# Introduction to lattice QCD simulations

Hideo Matsufuru

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

Email: hideo.matsufuru@kek.jp
WWW: http://www.rcnp.osaka-u.ac.jp/~matufuru/

7 September 2007, Ver.1.0

**Abstract**

Lattice QCD simulations have been a most powerful tool to analyze the nonperturbative nature of QCD. In this note, we introduce fundamental algorithms to perform numerical simulations of lattice QCD.

## Contents

# 1 Introduction

Quantum chromodynamics is the fundamental theory to describe the dynamics of quarks and gluons in hadrons [1]. The gluons are non-Abelian SU(3) gauge field interacting with quarks. At low-energy, QCD has tow distinct features: the confinement and the spontaneous breaking of the chiral symmetry. The former confines quarks and gluons in hadrons, and the latter is the origin of large effective masses of quarks and light pseudoscalar mesons such as pions. The QCD coupling has striking feature that it decreases at short length (the asymptotic freedom). Because of this property, the perturbation theory is applicable to high energy phenomena with momentum transfer $p \gg \Lambda_{QCD}$, where $\Lambda_{QCD} \sim 200$ MeV is a typical energy scale of QCD. On the other hand, as the energy scale approaching down to $\Lambda_{QCD}$, the coupling becomes too strong so that the perturbation theory breaks down. Thus alternative approach is called for.

Lattice QCD simulations have become a most powerful tool to analyze the nonperturbative nature of QCD. It is a gauge field theory on 4D Euclidean lattice spacetime, and quantized along the Feynman's path integral formalism. The quantized theory can be regarded as a statistical mechanics system, which enables us to perform numerical simulations by the Monte Carlo method. With development of computational equipments as well as numerical algorithms, lattice QCD simulations have become possible to provide quantitative predictions of hadronic matrix elements such as those needed for the flavor physics. Lattice QCD is also used to explore the natures of QCD dynamics, such as the confinement mechanism, the spontaneous chiral symmetry breaking, the instanton dynamics, the U(1) problem, the finite temperature phase transition, and so on. Similar approaches are of course also applicable to other field theories.

This note explains basic ingredients of numerical simulations of the lattice QCD. As an example of lattice QCD action, we adopt the Wilson gauge action and the Wilson fermion action which are most simple form of the lattice QCD actions, while essentially the same numerical techniques are applicable to other lattice actions. In particular, the hybrid Monte Carlo algorithm is explained in detail, which is now the most popular algorithm to generate the gauge configurations including quark loop effects. Also provided are the procedures to measure hadronic correlation functions constructed with quark propagators. To obtain the quark propagator, one needs to invert the fermion operator. This is also the most significant ingredient of dynamical QCD simulations.

The rest of this note is organized as follows. In the next section, basic setup of lattice QCD formulation is summarized. Since this note is devoted to issues of numerical simulations, we restrict ourselves in minimal materials necessary in later sections, while some comments on the fermion formulations are also included. For details, see for example [2, 4, 3]. In Sec. 3, overview of lattice QCD simulation is provided. The foundation of the Monte Carlo simulation is given in Sec. 4. In Sec. 5, the hybrid Monte Carlo algorithm is described for the case of quenched approximation. While for quenched simulations more efficient algorithms, such as pseudo-heat-bath algorithm, exist, this is a good preparation for the dynamical simulations. Section 6 describes the algorithm to invert the fermion operator so as to determine the quark propagator. The conjugate gradient algorithm is explained as an example. Section 7 describes the HMC algorithm with dynamical quarks.

# 2 QCD on the lattice

## 2.1 Continuum and lattice field theories

QCD action in the continuum Euclidean space-time is represented as

$$S_{QCD} = \int d^4x \frac{1}{4} F^a_{\mu\nu} F^a_{\mu\nu} + \sum_f \int d^4x \bar{\psi}_f(x)(\gamma_\mu D_\mu + m_f)\psi_f(x), \tag{1}$$

where $D_\mu = \partial_\mu + igA_\mu(x)$ is the covariant derivative,

$$F^a_{\mu\nu} = \partial_\mu A^a_\nu - \partial_\nu A^a_\mu + igf^{abc}A^b A^c \tag{2}$$

is the field strength, with $f^{abc}$ the structure constants of SU(3) group. The index $f$ labels the quark flavors. The Dirac matrices $\gamma_\mu$ satisfy the relation

$$\{\gamma_\mu, \gamma_\nu\} = 2\delta_{\mu\nu}. \tag{3}$$

The quark field $\psi_f$ has 3-color $\otimes$ 4-spinor components. By applying path integral quantization, an expectation value of a physical observable $O$ is represented as

$$\langle O \rangle = \frac{1}{Z} \int \mathcal{D}A\mathcal{D}\psi\mathcal{D}\bar{\psi} O(\bar{\psi}, \psi, A) \exp(-S_{QCD}[\bar{\psi}, \psi, A]). \tag{4}$$

In fact, to quantize a gauge theory in the continuum space-time we need to fix the gauge. In the path integral quantization of a nonabelian gauge theory, the gauge fixing is performed with a trick of Fadeev-Popov determinant which leads the ghost field. However, since the lattice regularization does not necessarily require the gauge fixing, we did not included the gauge fixing tern in the above argument, while in some cases the gauge fixing is convenient in practical simulations.

The lattice discretization provides a regularization of the theory. Finally the lattice spacing is taken to $a \to 0$: the continuum limit. In this limit, the correlation length (the inverse of the mass of the lightest mode) in lattice units diverges, i.e. the continuum limit is the second order phase transition. In lattice QCD simulations, all the quantities are measured in units of the lattice spacing $a$. The $a^{-1}$ provides lattice cutoff of the momentum and energy: they are restricted in the region less than $\pi/a$. However, the parameter being set in simulation is not the lattice spacing but the coupling between the neighboring sites (and the bare quark masses). The lattice spacing is determined by comparing a measured quantity, such as the proton mass, which is provided as a certain number, with the experimental value, $m_p = 938$ MeV. Thus the lattice scale depends on the quantity defining the scale.

Because of the asymptotic freedom, the limit of the coupling $g \to 0$ corresponds to the $a \to 0$ limit. On the other hand, in the strong coupling QCD exhibits the confinement phenomena (area low for the Wilson loops). It is essential to demonstrate that there is no phase transition during taking the continuum limit. This question is investigated first by Creutz [ref], and indeed the confinement region of the coupling is smoothly connected to the weak coupling region without any phase transition. This implies that we can take the extrapolation of the result obtained at strong coupling region to the continuum limit.

The lattice gauge theory is defined on 4D Euclidean lattices. The gauge field is defined on links connecting the nearest neighboring sites. The gauge degree of freedom is represented by an SU(3) matrix $U_\mu(x)$, called link variable, which is related to the gauge field $A_\mu(x)$ as

$U_\mu(x) \simeq \exp[igA_\mu(x+\hat{\mu}/2)]$. The quark field, $\psi(x)$, is defined as an anticommuting Grassmann variables on sites. The fields $U_\mu(x)$ and $\psi(x)$ are subject to the gauge transformation:

$$\psi(x) \rightarrow V(x)\psi(x), \quad \bar{\psi}(x) \rightarrow \bar{\psi}(x)V(x)^\dagger \tag{5}$$

$$U_\mu(x) \rightarrow V(x)U_\mu(x)V^\dagger(x+\hat{\mu}a) \tag{6}$$

where $\hat{\mu}$ is a unit vector along $\mu$-th direction. One can easily verify that these expressions for the gauge transformation indeed provides the continuum gauge transformation for the small enough $a$.

## 2.2 Lattice QCD actions

The actions for the gauge and quark field must be invariant under the above gauge transformation Eqs. (5) and (6). The following quantities are gauge invariant: (1) a closed loop composed of $U_\mu$; (2) $\bar{\psi}$ and $\psi$ connected with $U_\mu$. The action of the gauge and quark fields are constructed with these quantities.

Let us consider the smallest loop composed of the link variables,

$$U_{\mu\nu}(x) = U_\mu(x)U_\nu(x+\hat{\mu})U^\dagger(x+\hat{\nu})U_\nu^\dagger(x), \tag{7}$$

called a plaquette. Hereafter we put $a = 1$ for simplicity. For small lattice spacing, $U_{\mu\nu} \simeq \exp(iga^2F_{\mu\nu})$. If one take

$$S_G = \beta \sum_{x,\mu>\nu} \left(1 - \frac{1}{N_c}\mathrm{ReTr}U_{\mu\nu}(x)\right), \tag{8}$$

where $\beta = 2N_c/g^2$, in the limit of $a \rightarrow 0$

$$S_G \rightarrow \int d^4x \frac{1}{4} F^a_{\mu\nu}F^a_{\mu\nu} + O(a^2). \tag{9}$$

The Eq. (8) is called the Wilson gauge action which provide the simplest form of a gauge action on the lattice. As the lattice gauge action, Eq. (9) is not a unique choice; one can add an $O(a^2)$ term which disappears in the continuum limit. Appropriate choice of the additional terms would cancel the $O(a^2)$ error in (8) and can be used to construct improved actions [5]. For example, by adding rectangular loops, which has different $O(a^2)$ contributions from Eq. (7), an $O(a^2)$-improved actions is constructed. Such an improved action is important in numerical simulations, since the finite $a$ errors rapidly disappear and extrapolations to the continuum limit become more stable. However, we restrict ourselves in the case of the Wilson gauge action (9), since the essentially the same algorithm is also applicable to the improved actions.

The quark field action is more involved than the gauge field action. Let us start with a naive discretization:

$$D_\mu\psi(x) \rightarrow \frac{1}{2a}\left[\gamma_\mu U_\mu(x)\psi(x+\hat{\mu}) - \gamma_\mu U_\mu^\dagger(x-\hat{\mu})\psi(x-\hat{\mu})\right]. \tag{10}$$

The resultant action has severe problem that there are 16 poles while physical one is only one of them. (*more detail*) These unwanted modes are called ‘doublers’, and they must be eliminated so as to define a theory with correct number of the quark degrees of freedom.

The most simple way to avoid doublers is adding Wilson term,

$$S^W = \frac{ra}{2}\bar{\psi}D^2\psi = \frac{r}{2a}\psi(x)\left(U_\mu(x)\psi(x+\hat{\mu}) + U_\mu^\dagger(x-\hat{\mu})\psi(x-\hat{\mu}) - 2\psi(x)\right), \tag{11}$$

where $r$ is called Wilson parameter. This form is called the Wilson fermion action. The Wilson term kills 15 doublers except for one at $p^2 \sim m^2$, while this term is of $O(a)$ and hence irrelevant in the continuum limit. However, the Wilson term explicitly breaks the chiral symmetry. While the chiral symmetry is restored in the continuum limit, absence of the chiral symmetry make calculations more involved by causing complex operator mixing. Also the extrapolation to the chiral limit must be performed carefully, since the quark mass receives additive renormalization, which does not occur in the presence of the chiral symmetry. In spite of these shortages, the Wilson fermion and its improved versions are extensively studied because of its simpleness.

In this note, we adopt the Wilson fermion action as an example of the quark action. By redefining the quark field normalization as

$$\psi \to \sqrt{2\kappa}\psi, \qquad \kappa = \frac{1}{2(m+4r)}, \tag{12}$$

the quark action is represented as

$$S_F = \sum_{x,y}\bar{\psi}(x)D(x,y)\psi(y), \tag{13}$$

$$D(x,y) = \delta_{x,y} - \kappa\sum_\mu\left\{(r-\gamma_\mu)U_\mu(x)\delta_{x+\hat{\mu},y} + (r+\gamma_\mu)U_\mu^\dagger(x-\hat{\mu})\delta_{x-\hat{\mu},y}\right\}. \tag{14}$$

The $\kappa$ is called the hopping parameter. A popular choice of the Wilson parameter is $r = 1$, and hereafter we also adopt this value.

### 2.3 Quantization and numerical simulations

The quantization is performed by the path integral formalism. An expectation value of an observable is represented as

$$\langle O\rangle = \frac{1}{Z}\int \mathcal{D}U\mathcal{D}\psi\mathcal{D}\bar{\psi}\, O(\bar{\psi},\psi,U)\exp(-S_G[U] - S_F[\bar{\psi},\psi,U]). \tag{15}$$

Integration of the gauge field $U$ over group space is finite. Since the fermion field is represented as the anticommuting Granssmann field, which is not treated on computers, one first integrated out the fermion field. Then

$$\langle O\rangle \simeq \frac{1}{Z}\int \mathcal{D}U\prod_f \det(D_f[U])O(S_q,U)\exp(-S_G[U]), \tag{16}$$

where $S_q(x,y) = D^{-1}(x,y)$ is a quark propagator. $\det(D)$ is real, because of so-called $gamma_5$ hermiticity, $D^d ag = \gamma_5 D\gamma_5$, which implies $\det(D^\dagger) = \det(\gamma_5 D\gamma_5) = \det(D)$. In addition, it is proven that $\det D > 0$ for $\kappa < 1/8$.

Eq. (16) means that the physical observable is evaluated by integrating over gauge configuration $\{U\}$ with the weight of $\det(Du)e^{-S_G}$. If one can generate the gauge configurations with the probability of $\det(Du)e^{-S_G}$, one can evaluate physical quantity as

$$\langle O\rangle = \frac{1}{N_{sample}}\sum_i O(U_i) \tag{17}$$

(the important sampling). How the gauge configurations can be generated is discussed in the following in detail.

**Pseudofermion.** Direct computation of the fermionic determinant, $\det(D[U])$, is practically impossible. The standard trick employs pseudofermion field. For the case of two flavors of degenerate quark mass, $\prod_f \det(D_f) = \det(D^\dagger D)$ and $D^\dagger D$ is real and positive. Then making use of the property of Gaussian distribution,

$$\det D^\dagger D = \int \phi^\dagger \phi \exp(-\phi^\dagger (D^\dagger D)^{-1} \phi). \tag{18}$$

which represents the fermionic determinant as an effective action with bosonic (pseudofermion) field. Appearance of $(D^\dagger D)^{-1}$ requires inverting the fermion operator $D$, which makes dynamical simulations very time consuming.

Compared to the cases with even number of degenerate flavors, simulation with odd number of flavor is more involved. There are several procedures to express the odd $N_f$ fermionic determinant as integrations over bosonic fields with effective actions. These procedures are not described in this note.

### 2.4 Comments on fermion formulations

In Subsection 2.2, we discussed that the Wilson fermion action explicitly breaks the chiral symmetry. For the chiral symmetry on the lattice, there is celebrated Nielsen-Ninomiya's theorem:

*Nielsen-Ninomiya's theorem* [6]
Suppose a fermion action $S_F = \bar{\psi} D[U] \psi$ satisfies the following conditions:
(a) Translational invariance
(b) Chiral symmetry: $D\gamma_5 + \gamma_5 D = 0$
(c) Hermiticity
(d) Bilinear in fermion field
(e) Locality
Then, doublers exist.

Thus to avoid doublers, at least one of the above conditions (a)–(e) must be violated. The Wilson fermion (and its improved variants) breaks the chiral symmetry explicitly.

Another popular fermion formulation is the staggered fermion. In the staggered fermion formulation, the spinor degrees of freedom is scattered on the $2^d$ hypercubes so that the field on each original site has only one spinor component. This decreases the number of doublers to 4, and these 4 doublers are regarded as flavors (recently called 'tastes'). The staggered fermion holds remnant of the chiral symmetry, and hence applied to studies in which the chiral symmetry plays essential role. Without additive mass renormalization, small quark mass region is relatively easily explored, and smaller number of degrees of freedom per site makes simulation faster. However, complicated flavor structure makes analysis more involved. Furthermore, so-called 'fourth root trick' introduced to implement dynamical flavors less than 4 is still now under active debate.

Recently, great progress has been achieved in understanding of the chiral symmetry on the lattice [7]. The progress was started by the domain-wall fermion formulation, which originally intended to implement the chiral gauge theory by considering 5D space [8]. While this idea has difficulty in practical simulation, it was applied to a vector gauge theory such as QCD and succeeded to describe fermions keeping the chiral symmetry [9]. A large fermion mass (domain-wall mass) is introduced which makes the left and right modes appear on two edges in the 5th direction. In practice, with finite extent in 5-th dimension causes small overlap between the left and right modes, which generate residual mass in addition to the quark mass put by hand.

The understanding of the chiral symmetry on the lattice was also drastically proceeded. In particular, the exact chiral symmetry on the lattice was realized in the light of the Ginsparg-Wilson relation [10],

$$\gamma_5 D + D\gamma_5 = aRD\gamma_5 D, \tag{19}$$

where $R$ is a local operator commuting with $\gamma_5$. A fermion action satisfying this relation is symmetric under a transformation

$$\delta\psi = \gamma_5(1 - aRZD)\psi, \delta\bar{\psi} = \bar{\psi}(1 - aDR(1 - Z))\gamma_5, \tag{20}$$

where $Z$ is a parameter [11]. This transformation leads to Ward-Takahashi identity which implies that the pions behave as Nambu-Goldstone bosons [12]. Namely, Eq. 20 is realized as the exact chiral symmetry on the lattice, for lattice fermion action holding the relation (19).

Examples of fermion actions which satisfy the Ginsparg-Wilson relation are the fixed point action [13] and the overlap fermion action [14]. Here we consider the latter formulation. The overlap fermion operator is written as

$$D = \frac{1}{Ra}\left[1 + \gamma_5 \mathrm{sign}(H_W(-M_0))\right] \tag{21}$$

where $H_W(-M_0)$ is the hermitian Wilson-Dirac operator with large negative mass $H_W(-M_0) = \gamma_5 D_W(-M_0)$. This operator is $N_s \to \infty$ limit of the domain-wall fermion, and hence free from residual mass. However, numerical implementation costs high, because of the sign-function of operator $H_W$. Hence numerical simulations have become possible only recently with large computational resources and development of algorithms. Large-scale simulations with dynamical overlap fermions are in progress by JLQCD Collaboration [15].

# 3 Overview of lattice QCD simulations

The lattice QCD simulation is in general performed as follows.

(1) Generating gauge configurations.

(2) Measuring physical observables.

(3) Analyzing measured quantities.

Each step is discussed in some detail below.

## 3.1 Generating gauge configuration

To obtain the expectation values of some physical observables, in any case one first needs the gauge configuration. One of main goals of this note is to introduce an algorithm to generate the gauge configurations. An important point is how to treat the fermionic determinant, $\det(D[U])$.

**Quenched approximation.** In early days (last century), and also nowadays in exploratory studies, quenched simulations have been a convenient approach; it sets $\det(D[U]) = 1$. This makes update of gauge configuration considerably easy. However, the quenched approximation neglects the quark loop effects completely, and hence important part of the dynamics lacks. Surprisingly, even in this approximation, many quantities give rather good results. For example, low-lying light hadron spectrum is in agreement with experimental values within $O(10\%)$ accuracy, with appropriate choice of the lattice scale. Several quantities, such as charmonium hyperfine splitting, severely deviate from the experimental values, and of course it is difficult to compute quantities in which vacuum polarization effect plays essential roles, such as $\eta'$ meson. Also the finite temperature phase transition essentially depends on the number of dynamical flavors.

**Full QCD.** With recent developments of computers and algorithms, dynamical simulations have become popular compared to the last century. The dynamical simulations are usually implemented with HMC algorithms or its variants. The dynamical simulations, which are usually implemented with HMC algorithms, require large computational cost compared to the quenched cases because they contain inversions of quark operator at every step of molecular dynamical evolution. Since the number of iteration rapidly increases as the quark decreases, so is the cost of dynamical simulations. For example, Ukawa estimated the cost of $N_f = 2$ simulations with $O(a)$-improved Wilson quarks as

$$\frac{\text{cost}}{\text{TFlops}\cdot\text{year}} \sim 2.8 \left[\frac{\#\ \text{conf}}{1000}\right] \left[\frac{m_\pi/m_\rho}{0.6}\right]^{-6} \left[\frac{L}{3fm}\right]^5 \left[\frac{a^{-1}}{2GeV}\right]^7 \tag{22}$$

based on their run in the CP-PACS project [16]. This estimate has been considerably changed by a new simulation techniques, such as the Lüscher's domain decomposed HMC algorithm [17, 18]. Thus it is quite important to develop efficient algorithms for dynamical simulations. The computational cost also strongly depends on the fermion formulations.

**Sharing configurations.** Once the gauge configurations are generated, there are numbers of possible measurements. Thus there has been an idea to share the gauge configurations generated with dynamical simulations. MILC Collaboration's 'Gauge Connection' [19] is pioneering activity of this kind. Recently, there has been started international activity to share the lattice QCD data: the International Lattice DataGrid (ILDG) [20]. ILDG is composed of regional grids, and delegates to the ILDG board composed of the representatives of participating countries. ILDG has two working groups. The metadata working group has been investigating how the information of an ensemble and configurations is expressed, and defined a standard format to store the configuration data. They have defined QCDml, which is markup language based on XML. The middleware working group has been investigating how the data are opened public and downloaded with common interface. ILDG formally started at June 2006, and the number of public data as well as members of the community are increasing. Thus one may find and make use of the dynamical configurations from ILDG, if appropriate data are public, without performing expensive and time-consuming dynamical simulations.

## 3.2 Measurement

Once the gauge configurations are in hand, one can measure physical observables on them. Some quantities are composed of only the gauge field, i.e. link variables. Examples are the

Wilson loops, the glueball correlators, and so on. The Wilson loops at large temporal separation gives the static quark potential, which is useful to set the lattice scale $a$.

Lots of quantities contain the quark fields. Hadronic matrix elements are usually calculated using two- or three-point correlation functions. As an example, let us consider a meson two-point correlator,

$$M(t) = \sum_{\vec{x}} \langle O(x) O^{\dagger}(x) \rangle, \tag{23}$$

where

$$O(x) = \bar{q}(x) \Gamma q(x) \tag{24}$$

is an operator having the quantum number of the meson, which is represented by $4 \times 4$ spinor matrix *Gamma*. Foe example, $\Gamma = \gamma_5$ and $\gamma_i$ specify a pseudoscalar meson and a vecotor meson, respectively. By inserting a complete set of states Eq. (23) is expressed as

$$\begin{aligned} M(t) &= \sum_j \frac{|Z_j|^2}{2M_j} \exp(-M_j t) & (25) \\ &\to \frac{|Z_0|^2}{2M_0} \exp(-M_0 t) \qquad (t \gg 1) & (26) \end{aligned}$$

where $M_j$ is the mass of $j$-th excited state, and $Z_j = \langle j | O(x) | 0 \rangle$. Thus from the behavior of $M(t)$, one can extract the mass of the meson. While the decay constant is also determined from the same correlator, it is more involved and not treated as an example in this note.

Eq. (23) is represented using quark propagator $S_q(x, y) = D^{-1}(x, y)$ as

$$\begin{aligned} M(t) &= \sum_{\vec{x}} \mathrm{Tr}\left[ S_q(x, 0) \Gamma S_q(0, x) \Gamma^{\dagger} \right] & (27) \\ &= \sum_{\vec{x}} \mathrm{Tr}\left[ S_q(x, 0) \Gamma \gamma_5 S_q^{\dagger}(x, 0) \gamma_5 \Gamma^{\dagger} \right]. & (28) \end{aligned}$$

We have used in the second line the $gamma_5$ hermiticity, $D^{\dagger} = \gamma_5 D \gamma_5$. Thus one first needs to determine the quark propagator by solving a linear equation,

$$D(x, y) S_q(y, z) = \delta_{x,z}. \tag{29}$$

This inversion is performed using iterative solver algorithm, since $D$ is a huge sparse matrix. Since the inversions often dominate the simulation time, seeking for fast algorithms is of great importance. In Sec. 6, we describe fundamental algorithm of iterative solver, the conjugate gradient (CG) algorithm.

### 3.3 Analysis

After the measurements are performed on each configuration, they are analyzed. First of all, the correlation functions are averaged and their statistical errors are evaluated. The estimation of errors estimation is briefly discussed below. Usually the observed correlation functions do not directly give the desired quantities, but give them as a results analyses such as fits. For example, the meson masses are extracted from the behavior of meson correlator (23) at large $t$, by fitting the data to exponential form. The fitting method is easily found in literatures. For the hadronic matrix elements, such as decay constants and form factors, one also need to renormalize them. The renormalization is done either perturbatively or nonperturbatively.

**Statistical errors** With $O(U_i)$ being computed, one can compute the average value given in Eq. (17). With finite statistics, the expectation value is determined with some precision, and one needs to also evaluate the statistical error. For independent data $O_i$ $(i = 1, ..., M)$, average and its error is estimated as

$$\langle O \rangle = \frac{1}{M}\sum_{i}^{M} O_k \tag{30}$$

$$\delta\langle O \rangle = \sqrt{\frac{\langle (O - \langle O \rangle)^2 \rangle}{M-1}} = \sqrt{\frac{\langle O^2 \rangle - \langle O \rangle^2}{M-1}} \tag{31}$$

This estimate is based on the central limiting theorem, which claims that with sufficiently large number of data, the distribution of average values obeys Gaussian distribution. The error in Eq. (31) represent the width of this distribution.

Most quantities are not directly extracted from the measured data, but given by results of fit analyses. For such a quantity, a complicated function $f(O)$ of directly measured quantities $O$'s, the propagation low of errors often give incorrect estimate. In addition, the fit itself requires the error of the correlator, and hence estimating the error of the result of fit is nontrivial.

To estimate the statistical error of such quantities, the jackknife method provides a convenient solution. The jackknife method is summarized in Appendix A.

**Systematic errors** In order to compare the results of simulations with experimental values, one also need to estimate the systematic errors. Since the evaluation of systematic errors is quite involved, here we just list up the typical source of systematic errors to be considered.

- *Finite lattice spacing.* Numerical simulations are performed inevitably at finite lattice spacing $a$, while the QCD result is obtained in the $a \to 0$ limit. Thus extrapolation to the continuum limit is necessary for quantitative calculations. Since the Wilson quark action has $O(a)$ error, the extrapolation is done linearly in $a$. If one employs $O(a)$ improved action, the extrapolation is done in $O(a^2)$, and the uncertainty of the extrapolation is reduced. This shows practical significance of improved actions. If one does not perform the continuum extrapolation, the systematic errors are estimated with an order counting.

- *Finite lattice size.* Numerical simulation is done also inevitably at finite lattice size. The lattice size should be taken so as to be enough larger than typical hadronic size, say ∼1fm. For relatively extended objects, lattice sizes should be chosen carefully, for example for baryons and exotic hadrons. For studies of finite temperature phase transition, in particular for the second order transition, the finite size effect should be carefully treated.

- *Chiral extrapolations.* The dynamical simulations with physical $u$ and $d$ quark mass is quite difficult. Usually simulations are done at rather large quark mass region, around $m_s/2 \sim m_s$, and the result is extrapolated to the chiral limit. A guide of the extrapolation is provided by the chiral perturbation theory (ChPT). However, fit of the result according to the prediction of ChPT is highly nontrivial, and usually produces large systematic errors.

- *Number of dynamical flavors.* If one chooses quenched or $N_f = 2$ simulations, the result of course suffer from the errors from the number of dynamical flavors. Without extrapolation or interpolation, estimate of this systematic uncertainty is difficult to control. In particular, the finite temperature phase transition is a most severe case, since even the order of transition changes with the number of active dynamical flavors.

- *Renormalization constants.* The matrix elements are related to the continuum quantities with renormalization. If the renormalization is performed perturbatively, say at one-loop level, $\alpha^2$ error exists. When nonperturbative renormalization is applied, the renormalization constant suffer from the systematic errors as same as of other observed quantities.

# 4 Foundation of Monte Carlo simulation

Monte Carlo simulation in general means a class of numerical algorithms which use random numbers. (Remember numerical integration of analytically nonintegrable functions.) However, for statistical mechanical system, one cannot give a required distribution of states from the beginning. One can determine only the values of Hamiltonian for given states. Therefore, we need an algorithm to obtain an ensemble in which each configuration appears with the Boltzmann weight, by starting from arbitrary initial configuration. In this section, we establish the principles to construct such an algorithm.

## 4.1 Important sampling

Let us consider a general system with degrees of freedom $\phi$ (collective notation) and its Hamiltonian $H$. For a canonical ensemble, an expectation value of physical quantity $A$ is represented as

$$\langle A \rangle = \frac{1}{Z} \sum_{\text{states}} A[\phi] e^{-\beta H[\phi]}, \tag{32}$$

where the sum is taken over all possible states. As the number of the degrees of freedom increases, number of possible states rapidly increases, and taking the sum over all the states becomes unrealistic.

On the other hand, in the above equation, the Boltzmann weight $e^{-\beta H[\phi]}$ can be regarded as a probability measure. If the state configuration $\{\phi_i\}$ can be generated with this Boltzmann weight, one can obtain a statistical mechanical average $\langle A \rangle$ by averaging $A[\phi]$ over these configurations:

$$\langle A \rangle \simeq \frac{1}{N_{sample}} \sum_i A[\phi_i], \tag{33}$$

where $\phi_i$ is $i$-th field configuration generated with probability $e^{-\beta H[\phi]}$. In this method, configurations largely contributing to $\langle A \rangle$ are devotedly generated, so the efficient computation can be achieved. This kind of algorithm is called an important sampling.

## 4.2 Markov process

To perform the important sampling, one needs to generate field configurations appearing with Boltzmann weight, we need an algorithm such that we can start with arbitrary initial state, and then evolve the state so as to appear with probability of Boltzmann weight. Among this kind of process, so-called Markov process has particular importance. In this process, next generated state depends only on the present state. Namely, one do not need to store the history of previous states. If one can generate configurations as a Markov process, and if these configurations reach thermal equilibrium (appearing with Boltzmann weight), one can calculate statistical quantities by important sampling.

To construct such a process, the algorithm to change the states must satisfy two conditions, 'ergodicity' and 'detailed balance', as explained in the following.

### 4.3 Master equation and ergodicity

Let us consider a system composed of $N$ finite states. For example, $n$ Ising spin system is composed of $N = 2^n$ distinct states. To express the evolution of the system, we introduce a 'time' argument $t$, which not necessarily coincides with the physical time, but is just introduced to describe the simulation process. Expressing the probability of the system being a state $i$ at time $t$ as $P_i(t)$, the conservation of the probability requires

$$\sum_i P_i(t) = 1. \tag{34}$$

For a Markov process, the probability of the system being the state $i$ at the time $t + \Delta t$ (after 1 time step) is described by so-called Master equation:

$$P_i(t + \Delta t) = P_i(t) - \sum_{j \neq i} P_i(t) w_{i \to j} \Delta t + \sum_{j \neq i} P_j(t) w_{j \to i} \Delta t. \tag{35}$$

$w_{i \to j}$ expresses the transition probability in unit time from a state $i$ to a state $j$. The second term in RHS of Eq. (35) stands for the process from the state $i$ to one of other $(N - 1)$ states, and the third term stands for the process one of other $(N - 1)$ states to the state $i$.

In matrix representation, Eq. (35) is represented as

$$\vec{P}(t + \Delta t) = L \vec{P}(t) \tag{36}$$

where $\vec{P}(t)$ is $N$-dimensional vector having $P_i(t)$ as components, and $L$ is $N \times N$ matrix having components

$$\begin{aligned} L_{ij} &= w_{j \to i} \Delta t \quad (i \neq j) & (37) \\ L_{ii} &= 1 - \sum_{j \neq i} w_{i \to j} \Delta t. & (38) \end{aligned}$$

From the conservation of probability,

$$\sum_i L_{ij} = 1. \tag{39}$$

By definition of the transition probability,

$$L_{ij} \geq 0 \tag{40}$$

must hold. A matrix $L$ satisfying these two condition is called probability matrix. After $s$ steps, the probability distribution of the system is given by

$$\vec{P}(t + s\Delta t) = L^s \vec{P}(t). \tag{41}$$

Now let us consider the limit of $s \to \infty$. If the probability distribution approaches to certain single distribution $P^{(eq)}$, that Markov process is called stationary Markov process.

$$\lim_{s \to \infty} L^s \vec{P}(t) = \vec{P}^{(eq)} \tag{42}$$

Then

$$L\vec{P}^{(eq)} = \vec{P}^{(eq)} \tag{43}$$

holds.

The eigenvalues of $L$ have the following properties. For eigenvalues of $L$, $\lambda_i$ ($i = 1, ..., N$), $|\lambda_i| \leq 1$ holds. In addition, $\lambda_i = 1$ is always an eigenvalue of probability matrix $L$. With this property and Eq. (42), $\vec{P}^{(eq)}$ is (right) eigenvector of $L$ with eigenvalue 1. If the state corresponding to this eigenvalue 1 is nondegenerate, the stationary distribution $\vec{P}^{(eq)}$ uniquely exists and stationary Markov process is obtained.

Let us consider the case that for a step number $s$ larger than certain value, if all the components of $L^s$ have positive definite values. Then from the Perron-Frobenius theorem, the largest eigenvalue of $L$ is nondegenerate and for other eigenvalues $|\lambda| < 1$. This condition means that a transition from an arbitrary state to an arbitrary state is possible. This property is called 'ergodicity'.

### 4.4 Detailed balance condition

For stationary state, Eq. (43) holds. By expressing with components,

$$L_{ij}P_j^{(eq)} = P_i^{(eq)}. \tag{44}$$

Substituting Eqs. (37) and (38) into this equation, one obtains the detailed balance condition,

$$P_i^{(eq)} w_{i\to j} = P_j^{(eq)} w_{j\to i}, \tag{45}$$

where $P_i^{(eq)}$ is probability of the state $i$ in the equilibrium state, and given by Boltzmann weight (normalization is irrelevant)

$$P^{(eq)}(i) = \exp[-\beta H(i)]. \tag{46}$$

Thus constructing an algorithms for which the transition probability $w(i \to j)$ satisfies the relation (45), one can generate an ensemble with Boltzmann weight as a stationary Markov process. In the following, two such algorithms are introduced.

**Metropolis algorithm**

The Metropolis algorithm is a general procedure to generate an algorithm satisfying the detailed balance condition. The Metropolis algorithm is less efficient comparing the next heat-bath algorithm, but can be applied in the cases where the latter algorithm is not possible to construct.

In the Metropolis algorithm, for a configuration $\{\phi\}$ to be updated, one first consider a candidate configuration after update $\{\phi'\}$. This candidate configuration is generated with uniform probability. Computing the values of Hamiltonian for $\{\phi\}$ and $\{\phi'\}$, one accept this candidate configuration $\{\phi'\}$ with the probability

$$P = \min\left\{1, e^{-\beta(H[\phi'] - H[\phi])}\right\} \tag{47}$$

as the new configuration. If $\{\phi'\}$ is rejected, the new configuration is set to the same as the old one $\{\phi\}$.

**Heat-bath algorithm**

In some cases, such as when the Hamiltonian is expressed as a sum of local terms, the heat-bath method may be possible to be constructed. Let us consider a case that one wants to update the degree of freedom $\phi_i$. The Hamiltonian is decomposed into the the Hamiltonian includes $\phi_i$, expressed $H(\phi_i;\phi_{j\neq i})$, and the rest which plays no role in updating $\phi_i$. The local Hamiltonian $H(\phi_i;\phi_{j\neq i})$ includes the interaction with this degree of freedom. Then the degrees of freedom other than $\phi_i$ are treated as the heat-bath, and if $\phi_i$ is updated with probability $\exp(-\beta H[\phi_i;\phi_{j\neq i}])$, this method is called the heat-bath algorithm. By iterating such a local update (sweep), one can update whole the system.

# 5 Hybrid Monte Carlo algorithm for quenched simulation

In this section, we construct the hybrid Monte Carlo (HMC) algorithm to update the gauge field in the quenched approximation. To update quenched gauge configuration, one can employ (pseudo-) heat-bath algorithm. However, since HMC is a standard algorithm for the dynamical simulations, here we introduce HMC for quenched case, while the efficiency is less than the heat-bath algorithm.

The HMC algorithm is developed by combining two algorithms proposed for dynamical simulations: the Langevin algorithm and the microcanonical algorithm. In addition, a Metropolis test performed at the end of the evolution of the fields removes finite step size errors.

## 5.1 Langevin Algorithm

Langevin algorithm was originally proposed by Parisi and Wu (1981), and applied to full QCD simulation by Fukugita and Ukawa, Botrouni et al. In the following, first we consider a bosonic system with finite degree of freedom.

Let us consider a system with coordinate: $q_i (i = 1, .., N)$ described by action S[q]. An update algorithm is needed which satisfies the ergodicity and the detailed balance. We introduce a new 'time' variable (simulation time) $\tau$, which labels elements of a Markov chain. Let us consider the following differential equation which relates $\tau_{n+1} = (n+1)\epsilon_L$ and $\tau_{n+1} = n\epsilon_L$ ($\epsilon_L$ : Langevin time step):

$$q_i(\tau_{n+1}) = q_i(\tau_n) + \epsilon_L \left( -\frac{\partial S[q]}{\partial q_i} + \eta_i(\tau_n) \right). \tag{48}$$

where $\{\eta_i(\tau_n)\}$ is Gaussian distributed random variables independent of each other,

$$P\left(\{\eta_i(\tau_n)\}\right) = \prod_i \sqrt{\frac{\epsilon_L}{4\pi}} \exp\left[ -\frac{\epsilon_L}{4} \eta_i(\tau_n)^2 \right]. \tag{49}$$

In the limit of $\epsilon_L \to 0$, Eq. (48) approaches Langevin equation

$$\frac{dq_i}{d\tau} = -\frac{\partial S[q]}{\partial q_i} + \eta_i(\tau). \tag{50}$$

Setting $\tilde{\eta}_i = \sqrt{\frac{\epsilon_L}{2}}\eta_i$ Eqs. (48) and (49) is written as

$$\begin{aligned} q_i(\tau_{n+1}) &= q_i(\tau_n) - (2\epsilon_L)\frac{1}{2}\frac{\partial S[q]}{\partial q_i} + \sqrt{2\epsilon_L}\tilde{\eta}_i(\tau_n), && (51)\\ \tilde{P}\left(\{\tilde{\eta}_i\}\right) &= \prod_i \frac{1}{\sqrt{2\pi}} \exp\left[ -\frac{1}{2}\tilde{\eta}_i{}^2 \right]. && (52) \end{aligned}$$

This form is convenient for later use in the molecular dynamics.

It is not difficult to show that this algorithm satisfies the detailed balance in the limit of $\epsilon_L \to 0$. In practice, however, $\epsilon_L$ is finite and data should be extrapolated to $\epsilon_L \to 0$ limit. This is time consuming. On the other hand, this algorithm updates all the degrees of freedom simultaneously, and hence suited to update a system described by a non-local action.

### 5.2 Molecular Dynamics

The fundamental idea is based on the fact that the Euclidean path integral formulation and 4-spatial dimension classical mechanical system have the same form. The Hamiltonian is considered to govern an new 'time' variable (simulation time). Assuming the ergodicity to the classical mechanical system and remembering that in the thermodynamic limit a canonical ensemble becomes a micro-canonical ensemble, the expectation values of an observable is calculated as a time average along a 'classical' trajectory corresponding to an energy which is determined with the parameters of the system.

In the molecular dynamics (MD), configurations are generated in deterministic manner, and quantum fluctuation is represented by complexity of the motion in 4-spacial dimension. In the following, let us consider a system of scalar field $\phi$ described by the action $S[\phi;\beta]$ and a parameter $\beta$.

An expectation value of an observable is represented as

$$\begin{aligned} \langle O \rangle &= \frac{1}{Z}\int \mathcal{D}\phi O[\phi] e^{-S[\phi;\beta]} && (53) \\ Z &= \int \mathcal{D}\phi e^{-S[\phi;\beta]} && (54) \end{aligned}$$

To define Eq. (53), we introduce a lattice and label the degree of freedom with '$i$'. $Z$ is still not a partition function of the classical Hamiltonian system. Canonical conjugate momenta $\pi_i$ is introduced as

$$\begin{aligned} \langle O \rangle &= \frac{1}{Z}\int \mathcal{D}\phi\mathcal{D}\pi O[\phi] e^{-H[\phi,\pi;\beta]} && (55) \\ H[\phi,\pi;\beta] &= \sum_i \frac{1}{2}{\pi_i}^2 + S[\phi;\beta] && (56) \\ \bar{Z} &= \int \mathcal{D}\phi\mathcal{D}\pi e^{-S[\phi,\pi;\beta]}. && (57) \end{aligned}$$

Measure is defined as usual,

$$\mathcal{D}\phi\mathcal{D}\pi = \prod_i d\phi_i d\pi_i. \qquad (58)$$

Since $O[\phi]$ does not depend on $\pi$, Eqs. (55) and (54) are equivalent. With the above rewriting, the 4-dim Euclidean quantum system was represented as a classical canonical ensemble.

In the thermodynamic limit, the canonical ensemble becomes a micro-canonical ensemble on a 'energy' surface in 4-spatial dimension.

Note that while as a classical micro-canonical ensemble each degree of freedom evolves deterministically, it is fluctuating as an original quantum system. Quantities in a canonical ensemble and a micro-canonical ensemble are labeled with subscript 'can' and 'mic', respectively. An expectation value of an observable is expressed as

$$\langle O \rangle_{can}(\beta) = \frac{\int \mathcal{D}\phi\mathcal{D}\pi O[\phi] \int dE \delta(H[\phi,\pi;\beta]-E)e^{-E}}{\int \mathcal{D}\phi\mathcal{D}\pi \int dE \delta(H[\phi,\pi;\beta]-E)e^{-E}}. \qquad (59)$$

$$
\begin{aligned}
\langle O\rangle_{mic}(E,\beta) &= \frac{1}{Z_{mic}(E;\beta)}\int \mathcal{D}\phi\mathcal{D}\pi O[\phi]\delta(H[\phi,\pi;\beta]-E), &(60)\\
Z_{mic}(E;\beta) &= \int \mathcal{D}\phi\mathcal{D}\pi\delta(H[\phi,\pi;\beta]-E). &(61)
\end{aligned}
$$

where $Z_{mic}(E;\beta)$ is the state density at energy $E$. Defining the 'entropy' of the system as

$$s(E;\beta) = lnZ_{mic}(E;\beta), \tag{62}$$

the expectation value is represented as

$$\langle O\rangle_{can}(\beta) = \frac{\int dE\langle O\rangle_{mic}(E,\beta)\exp\{E-s(E;\beta)\}}{\int dE\exp\{E-s(E;\beta)\}}. \tag{63}$$

When the degree of freedom of the system is sufficiently large, the above exponential factor has sharp peak at $E=\bar{E}$. $\bar{E}$ is given implicitly by

$$\left(\frac{\partial s(E,\beta)}{\partial E}\right)_{E=\bar{E}} = 1. \tag{64}$$

In the thermodynamic limit,

$$\langle O\rangle_{can}(\beta) = [\langle O\rangle_{mic}]_{E=\bar{E}} \tag{65}$$

In the microcanonical ensemble, each degree of freedom and its canonical conjugate momentum evolve in the phase space according to the Hamiltonian equation of motion.

$$
\begin{aligned}
\dot{\phi}_i &= \frac{\partial H[\phi,\pi]}{\partial\pi_i}, &(66)\\
\dot{\pi}_i &= -\frac{\partial H[\phi,\pi]}{\partial\phi_i}. &(67)
\end{aligned}
$$

Assuming ergodicity, ensemble average can be replaced with the time average.

$$[\langle O\rangle_{mic}]_{E=\bar{E}} = \lim_{T\to\infty}\frac{1}{T}\int_0^T d\tau O[\phi(\tau)]. \tag{68}$$

The trajectory in the phase space is governed by

$$\frac{d^2\phi_i}{d\tau^2} = -\frac{\partial S[\phi]}{\partial\phi_i}. \tag{69}$$

Naively discretizing this equation in the 'time' direction,

$$\ddot{\phi}(\tau_n) = [\phi(\tau_{n+1})+\phi(\tau_{n-1})-2\phi(\tau_n)]/(\Delta\tau)^2 \tag{70}$$

leads to

$$
\begin{aligned}
\phi_i(\tau_{n+1}) &= \phi_i(\tau_n) - \epsilon^2\frac{1}{2}\frac{\partial S[\phi]}{\partial\phi_i(\tau_n)} + \epsilon\pi_i(\tau_n), &(71)\\
\pi_i(\tau_n) &= \frac{1}{2\epsilon}\left(\phi_i(\tau_{n+1})+\phi_i(\tau_{n-1})\right), &(72)
\end{aligned}
$$

where $\epsilon = \Delta\tau = \tau_{n+1}-\tau_n$ is micro-canonical time step. Note that the similarity between this equation and Langevin equation (51). These equations takes the same form by setting $\pi_i$ random variables and $\epsilon^2/2 = \epsilon_L$.

In practical simulation, Hamilton equation is solved with the Runge-Kutta algorithm. Comparing the traveling distance of configurations, ($\epsilon^2/2 = \epsilon_L$) the micro-canonical case is $O(n\epsilon)$ while the Langevin case $O(\sqrt{n}\epsilon)$. The latter is because of the stochastic nature. The micro-canonical travels further, while it also has uncertainties that thermodynamic limit is nontrivial for small lattices, and ergodicity is not incorporated in the algorithm. Combining advantages of these two algorithms, one reaches the hybrid algorithm mentioned next.

## 5.3 Hybrid Algorithm

Hybrid algorithm is an algorithm which has advantages of the Langevin and molecular dynamics algorithms. In equation of MD, setting the time step as $\epsilon_L = \epsilon^2/2$ and taking the momenta $\{\pi_i\}$ to be random variable with probability density

$$P\left(\{\tilde{\eta}_i\}\right) = \left(\prod_i \frac{1}{\sqrt{2\pi}}\right) \exp\left[-\sum_i \frac{1}{2}{\pi_i}^2\right], \tag{73}$$

it takes the same form as the Langevin equation. In fact if the 'time' evolution of the Hamiltonian (57) is ergodic, the momenta distribute with the above probability. Then the hybrid algorithm is obtained as: configurations are generate with MD with probability $\alpha$, or Langevin with $(1-\alpha)$.

In the following, let us consider in more detail in the case of scalar field. Equation of motion is Hamiltonian form, (66), (67). Discretizing with a time step $\epsilon$, Taylor expand $\phi_i(\tau+\epsilon)$ and $\pi_i(\tau+\epsilon)$ to the order of $\epsilon^2$.

$$\begin{array}{rcl} \phi_i(\tau+\epsilon) & = & \phi_i(\tau) + \epsilon\dot{\phi}_i(\tau) + \frac{\epsilon^2}{2}\ddot{\phi}_i(\tau) + O(\epsilon^3) \\ \pi_i(\tau+\epsilon) & = & \pi_i(\tau) + \epsilon\dot{\pi}_i(\tau) + \frac{\epsilon^2}{2}\ddot{\pi}_i(\tau) + O(\epsilon^3) \end{array} \tag{74}$$

Eq. of motion leads to $\dot{\phi}_i(\tau) = \pi_i(\tau)$, $\ddot{\phi}_i(\tau) = \dot{\pi}_i(\tau) = -\frac{\partial S}{\partial\phi_i(\tau)}$,

$$\ddot{\pi}_i(\tau) = -\sum_j \frac{\partial^2 S}{\partial\phi_i(\tau)\partial\phi_j(\tau)}\pi_j(\tau). \tag{75}$$

The RHS is represented in the leading order of $\epsilon$ as

$$\sum_j \frac{\partial^2 S}{\partial\phi_i(\tau)\partial\phi_j(\tau)}\pi_j(\tau) = \frac{1}{\epsilon}\left(\frac{\partial S}{\partial\phi_i(\tau+\epsilon)} - \frac{\partial S}{\partial\phi_i(\tau)}\right) + O(\epsilon^2). \tag{76}$$

Substituting this form into (74),

$$\begin{array}{rcl} \phi_i(\tau+\epsilon) & = & \phi_i(\tau) + \epsilon\left(\pi_i(\tau) - \frac{\epsilon}{2}\frac{\partial S}{\partial\phi_i(\tau)}\right) + O(\epsilon^3) \\ \left(\pi_i(\tau+\epsilon) - \frac{\epsilon}{2}\frac{\partial S}{\partial\phi_i(\tau+\epsilon)}\right) & = & \left(\pi_i(\tau) - \frac{\epsilon}{2}\frac{\partial S}{\partial\phi_i(\tau)}\right) - \epsilon\frac{\partial S}{\partial\phi_i(\tau+\epsilon)} + O(\epsilon^3) \end{array} \tag{77}$$

In the parenthesis, the momentum is at $+\epsilon/2$ step forward. Therefore up to $O(\epsilon^3)$, Eq. (77) is equivalent to

$$\begin{array}{rcl} \phi_i(\tau+\epsilon) & = & \phi_i(\tau) + \epsilon\pi_i(\tau+\frac{1}{2}\epsilon) \\ \pi_i(\tau+\frac{3}{2}\epsilon) & = & \pi_i(\tau+\frac{1}{2}\epsilon) - \epsilon\frac{\partial S}{\partial\phi_i(\tau+\epsilon)} \end{array} \tag{78}$$

This equation is solved iteratively.

In the hybrid algorithm, momentum variables is refreshed at the beginning of M.D. chain. Starting the iteration at time $\tau$, $\{\pi_i(\tau)\}$ is chosen randomly as a Gaussian ensemble. First $\pi_i(\tau+\epsilon/2)$ is determined by being evolved half-time step.

$$\pi_i(\tau+\frac{1}{2}\epsilon) = \pi_i(\tau) - \frac{\epsilon}{2}\frac{\partial S}{\partial\phi_i(\tau)} + O(\epsilon^2) \tag{79}$$

This generated $O(\epsilon^2)$ error. However, this occurs only at the beginning of M.D. step. The accumulated error after the evolution of the length $1/\epsilon$ is also $O(\epsilon^2)$, so the observable has $O(\epsilon^2)$ error.

Hybrid Monte Carlo algorithm is constructed by adding a Metropolis test at the end of molecular dynamics of the hybrid algorithm. By this Metropolis test, the detailed balance condition is hold for arbitrary finite time steps (for proof, see e.g. Rhote (1992)). Therefore systematic errors due to the finite time step disappear and the extrapolation in $\epsilon$ is no longer needed.

Hybrid Monte Carlo algorithm is summarized as follows.

(1) Choose the initial configuration $\{\phi_i\}$ in some way.
(2) $\{\pi_i\}$ is randomly chosen as an Gaussian ensemble
(3) $\pi_i$ is evolved by half step.
(4) $\phi_i and \pi_i$ are evolved by certain steps. (The last step of $\pi_i$ is with half the step size.)
(5) New configuration $\{\phi'_i, \pi'_i\}$ accepted with probability

$$P = \min\left\{ 1, \ \frac{e^{-H[\phi'_i,\pi'_i]}}{e^{-H[\phi_i,\pi_i]}} \right\}. \tag{80}$$

(6) Return to (2).

Note that when the candidate configuration is rejected at (5), the old configuration is adopted as the product of this trajectory.

In the hybrid Monte Carlo algorithm, whole the configuration is updated at the same time (in contrary to the local update algorithms such as heat-bath algorithms).

The value of Hamiltonian is in principle kept constant along the trajectory in the phase space. In practice, due to the finite step size, there is an $O(\epsilon^2)$ difference in the values of Hamiltonian on the old and new (candidate) configurations. The Metropolis acceptance test cures this breaking of energy conservation so as to ensures the detailed balance. Thus for not small enough $\epsilon$, acceptance rate would be small. For efficient numerical simulations, appropriate value of $\epsilon$ and length of molecular dynamics evolution should be chosen.

In the molecular dynamics evolution is not performed and momentum is refreshed on every time, $\{\phi_i\}$ is updated with

$$\phi_i(n+1) = \phi_i(n) - \frac{\epsilon^2}{2}\frac{\partial S}{\partial\phi_i(n)} + \epsilon\eta_i(n), \tag{81}$$

which is equivalent to the Langevin step with the $\epsilon_L = \epsilon^2/2$.

### 5.4 SU(3) case

In this section, let us consider a HMC algorithm for SU(3) gauge field in the quenched approximation. The action is given as

$$S_G = \beta\sum_n\sum_{\mu>\nu}(1-\frac{1}{3}\text{Re Tr}[U_\mu(n)U_\nu(n+\hat{\mu})U^\dagger_\mu(n+\hat{\nu})U^\dagger_\nu(n)]). \tag{82}$$

The first term gives a constant contribution and is neglected in the simulation. Thus hereafter we adopt the action

$$\begin{aligned} S_G &= -\frac{\beta}{3}\sum_n \sum_{\mu>\nu} \text{Re Tr}\left[U_\mu(n)U_\nu(n+\hat{\mu})U_\mu^\dagger(n+\hat{\nu})U_\nu^\dagger(n)\right] \\ &= -\frac{\beta}{6}\sum_n \sum_{\mu>\nu} \text{Tr}\left[U_{\mu\nu}(n)+U_{\mu\nu}^\dagger(n)\right], \end{aligned} \tag{83}$$

where $U_{\mu\nu}(n) = U_\mu(n)U_\nu(n+\hat{\mu})U_\mu^\dagger(n+\hat{\nu})U_\nu^\dagger(n)$.

The link variable is represented as $U_\mu(n) = \exp[iA_\mu^a(n)t^a]$. $t^a$ is SU(3) generator satisfying $Tr[t^a t^b] = \delta_{ab}$. Defining as

$$\frac{dU_\mu(n)}{d\tau} = i\dot{A}_\mu^a(n)\, t^a U_\mu(n) \equiv iH_\mu(n)\, U_\mu(n), \tag{84}$$

$$H_\mu(n) = h_\mu^a(n)\, t^a \qquad (\ h_\mu^a(n)t^a \in \text{R}\ ), \tag{85}$$

$H_\mu(n)$ is a hermitian and traceless matrix. Since $\dot{A}_\mu^a(n) = h_\mu^a(n)$, $h_\mu^a(n)$ is a conjugate field to $A_\mu^a(n)$. Kinetic part is

$$\sum_{n,\mu,a} \frac{1}{2}h_\mu^a(n) = \frac{1}{2}\sum_{n,\mu}\text{Tr}[HH^\dagger]. \tag{86}$$

Therefore Hamiltonian is represented as

$$\begin{aligned} H[H,U] &= \frac{1}{2}\sum_{n,\mu}\text{Tr}[HH^\dagger] + S_G \\ &= \frac{1}{2}\sum_{n,\mu}\text{Tr}[HH^\dagger] - \frac{\beta}{6}\sum_n\sum_{\mu>\nu}\text{Tr}\left[U_{\mu\nu}(n)+U_{\mu\nu}^\dagger(n)\right]. \end{aligned} \tag{87}$$

We need an equation of motion for $H_\mu(n)$. To obtain it, the above Hamiltonian is differentiated with respect to 'time'.

$$\dot{H} = \text{Tr}\left[\sum \dot{H}\, H - \sum \frac{\beta}{6}\left(\dot{U}_{\mu\nu}(n) + \dot{U}_{\mu\nu}^\dagger(n)\right)\right] \tag{88}$$

$$\begin{aligned} \sum_{n,\mu>\nu}\left(\dot{U}_{\mu\nu}(n)+\dot{U}_{\mu\nu}^\dagger(n)\right) &= \sum_{n,\mu}\left[\dot{U}_\mu(n)V_\mu^\dagger(n) + V_\mu(n)\dot{U}_\mu^\dagger(n)\right] \quad (89) \\ &= \sum_{n,\mu}\left[iH_\mu(n)U_\mu(n)V_\mu^\dagger(n) - iV_\mu(n)U_\mu^\dagger(n)H_\mu(n)\right]. \quad (90) \end{aligned}$$

Thus we have

$$\dot{H}_\mu(n) = -i\frac{\beta}{6}\left[U_\mu(n)V_\mu^\dagger(n) - h.c.\right], \tag{91}$$

for which hermiticity of $\dot{H}_\mu(n)$ is satisfied while the traceless is not ensured. Requiring the RHS being traceless, the degrees of freedom on both the sides become same, and the equation of motion for $H_\mu(n)$ becomes

$$i\dot{H}_\mu(n) = -\frac{\beta}{3}\left[\ U_\mu(n)\, V^\dagger(n)\ \right]_{AT}, \tag{92}$$

where subscript 'AT' denotes anti-hermitian and traceless and for arbitrary matrix $A$

$$[A]_{AT} = \frac{1}{2}(A - A^\dagger) - \frac{1}{6}\mathrm{Tr}(A - A^\dagger). \tag{93}$$

Since we have obtained the canonical equation of motion for SU(3) gauge field as

$$\begin{aligned} \dot{U}_\mu(n) &= iH_\mu(n)\, U_\mu(n) && (94) \\ i\dot{H}_\mu(n) &= -\frac{\beta}{3}\left[\, U_\mu(n)\, V^\dagger(n) \,\right]_{AT}, && (95) \end{aligned}$$

we can update the gauge field with the hybrid method. First, as the Langevin step, $h_\mu^a(n)$ is chosen randomly as a Gaussian ensemble with $\langle h^2 \rangle = 1$. Then as the MD step, $U$ and $H$ are updated according to the above equation of motion. Descretizing with a finite 'time' step $\epsilon$,

$$\begin{aligned} U(\tau) &= exp[i\epsilon H(\tau + \frac{1}{2}\epsilon)]U(\tau) && (96) \\ iH(\tau + \frac{3}{2}\epsilon) &= H(\tau + \frac{1}{2}\epsilon) - \epsilon\frac{\beta}{3}\left[\, U_\mu(n)\, V^\dagger(n) \,\right]_{AT}. && (97) \end{aligned}$$

Here we omit site and direction indices for simplicity. For $U$, one needs to approximate the exponential function of matrix by expanding to some degree.[1] To keep the error in MD evolution $O(\epsilon^2)$, one needs an expression at least correct to the order of $\epsilon^2$. If this approximation is not sufficient, the updated $U$ goes out of SU(3), then reunitarization is necessary. (The reunitarization is easily done by e.g. Gram-Schmidt orthogonalization.)

# 6 Solving quark propagator

To calculate the observables containing fermion fields, such as two-point correlation function of hadrons, one needs to determine the quark propagator. The quark propagator is the inverse of fermion operator. One needs to solve the linear equation

$$D(x,y)S_q(y,z) = \delta_{x,z}. \tag{98}$$

Since the Dirac operator $D(x,y)$ has degrees of freedom of color, spin, and site, and thus huge matrix. Thus holding all the component of the matrix $D$ is not realistic. Since $D(x,y)$ is sparce matrix, iterative solver is well applicable. Most typical approach is Conjugate Gradient (CG) algorithm.

## 6.1 CG algorithm

The CG algorithm is applicable to positive definite matrix. Instead of Eq.(98), one solves

$$(D^\dagger D)(x,y)S_q(y,z) = D^\dagger(x,y)\delta_{y,z}. \tag{99}$$

Conjugate Gradient (CG) algorithm is one of most popular algorithm to solve a linear equation,

$$Ax = b, \tag{100}$$

[1] Alternative way to compute this exponential has also been proposed.

where $b$ is the source vector $b \in V = R^N$, and $A$ is $N \times N$ matrix. To obtain $x = A^{-1}b$ is the problem to be solved.

CG algorithm is one of the Krylov subspace algorithm. The $k$-th Krylov subspace is define as

$$\mathcal{K}_k(A, v_0) = \mathrm{span}\{A^k v_0; k = 0, \dots, k-1\} \tag{101}$$

where $v_0$ is certain initial vector, and $A$ is a matrix. The family of the Krylov subspace method approximately solve Eq. (98) in $\mathcal{K}_k(A, v_0)$.

We assume that the vector space is real, and the matrix to be solved is symmetric and positive definite. The linear equation (98) can be solved by minimizing

$$S(x) = \frac{1}{2}||\hat{x} - x||_A^2 \equiv \frac{1}{2}(\hat{x} - x, A(\hat{x} - x)), \tag{102}$$

where $(a, b)$ is an inner product of vectors $a$ and $b$, and $\hat{x} = A^{-1}b$ is the true solution. We iteratively approximate the solution of Eq. (98) as $\{x_i\}$ $(i = 0, 1, \dots)$, and generate new $x_{i+1}$ from $x_i$ as

$$x_{i+1} = x_i + \lambda_i p_i, \tag{103}$$

where $p_i$ is a correction vector defined later. Here let us suppose that $p_i$ is given, and obtain $\lambda_i$ which minimizes Eq. (102).

From the extremum condition of $S(x_{i+1})$ for $\lambda_i$,

$$\lambda_i = \frac{(p_i, b - Ax_i)}{(p_i, Ap_i)} = \frac{(p_i, r_i}{(p_i, Ap_i)} \tag{104}$$

where we introduce the residual vector $r_i = b - Ax_i$. This type of algorithms which determine the solution based on minimization is called gradient method. The essential point for the CG method lies how to choose the correction vector $p_i$. Before constructing $p_i$, however, let us prove a useful relation,

$$(p_k, r_{k+1}) = 0. \tag{105}$$

*Proof:*

$$\begin{aligned} (p_k, r_{k+1}) &= (p_k, b - Ax_{k+1}) \\ &= (p_k, b - Ax_k - \lambda_k Ap_k) \\ &= (p_k, r_k) - \lambda(p_k, Ap_k) \\ &= 0. \end{aligned}$$

In the last line, $\lambda_k = (p_k, r_k)/(p_k, Ap_k)$ was used. (QED)

For linearly independent $p_0, \dots, p_{N-1}$,

$$x = x_0 + \lambda_0 p_0 + \cdots + \lambda_j p_j + \cdots + \lambda_{N-1} p_{N-1} \tag{106}$$

then $x_j = x_0 + \lambda_0 p_0 + \cdots + \lambda_j p_j$. CG method chooses $p_j$ such that $\{p_j\}$ are orthogonal to each other with respect to $A$, *i.e.,*

$$(p_i, Ap_j) = 0 \quad \text{for} \quad i \neq j. \tag{107}$$

First let us prove that if $(p_j, Ap_k) = 0$ for $j \neq k$, $(p_i, r_{k+1}) = 0$ for $i = 0, 1, \dots, k$. This is generalization of Eq. (105) to $i < k$.

*Proof:*

$$\begin{aligned}
x_{k+1} &= x_{i+1} + \sum_{j=i+1}^{k} \lambda_j p_j \\
r_{k+1} &= b - Ax_{k+1} \\
&= b - A(x_{i+1} + \sum_{j=i+1}^{k} \lambda_j p_j) \\
&= r_{i+1} - \sum_{j=i+1}^{k} \lambda_j Ap_j \\
(p_i, r_{k+1}) &= (p_i, r_{i+1}) - \sum_{j=i+1}^{k} \lambda_j (p_i, Ap_j) \quad \text{(second term vanishes by assumption)} \\
&= (p_i, b - Ax_i - \lambda_i Ap_i) \\
&= (p_i, r_i) - \lambda_i (p_i, Ap_i) \\
&= (p_i, r_i) - \frac{(p_i, r_i)}{(p_i, Ap_i)} (p_i, Ap_i) \\
&= 0.
\end{aligned}$$

(QED)

Let us take

$$p_k = r_k + \mu_{k-1} p_{k-1} \tag{108}$$

where from the condition $0 = (p_{k-1}, Ap_k) = (p_{k-1}, Ar_k) + \mu_{i-1}(p_{k-1}, Ap_{k-1})$,

$$\mu_{k-1} = -\frac{(p_{k-1}, Ar_k)}{(p_{k-1}, Ap_{k-1})}. \tag{109}$$

With this construction of $p_k$, the following relations hold.

$$\begin{aligned}
(r_i, r_j) &= 0 \quad \text{for} \quad (i \neq j), \\
(p_i, Ap_j) &= 0 \quad \text{for} \quad (i \neq j).
\end{aligned}$$

*Proof:* By induction. Assume that

$$(r_i, r_j) = 0 \quad \text{for} \quad i \neq j, \ i, j \leq k. \tag{110}$$

then using

$$(p_i, r_{k+1}) = 0 \quad \text{for} \quad i \leq k, \tag{111}$$

$$\begin{aligned}
0 = (p_i, r_{k+1}) &= (r_i + \mu_{i-1} p_{i-1}, r_{k+1}) \\
&= (r_i, r_{k+1}).
\end{aligned}$$

That is

$$(r_i, r_j) = 0 \quad \text{for} \quad i \neq j, \ i, j \leq k+1. \tag{112}$$

and Eq.(110) is proved by induction (QED).

### 6.2 Krylov subspace method

The CG algorithm drastically improved large linear problems. CG algorithm composes the approximate solution $x_i$ in the subspace spanned by $r_0$, $Ar_0$, $A^2r_0$, ..., $A^{i-1}r_0$. After invention of CG method, this kind of algorithms has been extensively investigated, and called Krylov subspace method [32]. The Krylov subspace provides powerful framework for iterative solver. While CG method is applicable to a positive definite matrix, there are other Krylov subspace methods which is applicable to nonhermitian matrices. In particular, GMRES, MR, BiCGStab methods are popular algorithms in lattice QCD simulations.

## 7 Dynamical simulations

In this section, HMC algorithm described in Sec. 5 is extended to the case with two dynamical flavors of degenerate masses.

### 7.1 Pseudofermion

Let us consider a QCD system containing the standard Wilson fermions. Lattice action:

$$S_{eff} = S_G - \sum_f \ln \det(\frac{1}{2\kappa} K_f[U]), \tag{113}$$

where $S_G$ is the gauge field action and the Wilson-Dirac operator is

$$K_{nm}[U] = \delta_{nm} \cdot 1 - \kappa \sum_\mu [(1-\gamma_\mu)U_\mu(n)\delta_{n+\hat{\mu},m} + (1+\gamma_\mu)U_\mu^\dagger(n-\hat{\mu})\delta_{n-\hat{\mu},m}]. \tag{114}$$

(The redefinition of the normalization of the fermion field is not done.) For two degenerated flavors,

$$S_{eff} = S_G - \ln \left( \det \frac{1}{2\kappa} K_f[U] \right)^2 . \tag{115}$$

As an important feature of $K[U]$,

$$\det K[U] = \det K^\dagger[U], \tag{116}$$

and for $\kappa < 1/8$,

$$\det K[U] > 0 \tag{117}$$

holds. This condition is with rigorous proof, and in practical case $\kappa < \kappa_c$ would replace. Then

$$\det K[U] = \sqrt{\det Q[U]}, \tag{118}$$

where

$$Q[U] = K^\dagger[U] K[U] \tag{119}$$

is positive definite and Hermite. Therefore an effective action for $N_f$ degenerate flavors is written as

$$S_{eff} = S_G - \frac{N_f}{2} \ln \det(\frac{1}{4\kappa^2} Q[U]). \tag{120}$$

In the following we consider $N_f = 2$ case:

$$\exp\{-S_{eff}\} = \det(\frac{1}{4\kappa^2}Q[U]) \cdot \exp\{-S_G\}. \tag{121}$$

$\det \frac{1}{4\kappa^2}Q$ is represented as

$$\det(\frac{1}{4\kappa^2}Q[U]) = \frac{1}{\det 4\kappa^2 Q^{-1}} = \int \mathcal{D}\phi^*\mathcal{D}\phi \exp\{-4\kappa^2 \sum_{ij} \phi_i^* Q^{-1}[U]_{ij}\phi_j\} \tag{122}$$

where $i$ and $j$ label site ⊗ color ⊗ spinor space component. Partition function is represented

$$\begin{aligned} Z &= \int \mathcal{D}U \det Q[U] e^{-S_G[U]} \\ &= \int \mathcal{D}U\mathcal{D}\phi^*\mathcal{D}\phi \exp\{-S_G[U] - 4\kappa^2 \sum_{ij} \phi_i^* Q^{-1}[U]_{ij}\phi_j\}. \end{aligned} \tag{123}$$

Changing the pseudo-fermion field (it is after all irrelevant) as $2\kappa\phi \to \phi$,

$$Z = \int \mathcal{D}U\mathcal{D}\phi^*\mathcal{D}\phi \exp\{-S_G - \sum_{ij} \phi_i^* Q^{-1}[U]_{ij}\phi_j\}. \tag{124}$$

Here we define

$$S_{PF} = \sum_{ij} \phi_i^* Q^{-1}[U]_{ij}\phi_j. \tag{125}$$

Update is done with the hybrid Monte Carlo algorithm. For this purpose, for each of $U$, $\phi$, and $\phi^\dagger$ canonical conjugate fields should be introduced so as to define the corresponding Hamiltonian. However, $\phi$, $\phi^\dagger$ can be treated as a background field during molecular dynamics evolution and can be given at the Langevin step by heat-bath algorithm. Namely, $U$ and its conjugate $H$ are evolved under fixed configuration of $\phi$ and $\phi^\dagger$. Since

$$Q^{-1} = (K^\dagger K)^{-1} = K^{-1}K^{\dagger -1}, \tag{126}$$

introducing

$$\xi = K^{\dagger -1}\phi, \tag{127}$$

$$e^{-S_{PF}} = e^{-\xi^*\xi} \tag{128}$$

holds. Thus generating $\xi$ with the distribution probability $e^{-\xi^*\xi}$, $\phi$ is given as $\phi = K^\dagger\xi$ under which $U$ and $H$ are evolved with molecular dynamics.

## 7.2 Hamilton equation

A conjugate matrix field $H$ to $U$ is introduced so as to satisfy $\dot{U} = iHU$. Hamiltonian is (with neglecting the constant term)

$$\begin{aligned} \mathcal{H}[U,H] &= \frac{1}{2}\sum_{n,\mu}\left[H_\mu(n)H_\mu^\dagger(n)\right] + S_G + S_{PF} && (129) \\ &= \frac{1}{2}\sum_{n,\mu}\left[H_\mu(n)H_\mu^\dagger(n)\right] - \frac{\beta}{6}\sum_{n,\mu>\nu}\mathrm{Tr}(U_{\mu\nu} + U_{\mu\nu}^\dagger) + \sum_{i,j}\phi_i^* Q_{ij}^{-1}\phi_j. && (130) \end{aligned}$$

Since $\phi$ is fixed during MD chain as background field,

$$\frac{d}{d\tau}\phi = 0. \tag{131}$$

Thus

$$\frac{d}{d\tau}\mathcal{H}[U,H] = \sum_{n,\mu} Tr[\dot{H}_\mu(n)H_\mu(n) - \frac{\beta}{6}(\dot{U}_\mu(n)V_\mu(n) + h.c.)] + \sum_{i,j}\phi_i^*(\frac{d}{d\tau}Q_{ij}^{-1})\phi_j. \tag{132}$$

One needs to compute $\frac{d}{d\tau}Q_{ij}^{-1}$. Since

$$\frac{d}{d\tau}(QQ^{-1}) = \frac{d}{d\tau}Q\cdot Q^{-1} + Q\cdot\frac{d}{d\tau}Q^{-1} = 0, \tag{133}$$

$$\frac{d}{d\tau}Q^{-1} = -Q^{-1}\frac{d}{d\tau}QQ^{-1}. \tag{134}$$

Defining as

$$\eta_i = Q_{ij}^{-1}\phi_j, \tag{135}$$

and with $Q^\dagger = Q$, one has

$$\sum_{i,j}\phi_i^*(\frac{d}{d\tau}Q_{ij}^{-1})\phi_j = \sum_{i,j}\eta_i^*(\frac{d}{d\tau}Q_{ij})\eta_j. \tag{136}$$

Then

$$\frac{d}{d\tau}Q = \frac{d}{d\tau}(K^\dagger K) = K^\dagger\frac{d}{d\tau}K + \frac{d}{d\tau}K^\dagger\cdot K, \tag{137}$$

$$\eta^\dagger\frac{dQ}{d\tau}\eta = \eta^\dagger K^\dagger\frac{dK}{d\tau}\eta + \eta^\dagger\frac{dK^\dagger}{d\tau}K\eta = \zeta^\dagger\frac{dK}{d\tau}\eta + \eta^\dagger\frac{dK^\dagger}{d\tau}\zeta. \tag{138}$$

where we set as $\zeta = K\eta$ . Since

$$K_{nm} = \delta_{nm} - \kappa\sum_\mu\left[(1-\gamma_\mu)U_\mu(n)\delta_{n+\hat{\mu}},m + (1+\gamma_\mu)U_\mu^\dagger(n-\hat{\mu})\delta_{n-\hat{\mu}},m\right] \tag{139}$$

$$K_{nm}^\dagger = \delta_{nm} - \kappa\sum_\mu\left[(1+\gamma_\mu)U_\mu(n)\delta_{n+\hat{\mu}},m + (1-\gamma_\mu)U_\mu^\dagger(n-\hat{\mu})\delta_{n-\hat{\mu}},m\right], \tag{140}$$

the time derivative of $K$ is represented as

$$\frac{dK}{d\tau} = -\kappa\sum_\mu\left[(1-\gamma_\mu)iH_\mu(n)U_\mu(n)\delta_{n+\hat{\mu}},m - (1+\gamma_\mu)U_\mu^\dagger(n-\hat{\mu})iH_\mu(n-\hat{\mu})\delta_{n-\hat{\mu}},m\right], \tag{141}$$

$$\begin{aligned}
\zeta^\dagger\frac{dK}{d\tau}\eta &= -\kappa\sum_{n,\mu}\left[\zeta_n^\dagger(1-\gamma_\mu)iH_\mu(n)U_\mu(n)\eta_{n+\hat{\mu}} - \zeta_n^\dagger(1+\gamma_\mu)U_\mu^\dagger(n-\hat{\mu})iH_\mu(n-\hat{\mu})\eta_{n-\hat{\mu}}\right] \\
&= -\kappa\sum_{n,\mu}\left[\zeta_{nb}^\dagger(1-\gamma_\mu)iH_\mu(n)_{ba}(U_\mu(n)\eta_{n+\hat{\mu}})_a - (\zeta_{n+\hat{\mu}}^\dagger U_\mu^\dagger(n))_b(1+\gamma_\mu)iH_\mu(n)_{ba}\eta_{n,a}\right] \\
&= -\kappa\sum_{n,\mu}iH_\mu(n)_{ba}\left[\zeta_{nb}^\dagger(1-\gamma_\mu)(U_\mu(n)\eta_{n+\hat{\mu}})_a - (\zeta_{n+\hat{\mu}}^\dagger U_\mu^\dagger(n))_b(1+\gamma_\mu)\eta_{n,a}\right].
\end{aligned} \tag{142}$$

Here $a$, $b$ are the color indices and the spinor components are understood to be contracted. The last line of the above equation has a form of Tr[$H \times$ (matrix)]. Similarly,

$$\frac{dK^\dagger}{d\tau} = -\kappa \sum_\mu \left[(1+\gamma_\mu) iH_\mu(n)U_\mu(n)\delta_{n+\hat{\mu}}, m - (1-\gamma_\mu)U_\mu^\dagger(n-\hat{\mu})iH_\mu(n-\hat{\mu})\delta_{n-\hat{\mu}}, m\right], \quad (143)$$

$$\begin{aligned}
\eta^\dagger \frac{dK}{d\tau}\zeta &= -\kappa \sum_{n,\mu} \left[\eta_n^\dagger (1+\gamma_\mu) iH_\mu(n)U_\mu(n)\zeta_{n+\hat{\mu}} - \eta_n^\dagger (1-\gamma_\mu)U_\mu^\dagger(n-\hat{\mu})iH_\mu(n-\hat{\mu})\zeta_{n-\hat{\mu}}\right] \\
&= -\kappa \sum_{n,\mu} iH_\mu(n)_{ba} \left[\eta_{nb}^\dagger (1+\gamma_\mu)(U_\mu(n)\zeta_{n+\hat{\mu}})_a - (\eta_{n+\hat{\mu}}^\dagger U_\mu^\dagger(n))_b (1-\gamma_\mu)\zeta_{n,a}\right] \\
&= -\kappa \sum_{n,\mu} iH_\mu(n)_{ba} \left[\left((\zeta_{n+\hat{\mu}}^\dagger U_\mu^\dagger(n))_a (1+\gamma_\mu)\eta_{n,b}\right)^* - \left(\zeta_{n,a}^\dagger (1-\gamma_\mu)(U_\mu(n)\eta_{n+\hat{\mu}})_b\right)^*\right] \\
&= -\kappa \sum_{n,\mu} iH_\mu(n)_{ba} \left[(\zeta_{n+\hat{\mu}}^\dagger U_\mu^\dagger(n))_b (1+\gamma_\mu)\eta_{n,a} - \zeta_{n,b}^\dagger (1-\gamma_\mu)(U_\mu(n)\eta_{n+\hat{\mu}})_a\right]^\dagger. \quad (144)
\end{aligned}$$

Therefore

$$\eta^\dagger \frac{dQ}{d\tau}\eta = -\kappa \sum_{n,\mu} iH_\mu(n)_{ba} \left[(\zeta_{n+\hat{\mu}}^\dagger U_\mu^\dagger(n))_b (1+\gamma_\mu)\eta_{n,a} - \zeta_{n,b}^\dagger (1-\gamma_\mu)(U_\mu(n)\eta_{n+\hat{\mu}})_a\right]_A \quad (145)$$

where $[\ldots]_A$ stands for the anti-hermitian operation, i.e., $[M]_A = [M - M^\dagger]/2$. Defining

$$R_\mu(n)_{ab} = \left[(\zeta_{n+\hat{\mu}}^\dagger U_\mu^\dagger(n))_b (1+\gamma_\mu)\eta_{n,a} - \zeta_{n,b}^\dagger (1-\gamma_\mu)(U_\mu(n)\eta_{n+\hat{\mu}})_a\right]_A, \quad (146)$$

$$\begin{aligned}
\eta^\dagger \frac{dQ}{d\tau}\eta &= -2\kappa \sum_{m,\mu} iH_\mu(n)_{ba} R_\mu(n)_{ab} \quad (147) \\
&= -2\kappa \sum_{m,\mu} \mathrm{Tr}[iH_\mu(n)R_\mu(n)]. \quad (148)
\end{aligned}$$

Thus

$$\frac{d}{d\tau}\mathcal{H} = \sum_{n,\mu} \mathrm{Tr}\left[\dot{H}_\mu(n)H_\mu(n) - \frac{\beta}{3}iH_\mu(n)[U_\mu(n)V_\mu]_A - 2\kappa H_\mu(n)R_\mu(n)\right]. \quad (149)$$

$H_\mu(n)$ satisfies a differential equation

$$-i\dot{H}_\mu(n) - \frac{\beta}{3}[U_\mu(n)V_\mu]_A - 2\kappa R_\mu(n) = 0. \quad (150)$$

Since $H$ is traceless,

$$i\dot{H}_\mu(n) = -\frac{\beta}{3}[U_\mu(n)V_\mu]_{TA} - 2\kappa R_\mu(n) \quad (151)$$

holds. Here, $[\ldots]_{TA}$ stands for the traceless and anti-hermitian operation.

# A Jackknife method

The jackknife method is used to estimate an error of a quantities expressed as a complicated function of measured observables or a result of fit. First, we consider the simplest form of the method with a bin size 1. Let us define the average value without $k$-th measured value,

$$\langle O \rangle_k \equiv \frac{1}{M-1} \sum_{i \neq k} A_i. \tag{152}$$

For a quantity $f(O)$ which is a function of $O$, the average and error of $f(O)$ can be calculated as

$$\langle f(O) \rangle = \frac{1}{M} \sum_{k}^{M} f(\langle O \rangle_k) \tag{153}$$

$$\delta \langle f(O) \rangle = \sqrt{(M-1)\{\langle f(O)^2 \rangle - \langle f(O) \rangle^2}. \tag{154}$$

Applying these formulae to $O$ itself, one easily reproduces Eqs. (30) and (31).

Similarly, the jackknife method with bin size $m$ is constructed as follows. Firstly all data is decomposed into $M_m = M/m$ bins, so that each bin having $m$ data is labeled with $b$ ($b = 1, ..., M_m$), and the set of bins is denoted by $B_b$. An average value without bin $b$ is determined as

$$\langle O \rangle_b \equiv \frac{1}{M-m} \sum_{i \notin B_b} O_i. \tag{155}$$

Then the average and error of $f(O)$ is evaluated as

$$\langle f(O) \rangle = \frac{1}{M_m} \sum_{b=1}^{M_m} f(\langle O \rangle_b) \tag{156}$$

$$\delta \langle f(O) \rangle = \sqrt{(M_m-1)\{\langle f(O)^2 \rangle - \langle f(O) \rangle^2\}}. \tag{157}$$

The jackknife method has several advantages:

- Applicable to arbitrary function of observable $f(O)$.
- Applicable to quantities related to the variance, such as a susceptibility and a specific heat.
- It incorporates the correlation between observables.
- By changing the size of bins, the autocorrelation autocorrelation can be estimated. This is done observing the dependence of errors on the bin size, which disappears in the absence of correlation.

On the other hand, jackknife method estimates a symmetric error in plus and minus, as same as with Eqs. (30), (31). This is valid only when the error is well small compared to the fluctuation of the observable. If it is not the case, jackknife method may fail, and more elaborated method such as bootstrap method should be employed.

## References

[1] T. Muta, *Fundations of Quantuim Chromodynamics*, 2nd ed. (World Scientific, 1998).

[2] M. Creutz, *Quarks, gluons, and lattices* (Cambridge Univ. Press, 1983).

[3] I.M.Montvay and G. Münster, *Quantum Fields on a Lattice* (Cambridge Univ. Press, 1994).

[4] H. J. Rothe, *LATTICE GAUGE THEORIES* 2nd ed., (World Scientific, 1992) (now 3rd ed. is available).

[5] K. Symanzik, Nucl. Phys. B **226** (1983) 187; Nucl. Phys. B **226** (1983) 205.

[6] H. B. Nielsen and M. Ninomiya, Nucl. Phys. B **185** (1981) 20 [Erratum-ibid. B **195** (1982) 541]; Nucl. Phys. B **193** (1981) 173; Phys. Lett. B **105** (1981) 219.

[7] For a review, see M. Luscher, “Chiral gauge theories revisited,” arXiv:hep-th/0102028.

[8] D. B. Kaplan, “A Method for simulating chiral fermions on the lattice,” Phys. Lett. B **288** (1992) 342 [arXiv:hep-lat/9206013].

[9] Y. Shamir, “Chiral fermions from lattice boundaries,” Nucl. Phys. B **406** (1993) 90 [arXiv:hep-lat/9303005]; V. Furman and Y. Shamir, “Axial Symmetries In Lattice QCD With Kaplan Fermions,” Nucl. Phys. B **439** (1995) 54 [arXiv:hep-lat/9405004].

[10] P. H. Ginsparg and K. G. Wilson, “A Remnant Of Chiral Symmetry On The Lattice,” Phys. Rev. D **25** (1982) 2649.

[11] M. Luscher, “Exact chiral symmetry on the lattice and the Ginsparg-Wilson relation,” Phys. Lett. B **428** (1998) 342 [arXiv:hep-lat/9802011].

[12] P. Hasenfratz, V. Laliena and F. Niedermayer, “The index theorem in QCD with a finite cut-off,” Phys. Lett. B **427** (1998) 125 [arXiv:hep-lat/9801021].

[13] P. Hasenfratz, Nucl. Phys. Proc. Suppl. **63** (1998) 53 [arXiv:hep-lat/9709110].

[14] 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].

[15] JLQCD Collaboration, `http://jlqcd.kek.jp/`.

[16] A. Ukawa [CP-PACS and JLQCD Collaborations], Nucl. Phys. Proc. Suppl. **106** (2002) 195.

[17] M. Luscher, JHEP **0305** (2003) 052 [arXiv:hep-lat/0304007].

[18] M. Luscher, Comput. Phys. Commun. **165** (2005) 199 [arXiv:hep-lat/0409106].

[19] The gauge Connection, `http://qcd.nersc.gov/`.

[20] The International Lattice DataGrid, `http://www.lqcd.org/ildg/`.

[21] M. Hasenbusch, Phys. Lett. B **519** (2001) 177 [arXiv:hep-lat/0107019].

[22] A. Ukawa and M. Fukugita, Phys. Rev. Lett. **55** (1985) 1854

[23] G. G. Batrouni, et al, Phys. Rev. **D32** (1985) 2736

[24] D. J. E. Callaway and A. Rahman, Phys. Rev. Lett. **49** (1982) 612

[25] D. J. E. Callaway and A. Rahman, Phys. Rev. **D28** (1983) 1506

[26] J. Polonyi, et al, Phys. Rev. Lett. **53** (1984) 644

[27] Duane, Nucl. Phys. **B257** (1985) 652

[28] Duane and Kogut, Phys. Rev. Lett. **55** (1985) 2774

[29] Duane and Kogut, Nucl. Phys. **B275** (1986) 398

[30] S. Gottlieb, et al, Phys. Rev. **D35** (1987) 2531

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

[32] H. A. van der Vorst, *Iterative Krylov Methods for Large Linear Systems* (Cambridge Univ. Press, 2003).