Eq. (37) replacing W_{oo} in Eq. (38) by A_{oo},

$$dS=|(A_{oo}[U'])^{-1}A_{oo}[U]\chi_o|^2-|\chi_o|^2. \tag{52}$$

The only complication is the use of the Taylor expansion Eq. (50) every time we need a multiplication with A_{oo}. For the inverse A^{-1}_{oo} we use another polynomial

$$A^{-1}_{oo}=1+\sum_k^\infty(-1)^k\frac{(2k-1)!!}{(2k)!!}\delta^k_{oo} =1-\frac{1}{2}\delta_{oo}+\frac{3}{8}\delta^2_{oo}-\frac{5}{16}\delta^3_{oo}\cdots. \tag{53}$$

In the numerical calculation, summation from the lower order to the higher should be avoided to reduce round-off errors. We therefore use the following (Clenshaw's type) expressions:

$$A_{oo}=\left[1+\frac{1}{2}\delta_{oo}\left[1+\frac{-1}{4}\delta_{oo}\left[1+\frac{-3}{6}\delta_{oo}\cdots\times\left[1+\frac{3-2k}{2k}\delta_{oo}\right]\cdots\right]\right]\right], \tag{54}$$

$$A'^{-1}_{oo}=\left[1+\frac{-1}{2}\delta'_{oo}\left[1+\frac{-3}{4}\delta'_{oo}\left[1+\frac{-5}{6}\delta'_{oo}\cdots\times\left[1+\frac{1-2k}{2k}\delta'_{oo}\right]\cdots\right]\right]\right]. \tag{55}$$

A shortcoming of this method is that we have to recalculate the entire expressions when we need to increase the order of truncation k in the Taylor expansion.

In order to avoid systematic errors from the truncation of the Taylor expansion, we monitor the residual

$$r_1=\frac{|A_{oo}[U](A_{oo}[U]\chi_o)-W_{oo}[U]\chi_o|}{|W_{oo}[U]\chi_o|}, \tag{56}$$

in the calculation of $A_{oo}[U]\chi_o$, and

$$r_2=\frac{|W_{oo}[U'](A_{oo}[U'])^{-1}\{(A_{oo}[U'])^{-1}\omega_o\}-\omega_o|}{|\omega_o|}, \tag{57}$$

in the calculation of $(A_{oo}[U'])^{-1}\omega_o$ with $\omega_o=A_{oo}[U]\chi_o$. We require that the residuals be smaller than 10^{-14} to keep the exactness of the algorithm. In the simulation program we always monitor the residuals, and when the residuals become larger than our condition we repeat the calculation increasing k until it becomes satisfied.

The necessary order of the Taylor expansion depends significantly on the order of polynomial N_{poly}. If N_{poly} is large enough, W_{oo} is very close to the identity and the Taylor expansion may be truncated at very low orders. Therefore, there is a complicated trade-off between N_{poly} and k to the computational cost in the algorithm. We consider briefly the computational cost to calculate the square root of the correction matrix and the noisy Metropolis acceptance probability as follows. In the case of the Chebyshev polynomial, the residual of the correction matrix is estimated as

$$W_{oo}-1=\delta_{oo}=(\hat{D}^S_{oo}-1)^{N_{poly}+1}. \tag{58}$$

If we take λ as the largest eigenvalue of $\hat{D}^S_{oo}-1$, this leads to $|\delta_{oo}|\simeq|\lambda|^{N_{poly}+1}$ where $|\lambda|<1$ is assumed. To keep the

TABLE VII. A comparison of the two- and $(1+1)$-flavor QCD simulations at $\beta=5.0$, $8^3\times16$, $\kappa=0.1415$, $c_{\rm sw}=1.855$.

	S-HMC $N_f=2$	C-PHMC(26) $N_f=2$	C-PHMC(26) $N_f=1+1$
$N_{\rm MD}$	32	32	32
Trajectories	5000	5000	5000
$\langle dH\rangle$	0.2634(107)	0.2236(106)	0.1262(70)
HMC acceptance	0.7172(79)	0.7444(70)	0.7994(78)
$\langle dS\rangle$ (quark 1)	-	0.0553(59)	0.0234(36)
Correction acceptance (quark 1)	-	0.8595(53)	0.9264(54)
$\langle dS\rangle$ (quark 2)	-	-	0.0167(37)
Correction acceptance (quark 2)	-	-	0.9370(58)
Total acceptance	0.7172(79)	0.6398(117)	0.6950(74)
Plaquette	0.43877(22)	0.43839(27)	0.43857(20)

residual of the square root Eq. (56) (for example) lower than a constant ϵ, we have the following inequality when the Taylor expansion is truncated at an order k:

$$r_1\propto|\delta_{oo}^{k+1}|=|\lambda|^{(k+1)(N_{poly}+1)}<C\epsilon, \tag{59}$$

with a coefficient C. Thus $(k+1)(N_{poly}+1)$ must be larger than a constant proportional to $\ln(\epsilon)$. When we fix ϵ as a stopping condition, the truncation order k is chosen so as to satisfy Eq. (59). The computational cost to calculate the square root of the correction matrix becomes a constant because the number of multiplication of $\hat{D}_{oo}^S$ is proportional to $k\times N_{poly}$, which is roughly $\sim(k+1)(N_{poly}+1)$. Consequently the total amount of the computational cost to calculate Eq. (52) becomes almost constant. Thus we conclude that the choice of N_{poly} does not affect the cost in the noisy Metropolis test, and that the efficiency of the whole algorithm is governed by the cost of the molecular dynamics step (proportional to N_{poly}) and by the acceptance rates of the HMC and the noisy Metropolis tests.

In order to evaluate the correction factor $\det[W_{oo}]$, Takaishi and de Forcrand [3] employed the idea of the unbiased stochastic estimator [40] using $\sqrt{\det[W_{oo}[U']^2/W_{oo}[U]^2]}=\sqrt{\langle e^{-dS}\rangle_{\chi_o}}$ from several estimates of $\langle e^{-dS}\rangle_{\chi_o}$ with dS defined in Eq. (38) for the $N_f=2$. Their method is faster than ours because they do not need to calculate the square root of the correction matrix as we did in Eqs. (50) and (53). On the other hand, the stochastic estimator may produce negative probabilities for the Metropolis test, which leads to systematic errors in the final results. In order to avoid this problem they keep dS sufficiently small with a high acceptance ratio so that the negative probabilities within a desired trajectory length do not appear. In our method these problems are avoided at the price of additional computational costs by taking explicitly the square root of the correction matrix.

C. Numerical test with $(1+1)$-flavor QCD

The algorithm for one-flavor of dynamical fermion can be tested by considering $(1+1)$-flavor QCD, which should be identical to two-flavor QCD. Since we already have results with established algorithms for two-flavor QCD, we check if we can reproduce the results with the $(1+1)$-flavor QCD simulation. For $(1+1)$ flavors, we introduce two sets of pseudofermion fields $\phi_o^{[f]}$ $(f=1,2)$ with the effective action $S_{poly}^S[\phi_o^{[f]}]=|T_{N_{poly}}[\hat{D}_{oo}^S]\phi_o^{[f]}|^2$. The correction factor $\det[W_{oo}]$ is evaluated twice with the noisy Metropolis test described in Sec. V B.

Simulation parameters and some results on our small heavy lattice are listed in Table VII. We employ the Chebyshev polynomial of order $N_{poly}=26$ both in the two-flavor simulation and in the $(1+1)$-flavor simulation with PHMC algorithms. Note that the order of the polynomial $T_{N_{poly}}$ in $N_f=1+1$ is $26/2=13$ by its definition for each pseudofermion. We also have a result with the standard S-HMC.

We observe in Table VII that the three algorithms give a consistent plaquette expectation value within the statistical error of less than 0.1%. It is evident that the algorithm for odd number of flavors works as we expected. The statistical error is evaluated with the binned jack-knife method and the bin size is increased until the error ceases to grow.

In the same table we find that $\langle dH\rangle$, which controls the HMC acceptance P_{acc}, is significantly smaller for the $(1+1)$-flavor simulation at the same MD step size dt. The size of $\langle dH\rangle$ depends on the precise form of the Hamiltonian we consider. While the formula described in Ref. [23] may be employed to examine this issue, we do not pursue it here because of the complication of the force contribution from the pseudofermion action. Note that this decrease of $\langle dH\rangle$ in the $N_f=1+1$ case does not immediately mean an increase of the efficiency. The reason is that we expect the duplication of the pseudofermion field to cause an extension of the autocorrelation time.

We find that the acceptance rate $P_{corr}^{N_f=2}$ in the correction factor for the two-flavor case is related to those of the $(1+1)$-flavor simulation as $P_{corr}^{N_f=2}\simeq(P_{corr}^{N_f=1})^2$. This property can be explained as follows: Expanding $dS^{N_f=2}$ in Eq. (38) in terms of $\delta_{oo}=W_{oo}[U]-1$ and $\delta'_{oo}=W_{oo}[U']-1$, we obtain $dS^{N_f=2}=2\,\mathrm{Re}[\chi_o^\dagger(\delta-\delta')_{oo}\chi_o]$ up to $O(\delta^2,\delta'^2,\delta\delta')$. On the other hand, $dS^{N_f=1}$ in Eq. (52) is expressed as

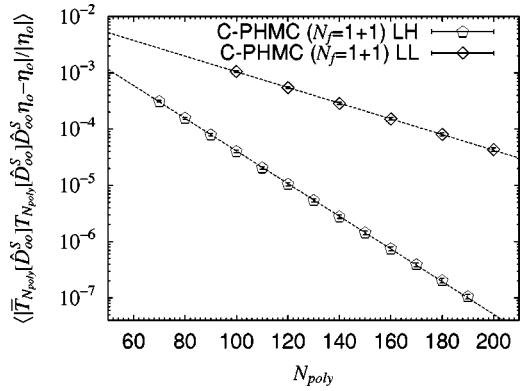

FIG. 20. N_{poly} dependence of the residual $\langle|\overline{T}_{N_{poly}}[\hat{D}^S_{oo}]T_{N_{poly}}[\hat{D}^S_{oo}]\eta_o-\eta_o|/|\eta_o|\rangle$ $(N_f=1+1)$.

$dS^{N_f=1}=\mathrm{Re}[\chi_o^\dagger(\delta-\delta')_{oo}\chi_o]$. Up to higher orders in δ_{oo} and δ'_{oo} we then obtain $dS^{N_f=2}\simeq 2dS^{N_f=1}$ and $P^{N_f=2}_{corr}\simeq(P^{N_f=1}_{corr})^2$.

We also test our algorithm on large heavy and large light lattices. The convergence of the polynomial $T_{N_{poly}}[\hat{D}^S_{oo}]$ and of the Taylor expansion of the correction factor is non-trivial on these large lattice sizes. To investigate the convergence of the polynomial $T_{N_{poly}}[\hat{D}^S_{oo}]$ we perform the same check as that made for $P_{N_{poly}}[\hat{D}^S_{oo}]$. In Fig. 20 we show the convergence behavior using

$$\frac{|\overline{T}_{N_{poly}}[\hat{D}^S_{oo}]T_{N_{poly}}[\hat{D}^S_{oo}]\hat{D}^S_{oo}\eta_o-\eta_o|}{|\eta_o|}, \qquad (60)$$

as the residual. Here η_o is a Gaussian noise vector and the measurement is made on 20 thermalized configurations separated by ten trajectories. Since $\overline{T}_{N_{poly}}[\hat{D}^S_{oo}]T_{N_{poly}}[\hat{D}^S_{oo}]$ should be $P_{N_{poly}}[\hat{D}^S_{oo}]$ by definition, Eq. (60) must be identical to Eq. (42) except for round-off errors. As shown in Fig. 20, Eq. (60) decreases exponentially as $N_{N_{poly}}$ increases, which is the same behavior as in Fig. 13. Thus we confirm that there is no unexpected accumulation of round-off errors

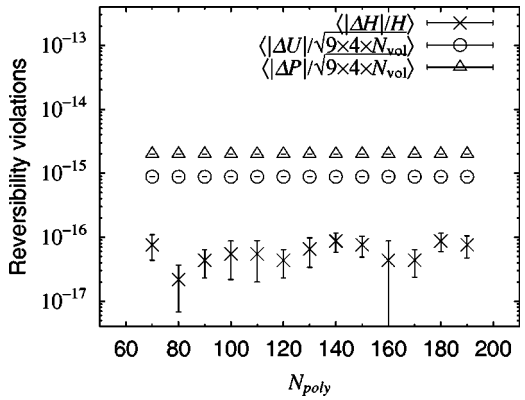

FIG. 21. N_{poly} dependence of the reversibility violation on the large heavy lattice $(N_f=1+1)$.

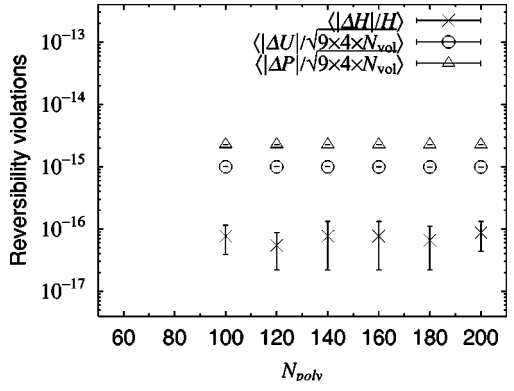

FIG. 22. Same as Fig. 21 but for the large light lattice.

in the calculation of $T_{N_{poly}}[\hat{D}^S_{oo}]$ with our choice of N_{poly} ($T_{N_{poly}}[\hat{D}^S_{oo}]$ is also evaluated with the Clenshaw's recurrence formula). The violation of reversibility is extremely small as plotted in Figs. 21 and 22. Their magnitude stays around the limit of the double precision arithmetic, which parallels our finding with the two-flavor case (Figs. 14 and 15).

Figures 23 (large heavy) and 24 (large light) show the convergence behavior of the Taylor expansion of the correction matrix as a function of the order of the expansion. The convergence is monitored with the residuals r_1 and r_2 defined in Eqs. (56) and (57), respectively. We also monitor the convergence of the weight dS defined in Eq. (52), by measuring $|dS-dS_{end}|$, where dS_{end} is the value of dS at the highest order of the expansion. These figures are also plotted with measurements on 20 configurations separated by 10 trajectories. Open symbols are obtained for the smallest N_{poly} (70 for large heavy, 100 for large light), and filled ones are for the largest N_{poly} (190 for large heavy, 200 for large light). The convergence of the residuals is almost exponential. The slope, however, becomes weaker near the limit of the double precision arithmetic. In the region where the exponential decay is observed, $k\times N_{poly}$ seems to behave as roughly constant irrespective of the choice of N_{poly}. This is the expected behavior discussed in Sec.V B. When the stopping condition for r_1 and r_2 is set to be 10^{-14}, the improve-

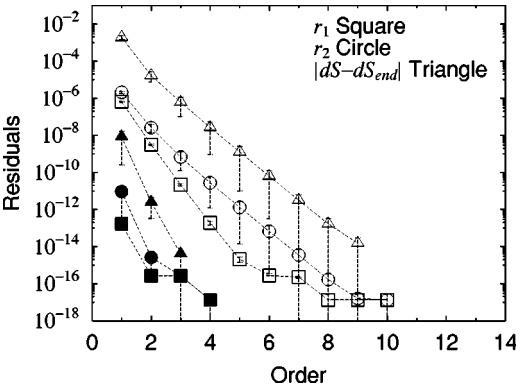

FIG. 23. Convergence behavior of the Taylor expansion of the correction matrix on the large heavy lattice. Open: $N_{poly}=70$; filled: $N_{poly}=190$.

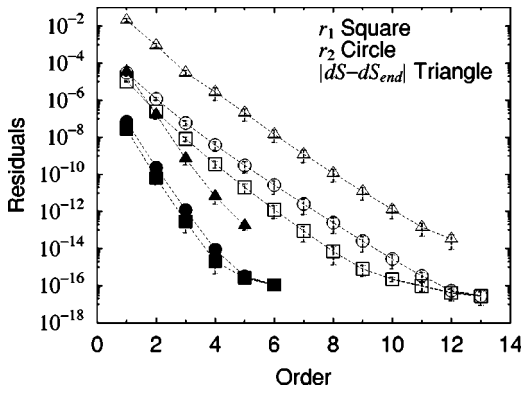

FIG. 24. Same as Fig. 23 but for the large light lattice. Open: $N_{poly}=100$; filled: $N_{poly}=200$.

ment of $|dS-dS_{end}|$ stops at $\sim 10^{-12}$. Since dS itself is of $O(10^{-2})$, we expect that dS has ~ 10 digits of significant figure, which we expect to be sufficient for current simulation trajectory lengths. The negative eigenvalue problem did not occur in these investigations, probably because of the intermediate quark mass we employed.

Table VIII shows the simulation statistics for C-PHMC with $N_f=1+1$ on both of the large lattices. We obtain results for the averaged plaquette value which are consistent with those for the $N_f=2$ case. The relation $P_{corr}^{N_f=2} \simeq (P_{corr}^{N_f=1})^2$ holds again for such large lattice sizes, and we did not encounter the negative eigenvalue problem during the long trajectories (~ 1000). We expect that the total efficiency has the same functional dependence on N_{poly} as that with the $N_f=2$ PHMC, since the behavior on N_{poly} is mostly ruled by the molecular dynamics. The actual value of the total efficiency is slightly worse than that with the $N_f=2$ PHMC algorithm due to the two pseudofermion generations, the Hamiltonian calculation, and monitoring of the residual in the noisy Metropolis test. We note that the autocorrelation time may be extended by the increase of the dynamical variable in the path integral. Examination of this point is left for future studies. With the numerical tests described here we conclude that our PHMC algorithm for one-flavor dynamical quark works well even for a moderately large lattice size $16^3\times 48$ at intermediate quark masses of $m_{PS}/m_V \sim 0.7$–0.8, at least in the $N_f=1+1$ case.

D. A (2+1)-flavor QCD simulation

Combining the two-flavor HMC algorithm with the one-flavor PHMC leads to an exact algorithm for $(2+1)$-flavor QCD. For the two-flavor part, we may also choose the PHMC if the usable amount of memory allows to store work vectors. A test of the algorithm can be performed comparing the results with those of the R algorithm [1] after an extrapolation to zero step size in the latter. In this section we show the results of such a comparison on some small lattices.

The numerical test is made with the following two sets of lattice parameters. One set uses a lattice of size $4^3\times 8$ at $\beta=4.8$, sea quark mass of $\kappa_{ud}=0.150$ for two light flavors, and $\kappa_s=0.140$ for the third flavor, and $c_{\rm sw}=1.0$ for all three flavors ($N_f=2+1$ small). The order of the polynomial is set to $N_{poly}=10$ for the single flavor. The second set uses a $8^3\times 16$ lattice, $\beta=5.0$, $\kappa_{ud}=0.1338$, and $\kappa_s=0.1330$, and $c_{\rm sw}=2.08$ ($N_f=2+1$ middle), where $N_{poly}=58$ is employed. For both lattice sizes we use the Chebyshev polynomial with unit circle convergence domain.

TABLE VIII. Simulation statistics with the $N_f=1+1$ C-PHMC algorithm on large lattices.

	Large heavy C-PHMC(80) $N_f=1+1$	Large light C-PHMC(140) $N_f=1+1$
$N_{\rm MD}$	80	100
Trajectories	1000	1500
$\langle dH\rangle$	0.081(14)	0.042(11)
HMC acceptance	0.829(14)	0.872(14)
$\langle dS\rangle$ (quark 1)	0.014(7)	0.042(10)
Correction acceptance (quark 1)	0.944(11)	0.878(9)
$\langle dS\rangle$ (quark 2)	0.0084(61)	0.047(10)
Correction acceptance (quark 2)	0.936(9)	0.876(16)
Total acceptance	0.733(20)	0.671(20)
Plaquette	0.52782(12)	0.53392(9)

The simulation statistics is tabulated in Tables IX and X together with the plaquette expectation value extracted from the R algorithm. Figures 25 and 26 show the plaquette expectation value from the runs with the R algorithm at several values of the MD step size dt (open symbols). Filled symbols are from the PHMC algorithm. The plaquette values with the R algorithm extrapolated to zero step size are plotted with dotted horizontal lines. We observe that our exact algorithm (filled symbols) gives results at a finite dt (see also Tables IX and X) consistent with the extrapolated value (horizontal dotted line) of the R algorithm. Because of the finite dt dependence, the cost to obtain reliable results with the R algorithm is higher than that of the PHMC algorithm.

For larger and realistic lattice sizes, we started a parameter search in order to realize a physical volume $L\sim 1.7-2.0$ fm, a lattice cutoff $a^{-1}\sim 1.5-2.0$ GeV, and pseudoscalar to vector meson mass ratios $m_{PS}/m_V\sim 0.7$–0.8. During the parameter search we found an unexpected first-order phase transition [13]. Details of this search, including the property of the PHMC algorithm with the realistic parameters in the $N_f=2+1$ case on large lattice sizes, will be reported elsewhere.

VI. CONCLUSIONS

In this paper, we introduced a polynomial hybrid Monte Carlo (PHMC) algorithm which is applicable to QCD with an odd number of flavors. The algorithm is an extension of the one by Takaishi and de Forcrand [3] to the $O(a)$-improved Wilson quark action. We also described a method to remove the systematic error from the non-Hermitian polynomial approximation to the invese of the Wilson-Dirac operator in the single flavor case.

An important technical point uncovered in our work con-

TABLE IX. Simulation parameters for a $(2+1)$-flavor QCD simulation. $\beta=4.8, 4^3\times 8, c_{\rm sw}=1.00, \kappa_{ud}=0.150, \kappa_s=0.140$ are used. The stopping conditions are defined as follows: (a) the force calculation from the $N_f=2$ pseudofermion action, (b) the calculation of the Hamiltonian of the $N_f=2$ pseudofermion action, (c) the generation of the pseudofermion field, and the calculation of the correction factor for the single flavor part.

	Hybrid-R (extrapolated) $N_f=2+1$	C-PHMC(10) $N_f=2+1$	C-PHMC(10) $N_f=2+1$
$N_{\rm MD}$	-	20	10
Stopping condition (a)	-	10^{-14}	10^{-14}
Stopping condition (b)	-	10^{-14}	10^{-14}
Stopping condition (c)	-	10^{-14}	10^{-14}
$\langle dH\rangle$	-	0.055(5)	0.839(28)
HMC acceptance ratio	-	0.877(7)	0.521(12)
$\langle dS\rangle$	-	0.00014(61)	0.00056(62)
Correction acceptance ratio	-	0.9843(21)	0.9861(22)
Total acceptance ratio	-	0.864(7)	0.514(12)
Plaquette	0.39702(13)	0.39669(38)	0.39695(32)

cerns the choice of the even-odd preconditioning to the $O(a)$-improved Wilson-Dirac operator. Asymmetric and symmetric even-odd preconditionings were introduced and investigated in the HMC algorithm with two-flavor dynamical quarks. We found that the HMC algorithm with the *symmetrically* even-odd preconditioned form of the lattice QCD partition function yields roughly a factor two gain in efficiency over the unpreconditioned one. This performance exceeds the gain of about 1.5 for the asymmetrical preconditioning employed in actual simulations so far. We, then, decided to use the symmetrically even-odd preconditioned form for the quark determinant for the PHMC algorithm.

We explored distinctive features of the PHMC algorithm using the case of two flavors of quarks where comparisons with the standard HMC are possible. Our findings are (i) the reversibility is much better with the PHMC algorithm because of the fully deterministic nature of multiplication with the Wilson-Dirac operator in the force calculation in the molecular dynamics step, (ii) for the order of the polynomial chosen sufficiently large, the total efficiency of the PHMC algorithm is almost identical to or rather better than that with the HMC algorithm. Hence the PHMC algorithm is an alternative for $N_f=2$ dynamical QCD simulations on moderately large lattice size in the intermediate quark mass region $m_{PS}/m_V\sim 0.7$–0.8.

We demonstrated the consistency and applicability of the PHMC algorithm for an odd number of flavors by considering the case of two single-flavor pseudofermions ($N_f=1+1$ QCD) and comparing it with the established algorithm for the two-flavor pseudofermion ($N_f=2$ QCD). The reversibility holds to almost the same degree as that with the $N_f=2$ PHMC algorithm. The noisy Metropolis test for single-flavor part, in which we have to take the square root of the correction matrix explicitly, works well on moderately large lattices with intermediate quark masses of $m_{PS}/m_V\sim 0.7$–0.8.

TABLE X. Simulation parameters for a $(2+1)$-flavor QCD simulation. $\beta=5.0, 8^3\times 16, c_{\rm sw}=2.08, \kappa_{ud}=0.1338, \kappa_s=0.1330$ are used. The definition of the stopping condition (a)–(c) is the same as those in Table IX.

	Hybrid-R (extrapolated) $N_f=2+1$	C-PHMC(58) $N_f=2+1$
$N_{\rm MD}$	-	32
Stopping condition (a)	-	10^{-9}
Stopping condition (b)	-	10^{-14}
Stopping condition (c)	-	10^{-14}
$\langle dH\rangle$	-	0.194(9)
HMC acceptance ratio	-	0.743(7)
$\langle dS\rangle$	-	0.019(3)
Correction acceptance ratio	-	0.926(5)
Total acceptance ratio	-	0.688(8)
Plaquette	0.53161(7)	0.53145(11)

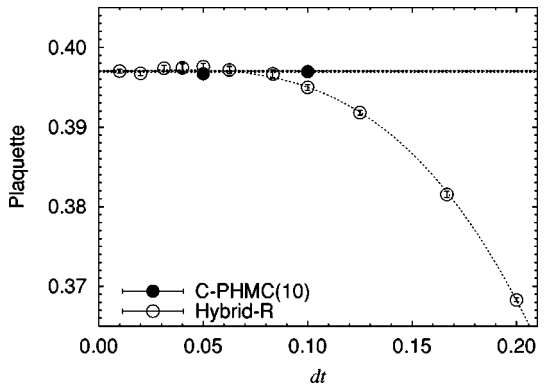

FIG. 25. MD step size dt dependence of the plaquette expectation value on the lattice of size $4^3\times 8$ at $\beta=4.8$, $c_{\rm SW}=1.00$, $\kappa_{ud}=0.150$, $\kappa_s=0.140$. Open circles are results of the R algorithm, and the filled circles are from our exact algorithm.

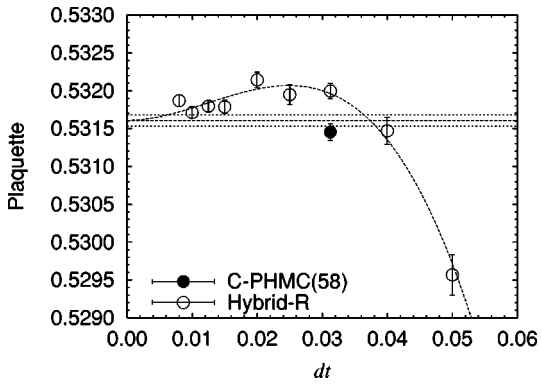

FIG. 26. MD step size dt dependence of the plaquette expectation value on the lattice of size $8^3\times16$ at $\beta=5.0$, $c_{\rm SW}=2.08$, $\kappa_{ud}=0.1338$, $\kappa_s=0.1330$. Open circles are results of the R algorithm, and the filled circle is from our exact algorithm.

Finally we constructed a PHMC algorithm for $2+1$ flavors of quarks by combining a two-flavored pseudofermion, which is employed in the usual HMC algorithm, and a single-flavored pseudofermion described by the polynomial approximation. Running the algorithm on two small lattice sizes we confirmed an agreement of plaquette values with those from the R algorithm after an extrapolation to the zero step size in the latter.

We conclude that the PHMC algorithm is a viable choice for realistic simulations of lattice QCD with $2+1$ flavors. Since our numerical tests show that the computational cost for two single-flavor pseudofermions is comparable to that of the two-flavor case, the cost for the single-flavor part of the $(2+1)$-flavor QCD is about a half of the two-flavor part. We thus expect that the simulation of the $(2+1)$-flavor QCD may be performed with a cost of a factor 1.5–2 compared to the two-flavor QCD simulation.

ACKNOWLEDGMENTS

This work is supported by the Supercomputer Project No.66 (FY2001) of High Energy Accelerator Research Organization (KEK), and also in part by the Grant-in-Aid of the Ministry of Education (Nos. 10640246, 11640294, 12014202, 12640253, 12640279, 12740133, 13640260, and 13740169). K-I.I. and N.Y. are supported by the JSPS.

APPENDIX: FORCE CALCULATION IN THE HMC ALGORITHMS

In this appendix we describe the explicit form of the quark force in the HMC algorithms for different preconditionings. Since most of the definitions and extractions of the quark force are common to the standard Wilson quark action, we only show the variation of the quark action under an infinitesimal change of the gauge link variable as defined in Eq. (12).

1. Without preconditioning

If we do not apply the even-odd preconditioning, the force from the pseudofermion field is simply written as

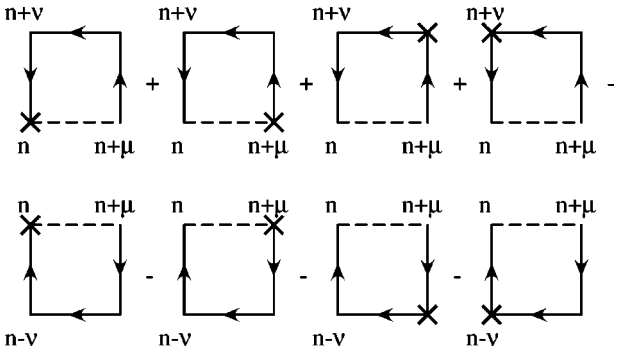

FIG. 27. Diagrams contributing to $F_\mu(n)$ from the SW term.

$$\delta S_q=\{-X^\dagger\delta D Y\}+\text{H.c.}, \tag{A1}$$

where

$$X=(D^\dagger)^{-1}D^{-1}\phi, \tag{A2}$$

$$Y=D^{-1}\phi, \tag{A3}$$

$$\delta D=\begin{pmatrix}\delta T_{ee} & \delta M_{eo}\\ \delta M_{oe} & \delta T_{oo}\end{pmatrix}. \tag{A4}$$

The contribution from the derivative of the hopping matrix $\delta M_{eo(oe)}$ is the same as that in the Wilson action. The contribution from the SW term $\delta T_{ee(oo)}$ is shown in Fig. 27, where $\times$ is a 3×3 matrix defined by

$$(\times)_{\mu\nu}(n)=\left\{-\frac{ic_{\rm sw}\kappa}{8}\,\mathrm{tr}_{\rm dirac}[\sigma_{\mu\nu}Y(n)X(n)^\dagger]\right\}+\text{H.c.} \tag{A5}$$

$\mathrm{tr}_{\rm dirac}[\cdots]$ means the trace over the spinor indices.

2. Asymmetric preconditioning

The force from the pseudofermion field with the asymmetric preconditioning is given by

$$\delta S_q^A=\{-X^{A\dagger}\delta D Y^A\}+\text{H.c.}, \tag{A6}$$

where

$$X^A=\begin{pmatrix}-(1+T)_{ee}^{-1}M_{oe}^\dagger\hat{X}_o^A\\ \hat{X}_o^A\end{pmatrix}, \tag{A7}$$

$$Y^A=\begin{pmatrix}-(1+T)_{ee}^{-1}M_{eo}\hat{Y}_o^A\\ \hat{Y}_o^A\end{pmatrix}, \tag{A8}$$

$$\hat{X}_o^A=(\hat{D}_{oo}^{A\ \dagger})^{-1}(\hat{D}_{oo}^A)^{-1}\phi_o, \tag{A9}$$

$$\hat{Y}_o^A=(\hat{D}_{oo}^A)^{-1}\phi_o. \tag{A10}$$

Note that $\hat{D}_{oo}^{A\ \dagger}=\gamma_5\hat{D}_{oo}^A\gamma_5$ and $M_{oe}^\dagger=\gamma_5 M_{eo}\gamma_5$.

In addition we need the force from the determinant of the SW term,

$$\delta S^A_{det} = -2\ \mathrm{Tr}[\,\delta T_{ee}(1+T)^{-1}_{ee}]. \tag{A11}$$

This is calculated only for even sites. The term $\times$ in Fig. 27 from the SW term is replaced by

$$(\times)_{\mu\nu}(n) = \Big\{ -\frac{ic_{\rm sw}\kappa}{8}\mathrm{tr}_{\rm dirac}[\,\sigma_{\mu\nu}Y^A(n)X^A(n)^\dagger] - \frac{ic_{\rm sw}\kappa}{8}\mathrm{tr}_{\rm dirac}[\,\sigma_{\mu\nu}(1+T)^{-1}(n)] \times \delta_{n,\rm evensite}\Big\} + \mathrm{H.c.} \tag{A12}$$

3. Symmetric preconditioning

For the symmetric preconditioning the force is separated into two parts as

$$\delta S^S_q = \{-X^{S\dagger}\delta M Y^S - X^{S\dagger}\delta T Z^S\} + \mathrm{H.c.}, \tag{A13}$$

where

$$\delta M = \begin{pmatrix} 0 & \delta M_{eo} \\ \delta M_{oe} & 0 \end{pmatrix}, \tag{A14}$$

$$\delta T = \begin{pmatrix} \delta T_{ee} & 0 \\ 0 & \delta T_{oo} \end{pmatrix}, \tag{A15}$$

$$X^S = \begin{pmatrix} -(1+T)^{-1}_{ee}M^\dagger_{oe}\hat{X}^S_o \\ \hat{X}^S_o \end{pmatrix}, \tag{A16}$$

$$Y^S = \begin{pmatrix} -(1+T)^{-1}_{ee}M_{eo}\hat{Y}^S_o \\ \hat{Y}^S_o \end{pmatrix}, \tag{A17}$$

$$Z^S = \begin{pmatrix} -(1+T)^{-1}_{ee}M_{eo}\hat{Y}^S_o \\ (1+T)^{-1}_{oo}M_{oe}(1+T)^{-1}_{ee}M_{eo}\hat{Y}^S_o \end{pmatrix}, \tag{A18}$$

$$\hat{X}^S_o = (1+T)^{-1}_{oo}(\hat{D}^{S\ \dagger}_{oo})^{-1}(\hat{D}^S_{oo})^{-1}\phi_o, \tag{A19}$$

$$\hat{Y}^S_o = (\hat{D}^S_{oo})^{-1}\phi_o. \tag{A20}$$

The γ_5 Hermiticity is slightly different for $\hat{D}^S_{oo}$, which is $\hat{D}^{S\ \dagger}_{oo} = \gamma_5(1+T)_{oo}\hat{D}^S_{oo}(1+T)^{-1}_{oo}\gamma_5$.

The force contribution from the determinant of the SW term is written as

$$\delta S^S_{det} = -2\ \mathrm{Tr}[\,\delta T(1+T)^{-1}], \tag{A21}$$

at every lattice site. The term $\times$ in Fig. 27 is replaced by

$$(\times)_{\mu\nu}(n) = \Big\{ -\frac{ic_{\rm sw}\kappa}{8}\mathrm{tr}_{\rm dirac}[\,\sigma_{\mu\nu}Z^S(n)X^S(n)^\dagger] - \frac{ic_{\rm sw}\kappa}{8}\mathrm{tr}_{\rm dirac}[\,\sigma_{\mu\nu}(1+T)^{-1}(n)]\Big\} + \mathrm{H.c.} \tag{A22}$$

4. PHMC

In the PHMC algorithm, the term $\times$ from the SW term δT in Fig. 27 is written as

$$(\times)_{\mu\nu}(n) = \Big\{ \frac{ic_{\rm sw}\kappa}{8}\sum_{j=1}^{N_{poly}}\{\mathrm{tr}_{\rm dirac}[\,\sigma_{\mu\nu}Z^{P(j)}(n)X^{P(j)}(n)^\dagger]\} - \frac{ic_{\rm sw}\kappa}{8}\mathrm{tr}_{\rm dirac}[\,\sigma_{\mu\nu}(1+T)^{-1}(n)]\Big\} + \mathrm{H.c.} \tag{A23}$$

[1] S. Gottlieb, W. Liu, D. Toussaint, R.L. Renken, and R.L. Sugar, Phys. Rev. D **35**, 2531 (1987).
[2] S. Duane, A.D. Kennedy, B.J. Pendleton, and D. Roweth, Phys. Lett. B **195**, 216 (1987).
[3] T. Takaishi and Ph. de Forcrand, hep-lat/0009024; Nucl. Phys. B (Proc. Suppl.) **94**, 818 (2001); hep-lat/0108012.
[4] Ph. de Forcrand and T. Takaishi, Nucl. Phys. B (Proc. Suppl.) **53**, 968 (1997).
[5] R. Frezzotti and K. Jansen, Phys. Lett. B **402**, 328 (1997).
[6] R. Frezzotti and K. Jansen, Nucl. Phys. **B555**, 395 (1999); **B555**, 432 (1999).
[7] A. Boriçi and Ph. de Forcrand, Nucl. Phys. **B454**, 645 (1995).
[8] C. Alexandrou, A. Boriçi, A. Feo, Ph. de Forcrand, A. Galli, F. Jegerlehner, and T. Takaishi, Phys. Rev. D **60**, 034504 (1999).
[9] Ph. de Forcrand, Parallel Comput. **25**, 1341 (1999).
[10] B. Sheikholeslami and R. Wohlert, Nucl. Phys. **B259**, 572 (1985).
[11] X-Q. Luo, Comput. Phys. Commun. **94**, 119 (1996).
[12] K. Jansen and C. Liu, Comput. Phys. Commun. **99**, 221 (1997).
[13] JLQCD Collaboration, S. Aoki *et al.*, Nucl. Phys. B (Proc. Suppl.) **106**, 263 (2002); and (in preparation).
[14] K. Symanzik, Nucl. Phys. **B226**, 187 (1983); **B226**, 205 (1983).
[15] M. Lüscher, Nucl. Phys. **B418**, 637 (1994).
[16] A. Borrelli, Ph. de Forcrand, and A. Galli, Nucl. Phys. **B477**, 809 (1996).
[17] T.A. DeGrand, Comput. Phys. Commun. **52**, 161 (1988).
[18] R. Gupta, A. Patel, C.F. Baillie, G. Guralnik, G.W. Kilcup, and S.R. Sharpe, Phys. Rev. D **40**, 2072 (1989).
[19] T.A. DeGrand and P. Rossi, Comput. Phys. Commun. **60**, 211 (1990).
[20] JLQCD Collaboration, S. Aoki *et al.*, Nucl. Phys. B (Proc. Suppl.) **94**, 233 (2001).

[21] A. Frommer, V. Hannemann, B. Nockel, T. Lippert, and K. Schilling, Int. J. Mod. Phys. C **5**, 1073 (1994).
[22] M. Creutz, Phys. Rev. D **38**, 1228 (1988).
[23] R. Gupta, G.W. Kilcup, and S.R. Sharpe, Phys. Rev. D **38**, 1278 (1988).
[24] S. Gupta, A. Irbäck, F. Karsch, and B. Petersson, Phys. Lett. B **242**, 437 (1990).
[25] T. Takaishi, Comput. Phys. Commun. **133**, 6 (2000).
[26] R.G. Edwards, I. Horvath, and A.D. Kennedy, Nucl. Phys. **B484**, 375 (1997).
[27] C. Liu, A. Jaster, and K. Jansen, Nucl. Phys. **B524**, 603 (1998).
[28] UKQCD Collaboration, B. Joo, B. Pendleton, A.D. Kennedy, A.C. Irving, J.C. Sexton, S.M. Pickles, and S.P. Booth, Phys. Rev. D **62**, 114501 (2000).
[29] M.J. Peardon, hep-lat/0011080.
[30] M. Hasenbusch, Phys. Lett. B **519**, 177 (2001); M. Hasenbusch and K. Jansen, Nucl. Phys. B (Proc. Suppl.) **106**, 1076 (2002); hep-lat/0110180.
[31] ALPHA Collaboration, R. Frezzotti, M. Hasenbusch, U. Wolff, J. Heitger, and K. Jansen, Comput. Phys. Commun. **136**, 1 (2001).
[32] ALPHA Collaboration, K. Jansen and R. Sommer, Nucl. Phys. **B530**, 185 (1998).
[33] ALPHA Collaboration, A. Bode *et al.*, Phys. Lett. B **515**, 49 (2001).
[34] I. Montvay, Comput. Phys. Commun. **109**, 144 (1998); R. Kirchner, S. Luckman, I. Montvay, K. Spanderen, and J. Westphalen, Nucl. Phys. B (Proc. Suppl.) **73**, 828 (1999); I. Montvay, hep-lat/9903029; hep-lat/9911014.
[35] Ph. de Forcrand, Nucl. Phys. B (Proc. Suppl.) **73**, 822 (1999).
[36] C. Alexandrou, Ph. de Forcrand, M. D'Elia, and H. Panagopoulos, Phys. Rev. D **61**, 074503 (2000); Nucl. Phys. B (Proc. Suppl.) **83-84**, 765 (2000).
[37] B. Bunk, S. Elser, R. Frezzotti, and K. Jansen, Comput. Phys. Commun. **118**, 95 (1999).
[38] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling, *Numerical Recipes* (Cambridge University Press, Cambridge, England, 1988).
[39] A.D. Kennedy and J. Kuti, Phys. Rev. Lett. **54**, 2473 (1985).
[40] L. Lin, K.F. Liu, and J. Sloan, Phys. Rev. D **61**, 074505 (2000).