Implementation of the overlap fermion simulation

Hideo Matsufuru for JLQCD Collaboration

High Energy Accelerator Research Organization (KEK),
Oho 1-1, Tsukuba 305-0801, Japan

Email: hideo.matsufuru@kek.jp

May 12, 2013, Ver.2.3.1

Abstract

This note explains how the simulation programs of the overlap fermions are implemented. The note contains the implementation of the overlap operator, solver algorithms, and the Hybrid Monte Carlo update with one and two flavors. Section for describing eigenvalues of overlap operator was added from Ver.2.3.

Contents

1 Introduction

This note describes the implementation of the simulation programs of the overlap fermions for the JLQCD project.

2 Lattice actions

The total action in the present project is represented as

$$S = S_{ov} + S_G + S_{Fukaya}, \tag{2.1}$$

where S_{ov} is the overlap fermion action, S_G the gauge field part, and S_{Fukaya} the extra Wilson fermion term[1] which is introduced to avoid near-zero modes of H_W. S_{Fukaya} does not describe physical fermion content, and considered as a part of the gauge action.

2.1 Overlap fermion

The overlap quark action [1] with quark mass m is defined as

$$D(m) = \left(1 - \frac{1}{2}\bar{a}m\right) D + m, \tag{2.2}$$

where

$$\begin{aligned} D &= \frac{1}{\bar{a}}\left[1 + \gamma_5 \cdot \mathrm{sign}(H_W)\right], && (2.3)\\ \bar{a} &= a/(1+s) \equiv 1/M_0, && (2.4) \end{aligned}$$

[1] Unlike the publication, this term is called ‘Fukaya term’ in this note after H. Fukaya who proposed to employ this term in our project.

where $H_W(-M_0)$ is the hermitian Wilson-Dirac operator,

$$H_W(-M_0) \equiv \gamma_5 a D_W(-M_0), \tag{2.5}$$

and D_W is the Wilson-Dirac operator,

$$D_W(-M_0;x,y) = 4 - M_0 - \frac{1}{2a}\sum_\mu \left[(1-\gamma_\mu)U_\mu(x)\delta_{x+\hat{\mu},y} + (1+\gamma_\mu)U_\mu^\dagger(x-\hat{\mu})\delta_{x-\hat{\mu},y}\right]. \tag{2.6}$$

As the kernel D_W, we take the simplest Wilson-Dirac operator, while improved operators are also candidates. In the following, we set $a = 1$ and use the hopping parameter representation of D_W, $D_W \to 2\kappa D_W$ with $\kappa = 1/2(4 - aM_0)$. The normalization of D_W is irrelevant to the overlap operator, since it appears only in the sign-function.[2]

The sign function $\text{sign}(H_W)$ is the most involved part for the practical implementation of the overlap fermion and will be described in the next section. Here we proceed as it is homehow given.

Two-flavor case. In the case of two degenerate flavors, the fermion action is treated in the standard prescription. By integrating the Grassmann variables, fermion determinant arizes and it is again exponentiated by introducing pseudofermion fields:

$$\int \mathcal{D}\bar{\psi}^\dagger \mathcal{D}\psi \exp[-S_{ov}] = \det[D(m)^\dagger D(m)] = \int \mathcal{D}\phi^\dagger \mathcal{D}\phi \exp[-S_{PF}], \tag{2.7}$$

$$S_{PF} = \phi^\dagger [D(m)^\dagger D(m)]^{-1}\phi. \tag{2.8}$$

If the Hasenbusch acceleration (multi-mass preconditioning) [2] is applied,

$$\begin{aligned} \det[D(m)^\dagger D(m)] &= \det[D(m')^\dagger D(m')] \det[D(m')^{\dagger -1} D(m)^\dagger D(m) D(m')^{-1}] && (2.9)\\ &= \int \mathcal{D}\phi_1^\dagger \mathcal{D}\phi_1 \mathcal{D}\phi_2^\dagger \mathcal{D}\phi_2 \exp\left[-S_{PF}^{(1)} - S_{PF}^{(2)}\right], && (2.10) \end{aligned}$$

$$\begin{aligned} S_{PF}^{(1)} &= \phi_1^\dagger [D(m')^\dagger D(m')]^{-1}\phi_1, && (2.11)\\ S_{PF}^{(2)} &= \phi_2^\dagger \left\{ D(m')[D(m)^\dagger D(m)]^{-1} D(m')^\dagger \right\} \phi_2, && (2.12) \end{aligned}$$

where m' is a mass of the preconditioner. Here we restrict ourselves to the case of single preconditioner.

One-flavor case. The algorithm of HMC with single flavor is based on Refs. [3, 4]. The squared hermitian overlap-Dirac operator, $H^2(m) = D^\dagger(m)D(m)$, commutes with γ_5, and therefore can have eigenstates with definite chirality. The projection operators $P_\pm = (1 \pm \gamma_5)/2$ decompose H^2 into two parts:

$$H^2 = P_+ H^2 P_+ + P_- H^2 P_- \equiv Q_+ + Q_-, \tag{2.13}$$

$$\det(H^2) = \det(Q_+)\det(Q_-) \tag{2.14}$$

[2] Caution: in the low-mode subtraction, the threshold parameter in the code is given by the hopping parameter representation, and differs in normalization to the value quoted in the paper.

and when there is no zero mode, the two factors on RHS of Eq. (2.14) give equal contribution to the path integral. Even when there are zero modes, since they give constant contributions to the path integral which is easily evaluated, only the non-zero modes enters the molecular dynamical evolution. Therefore by choosing one of the chiral sectors one can simulate one flavor of the overlap fermion.

In the following, let us choose for simplicity the positive chirality sector assuming no zero modes. The cases with the negative chirality sector and with zero modes are straightforward, while the simulation becomes complicated when the topological charge can changes during HMC [4], which is absent in our case. The pseudo-fermion action for one flavor of overlap fermion reads as

$$S_{PF} = S_{PF}^{(1)} + S_{PF}^{(2)}, \tag{2.15}$$

where

$$\begin{aligned} S_{PF}^{(1)} &= \phi_{1+}^{\dagger} Q_{+}^{-1}(m') \phi_{1+}, & (2.16) \\ S_{PF}^{(2)} &= \phi_{2+}^{\dagger} \left(\frac{Q_{+}(m')}{Q_{+}(m)} \right) \phi_{2+} = \phi_{2+}^{\dagger} \frac{1}{2} \left[Q_{+}(m') Q_{+}^{-1}(m) + Q_{+}^{-1}(m) Q_{+}(m') \right] \phi_{2+} & (2.17) \end{aligned}$$

are the terms of the preconditioner and the preconditioned dynamical fermion, respectively. For the latter, we adopt the symmetric form so as to asure that $S_{PF}^{(2)}$ is real. ϕ_{1+} and ϕ_{2+} are invariant under multiplication of projection operator P_{+}.

2.2 Gauge action

We consider either the admissible (topology conserving) action or rectangular improved action.

- Admissible action:

$$S_G = \beta \sum_{x,\mu>\nu} \frac{1 - \frac{1}{3}\mathrm{ReTr}P_{\mu\nu}(x)}{1 - \frac{1}{\epsilon}\left[1 - \frac{1}{3}\mathrm{ReTr}P_{\mu\nu}(x)\right]} \tag{2.18}$$

  which satisfies the admissibility condition

$$|1 - P_{\mu\nu}(x)| < \epsilon, \qquad \forall x, \mu, \nu. \tag{2.19}$$

- Rectangular improved action:

$$S_G = \beta \left\{ c_{plq} \sum_{x,\mu>\nu} \left(1 - \frac{1}{3}\mathrm{ReTr}P_{\mu\nu}(x) \right) + c_{rct} \sum_{x,\mu\neq\nu} \left(1 - \frac{1}{3}\mathrm{ReTr}R_{\mu\nu}(x) \right) \right\} \tag{2.20}$$

  where $R_{\mu}\nu(x)$ is 2×1 rectangular Wilson loop. Several improved actions are categorized to this class of actions, such as Iwasaki, Lüscher-Weisz, DBW2, depending on the the choices of the coefficients c_{plq} and c_{rct}.

In our productive run, we adopt the rectangular improved action with the values $c_{plq} = 3.648$ and $c_{rct} = -0.331$, *i.e.* the Iwasaki gauge action [5]. In the present version of the Fortran code, the admissible action is not involved.[3]

[3] The admissible action had been included in the Fortran overlap HMC code Ver.3.1.0 and the preceding versions.

2.3 Extra Wilson (Fukaya) term

To suppress the near-zero modes of H_W, the Wilson/twisted mass ghost term is introduced as [6]

$$\det\left(\frac{H_W^2}{H_W^2+\mu^2}\right) = \int \mathcal{D}\phi_W^\dagger \mathcal{D}\phi_W \exp(-S_{Fukaya}) \tag{2.21}$$

with

$$S_{Fukaya} = \phi_W^\dagger \left[D_W^{tm}(\mu)(D_W^\dagger D_W)^{-1} D_W^{tm}(\mu)^\dagger\right]\phi_W, \tag{2.22}$$

where $D_W^{tm}(\mu) = D_W + i\gamma_5\mu$ and μ is the mass of the twisted mass ghost. Note that $H_W(\mu)^\dagger = H_W(-\mu)$.

3 Overlap Dirac operator

3.1 Sign function

The overlap operator with the bare mass m introduced in the previous section is written as

$$D(m) = \left(M_0 + \frac{m}{2}\right) + \left(M_0 - \frac{m}{2}\right)\gamma_5 \cdot \mathrm{sign}(H_W). \tag{3.1}$$

As already noted, the implementation of the sign function sign(H_W) is a quite involved issue. In practice, one needs to compute a result of multiplication of $D(m)$ to a vector v. v is a complex vector having the degrees of freedom of color, spinor, and site. When sign(H_W) is applied to v, it expands v in terms of the eigenmodes of H_W, and assign ±1 according the sign of the eigenvalues:

$$\mathrm{sign}(H_W)\cdot v = \sum_\lambda \mathrm{sign}(\lambda)(\psi_\lambda, v)\psi_\lambda. \tag{3.2}$$

(ψ_λ, v) is a inner product and given by $\psi_\lambda^\dagger \cdot v$. In numerical application, it is not realistic to determine all the eigenmodes. A standard procedure therefore applies the eigenmode decomposition only to low-lying modes, and employs some approximation formula to sign(H_W). As the latter, we adopt the Zolotarev's rational approximation [9, 10]. The polynomial approximation is also widely applied procedure. Since these approximations are valid in certain region of λ, highest eigenvalue is also needed to be computed (or to be set to certain value).

The steps to compute sign$H_W \cdot v$ is as follows:

(i) Determine the low-lying (and highest) eigenmodes of H_W.
(ii) Subtract low-lying eigenmodes from the vector v:

$$\tilde{v} = v - \sum_{i=1}^{n_\lambda}(\psi_{\lambda_i}, v)\psi_{\lambda_i} \tag{3.3}$$

(iii) Multiply approximate sign-function $\epsilon(H_W)$ to $\tilde{v}$.
(iv) Then the total sign function, and thus $D(m)v$, is calculated as

$$\mathrm{sign}(H_W)v \simeq \sum_{i=1}^{n_\lambda}\mathrm{sign}(\lambda_i)(\psi_{\lambda_i}, v)\psi_{\lambda_i} + \epsilon(H_W)\tilde{v}. \tag{3.4}$$

Implementation: In the Fortran code, the Wilson fermion kernel is implemented in file `oprw5_chiral.f` with the chiral representation of the γ matrices. (The Dirac representation os also adopted in early stage of the project, but not not maintained. There are two versions, one is with generic MPI version and the other is tuned for the Blue Gene/L machine.[4]

The overlap fermion kernel is implemented in file `opr_overlap_zolotarev.f`. The input parameters `Rm`, `Rm0` correspond to m and M_0.

3.2 Low-mode subtraction

It is necessary to determine the low-lying modes to subtract them from H_W. As an eigenvalue solver, we currently adopt the implicitly restarted Lanczos method [7]. This method is based on the Lanczos algorithm. When N_k eigenmodes are desired to determine, this method extends the Krylov subspace $\mathcal{K}_k$ to $(N_k + N_p)$-dimension. Then information of N_p vectors in extra-space is compressed into the N_k-dimension space by applying implicitly shifted QR algorithm. Repeated application of extension and compression of Krylov subspace causes that the N_k vectors approach to the lowest N_k eigenvectors. After enough application of this step, the tridiagonal Lanczos matrix is diagonalized by QR algorithm. The same algorithm is also applied to determine the highest eigenmodes.

In practice, it is useful to determine the eigenvalues whose absolute values are less than certain threshold, V_{th}. Then n_λ is defined as the number of eigenvalues which satisfies $|\lambda| < V_{th}$. This is the policy adopted by the present program. Detailed description of eigenvalue solver will be presented in a separate note.

Implementation:
The eigenvalues of H_W are determined by routines in the files `eigen_wilson5_lex.f` and `qris.f`. The former implements the main part of the implicitly restated Lanczos algorithm, and the latter contains routines for implicitly shifted QR algorithm. In the common file, `eigen_wlex.h`, contains parameter `Nkmax`. This parameter specifies the maximum size of Krylov subspace, $N_k + N_p$, and hence it must be larger than the sum of the input parameters `Nkxmin` and `Nkxmin`, which are respectively N_k and N_p for the determination of low-lying eigenmodes. This is also true for `Nkxmax` and `Nkxmax` for the determination of highest eigenmodes (while practically the highest mode can be easily determined compared to the low-lying ones). The parameter `Nkmax` also appears in other common files, and must be changed simultaneously.

The parameter `Vthrs` corresponds to V_{th}. `Nsbt`, the number of subtracted eigenvectors, is counted accordingly in the program. The parameters `Enorm_eigen` specifies the precision of the eigenvalue relation, $H_W \psi_\lambda = \lambda \psi_\lambda$. The determined eigenvalues and eigenvectors are stored in common variables `TDa(Nkmax)` and `Vk(Nvst,Nkmax)`.

For an acceleration technique, the Chebychev acceleration is implemented. (More detailed description will be supplyed.)

Comment: If the low-lying eigenmodes of H_W are monitored during MD steps in HMC, the cost of eigenvalue solver is not negligible. Improving this part is highly desired. CG (conjugate gradient) algorithm [8] is a potential alternative to the Lanczos type algorithm.

[4] We thank IBM staffs for tuning the Wilson Dirac kernel, in particular Jun Doi for continuous effort.

3.3 Approximate sign function

There are several procedures to approximate the sign function sign(H_W).

In this project, we adopt the Zolotarev's partially fractional approximation [9, 10].

3.4 Zolotarev's rational approximation

The Zolotarev's rational approximation is represented as [9, 10]

$$\frac{1}{\sqrt{H_W^2}} = \frac{d_0}{\lambda_{min}}(h_W^2 + c_{2n}) \sum_{l=1}^{n} \frac{b_l}{h_W^2 + c_{2l-1}} \tag{3.5}$$

where $h_W = H_W/\lambda_{min}$. Multiplying H_W to this function, sign(H_W) is computed. This formula is valid for the interval $h_W \in [1, b]$ with $b = \lambda_{max}/\lambda_{min}$. The parameters d_0 and b_l are related to the coefficients c_l as follows.

$$d_0 = \frac{2\lambda}{1+\lambda} \prod_{l=1}^{n} \frac{1 + c_{2l-1}}{1 + c_{2l}} \tag{3.6}$$

$$b_l = d_0 \frac{\prod_{i=1}^{n-1}(c_{2i} - c_{2l-1})}{\prod_{i=1, i\neq l}^{n}(c_{2i-1} - c_{2l-1})} \tag{3.7}$$

To determine d_0 precisely, the parameter λ (which is defined in terms of ϑ function) must also be determined. However, d_0 can practically be determined by enforcing that the approximate sign function exhibits least deviation from unity in the range $[1, b]$.

Alternatively, the following approximate expresseion is practically useful. By enforcing the sign function is unity at $h_W = 1$, $2\lambda/(1 + \lambda) = 1$ follows. (After numerically obtaining approximate function, this factor can easily obtained by maximum and minimum in the range $[\lambda_{min}, \lambda_{max}]$. Precise value of this factor maximally halves the deviation from 1.) Then

$$d_0 = \prod_{l=1}^{n} \frac{1 + c_{2l-1}}{1 + c_{2l}}. \tag{3.8}$$

The coefficients c_l is given as [10][5]

$$c_l = \frac{\mathrm{sn}^2(lK'/(2n+1); \kappa')}{1 - \mathrm{sn}^2(lK'/(2n+1); \kappa')} \tag{3.9}$$

$$K' = u(1) = \int_0^1 \frac{dt}{\sqrt{(1-t^2)(1-\kappa'^2 t^2)}}, \tag{3.10}$$

where $\kappa' = \sqrt{1 - \kappa^2}$, $\kappa = \lambda_{min}/\lambda_{max}$. K' is the complete elliptic integral of the first kind with modulus κ', *i.e.*, the value of u such that $\mathrm{sn}(u; \kappa') = 1$.

One needs to determine c_l and K' for given $b = \lambda_{max}/\lambda_{mim}$ and n. $\mathrm{sn}(u, \kappa')$ must be computed somehow.

[5] In Ref. [9], c_l is defined $\mathrm{sn}(lK'/2n; K')$ instead of $\mathrm{sn}(lK'/(2n+1); K')$ in Eq. (3.10).

Implementation: First of all, elliptic function sn must be computed. We make use of a routine in Numerical Recipes [11]. In Fortran code, subroutine `Jacobi_elliptic` (`sncndn` in Numerical Recipes) compute $\mathrm{sn}(u, k_c)$, $\mathrm{cn}(u, k_c)$, and $\mathrm{dn}(u, k_c)$ for given u and $k_c = 1 - k^2$. In subroutine `Poly_Zolotarev`, first the value of u which satisfies $\mathrm{cn}(u, k_c) = 0$ ($\mathrm{sn}(u, k_c)$=1) is determined to the 14-th digit precision. Then c_l, d_0, and b_l are determined according to Eqs. (3.10), (3.8), and (3.7), respectively. A program to check the Zolotarev approximation formula by giving $\mathrm{sign}(x)$ for real number x is also available.

3.5 Multi-shift CG solver

In the rational approximation of $\mathrm{sign}(H_W)$, one needs to solve equations

$$\left(H_W^2 + c_j\right) x = b \qquad j = 1, \ldots n \tag{3.11}$$

for single source vector b. If there is no efficient way to solve these equations, the rational approximation is not tractable.

Multi-shift solver [12, 13] enables to solve these n equations at once. We employ the multi-shift CG algorithm (for the standard CG algorithm, see Sec. A.1). The multi-shift solver algorithms make use of the fact that the Krylov subspace $\mathcal{K}_k(A, v_0)$ for matrix A and initial vector v_0 is unchanged against a shift $A \to A + \sigma$, *i.e.*, $\mathcal{K}_k(A, v_0) = \mathcal{K}_k(A + \sigma, v_0)$. This implies that the residual vector, which is perpendicular to the previous Klylov subspace, can be shared by n equations in Eq. (3.11). Then n solutions of Eq. (3.11) can be determined simultaneously. The algorithm is summarized in A.2.

Comment: On scalar machines, the double-path solver [14, 15] is worth to be considered as an alternative. There are alternative method to determine the solution of Eq. (3.11) by extending the dimension of vectors.

4 Overlap solver algorithm

The inversion of the overlap operator $D(m)$ is required in every step of the HMC update. This is usually most time consuming part of the simulation. Therefore, improvement of the solver algorithm for this inversion is quite important to reduce the simulation cost.

Besides HMC, solver algorithm is also needed to determine the quark propagators. To determine the quark propagator for several values of mass simultaneously, multishift CG solver is applicable. If the low-lying eigenmodes of D_{ov} are determined, they can be used for the low-mode preconditioning, in which these low-mode part is subtracted from D_{ov} for decreasing the condition number.

4.1 Nested conjugate Gradient algorithm

Once the approximate $\mathrm{sign}(H_W)$ is at hand, overlap quark propagator can be computed by standard solver algorithms for hermitian or nonhermitian matrices. In HMC algorithm, since one needs to invert only $D(m)^\dagger D(m)$, which is hermitian and positive definite, the standard conjugate gradient (CG) algorithm is applicable. The CG algorithm is summarized in Appendix A.1. Of course it is worth to examine which of hermitian algorithms and nonhermitian algorithms (BiCGStab, MR, GMRES, etc.) are efficient (see T. Kaneko's report [25, 27]). For the latter, one needs to solve the linear equations twice in HMC algorithm.

With the rational approximation to the sign function, the CG algorithm is also necessary to implement D_{ov}. In this sense, it is called 'nested' CG algorithm.

Relaxed CG algorithm. The relaxed stopping condition method [16] is based on an idea that, as the outer solver proceeds, the correction to the solution vector, $|x_i - x_{i-1}|$, becomes smaller and one does not have to evaluate $D(m)$ with too much accuracy. Its implementation depends on the outer solver algorithm, and for CG(NE), the condition is loosened as

$$\epsilon_i^{ms} \propto \sqrt{\zeta_i}, \qquad \zeta_i = \zeta_{i-1} + \frac{1}{|r_{i-1}|^2}, \tag{4.1}$$

where $r_{i-1} = Dx_{i-1} - b$ and ϵ_i^{ms} the stopping condition for the inner (multishift) solver.

This technique accelerate the convergence measured in multiplication of D_W almost a factor of 2 [17, 26]. The relaxed CG algorithm is summarized as follows.

outer loop:

```
1: set initial guess $x_0$
2: $r_0 = b - Ax_0$
3: $p_0 = r_0$
4: $\zeta = 1/|r_0|^2$
5: repeat until $|r_k|/|b| < \epsilon_{out}$
    5-1: inner loop: calculate $q_k$ so that $|q_k - Ap_k| < \epsilon_{out}|b||p_k|\sqrt{\zeta}/k$
    5-2: $\alpha = \langle r_k, r_k\rangle / \langle p_k, q_k\rangle$
    5-3: $x_{k+1} = x_k + \alpha p_k$
    5-4: $r_{k+1} = r_k - \alpha p_k$
    5-5: convergence check: exit outer loop if $|r_k|/|k| < \epsilon_{out}$
    5-6: $\beta = \langle r_{k+1}, r_{k+1}\rangle / \langle r_k, r_k\rangle$
    5-7: $p_{k+1} = r_k + \beta p_k$
    5-7: $\zeta = \zeta + 1/|r_{k+1}|^2$
```

Comment: There are possible improvements:
- Other outer solver algorithm.
- Adoptive precision.
- Guess of initial approximate solution (chronological estimator or preconditioning).
- Chiral projection.

4.2 5-dimensional solver

The 5-dimensional CG solver is based on the Schur decomposition [20, 21]. Let us consider a 5-dimensional block matrix

$$M_5 = \begin{pmatrix} A & B \\ C & D \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ CA^{-1} & 1 \end{pmatrix} \begin{pmatrix} A & 0 \\ 0 & S \end{pmatrix} \begin{pmatrix} 1 & A^{-1}B \\ 0 & 1 \end{pmatrix} \equiv \tilde{L}\tilde{D}\tilde{U}, \tag{4.2}$$

where $S = D - CA^{-1}B$. S is called the Schur complement. Consider a linear equation

$$M_5 \begin{pmatrix} \phi \\ \psi_4 \end{pmatrix} = \begin{pmatrix} 0 \\ \chi_4 \end{pmatrix}. \tag{4.3}$$

Using $M_5 = \tilde{L}\tilde{D}\tilde{U}$ and multiplying $\tilde{L}^{-1}$ from the left to the above equation, one arrives at

$$\begin{pmatrix} A & 0 \\ 0 & S \end{pmatrix} \begin{pmatrix} \phi + A^{-1}B\psi_4 \\ \psi_4 \end{pmatrix} = \begin{pmatrix} 0 \\ \chi_4 \end{pmatrix}. \tag{4.4}$$

Namely, by solving 5-dimensional equation (4.3), one can solve $S\psi_4 = \chi_4$.

Hereafter an example for $N(= N_{pole}) = 2$ case is shown explicitly. Let us consider the 5D operator of a form

$$M_5 = \left(\begin{array}{cccc|c} H_W & -\sqrt{q_2} & & & 0 \\ -\sqrt{q_2} & -H_W & & & \sqrt{p_2} \\ & & H_W & -\sqrt{q_1} & 0 \\ & & -\sqrt{q_1} & -H_W & \sqrt{p_1} \\ \hline 0 & \sqrt{p_2} & 0 & \sqrt{p_1} & R\gamma_5 + p_0 H \end{array}\right) = \left(\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right). \tag{4.5}$$

In general, A B, and C has the following structure.

$$A = \begin{pmatrix} A_N & & \\ & A_{N-1} & \\ & & \ddots \end{pmatrix}, \quad B = \begin{pmatrix} B_N \\ B_{N-1} \\ \vdots \end{pmatrix}, \quad C = (C_N, C_{N-1}, \cdots) \tag{4.6}$$

$$A^{-1} = \begin{pmatrix} A_N^{-1} & & \\ & A_{N-1}^{-1} & \\ & & \ddots \end{pmatrix} \tag{4.7}$$

$$A_i^{-1} = \frac{1}{H_W^2 + q_i} \begin{pmatrix} H_W & -\sqrt{q_i} \\ -\sqrt{q_i} & -H_W \end{pmatrix} \qquad (i = 1, \ldots, n) \tag{4.8}$$

Then

$$S = D - CA^{-1}B = D - \sum_i C_i A_i^{-1} B_i \tag{4.9}$$

$$= R\gamma_5 + p_0 H_W + H_W \sum_{i=1}^{N} \frac{p_i}{H_W^2 + q_i}. \tag{4.10}$$

The parameters R, p_0, p_i and q_i ($i = 1, \ldots, N$) are determined appropriately by comparing with the Zolotarev's partially fractional approximation formula (see below). This technique also applies to other partial fractional approximation and continuum fractional approximation [20, 21].

Low-mode subtraction [18]**.** The low-mode preconditioned hermitian overlap operator is written as

$$H_{ov} = f_1\gamma_5 + f_2 \sum_{j=1}^{N_{sbt}} \mathrm{sign}(\lambda_j) v_j \times v_j^\dagger + f_2 P_H \,\mathrm{sign}(H_W) P_H, \tag{4.11}$$

where $f_1 = M_0 + \frac{m}{2}$, $f_2 = M_0 - \frac{m}{2}$, v_j is an eigenvector of H_W associated to an eigenvalue λ_j, and $P_H = 1 - \sum_{j=1}^{N_{sbt}} v_j \times v_j^\dagger$ is the projector onto the space spanned by the eigenvectors whose eigenvalue $|\lambda| > \lambda_{thrs}$. This implies that the 5D operator is modified as

$$D \;=\; R\gamma_5 + p_0 P_H H_W P_H + f_2 \sum_{j=1}^{N_{sbt}} \mathrm{sign}(\lambda_j) v_j \times v_j^\dagger, \tag{4.12}$$

$$B_i \;=\; \begin{pmatrix} 0 \\ \sqrt{p_i} P_H \end{pmatrix}, \qquad C_i \;=\; (\, 0, \sqrt{p_i} P_H \,). \tag{4.13}$$

The parameters of the 5D matrix are determined as

$$R \;=\; f_1, \tag{4.14}$$

$$p_0 \;=\; f_2 \frac{d_0}{\lambda_{thrs}} \sum_l b_l, \tag{4.15}$$

$$p_l \;=\; f_2 d_0 b_l (c_{2n} - c_{2l-1}) \lambda_{thrs}, \tag{4.16}$$

$$q_l \;=\; c_{2l-1} \cdot \lambda_{thrs}^2. \tag{4.17}$$

Note that in the approximation of sign(H_W) in Eq. (4.11), λ_{thrs} replaces λ_{min} in Eq. (3.5).

Even-odd preconditioning. 5D solver is not efficient without applying a preconditioning technique. One can apply the even-odd preconditioning by decomposing 4D lattice sites into even and odd sites as

$$M_5 x = \begin{pmatrix} M_{ee} & M_{eo} \\ M_{oe} & M_{oo} \end{pmatrix} \begin{pmatrix} x_e \\ x_o \end{pmatrix} = \begin{pmatrix} b_e \\ b_o \end{pmatrix}. \tag{4.18}$$

By multiplying

$$\begin{pmatrix} M_{ee}^{-1} & -M_{ee}^{-1} M_{eo} M_{oo}^{-1} \\ 0 & M_{oo}^{-1} \end{pmatrix} \tag{4.19}$$

from the left, one has a closed equation for x_e,

$$(1 - M_{ee}^{-1} M_{eo} M_{oo}^{-1} M_{oe}) x_e = b_e' \equiv M_{ee}^{-1} b_e - M_{ee}^{-1} M_{eo} M_{oo}^{-1} b_o, \tag{4.20}$$

and after solving Eq. (4.20), x_o is provided as

$$x_o = b_o - M_{oo}^{-1} M_{oe} x_e. \tag{4.21}$$

The even-even and odd-odd block matrices must be inverted. The even-even matrix has the form

$$M_{ee} = \left(\begin{array}{cccc|c} \gamma_5 & -\sqrt{q_2} & & & 0 \\ -\sqrt{q_2} & -\gamma_5 & & & \sqrt{p_2} P_{Hee} \\ & & \gamma_5 & -\sqrt{q_1} & 0 \\ & & -\sqrt{q_1} & -\gamma_5 & \sqrt{p_1} P_{Hee} \\ \hline 0 & \sqrt{p_2} P_{Hee} & 0 & \sqrt{p_1} P_{Hee} & D_{ee} \end{array}\right), \tag{4.22}$$

$$D_{ee} = f_1\gamma_5 + f_2 \sum_j \mathrm{sign}\lambda_j v_{je} \times v_{je}^\dagger + p_0 P_{Hee}\gamma_5 P_{Hee}, \tag{4.23}$$

where v_{je} is even-part of the eigenvector v_j. M_{oo}, M_{eo}, and M_{oe} are similarly defined.

The matrix M_{ee} (and also M_{oo}) can be decomposed into left and right triangular matrices,

$$M_{ee} = L_e U_e = \left(\begin{array}{cccc|c} 1 & & & & \\ r_2 & 1 & & & \\ & & 1 & & \\ & & r_1 & 1 & \\ \hline 0 & s_2 & 0 & s_1 & 1 \end{array}\right) \left(\begin{array}{cccc|c} \gamma_5 & -\sqrt{q_2} & & & \\ & v_2 & & & \sqrt{p_2}P_{Hee} \\ & & \gamma_5 & -\sqrt{q_1} & \\ & & & v_1 & \sqrt{p_1}P_{Hee} \\ \hline & & & & u_0 \end{array}\right) \tag{4.24}$$

where

$$r_l = -\sqrt{q_l}\gamma_5 \tag{4.25}$$

$$v_l = -(1+q_l)\gamma_5 \tag{4.26}$$

$$s_l = -\frac{\sqrt{p_l}}{1+q_l} P_{Hee}\gamma_5 \tag{4.27}$$

$$u_0 = D_{ee} + \left(\sum_l \frac{p_l}{1+q_l}\right) P_{Hee}\gamma_5 P_{Hee} \tag{4.28}$$

Multiplication of U^{-1} and L^{-1} is easily implemented by forward and backward substitution. However, one need to solve $x = u_0^{-1}z$ for a given 4D even-vector z at each iteration step of the solver algorithm for Eq. (4.20).

Inversion of u_0. The standard method to solve a linear equation is the iterative Krylov subspace method. The Krylov subspace is, starting with initial residual vector $r_0 = z$, composed as $\mathcal{K}^k(A;z) = \mathrm{span}\{z, Az, \ldots, A^{k-1}z\}$. Because of the structure of the matrix u_0, the Krylov subspace closes at most at $2(N_{sbt}+1)$ dimension, and then is spanned by a non-orthogonal basis $\{z, \gamma_5 z, v_{je}, \gamma_5 v_{je}\}$ $(j = 1, \ldots N_{sbt})$. Thus $u_0^{-1}z$ is also expanded in this basis.

Denoting $w_j = v_{je}$, $w_{N_{sbt}+j} = \gamma_5 v_{je}$ $(j = 1, \ldots, N_{sbt})$, u_0 is expressed as

$$u_0 = \left(a + \sum_{i,j=1}^{2N_{sbt}} c_{i,j} w_i \times w_j^\dagger\right)\gamma_5. \tag{4.29}$$

The coefficients are given as $a = f_1 + p_0 + u_H$,

$$c_{i,j} = -u_H\delta_{i,j} \equiv \left(\sum_l \frac{p_l}{1+q_l}\right)\delta_{i,j} \tag{4.30}$$

$$c_{i+N_{sbt},j+N_{sbt}} = -u_H\delta_{i,j} \tag{4.31}$$

$$c_{i,j+N_{sbt}} = [f_2\mathrm{sign}(\lambda_i) - p_0\lambda_i]\delta_{i,j} + u_H(v_{ie}^\dagger\gamma_5 v_{je}). \tag{4.32}$$

for $i,j = 1, \ldots, N_{sbt}$. u_0 is also be expanded as

$$u_0^{-1} = \gamma_5\left(\bar{a} + \bar{a}'\gamma_5 + \sum_{i,j=1}^{2N_{sbt}} \bar{c}_{i,j} w_i \times w_j^\dagger\right). \tag{4.33}$$

		5D solver		4D solver	
m_q	N_{poly}	time [sec]	D_W mult [k]	time [sec]	D_W mult [k]
0.400	10	16.7	27	41.3	119
0.100	10	41.0	65	161.2	467
0.050	10	75.5	119	322.2	932
0.035	10	104.7	165	458.8	1328
0.025	10	141.2	222	621.5	1801
0.015	10	203.8	321	787.9	2281
0.400	20	28.7	56	55.4	125
0.100	20	65.2	127	216.2	489
0.050	20	121.4	236	431.2	976
0.035	20	168.8	329	614.3	1,390
0.025	20	228.0	444	832.1	1,883
0.015	20	330.1	642	1050.0	2,377

Table 1: The convergence of 5D and 4D solvers on Blue Gene 1024-node class. The convergence criterion is $|r|^2/|b|^2 < 10^{-20}$. Note that the $|b - (D^\dagger D)x|^2$ of the 5D solver is of one order of magnitude larger that of the 4D solver.

From the condition $u_0 u_0^{-1} = 1$, one easily obtains that $\bar{a} = 1/a$, $\bar{a}' = 0$, and $\bar{c}_{i,j}$ is determined by solving a linear equation

$$\sum_l \left[a\delta_{il} + \sum_k c_{ik}(w_k^\dagger w_l) \right] \bar{c}_{lj} = -\frac{1}{a} c_{ij}. \tag{4.34}$$

This inversion must be done only once before starting the 5D solver. In each step of iteration, required operation is the $2N_{sbt}$ inner products $(w_j^\dagger b)$ and vector operations related to w_j. This is not a heavy operation compared to other parts of the 5D solver.

4.3 Performance test

A comparison of performance of the 5D and 4D solver is performed on a single configuration of $16^3 \times 48$ lattice generated at $\beta = 2.30$, $N_f = 2 + 1$ with $m_{ud} = 0.100$, $m_s = 0.100$, and at $Q = 0$. The result on Blue Gene 1024-node job class is shown in Table 1 as well as in Fig. 1 for $m_q \leq 0.1$. The convergence time and numbers of D_W mult were observed for various quark masses and $N_{poly} = 10$ and 20. The former N_{poly} corresponding the value used in the productive run. The convergence criterion is $|r|^2/|b|^2 < 10^{-20}$. Note that the $|b - (D^\dagger D)x|^2$ of the 5D solver is of one order of magnitude larger than that of the 4D solver. The threshold of eigenvalues for low-mode subtraction is set as $\lambda_{thrs} = 0.045$, which leads on this configuration $N_{sbt} = 8$.

There are specific features for each of these two algorithms.

- Increasing the degree of polynomial, N_{poly}, Relaxed CG increases the cost only gradually, because the multishift CG mothod is applied for the inner loop. In 5D solver, since the size of vector is almost propotional to N_{poly}, the numerical cost increases linearly. (However, these features are not apparent in the above performance test. This may depend also on the machine archtecture.)

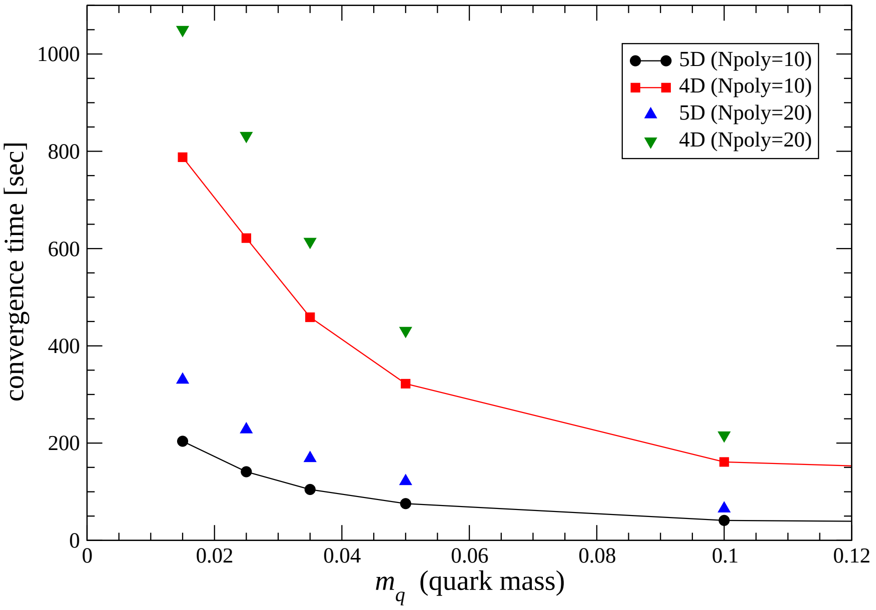

Figure 1: The quark mass dependence of the convergence time of 5D and 4D solvers on Blue Gene 1024-node class. See also the comments of Table 1.

- By relaxed CG, several masses of overlap fermion can be solver simultaneously by making use of multishift CG for outer loop. The 5D solver can be applied to single quark mass mass at once. Thus we adopt the relaxed CG to the spectroscopy (partially quenched), in which the cost can be largely reduced by solving quark propagator for several masses simultaneously. On the other hand, in HMC the quark mass is fixed to a single value, and hence the 5D solver is employed.

4.4 Additional topics on the 5D solver

Approximate solution Suppose that one has an approximate solution $\tilde{\psi}_4$ of a 4D equation,

$$D_{ov}\psi_4 = \chi_4. \tag{4.35}$$

If one can also compose an approximate solution of 5D equation from $\tilde{\psi}_4$, it can provide a good initial guess for the 5D linear problem.

Because of Eq. (4.4), when ψ_4 is already known, ϕ is given by solving

$$A\phi = -B\psi_4. \tag{4.36}$$

More explicitly,

$$\phi = \begin{pmatrix} \phi_N \\ \phi_{N-1} \\ \vdots \end{pmatrix}, \quad \phi_i = \begin{pmatrix} \sqrt{q_i} \\ H_W \end{pmatrix} \frac{\sqrt{p_i}}{H_W^2 + q_i}\psi_4, \tag{4.37}$$

which is easily determined simultaneously by using the multishift CG solver.

There are two straightforward applications:

- Adaptive 5D solver
  At early stage of the CG iteration, one does not require the full precision to D_{ov}. The above method enables us to change the value of N_{poly} at an intermediate stage of CG iteration. Namely, the smaller N_{poly} is used at the earlier CG iterations. The precisions at which N_{poly} is changed are selected in accord with the precisions of the approximation to the sign function with that N_{poly}. This idea was examined in a reort [28] on a N_f =2+1 configuration with $m_q = 0.015$ and $m_s = 0.100$. In summary, the adoptive 5D solver achieves about 15% acceleration for the dynamical quark, while almost no improvement for the preconditioner. Present simulation code does not include this technique.

- Chronological estimator in HMC update.
  The chronological estimator technique compose an initial guess of linear equation by previous solutions. To apply it to the 5D solver, one needs to store large 5D solutions. The above method reduces the size of stored vectors from $(2N_{poly}+1) \times N_q$ to N_q, where $N_q = 2 \cdot N_{color} \cdot N_{spinor} \cdot N_{site}$

Hermitian version of 5D solver. In the implementation in Subsec. 4.2, the even-odd preconditioned matrix in Eq. 4.20 was not hermitian. Therefore the CGNE algorithm is employed to solve it. However, the hermitian (while not positive definite) version of the even-odd preconditioned matrix can also be constructed [30]. Then the algorithms for hermitian matrices, *e.g.* MINRES, are applicable. Unfortunately, this implementation does not improve the cost [30].

5 Hybrid Monte Carlo algorithm

The hybrid Monte Carlo algorithm introduces the conjugate momenta to the gauge field, and update them with molecular dynamical evolution based on the Hamiltonian formalism.

- At the beginning of this evolution period, the conjugate momenta and pseudofermion fields are refreshed according to the Gaussian weight (Langevin algorithm).
- The molecular dynamical evolution is performed with the leap-frog algorithm so as to conserve the total Hamiltonian.
- At the end of evolution, the Metropolis test is performed so that the step size error is statistically corrected. For this purpose, total Hamiltonian is computed at the beginning and at the end of molecular dynamical evolution.

These whole steps compose one ‘trajectory’.

5.1 Hamiltonian

The Hamiltonian is defined as

$$\mathcal{H}[H,U;\phi] = \sum_{x,\mu} \frac{1}{2} \mathrm{Tr}\left[P_\mu(x)^2\right] + S[U,\phi]. \tag{5.1}$$

$P_\mu(x)$ is conjugate momenta to the link variable $U_\mu(x)$. Writing link variable as

$$U_\mu(x) = \exp\left[iA_\mu^a(x)t^a\right], \tag{5.2}$$

where t^a is SU(3) generator satisfying $\mathrm{Tr}(t^a t^b) = \delta_{ab}$,

$$\begin{aligned} \frac{dU_\mu(x)}{d\tau} &= i\frac{dA_\mu^a(x)}{d\tau} t^a U_\mu(x) \equiv iP_\mu(x)U_\mu(x), \\ P_\mu(x) &= p_\mu^a(x) t^a. \end{aligned} \tag{5.3}$$

Therefore $p_\mu^a(x)$ is conjugate to $A_\mu^a(x)$, and $P_\mu(x)$ is hermitian and traceless. The kinetic term of Eq. (5.1) implies

$$\frac{1}{2}\mathrm{Tr}\left[P_\mu(x)^2\right] = \frac{1}{2}\sum_a \left[p_\mu^a(x)\right]^2. \tag{5.4}$$

Implementation: In the Fortran program, variable `H` corresponds to iH.

5.2 Langevin step

At the beginning of trajectory, p_μ^a is randomly refreshed with Gaussian probability measure,

$$P(p_\mu^a) = \exp\left[-\frac{1}{2}(p_\mu^a)^2\right]. \tag{5.5}$$

The pseudo-fermion field is given by the heat-bath algorithm. During the molecular dynamical evolution, the pseudo-fermion field plays a role of external field. All pseudo-fermionic terms appearing in the present action is written as the form of

$$\begin{aligned} \exp(-S_{PF}) &= \exp\left[-\phi^\dagger(\mathcal{D}^\dagger\mathcal{D})^{-1}\phi\right] \\ &= \exp(-\xi^\dagger\xi), \end{aligned} \tag{5.6}$$

and hence easily generated by Gaussian random number generator. After generating ξ, ϕ is given as

$$\phi = \mathcal{D}^\dagger\xi. \tag{5.7}$$

One-flavor case. The pseudo-fermion fields ϕ_{1+} and ϕ_{2+} must be given according to a probability density $\exp(-S_{PF1})$ and $\exp(-S_{PF2})$. This is achieved with Gaussian distributed field ξ_{1+} and $\xi = 2_+$ by transformations:

$$\phi_{1+} = \sqrt{Q_+(m')}\cdot\xi_{1+}, \tag{5.8}$$

$$\phi_{2+} = \sqrt{\frac{Q_+(m)}{Q_+(m')}}\cdot\xi_{2+}. \tag{5.9}$$

The operator $\sqrt{Q_+(m')}$ can be implemented by, e.g., partial fraction expansion (Zolotarev's approximation) by expressing as [4]

$$\phi_{1+} = \frac{Q_+(m')}{\sqrt{Q_+(m')}}\cdot\xi_{1+}. \tag{5.10}$$

This part is more time consuming than the $N_f = 2$ HMC. Other approaches, for example Chebyshev polynomial approximation, should be compared.

5.3 Molecular dynamical evolution

Let us start with general discussion. For dynamical variable p and its conjugate momentum p, the evolution operator V_p and V_q is defined so as to evolve $\{p, q\}$ as

$$\begin{aligned} V_q(\Delta\tau): &\qquad \{p, q\} \to \{p, q + \Delta\tau \cdot p\} \\ V_p(\Delta\tau): &\qquad \{p, q\} \to \{p - \Delta\tau \frac{\partial S}{\partial q}, q\}. \end{aligned} \tag{5.11}$$

By requiring reversibility, the leap-frog integrator is constructed as

$$V_{PUP}(\Delta\tau) = V_p(\frac{\Delta\tau}{2})V_q(\Delta\tau)V_p(\frac{\Delta\tau}{2}). \tag{5.12}$$

This form is called 'PUP-type' integrator, since variable q is the link variable in the present case. Alternatively, 'UPU-type' integrator is also defined as

$$V_{UPU}(\Delta\tau) = V_q(\frac{\Delta\tau}{2})V_p(\Delta\tau)V_q(\frac{\Delta\tau}{2}). \tag{5.13}$$

Standard leap-frog evolution is implemented as, for example by adopting UPU-type integrator,

$$V(\Delta\tau)^n = V_q(\frac{\Delta\tau}{2}) \left(\prod_i^{n-1} [V_p(\Delta\tau)V_q(\Delta\tau)] \right) V_p(\Delta\tau)V_q(\frac{\Delta\tau}{2}). \tag{5.14}$$

Practically, UPU is known to be more efficient than PUP.

In the present case, p and q correspond to $P_\mu(x)$ and $U_\mu(x)$, respectively. By differentiating the Hamiltonian $\mathcal{H}$ by simulation time τ,

$$\frac{d\mathcal{H}}{d\tau} = \sum_{\mu,x} \mathrm{Tr}\left[\frac{dP_\mu(x)}{d\tau}P_\mu(x) + iP_\mu(x)R_\mu(x)\right] \tag{5.15}$$

where the force $R_\mu(x)$ satisfying

$$\frac{dS}{d\tau} = \sum \mathrm{Tr}[iP_\mu(x)R_\mu(x)] \tag{5.16}$$

is anti-hermitian and traceless. The evolution of $P_\mu(x)$ is

$$\begin{aligned} iP_\mu(x) &\to iP_\mu(x) + \Delta\tau R_\mu(x), \\ U_\mu(x) &\to \exp(\Delta\tau \cdot iH)U_\mu(x). \end{aligned} \tag{5.17}$$

Sexton-Weingarten (multi-time step) acceleration Sexton and Weingarten introduced multi-time step into the above molecular dynamical evolution.

$$\mathcal{H} = T(p) + S_1(q) + S_2(q) \equiv \mathcal{H}_1 + \mathcal{H}_2 \tag{5.18}$$

where

$$\mathcal{H}_1 = T(p) + S_1(q), \qquad \mathcal{H}_2 = S_2(q). \tag{5.19}$$

Then the evolution operator can be composed as

$$V(\Delta\tau) = V_2(\frac{\Delta\tau}{2})\left[V_1(\frac{\Delta\tau}{m})\right]^m V_2(\frac{\Delta\tau}{2}). \tag{5.20}$$

V_2 includes only the evolution of p. This procedure is efficient when the force $R_1 = \partial S_1/\partial q$ is much larger than $R_2 = \partial S_2/\partial q$.

The multi-time step acceleration can be generalized to cases of more than two time steps. In the case of three time step,

$$V(\Delta\tau) = V_3(\frac{\Delta\tau}{2}) \left\{ V_2(\frac{\Delta\tau}{2n}) \left[V_1(\frac{\Delta\tau}{mn}) \right]^m V_2(\frac{\Delta\tau}{2n}) \right\}^n V_3(\frac{\Delta\tau}{2}). \tag{5.21}$$

We set $S_1 = S_G + S_{Fukaya}$, $S_2 = S_{PF}^{(1)}$, and $S_3 = S_{PF}^{(2)}$.

Improved integrator. Instead of the integrator (5.12), Omelyan *et al.* considered the following type of the integrator [23, 24]:

$$V_{PUPUP}(\Delta\tau) = V_p(\lambda\Delta\tau)V_q(\frac{\Delta\tau}{2})V_p[(1-2\lambda)\Delta\tau]V_q(\frac{\Delta\tau}{2})V_p(\lambda\Delta\tau) \tag{5.22}$$

The leading error of this integrator is $C\Delta t^3$, and the coefficient C is represented as

$$C = \alpha(\lambda)[T,[V,T]] + \beta(\lambda)[V,[V,T]], \tag{5.23}$$

$$\alpha(\lambda) = \frac{1-6\lambda+6\lambda^2}{12}, \qquad \beta(\lambda) = \frac{1-6\lambda}{24}. \tag{5.24}$$

where T and V are the evolution operator for the kinetic and potential part of the Hamitonian. Minimizing

$$\mathcal{E}_3 = \sqrt{\alpha(\lambda)^2 + \beta(\lambda)^2}, \tag{5.25}$$

they found

$$\lambda_c = \frac{1}{2} - \frac{(2\sqrt{326}+36)^{1/3}}{12} + \frac{1}{(6\sqrt{326}+36)^{1/3}} \simeq 0.193\ 183\ 327\ 503\ 7836. \tag{5.26}$$

The Omelyan integrator can also be implemented for the nested integrator with multi-time step.

Comment on the performance:

5.4 Metropolis test

At the end of the molecular dynamical steps, Metropolis test is performed so as to ensure the detailed balance condition. The candidate of new configuration q' is accepted with probability

$$P = \min\left\{ 1, \frac{e^{-\mathcal{H}(p',q')}}{e^{-\mathcal{H}(p,q)}} \right\}. \tag{5.27}$$

If q' is rejected, the initial configuration q is adopted as the new configuration.

5.5 Gauge field part

Let us consider the admissible action first.

$$\frac{dS_G}{d\tau} = \beta \sum_{x,\mu>\nu} \left(-\frac{1}{3}\mathrm{ReTr}\frac{dP_{\mu\nu}(x)}{d\tau} \right) \frac{1}{[1-\frac{1}{\epsilon}\mathrm{ReTr}P_{\mu\nu}(x)]^2}. \tag{5.28}$$

Then the gauge force $R_{G\mu}(x)$ is expressed as

$$R_\mu(x) = -\frac{\beta}{3}\left[\sum_{\pm\nu} \frac{U_\mu(x)V^\dagger_{\mu,\nu}(x)}{[1-\frac{1}{\epsilon}S_{\mu,\nu}(x)]^2} \right]_{AT} \tag{5.29}$$

where

$$S_{\mu,\pm\nu}(x) = \mathrm{ReTr}[U_\mu(x)V^\dagger_{\mu,\pm\nu}(x)], \tag{5.30}$$

$$\begin{aligned} V_{\mu,+\nu}(x) &= U_\nu(x)U_\mu(x+\hat{\nu})U^\dagger_\nu(x+\hat{\mu}), \\ V_{\mu,-\nu}(x) &= U^\dagger_\nu(x-\hat{\nu})U_\mu(x-\hat{\nu})U_\nu(x-\hat{\nu}+\hat{\mu}), \end{aligned} \tag{5.31}$$

and $[\ldots]_{AT}$ means anti-hermitian-traceless.

In the case of rectangular improved gauge action, $R_{G\mu}(x)$ is obtained similarly.

5.6 Overlap quark part

We first describe the two flavor case in detail, and finally briefly commnet on the one flavor case which is almost the same as for the former. We consider the Hasenbusch preconditioned version, Eq.(2.10), since the unpreconditioned action is the same as the first term of the preconditioned one, Eq.(2.11).

$$S_{PF} = \phi^\dagger[D(m)^\dagger D(m)]^{-1}\phi. \tag{5.32}$$

$$\frac{d}{d\tau}S^{(1)}_{PF} = -\psi^\dagger\frac{d}{d\tau}[D(m)^\dagger D(m)]\psi, \tag{5.33}$$

where $\psi = [D(m)^\dagger D(m)]^{-1}\phi$.

$$\begin{aligned} \frac{d}{d\tau}D(m)^\dagger D(m) &= F^2\frac{d}{d\tau}\left\{R^2+1+R[\epsilon(H_W)\gamma_5+\gamma_5\epsilon(H_W)]\right\} \\ &= f\,[\,\dot{\epsilon}(H_W)\gamma_5+\gamma_5\dot{\epsilon}(H_W)\,] \end{aligned} \tag{5.34}$$

where $F = m_0 - m/2$, $R = (m_0 + \frac{m}{2})/(m_0 - \frac{m}{2})$, and $f = m_0^2 - m^2/4$. $\epsilon(x)$ is approximation to the sign function. Defining as $\psi_5 = \gamma_5\psi$, Eq. (5.33) is written as

$$\frac{d}{d\tau}S^{(1)}_{PF} = -f\left[\,\psi^\dagger\dot{\epsilon}(H_W)\psi_5 + \psi_5^\dagger\dot{\epsilon}(H_W)\psi\,\right] \tag{5.35}$$

Using the formula

$$\epsilon(x) = x(x^2+c_{2n})\sum_{l=1}^{n}\frac{b_l}{x^2+c_{2l-1}}, \tag{5.36}$$

where $x = h_W = H_W/\lambda_{min}$,

$$\begin{aligned} \frac{d\epsilon}{d\tau} &= \frac{dx}{d\tau}(x^2 + c_{2n})\sum_{l=1}^{n} \frac{b_l}{x^2 + c_{2l-1}} \\ &\quad + \sum_{l=1}^{n}(c_{2l-1} - c_{2n})\frac{x}{x^2 + c_{2l-1}}\left(x\frac{dx}{d\tau} + \frac{dx}{d\tau}x\right)\frac{b_l}{x^2 + c_{2l-1}}. \end{aligned} \quad (5.37)$$

Defining as

$$\begin{aligned} \psi_l &= \frac{1}{x^2 + c_{2l-1}}\psi, \\ \psi_{5l} &= \frac{1}{x^2 + c_{2l-1}}\gamma_5\psi, \end{aligned} \quad (5.38)$$

which can be determined by, for example, applying multi-shift CG method twice. The first term of Eq. (5.35) is expressed as

$$\begin{aligned} \left[\frac{d}{d\tau}S_{PF}^{(1)}\right]_{1st-term} &= -f\psi_5^\dagger\frac{dh_W}{d\tau}(h_W^2 + c_{2n})\sum_{l=1}^{n} b_l\psi_l \\ &\quad - f\sum_{l=1}^{n}(c_{2l-1} - c_{2n})\psi_{5l}^\dagger h_W\left(h_W\frac{dh_W}{d\tau} + \frac{dh_W}{d\tau}h_W\right)\psi_l. \end{aligned} \quad (5.39)$$

The second term of Eq. (5.35) is obtained by replacement

$$\begin{aligned} &\psi^\dagger \to \psi_5^\dagger, \quad \psi_l^\dagger \to \psi_{5l}^\dagger, \\ &\psi_5 \to \psi, \quad \psi_{5l} \to \psi_l. \end{aligned} \quad (5.40)$$

Force of Wilson kernel:
The derivative of the hermitian Wilson kernel, $dH_W/d\tau$, is a standard ingredient of HMC with Wilson-type fermions:

$$\begin{aligned} \zeta^\dagger\frac{dH_W}{d\tau}\eta &= -\kappa\cdot\zeta^\dagger(x)\gamma_5\sum_{x,\mu}\{(1-\gamma_\mu)iP_\mu(x)U_\mu(x)\eta(x+\hat{\mu}) \\ &\qquad -(1+\gamma_\mu)U_\mu^\dagger(x-\hat{\mu})iP_\mu(x-\hat{\mu})\eta(x-\hat{\mu})\Big\} \\ &= -\kappa\sum_{x,\mu}iP_\mu(x)_{ab}\Big\{\zeta^\dagger(x)_b\gamma_5[(1-\gamma_\mu)U_\mu(x)\eta(x+\hat{\mu})]_a \\ &\qquad -[(1-\gamma_\mu)U_\mu(x)\zeta(x+\hat{\mu})]_b^\dagger\gamma_5\eta(x+\hat{\mu})]_a\Big\}. \end{aligned} \quad (5.41)$$

Therefore, defining $T_{+\mu}\eta(x) \equiv (1-\gamma_\mu)U_\mu(x)\eta(x+\hat{\mu})$,

$$R_\mu(x)_{ab} = -\kappa\left[\zeta^\dagger(x)_b\gamma_5[T_{+\mu}\eta(x)]_a - [T_{+\mu}\zeta(x)]_b^\dagger\gamma_5\eta(x)_a\right]_{AT}. \quad (5.42)$$

Force of preconditioned overlap kernel:
For the preconditioned overlap operator, Eq. (2.12), the force is easily obtained.

$$\frac{d}{d\tau}S_{PF}^{(2)} = -\left[Z^\dagger\dot{\epsilon}(H_W)Y + Y^\dagger\dot{\epsilon}(H_W)Z\right] \quad (5.43)$$

where

$$
\begin{aligned}
Y &= [D^\dagger(m)D(m)]^{-1}D^\dagger(m')\phi_2, \\
Z &= \gamma_5[fY - F'\phi_2],
\end{aligned}
\tag{5.44}
$$

and $f = m_0^2 - m^2/4$, $F' = m_0 - m'/2$. Then computation of Eq. (5.43) can be done by the same routine to compute the preconditioner, Eq. (5.35), *i.e.* the standard overlap force.

One flavor case:
Once the pseudo-fermion fields is provided, the molecular dynamical evolution for the one flavor is straightforward, since

$$\frac{dS_{PF}^{(1)}}{d\tau} = \phi_{1+}^\dagger P_+ \left(\frac{dQ(m')^{-1}}{d\tau}\right) P_+\phi_{1+}, \tag{5.45}$$

and so on can be implemented as almost same as the $N_f = 2$ case.

5.7 Fukaya term

The force of Fukaya term is easily computed making use of the routine to compute the force of Wilson kernel, Eq. (5.42).[6]

$$
\begin{aligned}
\frac{dS_{Fukaya}}{d\tau} &= \frac{d}{d\tau}\phi_W D_W(\mu)[D_W^\dagger D_W]^{-1}D_W^\dagger(\mu)\phi_W \\
&= -\left[\zeta^\dagger\gamma_5\frac{dH_W}{d\tau}\eta + \eta^\dagger\gamma_5\frac{dH_W}{d\tau}\zeta\right],
\end{aligned}
\tag{5.46}
$$

where

$$
\begin{aligned}
\eta &= [D_W^\dagger D_W]^{-1}D_W(\mu)^\dagger\phi_W, \\
\zeta &= D_W\eta - \phi_W.
\end{aligned}
\tag{5.47}
$$

5.8 Additional issues in HMC algorithm

In this subsection, we describe the algorithms which are not used in the latest version of our simulation.

5.8.1 Reflection/refraction prescription

During update of link variable, the lowest eigenvalue of H_W may change the sign, where the value of S_{PF} discontinuously changes. Ref. [19] introduced reflection/refraction.

Suppose that the lowest eigenmode λ acrosses zero at time τ_0.

The antihermitian normal vector N of the zero-eigenvalue surface is expressed as

$$N = \left.\frac{D\lambda}{\sqrt{\mathrm{Tr}[(D\lambda)^2]}}\right|_{\lambda=0}, \tag{5.48}$$

[6] Cf. Fukaya's memo.

where $D\lambda = \langle\lambda|DH_W|\lambda\rangle$.

$$\Delta S_{PF} = S_{PF}[\lambda(\tau = \tau_0 + \epsilon)] - S_{PF}[\lambda(\tau = \tau_0 - \epsilon)] \tag{5.49}$$

Then reflection or refraction occurs depending on the values of ΔS_{PF} and $\mathrm{Tr}[NH]$:

$$\text{Reflection:} \quad \frac{1}{2}\mathrm{Tr}[NH]^2 < \Delta S_{PF} \quad \Rightarrow \quad H = 2\mathrm{Tr}[NH] \cdot N \tag{5.50}$$

$$\text{Refraction:} \quad \frac{1}{2}\mathrm{Tr}[NH]^2 > \Delta S_{PF} \quad \Rightarrow \quad H = -\mathrm{Tr}[NH] \cdot N \left(1 - \sqrt{1 - 2\Delta S/\mathrm{Tr}[NH]^2}\right). \tag{5.51}$$

Implementation: To detect the zero-crossing, the lowest eigenmode is monitored during every step of evolution of U_μ. If the lowst eigenvalue changes the sign, whether $\lambda_{before} \cdot \lambda_{after}$ is less than certain value (input parameter `E_crit_wall`) is checked. Then 'wall'-finding process starts.

Firstly, a value of τ_0, at which the lowest eigenvalue vanishes, is estimated by linear interpolation. If the value of $|\lambda|$ is larger than $|\lambda_{before}|$ or $|\lambda_{after}|$, this signals that the lowest eigenvalues before and after evolution has no relation. In this case, the program quits the wall-finding process.

If the program is treating a true zero-crossing, the value of τ_0 is determined until $|\lambda|$ is less than certain value (input parameter `E_conv_wall`). Then ΔS_{PF} is evaluated, and accordingly reflection or refraction occurs.

5.8.2 Noisy Metropolis test

For the $N_f = 2$ simulation at $\beta = 2.3$ [17], the 5D solver is adopted without the low-mode preconditioning. This makes the HMC update inaccurate when very low eigenvalue of H_W appears. To correct this inaccuracy, the nisy Metropolis algorithm is employed in $N_f = 2$ simulation.

6 Eigenmodes of ovelap fermion

This section summarize the nature of eigenvalues of the Ginsparg-Wilson fermions following the textbook by S. Aoki [31].

Let us consider a fermion operator which satisfies the Ginsparg-Wilson relation [32]

$$D\gamma_5 + \gamma_5 D = aDR\gamma_5 D, \tag{6.1}$$

where R is Hermitian and commutable with γ_5, and γ_5-Hermiticity

$$\gamma_5 D \gamma_5 = D^\dagger. \tag{6.2}$$

In addition, we assume locality of R, $R_{xy} = R\delta_{xy}$. Hereafter a is set to 1. In the case of overlap fermion of Eq. (2.3), $R = 1/M_0$.

With these conditions, there exists a unitary operator V satisfying

$$D = \frac{1}{R}(1 + V), \qquad V^\dagger V = VV^\dagger = 1, \qquad \gamma_5 V \gamma_5 = V^\dagger. \tag{6.3}$$

It is also easy to show that D and $D^\dagger$ are commutable:

$$DD^\dagger = D^\dagger D. \tag{6.4}$$

Eigenmodes of D. Consider an eigenmode $(\lambda, |\lambda\rangle)$ satisfying eigenvalue equation

$$D|\lambda\rangle = \lambda|\lambda\rangle. \tag{6.5}$$

Then the commutability of D and $D^\dagger$ indicates that $D^\dagger|\lambda\rangle$ is also an eigenstate belonging to λ. Considering degeneracy, one can express it as

$$D^\dagger|\lambda, i\rangle = \sum_j c_{ij}|\lambda, j\rangle. \tag{6.6}$$

Then

$$c_{ki} = \langle\lambda, k|D^\dagger|\lambda, i\rangle = \langle\lambda, k|\bar{\lambda}_k|\lambda, i\rangle = \bar{\lambda}_k\delta_{ki} \tag{6.7}$$

leading to

$$D^\dagger|\lambda, i\rangle = \bar{\lambda}|\lambda, i\rangle, \tag{6.8}$$

i.e. $|\lambda, i\rangle$ is a simultaneous eigenstate of D and $D^\dagger$.

From the Ginsparg-Wilson relation, $\lambda + \bar{\lambda} = R\bar{\lambda}\lambda$. Setting $\lambda = x + iy$,

$$\left(x - \frac{1}{R}\right)^2 + y^2 = \left(\frac{1}{R}\right)^2, \tag{6.9}$$

namely eigenvalues of D distribute on a circle of radius R^{-1} and center $(1/R, 0)$. Thus λ takes real values only when $\lambda = 0$ or $\lambda = 2/R$.

$\gamma_5|\lambda\rangle$ is an eigenfunction of D with eigenvalue $\bar{\lambda}$:

$$D\gamma_5|\lambda\rangle = \gamma_5 D^\dagger|\lambda\rangle = \bar{\lambda}\gamma_5|\lambda\rangle. \tag{6.10}$$

Considering a degenerate case, expanding as

$$\gamma_5|\lambda, i\rangle = \sum_j U_{ij}|\bar{\lambda}, j\rangle, \tag{6.11}$$

$$\delta_{ji} = \langle\lambda, j|\gamma_5\gamma_5|\lambda, i\rangle = (UU^\dagger)_{ji} \tag{6.12}$$

namely U is unitary. Without degeneracy, one finds

$$\gamma_5|\lambda\rangle = e^{-i\theta}|\bar{\lambda}\rangle. \tag{6.13}$$

Eigenmodes of H^2. For $H^2 = D^\dagger D$ ($H \equiv \gamma_5 D$ in this note),

$$[H^2, \gamma_5] = 0 \tag{6.14}$$

is derived from the Ginsparg-Wilson relation. Thus the eigenvector of H^2 can be set to an simultaneous eigenvector of γ_5, *i.e.* a state with definite chirality. The eigenvalue of H^2 is $\Lambda = \bar{\lambda}\lambda$, and chiral eigenstates are composed as

$$|\Lambda, \pm\rangle = C\frac{1 \pm \gamma_5}{2}|\lambda\rangle \tag{6.15}$$

where C is a normalization constant. With Eq. (6.13),

$$|\Lambda, \pm\rangle = \frac{C}{2}\left(|\lambda\rangle \pm e^{-i\theta}|\bar{\lambda}\rangle\right). \tag{6.16}$$

This gives a relation between the chiral eigenstates of H^2 and the eigenstates of D.

Eigenmodes of H. Since $H \equiv \gamma_5 D$ is Hermitian, its eigenvalue λ_H is real.

$$H|\lambda_H\rangle = \lambda_H|\lambda_H\rangle \tag{6.17}$$

For eigenvalue of H^2, $\Lambda = \lambda_H^2 = \bar{\lambda}\lambda < (2/R)^2$ meands

$$-\frac{2}{R} \le \lambda_H \le \frac{2}{R}. \tag{6.18}$$

Let us consider a condition that $|\lambda_H\rangle$ is chiral:

$$\gamma_5|\lambda_H\rangle = \pm|\lambda_H\rangle. \tag{6.19}$$

Applying $|\lambda_H\rangle$ from left and right to the Gisparg-Wilson relation, one obtains $2\lambda_H = \pm R\lambda_H^2$. Thus

$$\lambda_H = 0, +\frac{2}{R} \qquad \text{for} \qquad \gamma_5|\lambda_H\rangle = \ |\lambda_H\rangle \tag{6.20}$$

$$\lambda_H = 0, -\frac{2}{R} \qquad \text{for} \qquad \gamma_5|\lambda_H\rangle = -|\lambda_H\rangle \tag{6.21}$$

Defining

$$\Gamma_5 \equiv \gamma_5\left(1 - \frac{R}{2}D\right), \tag{6.22}$$

The Ginsparg-Wilson relation indicates that H and Γ_5 are anticommutative:

$$\{H, \Gamma_5\} = 0. \tag{6.23}$$

Since

$$H\Gamma_5|\lambda_H\rangle = -\Gamma_5 H|\lambda_H\rangle = -\lambda_H\Gamma_5|\lambda_H\rangle, \tag{6.24}$$

if $\lambda_H \neq 0$ and $\Gamma_5|\lambda_H\rangle \neq 0$, $-\lambda_H$ is also an eigenvalue of H with eigenvector $\Gamma_5|\lambda_H\rangle$. $\Gamma_5|\lambda_H\rangle = 0$ occurs when

$$\langle\lambda_H|\Gamma_5\Gamma_5|\lambda_H\rangle = 1 - \frac{R^2}{4}\lambda_H^2 = 0, \tag{6.25}$$

namely $\lambda_H = \pm 2/R$. These condition means that paired eigenvalues $(\lambda_H, -\lambda_H)$ appears in non-chiral cases.

For paired eigenmodes,

$$|-\lambda_H\rangle = C\Gamma_5|\lambda_H\rangle = \frac{1}{\sqrt{1 - R^2\lambda_H^2/4}}\left(\gamma_5 - \frac{R}{2}\lambda_H\right)|\lambda_H\rangle. \tag{6.26}$$

Since $|\lambda_H\rangle$ and $|-\lambda_H\rangle$ are orthogonal, $\langle\lambda_H|\gamma_5|\lambda_H\rangle = R\lambda_H/2$. Similarly, $\langle-\lambda_H|\gamma_5|-\lambda_H\rangle = -R\lambda_H/2$, and $\langle-\lambda_H|\gamma_5|\lambda_H\rangle = \langle\lambda_H|\gamma_5|-\lambda_H\rangle = \sqrt{1-(R\lambda_H/2)^2}$ hold.[7] Thus with the basis of eigenstates of H, D is expressed as

$$D = \left(\begin{array}{cc} \frac{R}{2}\lambda_H^2 & -\lambda_H\sqrt{1-(\frac{R}{2}\lambda_H)^2} \\ \lambda_H\sqrt{1-(\frac{R}{2}\lambda_H)^2} & \frac{R}{2}\lambda_H^2 \end{array}\right) = \lambda_H\left(\begin{array}{cc} x & -\sqrt{1-x^2} \\ \sqrt{1-x^2} & x \end{array}\right) \tag{6.27}$$

[7] In numerical simulations, these states are determined by some kind of eigenvalue solver. Then a relative phase of obtained $|\lambda_H\rangle$ and $|-\lambda_H\rangle$ may be arbitrary, and to be adjusted so that this condition is satisfied.

where $x = R\lambda_H/2$. Solving eigenvalue equation,

$$\lambda_{\pm} = \lambda_H \left\{ \frac{R\lambda_H}{2} \pm i\sqrt{1 - \left(\frac{R\lambda_H}{2}\right)^2} \right\}. \tag{6.28}$$

Eigenstates are

$$\begin{aligned} |\lambda \equiv \lambda_+\rangle &= \frac{1}{\sqrt{2}}\left[|\lambda_H\rangle - i| - \lambda_H\rangle\right] && (6.29) \\ |\bar{\lambda} \equiv \lambda_-\rangle &= \frac{1}{\sqrt{2}}\left[|\lambda_H\rangle + i| - \lambda_H\rangle\right] && (6.30) \end{aligned}$$

With the same basis, γ_5 is represented as

$$\gamma_5 = \begin{pmatrix} x & \sqrt{1-x^2} \\ \sqrt{1-x^2} & -x \end{pmatrix} \tag{6.31}$$

The eigenvalue of this matrix is surely ±1. Eigenvectors are simultaneous eigenvectors of $\Lambda = \lambda_H^2$,

$$\begin{aligned} |\Lambda, +\rangle &= \frac{1}{\sqrt{2}}\left[\sqrt{1+x^2}\,|\lambda_H\rangle + \sqrt{1-x^2}\,| - \lambda_H\rangle\right] && (6.32) \\ |\Lambda, -\rangle &= \frac{1}{\sqrt{2}}\left[\sqrt{1-x^2}\,|\lambda_H\rangle - \sqrt{1+x^2}\,| - \lambda_H\rangle\right]. && (6.33) \end{aligned}$$

θ apearing in the relation (6.13) satisfies $\tan\theta = \sqrt{1-x^2}/x$.
The chiral eigenmodes are as follows.

(1) $\lambda_H = 0$:

$$\gamma_5|\lambda_H = 0, \pm\rangle = \pm|\lambda_H = 0, \pm\rangle \tag{6.34}$$

is also the simultaneous eigenvector corresponding to $\Lambda = 0$ and $\lambda = 0$.

(2) $\lambda_H = \pm\frac{2}{R}$:

$$\gamma_5|\lambda_H = \pm 2/R\rangle = \pm|\lambda_H = \pm 2/R\rangle. \tag{6.35}$$

The eigenvectors $|\lambda_H = \pm 2/R\rangle$ is also an eigenvector of D with eigenvalue $\lambda = |\lambda_H|$, since

$$D|\lambda_H = \pm 2/R\rangle = \gamma_5(\pm)\frac{2}{R}|\lambda_H = \pm 2/R\rangle = \frac{2}{R}|\lambda_H = \pm 2/R\rangle. \tag{6.36}$$

Massive operators. We obtained the relation between eigenmodes of D and eigenmodes of H. In practical simulation, eigenvalue solver sometimes become unstable without small mass term. The overlap Dirac operator (2.2) contains the mass as an additive term, and thus the eigenvectors are unchenged with modified eigenvalues

$$\lambda(m) = \left(1 - \frac{m}{2M_0}\right)\lambda + m, \tag{6.37}$$

and similar relation for $\bar{\lambda}$. For $H(m) = \gamma_5 D(m)$, the mass enters as a coefficient of γ_5. For the chiral modes, $\lambda_H = 0$ and $\pm 2M_0$, they are eigenstates of γ_5 and the relation to $\lambda_H(m)$ is the same as Eq. (6.37). For the non-chiral modes, using Eq. (6.31), one can again diagonalize the matrix $H(m)$ and obtains two eigenvalues $\pm\lambda(m)$ as

$$\lambda_H(m) = \sqrt{\left(1 - m^2/4M_0^2\right)\lambda_H^2 + m^2}, \tag{6.38}$$

or inversely

$$\lambda_H = \sqrt{(\lambda_H^2(m) - m^2)/\left(1 - m^2/4M_0^2\right)}. \tag{6.39}$$

The corresponding eigenvectors are

$$\begin{aligned} |\lambda(m)\rangle &= \frac{1}{\sqrt{2\lambda_H(m)}}\left[\ \sqrt{\lambda_H(m) + \lambda_H}\,|\lambda_H\rangle + \sqrt{\lambda_H(m) - \lambda_H}\,|-\lambda_H\rangle\right] \\ |-\lambda(m)\rangle &= \frac{1}{\sqrt{2\lambda_H(m)}}\left[-\sqrt{\lambda_H(m) - \lambda_H}\,|\lambda_H\rangle + \sqrt{\lambda_H(m) + \lambda_H}\,|-\lambda_H\rangle\right] \end{aligned} \tag{6.40}$$

Since Eq. (6.40) is orthogonal transformation, its inverse is obviously

$$\begin{aligned} |\lambda_H\rangle &= \frac{1}{\sqrt{2\lambda_H(m)}}\left[\sqrt{\lambda_H(m) + \lambda_H}\,|\lambda_H(m)\rangle - \sqrt{\lambda_H(m) - \lambda_H}\,|-\lambda_H(m)\rangle\right] \\ |-\lambda_H\rangle &= \frac{1}{\sqrt{2\lambda_H(m)}}\left[\sqrt{\lambda_H(m) - \lambda_H}\,|\lambda_H(m)\rangle + \sqrt{\lambda_H(m) + \lambda_H}\,|-\lambda_H(m)\rangle\right] \end{aligned} \tag{6.41}$$

Thus eigenmodes of $H(m)$ can be converted to the eigenmodes of massless overlap Dirac operator.

A Algorithms for linear systems

A.1 Conjugate Gradient (CG) algorithm

The conjugate gradient(CG) algorithm is common and often most powerful algorithm for solving linear equations, in particular for large and sparse matrices. The CG algorithm applies to a hermitial and positive definite matrix. If the matrix A is not a hermitial and positive definite, the CG algorithm is applicable to $A^\dagger A$. This is called CGNE (CG for normal equation).

CG algorithm is an iterative solver which approximate the solution vector x to an equation

$$Ax = b \quad \text{(CG)} \qquad \text{or} \qquad A^\dagger A x = A^\dagger b \quad \text{(CGNE)} \tag{A.1}$$

by iteratively constructing x_i by multiplying A to the previously obtained vectors. After invention of the CG algorithm, which was one of the greatest algorithmic progress in 20th century, its mathematical background has extensively been investigated. The CG-type algorithms are now called Krylov subspace method, which obtain an approximate solution in the Krylov subspace $\mathcal{K}_k(v_0) = \text{span}\{v_0, Av_0, A^2v_0, \ldots, A^{k-1}v_0\}$. This family of algorithms has been extended to nonhermitian matrices, and include GMRES, BiCG, CGS, BiSGStab, etc.

Here we quote the CG algorithm for a hermitian and positive definite matrix matrix A.

CG algorithm:

```
(i) initial step
    $p_0 = r_0 = b$
    $x_0 = b$
```

```
(ii) iteration step
        for $i = 0, 1, 2, \dots$ (repeat until convergence)
        $\beta_i = -\frac{(r_i, r_i)}{(p_i, Ap_i)}$
        $x_{i+1} = x_i - \beta_i p_i$
        $r_{i+1} = r_i + \beta_i A p_i$
        $\alpha_i = \frac{(r_{i+1}, r_{i+1})}{(r_i, r_i)}$
        $p_{i+1} = r_{i+1} + \alpha_i p_i$
```

A.2 Multishift CG algorithm

Multi-shift solver [12, 13] enables to solve these n equations at once. We employ the multi-shift CG algorithm (for the standard CG algorithm, see Sec. A.1). The multi-shift solver algorithms make use of the fact that the Krylov subspace $\mathcal{K}_k(A, v_0)$ for matrix A and initial vector v_0 is unchanged against a shift $A \to A + \sigma$, *i.e.*, $\mathcal{K}_k(A, v_0) = \mathcal{K}_k(A + \sigma, v_0)$. This implies that the residual vector, which is perpendicular to the previous Klylov subspace, can be shared by n equations in Eq. (3.11). Then n solutions of Eq. (3.11) can be determined simultaneously. The algorithm is written as follows.

Multi-shift CG algorithm:

```
(i) initial step
        $x_0 = 0$                 $x_0^\sigma = 0$
        $r_0 = p_0 = b$           $p_0^\sigma = b$
                                  $\zeta_0^\sigma = \zeta_{-1}^\sigma = 1$

(ii) iteration step
        for $i = 0, 1, 2, \dots$ (repeat until convergence)
          $\beta_i = -\frac{(r_i, r_i)}{(p_i, Ap_i)}$
          $x_{i+1} = x_i - \beta_i p_i$
          $r_{i+1} = r_i + \beta_i A p_i$
          $\alpha_i = \frac{(r_{i+1}, r_{i+1})}{(r_i, r_i)}$
          $p_{i+1} = r_{i+1} + \alpha_i p_i$
                                  $\hat{\alpha}_i = 1 + \frac{\alpha_{i-1}\beta_i}{\beta_{i-1}}$
                                  $\zeta_{i+1}^\sigma = \left[(\hat{\alpha}_i - \sigma\beta_i)/\zeta_i^\sigma + (1 - \hat{\alpha}_i)/\zeta_{i-1}^\sigma\right]^{-1}$
                                  $\beta_i^\sigma = \frac{\zeta_{i+1}^\sigma}{\zeta_i^\sigma}\beta_i$
                                  $\alpha_i^\sigma = \left(\frac{\zeta_{i+1}^\sigma}{\zeta_i^\sigma}\right)\alpha_i$
                                  $x_{i+1}^\sigma = x_i^\sigma - \beta_i^\sigma p_i^\sigma$
                                  $p_{i+1}^\sigma = \zeta_{i+1}^\sigma r_{i+1} + \alpha_i^\sigma p_i^\sigma$
```

The operations in the right column are for the shifted equations.

Implementation: The solutions of Eq. (3.11) for larger c_j converges faster. For the convergence criterion, the squared norm of the vector p^σ is monitored and the itereation is stopped when it become less than the given criterion.

Acknowledgment

I would like to thank Satoru Ueda for pointing out many typos. This note is based on the work supported in part by the Grant-in-Aid of the Ministry of Education (Nos. 19740160, 20105005). Numerical simulations were performed on Hitachi SR11000 and IBM System Blue Gene Solution at High Energy Accelerator Research Organization (KEK) under a support of its Large-scale Simulation Program.

References

[1] H. Neuberger, "Exactly massless quarks on the lattice," Phys. Lett. B **417** (1998) 141 [arXiv:hep-lat/9707022];
"More about exactly massless quarks on the lattice," Phys. Lett. B **427** (1998) 353 [arXiv:hep-lat/9801031].

[2] M. Hasenbusch, "Speeding up the Hybrid-Monte-Carlo algorithm for dynamical fermions," Phys. Lett. B **519** (2001) 177 [arXiv:hep-lat/0107019].

[3] A. Bode, U. M. Heller, R. G. Edwards and R. Narayanan, "First experiences with HMC for dynamical overlap fermions," arXiv:hep-lat/9912043.

[4] T. DeGrand and S. Schaefer, "Simulating an arbitrary number of flavors of dynamical overlap fermions," JHEP **0607** (2006) 020 [arXiv:hep-lat/0604015].

[5] Y. Iwasaki, "Renormalization Group Analysis Of Lattice Theories And Improved Lattice Action. 2. Four-Dimensional Nonabelian SU(N) Gauge Model,"

[6] H. Fukaya, S. Hashimoto, K. I. Ishikawa, T. Kaneko, H. Matsufuru, T. Onogi and N. Yamada [JLQCD Collaboration], "Lattice gauge action suppressing near-zero modes of H(W)," Phys. Rev. D **74** (2006) 094505 [arXiv:hep-lat/0607020].

[7] ARPACK – Arnordi Package –, `http://www.caam.rice.edu/software/ARPACK/`

[8] T. Kalkreuter and H. Simma, "An Accelerated conjugate gradient algorithm to compute low lying eigenvalues: A Study for the Dirac operator in SU(2) lattice QCD," Comput. Phys. Commun. **93**, 33 (1996) [arXiv:hep-lat/9507023].

[9] J. van den Eshof, A. Frommer, T. Lippert, K. Schilling and H. A. van der Vorst, "Numerical methods for the QCD overlap operator. I: Sign-function and error bounds," Comput. Phys. Commun. **146**, 203 (2002) [arXiv:hep-lat/0202025].

[10] T. W. Chiu, T. H. Hsieh, C. H. Huang and T. R. Huang, "A note on the Zolotarev optimal rational approximation for the overlap Dirac operator," Phys. Rev. D **66**, 114502 (2002) [arXiv:hep-lat/0206007].

[11] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *NUMERICAL RECIPES in FORTRAN, 2nd ed.* (Cambridge Univ. Press, 1986,1992).

[12] A. Frommer, B. Nockel, S. Gusken, T. Lippert and K. Schilling, "Many masses on one stroke: Economic computation of quark propagators," Int. J. Mod. Phys. C **6**, 627 (1995) [arXiv:hep-lat/9504020].

[13] B. Jegerlehner, “Krylov space solvers for shifted linear systems,” arXiv:hep-lat/9612014.

[14] H. Neuberger, “Minimizing storage in implementations of the overlap lattice-Dirac operator,” Int. J. Mod. Phys. C **10**, 1051 (1999) [arXiv:hep-lat/9811019].

[15] T. W. Chiu and T. H. Hsieh, “A note on Neuberger’s double pass algorithm,” Phys. Rev. E **68**, 066704 (2003) [arXiv:hep-lat/0306025].

[16] N. Cundy, J. van den Eshof, A. Frommer, S. Krieg, T. Lippert and K. Schafer, “Numerical methods for the QCD overlap operator. III: Nested iterations,” Comput. Phys. Commun. **165** (2005) 221 [arXiv:hep-lat/0405003].

[17] S. Aoki *et al.* [JLQCD Collaboration], “Two-flavor QCD simulation with exact chiral symmetry,” Phys. Rev. D **78** (2008) 014508 [arXiv:0803.3197 [hep-lat]].

[18] S. Hashimoto *et al.* [JLQCD collaboration], “Lattice simulation of 2+1 flavors of overlap light quarks,” PoS **LAT2007** (2007) 101 [arXiv:0710.2730 [hep-lat]].

[19] Z. Fodor, S. D. Katz and K. K. Szabo, “Dynamical overlap fermions, results with hybrid Monte-Carlo algorithm,” JHEP **0408**, 003 (2004) [arXiv:hep-lat/0311010].

[20] A. Borici, “Computational methods for the fermion determinant and the link between overlap and domain wall fermions,” arXiv:hep-lat/0402035.

[21] R. G. Edwards, B. Joo, A. D. Kennedy, K. Orginos and U. Wenger, “Comparison of chiral fermion methods,” PoS **LAT2005** (2006) 146 [arXiv:hep-lat/0510086].

[22] T. Takaishi and P. de Forcrand, “Testing and tuning new symplectic integrators for hybrid Monte Carlo algorithm in lattice QCD,” Phys. Rev. E **73** (2006) 036706 [arXiv:hep-lat/0505020].

[23] I. P. Omelyan, I.M.Mryglod, and R. Folk, “Optimized Verlet-like algorithms for molecular dynamics simulation,” Phys. Rev. E **65** (2002) 056706.

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

[25] T. Kaneko, “GMRES(m) for overlap solver”, JLQCD internal report on 8 Dec 2005.

[26] T. Kaneko, “Relaxed overlap solver”, JLQCD internal report on 19 Jan 2006.

[27] T. Kaneko, “overlap solvers”, JLQCD internal report on 9 Feb 2006.

[28] H. Matsufuru, “Adaptive 5D solver”, JLQCD internal report on 12 Dec 2007.

[29] H. Matsufuru, “Omelyan integrator”, JLQCD internal report on 12 Nov 2008.

[30] H. Matsufuru, “5D solver with hermitian even-odd preconditioning”, JLQCD internal report on 14 Nov 2007.

[31] S. Aoki, *Field Theory on the lattice* (Springer Verlag Japan, 2005) [*in Japanese*].

[32] P. H. Ginsparg and K. G. Wilson, “A Remnant of Chiral Symmetry on the Lattice,” Phys. Rev. D **25** (1982) 2649.